

Action Translation in Extensive-Form Games with Large Action Spaces: Axioms, Paradoxes, and the Pseudo-Harmonic Mapping*

Sam Ganzfried and Tuomas Sandholm

Computer Science Department

Carnegie Mellon University

{sganzfri, sandholm}@cs.cmu.edu

Abstract

When solving extensive-form games with large action spaces, typically significant abstraction is needed to make the problem manageable from a modeling or computational perspective. When this occurs, a procedure is needed to interpret actions of the opponent that fall outside of our abstraction (by mapping them to actions in our abstraction). This is called an action translation mapping. Prior action translation mappings have been based on heuristics without theoretical justification. We show that the prior mappings are highly exploitable and that most of them violate certain natural desiderata. We present a new mapping that satisfies these desiderata and has significantly lower exploitability than the prior mappings. Furthermore, we observe that the cost of this worst-case performance benefit (low exploitability) is not high in practice; our mapping performs competitively with the prior mappings against no-limit Texas Hold'em agents submitted to the 2012 Annual Computer Poker Competition. We also observe several paradoxes that can arise when performing action abstraction and translation; for example, we show that it is possible to improve performance by including suboptimal actions in our abstraction and excluding optimal actions.

1 Introduction

Abstraction has emerged as a necessary component in solving large games. There are several reasons abstraction may be necessary. First, the model that one creates of the real world or of a complex artificial system is typically an abstraction. Game-theoretic modeling of security games and trading agent competitions are examples of this [Wellman, 2006]. Second, the model may be too computationally complex to solve, and thus needs to be abstracted further. For example, this is the typical way the top programs approach Texas Hold'em poker. Third, the solver that is used to find the game-theoretic strategies in the model may assume a certain kind of game, and

the model may not fall within that class without further abstraction. For example, the solver may assume that there is a countable or finite number of actions. Fourth, in certain kinds of game models a game-theoretic equilibrium might not even exist, and to be guaranteed existence, one may want to abstract the model further. For example, this has been discussed in the context of computational billiards [Archibald and Shoham, 2009].

In many domains, significant abstraction is necessary in order to produce software agents. For example, the variant of no-limit Texas Hold'em currently used in the Annual Computer Poker Competition has approximately 10^{165} states in its game tree [Johanson, 2013], while the best approximate equilibrium-finding algorithms “only” scale to games with about 10^{12} states [Hoda *et al.*, 2010; Zinkevich *et al.*, 2007]. In general, extensive-form games can have enormous strategy spaces for two primary reasons: the game tree has many *information sets* (i.e., game states where players must choose an action), or players have many actions available at each information set (e.g., when actions correspond to real numbers from some large set). There are two kinds of abstraction to deal with these two sources of complexity: *information abstraction* and *action abstraction* [Billings *et al.*, 2003; Gilpin and Sandholm, 2006; Sandholm, 2010]. In information abstraction, one groups information sets of a player together in order to reduce the total number of information sets. (Essentially this forces the player to play the game the same way in two different states of knowledge.) In action abstraction, one reduces the size of the action space. The typical approach for performing action abstraction is to discretize an action space into a smaller number of allowable actions; for example, instead of allowing agents to bid any integral amount between \$1 and \$1000, perhaps we limit the actions to only multiples of \$10 or \$100. This approach applies to almost any game where action sizing is an issue, such as bet sizing in poker, bid sizing in auctions, offer sizing in negotiations, allocating different quantities of attack resources or defense resources in security games, and so on.

One issue that can arise when performing action abstraction is that the opponent might take an action that we have removed from the model. For example, we may have limited bids to multiples of \$100, but the opponent makes a bid of \$215. We need an intelligent way of interpreting and responding to such actions which are not in our abstraction. The

*This material is based upon work supported by the National Science Foundation under grants IIS-0964579 and CCF-1101668. We also acknowledge Intel Corporation and IBM for their machine gifts.

standard approach for doing this is to apply an *action translation mapping* (aka *reverse mapping* [Gilpin *et al.*, 2008], *state translation* [Schnizlein *et al.*, 2009]), which maps the observed action a of the opponent to an action a' in the abstraction; then we simply respond as if the opponent had played a' instead of a . A natural action translation mapping would be to map the observed action to the closest action in our abstraction (according to a natural distance metric); in the example just described, this mapping would map the bid of \$215 to \$200. However, this is just one possible mapping, and significantly more sophisticated ones are possible.

Several prior action translation mappings have been proposed for the domain of no-limit Texas Hold'em [Andersson, 2006; Gilpin *et al.*, 2008; Rubin and Watson, 2012; Schnizlein *et al.*, 2009]. However, these have all been based on heuristics and lack any theoretical justification. We show that most of the prior approaches violate certain natural desiderata and that all of them are highly exploitable in simplified games. (Exploitability in such simplified games is a standard evaluation technique since it cannot be computed in the large.) We present a new mapping, called the *pseudo-harmonic mapping*, that satisfies these desiderata and has significantly lower exploitability than the prior mappings. Thus, we expect our mapping to perform much better than the prior ones against sophisticated adaptive opponents who are specifically trying to exploit our mapping. (For one, any strong human poker player would try this against a computer program.) Furthermore, we observe that the cost of this worst-case performance benefit (low exploitability) is not high in practice; our mapping performs competitively with the prior mappings against no-limit Texas Hold'em agents submitted to the 2012 Annual Computer Poker Competition.

2 Action Translation

Suppose the set of allowable actions at a given information set is some subset of the real interval $S = [\underline{T}, \bar{T}]$. (In no-limit poker, \underline{T} will be zero and \bar{T} will be the stack size of the player to act.) An action abstraction at this information set will correspond to a finite increasing sequence (A_0, \dots, A_k) with $\underline{T} \leq A_0$ and $A_k \leq \bar{T}$. (In our experiments we will set $A_0 = \underline{T}$ and $A_k = \bar{T}$; that is, the interval boundaries will be in our abstraction. In abstractions where that is not the case, actions that fall outside of $[A_0, A_k]$ can simply be mapped to A_0 or A_k .)

