Safe Opponent Exploitation

SAM GANZFRIED, Carnegie Mellon University
TUOMAS SANDHOLM, Carnegie Mellon University

We consider the problem of playing a finitely-repeated two-player zero-sum game safely—that is, guaranteeing at least the value of the game per period in expectation regardless of the strategy used by the opponent. Playing a stage-game equilibrium strategy at each time step clearly guarantees safety, and prior work has conjectured that it is impossible to simultaneously deviate from a stage-game equilibrium (in hope of exploiting a suboptimal opponent) and to guarantee safety. We show that such profitable deviations are indeed possible—specifically, in games where certain types of 'gift' strategies exist, which we define formally. We show that the set of strategies constituting such gifts can be strictly larger than the set of iteratively weakly-dominated strategies; this disproves another recent conjecture which states that all non-iteratively-weakly-dominated strategies are best responses to each equilibrium strategy of the other player. We present a full characterization of safe strategies, and develop efficient algorithms for exploiting suboptimal opponents while guaranteeing safety. We also provide analogous results for sequential perfect and imperfect-information games, and present safe exploitation algorithms and full characterizations of safe strategies for those settings as well. We present experimental results in Kuhn poker, a canonical test problem for game-theoretic algorithms. Our experiments show that 1) aggressive safe exploitation strategies significantly outperform adjusting the exploitation within equilibrium strategies and 2) all the safe exploitation strategies significantly outperform a (non-safe) best response strategy against strong dynamic opponents.

Categories and