Now suppose the opponent takes some action $x \in S$. Let $A = \max\{A_i : A_i \leq x\}$, and let $B = \min\{A_i : A_i \geq x\}$. Then $x \in [A, B]$, where $\underline{T} \leq A \leq B \leq \bar{T}$. The action translation problem is to determine whether we should map x to A or to B (perhaps probabilistically). Thus, our goal is to construct a function $f_{A,B}(x)$, which denotes the probability that we map x to A ($1 - f_{A,B}(x)$ denotes the probability that we map x to B). This is our *action translation mapping*. Ideally we would like to find the mapping that produces the lowest exploitability when paired with a given action abstraction and equilibrium-finding algorithm. We call the value x^* for which $f_{A,B}(x^*) = \frac{1}{2}$ the *median* of f (if it exists).

3 No-Limit Poker

We will evaluate different action translation mappings empirically in several variants of two-player no-limit poker. In all variants, both players sit down at a table with a *stack* of chips worth some monetary amount. For example, suppose each player has 400 chips worth \$1 each. At each hand, the players must put some number of chips initially into a *pot* in the middle of the table. In some variants, both players put in an *ante* of the same amount—e.g., \$1 each. In other variants, players put in different amounts; e.g., the *small blind* puts in \$1 and the *big blind* puts in \$2. After these initial investments have been made, both players are dealt some number of private cards (that only they can see) at random from a deck.

Next, there is an initial round of betting. The player whose turn it is to act can choose from three available options:

- *Fold*: Give up on the hand, surrendering the pot to the opponent.
- *Call*: Put in the minimum number of chips needed to match the number of chips put into the pot by the opponent. For example, if the opponent has put in \$5 and we have put in \$2, a call would require putting in \$3 more. A call of zero chips is also known as a *check*.
- *Bet*: Put in additional chips beyond what is needed to call. A bet can be of any size up to the number of chips a player has in his stack (provided it exceeds some minimum size). A bet of all of one's remaining chips is called an *all-in bet*. If the opponent has just bet, then our additional bet is also called a *raise*. In some variants, the number of raises in a given round is limited, and players are forced to either fold or call beyond that limit.

The initial round of betting ends if a player has folded, if there has been a bet and a call, or if both players have called or checked. Depending on the variant, there may be public cards revealed face-up on the table and additional rounds of betting (with the same rules, except potentially with a different player going first). If a player ever folds, the other player wins all the chips in the pot. If the final betting round is completed without a player folding, then both players reveal their private cards, and the player with the best hand wins the pot (it is divided equally if there is a tie).

3.1 Clairvoyance Game

In the *clairvoyance game* [Ankenman and Chen, 2006], player P2 is given no private cards, and P1 is given a single card drawn from a distribution that is half winning hands and half losing hands. Both players have stacks of size n , and they both ante \$0.50 (so the initial size of the pot is \$1). P1 is allowed to bet any amount $x \in [0, n]$. Then P2 is allowed to call or fold (but not raise).

3.2 Kuhn Poker

No-limit Kuhn poker is similar to the clairvoyance game, except that both players are dealt a single private card from a three-card deck containing a King, Queen, and a Jack [Ankenman and Chen, 2006; Kuhn, 1950].¹ For Kuhn

¹In limit Kuhn poker, player 2 is allowed to bet following a check of player 1; this is not allowed in no-limit Kuhn poker.

poker and the clairvoyance game, we restrict all bets to be multiples of \$0.10.

3.3 Leduc Hold'em

In Leduc Hold'em, both players are dealt a single card from a 6-card deck with two Kings, two Queens, and two Jacks. Both players start with \$12 in their stack, and ante \$1 [Vaughn *et al.*, 2009; Schnizlein *et al.*, 2009]. There is initially a round of betting, then one community card is dealt and there is a second round of betting. Any number of bets and raises is allowed (up to the number of chips remaining in one's stack).

3.4 Texas Hold'em

In Texas Hold'em, both players are dealt two private cards from a 52-card deck. Using the parameters of the Annual Computer Poker Competition, both players have initial stacks of size 20,000, with a small blind of 50 and big blind of 100. The game has four betting rounds. The first round takes place before any public information has been revealed. Then three public cards are dealt, and there is a second betting round. One more public card is then dealt before each of the two remaining betting rounds.

4 Action Translation Desiderata

Before presenting an analysis of action translation mappings for the domain of poker, we first introduce a set of natural domain-independent properties that any reasonable action translation mapping should satisfy.

- Boundary Constraints.** If the opponent takes an action that is actually in our abstraction, then it is natural to map his action to the corresponding action with probability 1. Hence we require that $f(A) = 1$ and $f(B) = 0$.
- Monotonicity.** As the opponent's action moves away from A towards B , it is natural to require that the probability of his action being mapped to A does not increase. Thus we require that f be non-increasing.
- Action Robustness.** We want f to be robust to small changes in x . If f changes abruptly at some x^* , then the opponent could potentially significantly exploit us by betting slightly above or below x^* . Thus, we require that $f_{A,B}$ is continuous in x , and preferably Lipschitz continuous as well.²
- Boundary Robustness.** We also want f to be robust to small changes in A and B . If a tiny change in A (say from A_1 to A_2) caused $f_{A,B}(x)$ to change dramatically, then it would mean that f was incorrectly interpreting a bet of size x for either $A = A_1$ or $A = A_2$, and could be exploited if the boundary happened to be chosen poorly. Thus, we require that f be continuous and ideally Lipschitz continuous in A and B .

²A function $f : X \rightarrow Y$ is *Lipschitz continuous* if there exists a real constant $K \geq 0$ such that, for all $x_1, x_2 \in X$, $d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$.

5 Prior Mappings

Several action translation mappings have been proposed in the literature for no-limit Texas Hold'em [Andersson, 2006; Gilpin *et al.*, 2008; Rubin and Watson, 2012; Schnizlein *et al.*, 2009]. In this section we describe them briefly. In later sections, we will analyze the mappings in more detail, both empirically and theoretically. For all the mappings, we assume that the pot initially has size 1 and that all values have been scaled accordingly.

5.1 Deterministic Arithmetic

The deterministic arithmetic mapping is the simple mapping described in the introduction. If $x < \frac{A+B}{2}$, then x is mapped to A ; otherwise x is mapped to B . In poker, this mapping can be highly exploitable. For example, suppose A is a pot-sized bet (e.g., of 1) and B is an all-in (e.g., of 100). Then the opponent could significantly exploit us by betting slightly less than $\frac{A+B}{2}$ with his strong hands. Since we will map his bet to A , we will end up calling much more often than we should with weaker hands. For example, suppose our strategy calls a pot-sized bet of 1 with probability $\frac{1}{2}$ with a medium-strength hand. If the opponent bets 1 with a very strong hand, his expected payoff will be $1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{2} = 1.5$. However, if instead he bets 50, then his expected payoff will be $1 \cdot \frac{1}{2} + 51 \cdot \frac{1}{2} = 26$. In fact, this phenomenon was observed in the 2007 Annual Poker Competition when the agent Tartanian1 used this mapping [Gilpin *et al.*, 2008].

5.2 Randomized Arithmetic

This mapping improves upon the deterministic mapping by incorporating randomness [Andersson, 2006; Gilpin *et al.*, 2008]:

$$f_{A,B}(x) = \frac{B-x}{B-A}$$

Now a bet at $x^* = \frac{A+B}{2}$ is mapped to both A and B with probability $\frac{1}{2}$. While certainly an improvement, it turns out that this mapping is still highly exploitable for similar reasons. For example, suppose the opponent bets 50.5 in the situation described above, and suppose that we will call an all-in bet with probability $\frac{1}{101}$. Then his expected payoff will be

$$\frac{1}{2}(1 \cdot \frac{1}{2} + 51.5 \cdot \frac{1}{2}) + \frac{1}{2}(1 \cdot \frac{100}{101} + 51.5 \cdot \frac{1}{101}) = 13.875.$$

This mapping was used by the agent AggroBot [Andersson, 2006].

5.3 Deterministic Geometric

In contrast to the arithmetic approaches, which consider differences from the endpoints, the deterministic geometric mapping uses a threshold x^* at the point where the ratios of x^* to A and B to x^* are the same [Gilpin *et al.*, 2008]. In particular, if $\frac{A}{x} > \frac{x}{B}$ then x is mapped to A ; otherwise x is mapped to B . Thus, the threshold will be $x^* = \sqrt{AB}$ rather than $\frac{A+B}{2}$. This will diminish the effectiveness of the exploitation described above; namely to make a large value bet just below the threshold. This mapping was used by the agent Tartanian2 in the 2008 Annual Computer Poker Competition [Gilpin *et al.*, 2008].

5.4 Randomized Geometric 1

Two different randomized geometric approaches have also been used by strong poker agents. Both behave similarly and satisfy $f_{A,B}(\sqrt{AB}) = \frac{1}{2}$. The first has been used by at least two strong agents in the competition, Sartre and Hyperborean [Rubin and Watson, 2012; Schnizlein *et al.*, 2009]:

$$g_{A,B}(x) = \frac{\frac{A}{x} - \frac{A}{B}}{1 - \frac{A}{B}} \quad h_{A,B}(x) = \frac{\frac{x}{B} - \frac{A}{B}}{1 - \frac{A}{B}}$$

$$f_{A,B}(x) = \frac{g_{A,B}(x)}{g_{A,B}(x) + h_{A,B}(x)} = \frac{A(B-x)}{A(B-x) + x(x-A)}$$

5.5 Randomized Geometric 2

The second one was used by another strong agent, Tartanian4, in the 2010 competition:

$$f_{A,B}(x) = \frac{A(B+x)(B-x)}{(B-A)(x^2 + AB)}$$

6 Our New Mapping

The prior mappings have all been based on heuristics without theoretical justification. We propose a new mapping that is game-theoretically motivated as the generalization of the solution to a simplified game—specifically, the clairvoyance game described in Section 3.1. The clairvoyance game is small enough that its solution can be computed analytically (a derivation is given in Appendix A):

- P1 bets n with probability 1 with a winning hand.
- P1 bets n with probability $\frac{n}{1+n}$ with a losing hand (and checks otherwise).
- For all $x \in [0, n]$, P2 calls a bet of size x with probability $\frac{1}{1+x}$.

In fact, these betting and calling frequencies have been shown to be optimal in many other poker variants as well [Ankenman and Chen, 2006].

Using this as motivation, our new action translation mapping will be the solution to

$$f_{A,B}(x) \cdot \frac{1}{1+A} + (1 - f_{A,B}(x)) \cdot \frac{1}{1+B} = \frac{1}{1+x}.$$

Specifically, our mapping is

$$f_{A,B}(x) = \frac{(B-x)(1+A)}{(B-A)(1+x)}.$$

This is the only mapping consistent with player 2 calling a bet of size x with probability $\frac{1}{1+x}$ for all $x \in [A, B]$.

This mapping is not as susceptible to the exploitations previously described. The median of f is

$$x^* = \frac{A+B+2AB}{A+B+2}.$$

As for the arithmetic and geometric mappings, we define both deterministic and randomized versions of our new mapping. The randomized mapping plays according to f as described above, while the deterministic mapping plays deterministically using the threshold x^* .

If we assumed that a player would call a bet of size x with probability $\frac{1}{x}$ instead of $\frac{1}{1+x}$, then the median would be the *harmonic mean* of the boundaries A and B : $\frac{2AB}{A+B}$. Because of this resemblance,³ we will call our new mapping the *pseudo-harmonic mapping*. We will abbreviate the deterministic and randomized versions of the mapping as Det-psHar and Rand-psHar.

7 Graphical examples

In Figure 1 we plot all four randomized mappings using $A = 0.01$ and $B = 1$. As the figure shows, both of the randomized geometric mappings have a median of 0.1 pot, while the median of the arithmetic mapping is around 0.5 pot and the median of the pseudo-harmonic mapping is around 0.34 pot. In this case, the mappings differ significantly.

In Figure 2, we plot the mappings using $A = 1$ and $B = 4$. In this case the pseudo-harmonic mapping is relatively similar to the geometric mappings, while the arithmetic mapping differs significantly from the others.

8 Theoretical Analysis

Before we present an axiomatic analysis of the mappings, we first note that $A = 0$ is somewhat of a degenerate special case. In particular, the geometric mappings are the constant function $f = 0$ for $A = 0$, and they behave much differently than they do for $A > 0$ (even for A arbitrarily small). So we will analyze these mappings separately for the $A = 0$ and $A > 0$ cases. In many applications it is natural to have $A = 0$; for example, for the interval between a check and a pot-sized bet in poker, we will have $A = 0$ and $B = 1$. So the degenerate behavior of the geometric mappings for $A = 0$ can actually be a significant problem in practice.⁴

All of the mappings satisfy the Boundary Conditions for $A > 0$, while the geometric mappings violate them for $A = 0$, since they map A to 0 instead of 1. All of the mappings satisfy (weak) Monotonicity (though the deterministic ones violate strict Monotonicity, as do the geometric ones for $A = 0$).

It is easy to see that the deterministic mappings violate Action Robustness, as they are clearly discontinuous at the threshold (this is true for any deterministic mapping). The randomized mappings satisfy Action Robustness, as their derivatives are bounded. The deterministic mappings all violate Boundary Robustness as well, since increasing A from A_1 to A_2 will cause $f(x)$ to change abruptly from 0 to 1 for some values of x near the threshold. It is natural to use the L^∞ norm to define distances between mappings, since a mapping could be exploited if it behaves poorly on just a single

³We call our mapping *pseudo-harmonic* because it is actually quite different from the one based on the harmonic series. For example, for $A = 0$ and $B = 1$ the median of the new mapping is $\frac{1}{3}$, while the harmonic mean is 0.

⁴Some poker agents never map a bet to 0, and map small bets to the smallest positive betting size in the abstraction (e.g., $\frac{1}{2}$ pot). This approach could be significantly exploited by an opponent who makes extremely small bets as bluffs, and is not desirable.

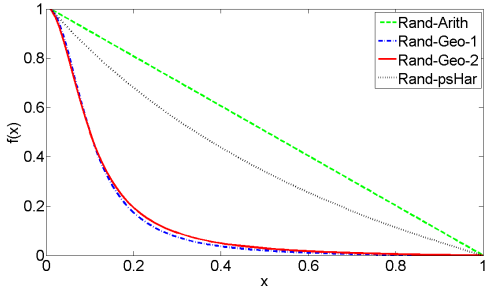


Figure 1: Randomized mappings with $A = 0.01$, $B = 1$.

action. Formally,

$$d(f_{A_1, B_1}, f_{A_2, B_2}) = \max_{x \in S} |f_{A_1, B_1}(x) - f_{A_2, B_2}(x)|,$$

where $S = [A_1, B_1] \cap [A_2, B_2]$ is nonempty. Using this definition, Rand-Arith and Rand-psHar are Lipschitz continuous in both A and B (even for $A = 0$), while Rand-Geo-1 and Rand-Geo-2 are discontinuous in A for $A = 0$, and Lipschitz discontinuous in A for $A > 0$. We present proofs for Rand-psHar and Rand-Geo-2 in Appendix B (the proofs of the results for Rand-Geo-1 are analogous to the proofs for Rand-Geo-2).

Proposition 1. *Rand-psHar is Lipschitz continuous in A .*

Proposition 2. *For any $B > 0$, Rand-Geo-1 and Rand-Geo-2 are not continuous in A , where A has domain $[0, B)$.*

Proposition 3. *For any $B > 0$, Rand-Geo-1 and Rand-Geo-2 are not Lipschitz continuous in A , where A has domain $(0, B)$.*

To give some intuition for why Boundary Robustness is important, we examine the effect of increasing A gradually from 0 to 0.1, while holding $B = 1$ and $x = 0.25$ fixed. Table 1 shows the value of $f_{A, B}(x)$ for several values of A , for each of the randomized mappings. For the two mappings that satisfy Boundary Robustness—Rand-Arith and Rand-psHar—the values increase gradually as A is increased: Rand-Arith increases from 0.75 at $A = 0$ to 0.833 at $A = 0.1$, while Rand-psHar increases from 0.6 to 0.733. The two geometric mappings increase much more sharply, from 0 to 0.667 and 0.641 respectively. In practice, we may not know the optimal values to use in our abstraction ex ante, and may end up selecting them somewhat arbitrarily. If we end up making a choice that is not quite optimal (for example, 0.01 instead of 0.05), we would like it to not have too much of an effect. For non-robust mappings, the effect of making poor decisions in these situations could be much more severe than desired.

	A				
	0	0.001	0.01	0.05	0.1
Rand-Arith	0.75	0.751	0.758	0.789	0.833
Rand-Geo-1	0	0.012	0.111	0.429	0.667
Rand-Geo-2	0	0.015	0.131	0.439	0.641
Rand-psHar	0.6	0.601	0.612	0.663	0.733

Table 1: Effect of increasing A while holding $B = 1$ and $x = 0.25$ fixed.

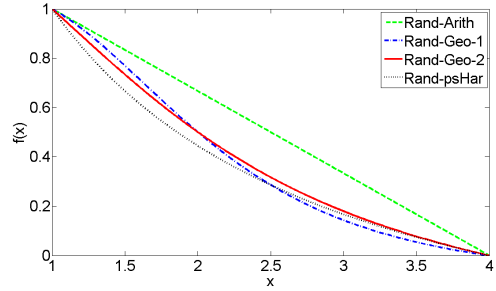


Figure 2: Randomized mappings with $A = 1$, $B = 4$.

9 Comparing Exploitability

The *exploitability* of a strategy is the difference between the value of the game and worst-case performance against a nemesis. In particular, Nash equilibrium strategies are precisely those that have zero exploitability. Since our main goal is to approximate equilibrium strategies, minimizing exploitability is a natural metric for evaluation. The clairvoyance game, Kuhn poker, and Leduc Hold'em are small enough that exploitability can be computed exactly.

9.1 Clairvoyance Game

In Table 2, we present the exploitability of the mappings described in Section 5 in the clairvoyance game. We varied the starting stack from $n = 1$ up to $n = 100$, experimenting on 7 games in total. (A wide variety of stack sizes relative to the blinds are encountered in poker in practice, so it is important to make sure a mapping performs well for many stack sizes.) For these experiments, we used the betting abstraction {fold, check, pot, all-in} (fcpa). This abstraction is a common benchmark in no-limit poker [Gilpin *et al.*, 2008; Hawkin *et al.*, 2011; 2012; Schnizlein *et al.*, 2009]: “previous expert knowledge [has] dictated that if only a single bet size [in addition to all-in] is used everywhere, it should be pot sized” [Hawkin *et al.*, 2012].

For the abstract equilibrium, we used the equilibrium strategy described in Section 6.⁵ The entries in Table 2 give player 2’s exploitability for each mapping. The results show that the exploitability of Rand-psHar stays constant at zero, while the exploitability of the other mappings steadily increases as the stack size increases. As we have predicted, the arithmetic mappings are more exploitable than the geometric ones, and the deterministic mappings are more exploitable than the corresponding randomized ones.

9.2 Kuhn Poker

We conducted similar experiments on the more complex game of Kuhn poker; the results are given in Table 3. As in the clairvoyance game, Rand-psHar significantly outperformed the other mappings, with an exploitability near zero for all stack sizes. Interestingly, the relative performances of

⁵We also experimented using the Nash equilibrium at the other extreme (see Appendix A), and the relative performances of the mappings were very similar. This indicates that our results are robust to the abstract equilibrium strategies selected by the solver.

	Stack Size (n)						
	1	3	5	10	20	50	100
Det-Arith	0.01	0.24	0.49	1.12	2.38	6.12	12.37
Rand-Arith	0.00	0.02	0.09	0.36	0.96	2.82	5.94
Det-Geo	0.23	0.28	0.36	0.63	0.99	1.68	2.43
Rand-Geo-1	0.23	0.23	0.23	0.24	0.36	0.66	1.01
Rand-Geo-2	0.23	0.23	0.23	0.25	0.36	0.65	1.00
Det-psHar	0.15	0.19	0.33	0.47	0.59	0.67	0.71
Rand-psHar	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 2: Exploitability of mappings for the clairvoyance game, using betting abstraction {fold, check, pot, all-in}.

the other mappings differ significantly from the results in the clairvoyance game. Rand-Arith performed second-best while Det-psHar performed the worst.⁶

It turns out that for each stack size, player 1 has a unique equilibrium strategy that uses a bet size of 0.4 times the pot (recall that we only allow bets that are a multiple of 0.1 pot). So we thought it would be interesting to see how the results would change if we used the bet size of 0.4 pot in our abstraction instead of pot. Results for these experiments are given in Table 4. Surprisingly, all of the mappings became more exploitable (for larger stack sizes) when we used the “optimal” bet size, sometimes significantly so (for $n = 100$ Det-Arith had exploitability 0.301 using the first abstraction and 3.714 using the second abstraction)! This is very counterintuitive, as we would expect performance to improve as we include “better” actions in our abstraction. It also casts doubt on the typical approach for selecting an action abstraction for poker-playing programs; namely, emulating the bet sizes that human professional poker players use.

We decided to investigate this paradox further, and computed the bet size that minimized exploitability for each of the mappings. The results are given in Table 5.⁷ Interestingly, the unique full equilibrium bet size of 0.4 was very rarely the optimal bet size to use. The optimal bet size varied dramatically as different stack sizes and mappings were used. In some cases it was quite large; for example, for $n = 100$ it was 71.5 for Det-psHar and 29.8 for Det-Geo. The results indicate that the optimal action abstraction to use may vary considerably based on the action translation mapping used, and can include surprising actions while excluding actions that are played in the full equilibrium (even when these are the only actions played in any full equilibrium). This suggests that when using multiple actions in an abstraction, a mix of both “optimal” offensive actions (which are actually taken by the agent) and defensive actions (which are not taken themselves, but reduce exploitability due to an imperfect abstraction) may be more successful than focusing exclusively on the offensive

⁶We do not allow player 2 to fold when player 1 checks for these experiments, since he performs at least as well by checking. The results are even more favorable for Rand-psHar if we remove this restriction because player 2 is indifferent between checking and folding with a Jack, and the abstract equilibrium strategy our solver output happened to select the fold action. The geometric mappings are unaffected by this because they never map a bet down to a check, but the other mappings sometimes do and will correctly fold a Jack more often to a small bet. In particular, Rand-psHar obtained exploitability 0 for all stack sizes using fcpa.

⁷For ties, we reported the smallest size.

ones. This is consistent with the approach that some teams in the competition have been using where they insert large defensive actions into the abstraction on the opponent’s side.

	Stack Size (n)						
	1	3	5	10	20	50	100
Det-Arith	0.205	0.205	0.244	0.271	0.287	0.298	0.301
Rand-Arith	0.055	0.055	0.055	0.055	0.055	0.055	0.055
Det-Geo	0.121	0.121	0.121	0.217	0.297	0.366	0.399
Rand-Geo-1	0.121	0.121	0.121	0.121	0.121	0.121	0.121
Rand-Geo-2	0.121	0.121	0.121	0.121	0.121	0.121	0.121
Det-psHar	0.171	0.171	0.233	0.365	0.454	0.520	0.545
Rand-psHar	0.029	0.029	0.029	0.029	0.029	0.029	0.029

Table 3: Exploitability of mappings for no-limit Kuhn poker, using betting abstraction {fold, check, pot, all-in}.

	Stack Size (n)						
	1	3	5	10	20	50	100
Det-Arith	0.088	0.110	0.285	0.485	0.848	1.926	3.714
Rand-Arith	0.012	0.033	0.068	0.157	0.336	0.871	1.764
Det-Geo	0.086	0.114	0.294	0.425	0.548	0.714	0.873
Rand-Geo-1	0.071	0.085	0.095	0.116	0.145	0.203	0.269
Rand-Geo-2	0.071	0.083	0.094	0.114	0.144	0.203	0.269
Det-psHar	0.064	0.090	0.302	0.420	0.500	0.556	0.574
Rand-psHar	0.008	0.010	0.017	0.027	0.037	0.047	0.054

Table 4: Exploitability of mappings for no-limit Kuhn poker, using betting abstraction {fold, check, 0.4 pot, all-in}.

	Stack Size (n)						
	1	3	5	10	20	50	100
Det-Arith	0.3	0.4	0.7	0.9	0.9	1.0	1.0
Rand-Arith	0.3	0.5	0.6	0.8	0.9	1.0	1.0
Det-Geo	0.3	0.2	1.0	2.6	2.5	14.6	29.8
Rand-Geo-1	0.2	0.1	0.3	0.4	1.0	1.0	1.0
Rand-Geo-2	0.2	0.1	0.3	0.3	1.0	1.0	1.0
Det-psHar	0.4	0.3	2.3	7.9	4.5	49.9	71.5
Rand-psHar	0.1	0.4	0.5	0.6	0.6	0.7	0.7

Table 5: Optimal bet sizes for player 2 of action translation mappings for no-limit Kuhn poker, using betting abstraction with fold, check, all-in, and one additional bet size.

9.3 Leduc Hold’em

We also compared exploitability on Leduc Hold’em—a much larger poker variant than the Clairvoyance Game and Kuhn Poker. Unlike these smaller variants, Leduc Hold’em allows for multiple bets and raises, multiple rounds of betting, and shared community cards. Thus, it contains many of the same complexities as the variants of poker commonly played by humans—most notably Texas Hold’em—while remaining small enough that exploitability computations are tractable.

Exploitabilities for both players using the fcpa abstraction are given in Table 6. The results indicate that Rand-psHar produces the lowest average exploitability by a significant margin, while Det-Arith produces the highest exploitability. Interestingly, Rand-psHar did not produce the lowest exploitability for player 1; however, its exploitability was by far the smallest for player 2, making its average the lowest. Player 2’s exploitability was higher than player 1’s in general because player 1 acts first in both rounds, causing player 2 to perform more action translation to interpret bet sizes.

	P1 exploitability	P2 exploitability	Avg. exploitability
Det-Arith	0.427	0.904	0.666
Rand-Arith	0.431	0.853	0.642
Det-Geo	0.341	0.922	0.632
Rand-Geo-1	0.295	0.853	0.574
Rand-Geo-2	0.296	0.853	0.575
Det-psHar	0.359	0.826	0.593
Rand-psHar	0.323	0.603	0.463

Table 6: Exploitability of mappings for each player in no-limit Leduc Hold’em using the fcpa betting abstraction.

10 Experiments in Texas Hold’em

We next tested the mappings against the agents submitted to the no-limit Texas Hold’em division of the 2012 Annual Computer Poker Competition. We started with our submitted agent, Tartanian5 [Ganzfried and Sandholm, 2012], and varied the action translation mapping while keeping everything else about it unchanged.⁸ Then we had it play against each of the other entries.

The results are in Table 7 (with 95% confidence intervals included). Surprisingly, Det-Arith performed best using the metric of average overall performance, despite the fact that it was by far the most exploitable in simplified games. Det-psHar, Rand-Arith, and Rand-psHar followed closely behind. The three geometric mappings performed significantly worse than the other four mappings, (and similarly to each other).

One interesting observation is that the performance rankings of the mappings differed significantly from their exploitability rankings in simplified domains (with Det-Arith being the most extreme example). The results can be partially explained by the fact that none of the programs in the competition were attempting any exploitation of bet sizes or of action translation mappings (according to publicly-available descriptions of the agents available on the competition website). Against such unexploitative opponents, the benefits of a defensive, randomized strategy are much less important.⁹ As agents become stronger in the future, we would expect action exploitation to become a much more important factor in competition performance, and the mappings with high exploitability would likely perform significantly worse. In fact, in the 2009 competition, an entrant in the bankroll category (Hyperborean) used a simple strategy (that did not even look at its own cards) to exploit opponents’ betting boundaries and came in first place [Schnizlein *et al.*, 2009].

11 Conclusions and Future Research

We have formally defined the action translation problem and analyzed all the prior action translation mappings which have been proposed for the domain of no-limit poker. We have developed a new mapping which achieves significantly lower exploitability than any of the prior approaches in the clairvoyance game, Kuhn poker, and Leduc Hold’em for a wide

⁸Tartanian5 used Det-psHar in the actual competition.

⁹This is similar to a phenomenon previously observed in the poker competition, where an agent that played a fully deterministic strategy outperformed a version of the same agent that used randomization [Ganzfried *et al.*, 2012].

variety of stack sizes. In no-limit Texas Hold’em, our mapping significantly outperformed the mappings used by the strongest agents submitted to the most recent Annual Computer Poker Competitions (Det-Geo, Rand-Geo-1, and Rand-Geo-2). It did not outperform the two less sophisticated (and highly exploitable) mappings Det-Arith and Rand-Arith because the opponents were not exploitative (though the performance differences were small). We also introduced a set of natural domain-independent desiderata and showed that only our new randomized mapping (and Rand-Arith, which we showed to be highly exploitable) satisfy all of them.

In the course of this work, we observed several paradoxical and surprising results. In Kuhn poker, all of the action translation mappings had lower exploitability for large stack sizes when using an abstraction with a suboptimal action (a pot-sized bet) than when using an abstraction that contained the optimal action (a 0.4 times pot bet), even when all equilibrium strategies use the latter bet size. When we computed what the optimal action abstractions would have been for each mapping, they often included actions that differed significantly from the unique equilibrium actions. In addition, we observed that the naïve deterministic arithmetic mapping actually outperformed all the other mappings against agents submitted to the 2012 Annual Computer Poker Competition despite the fact that it had by far the highest exploitability in simplified domains (and violated many of the desiderata).

This work suggests many avenues for future research. One idea would be to consider more complex action translation mappings in addition to the ones proposed in this paper. For example, one could consider mappings that take into account game-specific information (as opposed to the game-independent ones considered here which only take as input the action size x and the adjacent abstract actions A and B). It also might make sense to use different mappings at different information sets (or even between different actions at the same information set). For example, we may want to use one mapping to interpret smaller bets (e.g., between 0 and a pot-sized bet), but a different one to interpret larger bets. In addition, our paradoxical results in Kuhn poker suggest that, when using multiple actions in an abstraction, a mix of both “optimal” offensive actions and defensive actions may be more successful than focusing exclusively on the offensive ones. Finally, we would like to use our framework (and the new mapping) in applications other than poker, such as those discussed in the introduction.

A Analysis of the Clairvoyance Game

It was shown by Ankenman and Chen [2006] that the strategy profile presented in Section 6 constitutes a Nash equilibrium. Here is a sketch of that argument.

Proposition 4. *The strategy profile presented in Section 6 is a Nash equilibrium of the clairvoyance game.*

Proof. First, it is shown that player 2 must call a bet of size x with probability $\frac{1}{1+x}$ in order to make player 1 indifferent between betting x and checking with a losing hand. For a given x , player 1 must bluff $\frac{x}{1+x}$ as often as he value bets for player 2 to be indifferent between calling and folding. Given

	Action Translation Mapping						
	Det-Arith	Rand-Arith	Det-Geo	Rand-Geo-1	Rand-Geo-2	Det-psHar	Rand-psHar
azure.sky	3135 ± 106	3457 ± 90	2051 ± 96	2082 ± 97	2057 ± 97	2954 ± 96	3041 ± 109
dcubot	880 ± 52	752 ± 51	169 ± 47	156 ± 47	141 ± 46	754 ± 36	622 ± 47
hugh	137 ± 84	122 ± 86	-103 ± 50	-98 ± 52	-117 ± 52	-102 ± 30	42 ± 72
hyperborean	-189 ± 79	-272 ± 77	-216 ± 78	-203 ± 77	-161 ± 75	-161 ± 36	-276 ± 77
little.rock	-115 ± 100	-107 ± 95	-48 ± 92	-22 ± 91	-85 ± 89	165 ± 63	93 ± 87
lucky7.12	772 ± 104	510 ± 105	465 ± 82	471 ± 78	462 ± 78	536 ± 94	565 ± 74
neo.poker.lab	6 ± 97	-37 ± 106	11 ± 101	24 ± 98	31 ± 100	8 ± 31	-9 ± 103
sartre	94 ± 65	-3 ± 65	51 ± 64	86 ± 64	26 ± 64	56 ± 38	50 ± 65
spewy.louie	457 ± 118	421 ± 116	572 ± 102	530 ± 106	475 ± 103	614 ± 60	484 ± 109
uni.mb.poker	856 ± 84	900 ± 87	1588 ± 102	1567 ± 101	1657 ± 104	1148 ± 61	1103 ± 90
Avg.	609	571	454	459	449	597	568

Table 7: No-limit Texas Hold'em results in milli big blinds per hand. The entry is the profit of our agent Tartanian5 using the mapping given in the column against the opponent listed in the row.

these quantities, the expected payoff to player 1 of betting size x will be $v(x) = \frac{x}{2(1+x)}$. This function is monotonically increasing, and therefore player 1 will maximize his payoff by setting $x = n$ and going all-in. \square

It turns out that player 2 does not need to call a bet of size $x \neq n$ with exact probability $\frac{1}{1+x}$: he need only not call with such an extreme probability that player 1 has an incentive to change his bet size from n to x (with either a winning or losing hand). In particular, it can be shown that player 2 need only call a bet of size x with any probability (which can be different for different values of x) in the interval $\left[\frac{1}{1+x}, \min\left\{\frac{n}{x(1+n)}, 1\right\}\right]$ in order to remain in equilibrium. Only the initial equilibrium is reasonable, however, in the sense that we would expect a rational player 2 to maintain the calling frequency $\frac{1}{1+x}$ for all x so that he continues to play a properly-balanced strategy in case player 1 happens to bet x .

B Proofs of Results from Section 8

Proposition 5. *Rand-psHar is Lipschitz continuous in A .*

Proof. Let $A_1, A_2 \in (0, B]$, $A_1 \neq A_2$ be arbitrary, and without loss of generality assume $A_1 < A_2$. Let

$$K = \frac{1+B}{(B-A_1)(1+A_2)}.$$

Then

$$\begin{aligned} & \max_{x \in [A_2, B]} \left| \frac{(B-x)(1+A_1)}{(B-A_1)(1+x)} - \frac{(B-x)(1+A_2)}{(B-A_2)(1+x)} \right| \\ &= \frac{(A_2-A_1)(1+B)}{(B-A_1)(B-A_2)} \max_{x \in [A_2, B]} \left| \frac{B-x}{1+x} \right| \\ &= \frac{(A_2-A_1)(1+B)}{(B-A_1)(B-A_2)} \cdot \frac{B-A_2}{1+A_2} = K|A_2-A_1| \end{aligned}$$

\square

Proposition 6. *For any $B > 0$, Rand-Geo-2 is not continuous in A , where A has domain $[0, B)$.*

Proof. Let $B > 0$ be arbitrary, let $\epsilon = 0.5$, and let $\delta > 0$ be arbitrary. Let $A_1 = 0$ and $A_2 = \frac{\delta}{2}$. Then $f_{A_1, B}(A_2) = 0$ and $f_{A_2, B}(A_2) = 1$. So we have

$$\begin{aligned} & \max_{x \in [A_2, B]} |f_{A_2, B}(x) - f_{A_1, B}(x)| \\ & \geq |f_{A_2, B}(A_2) - f_{A_1, B}(A_2)| = 1 > \epsilon. \end{aligned}$$

But $|A_2 - A_1| = \frac{\delta}{2} < \delta$. So Rand-Geo-2 is not continuous in A at $A = 0$. \square

Proposition 7. *For any $B > 0$, Rand-Geo-2 is not Lipschitz continuous in A , where A has domain $(0, B)$.*

Proof. Let $B > 0, K > 0$ be arbitrary. For now, assume that $0 < A < A' < B$. Then

$$\begin{aligned} & \frac{\max_{x \in [A', B]} |f_{A, B}(x) - f_{A', B}(x)|}{|A' - A|} \\ & \geq \frac{|f_{A, B}(A') - f_{A', B}(A')|}{|A' - A|} = \frac{1 - f_{A, B}(A')}{|A' - A|} \\ & = \frac{B(A' + A)}{(B - A)(A'^2 + AB)} \end{aligned}$$

This quantity is greater than K if and only if

$$A^2(BK) + A(B + A'^2K - KB^2) + (A'B - A'^2BK) > 0.$$

Let $\mu(A)$ denote the LHS of the final inequality. Note that $\mu(A) \rightarrow (A'B - A'^2BK)$ as $A \rightarrow 0$. Since $\mu(A)$ is continuous, there exists some interval $I = (\underline{A}, \bar{A})$ with $0 < \underline{A} < \bar{A} < \min\{\frac{1}{4K}, \frac{B}{2}\}$ such that $\mu(A) > 0$ for all $A \in I$. Let A be any value in I , and let $A' = 2A$. Then we have found A, A' satisfying $0 < A < A' < B$ such that

$$\frac{\max_{x \in [A', B]} |f_{A, B}(x) - f_{A', B}(x)|}{|A' - A|} > K.$$

So Rand-Geo-2 is not Lipschitz continuous in A . \square

References

- [Andersson, 2006] Rickard Andersson. Pseudo-optimal strategies in no-limit poker. Master’s thesis, Umeå University, May 2006.
- [Ankenman and Chen, 2006] Jerrod Ankenman and Bill Chen. *The Mathematics of Poker*. ConJelCo LLC, Pittsburgh, PA, USA, 2006.
- [Archibald and Shoham, 2009] C. Archibald and Y. Shoham. Modeling billiards games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Budapest, Hungary, 2009.
- [Billings *et al.*, 2003] 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.
- [Ganzfried and Sandholm, 2012] Sam Ganzfried and Tuomas Sandholm. Tartanian5: A heads-up no-limit Texas Hold’em poker-playing program. In *Computer Poker Symposium at the National Conference on Artificial Intelligence (AAAI)*, 2012.
- [Ganzfried *et al.*, 2012] Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012.
- [Gilpin and Sandholm, 2006] 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.
- [Gilpin *et al.*, 2008] 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 *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.
- [Hawkin *et al.*, 2011] 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)*, pages 681–687, 2011.
- [Hawkin *et al.*, 2012] 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.
- [Hoda *et al.*, 2010] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.
- [Johanson, 2013] Michael Johanson. Measuring the size of large no-limit poker games. Technical report, University of Alberta, 2013.
- [Kuhn, 1950] H. W. Kuhn. Simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies*, 24, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950.
- [Rubin and Watson, 2012] Jonathan Rubin and Ian Watson. Case-based strategies in computer poker. *AI Communications*, 25(1):19–48, 2012.
- [Sandholm, 2010] Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, pages 13–32, Winter 2010. Special issue on Algorithmic Game Theory.
- [Schnizlein *et al.*, 2009] 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.
- [Waugh *et al.*, 2009] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2009.
- [Wellman, 2006] Michael Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1552–1555, 2006.
- [Zinkevich *et al.*, 2007] 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.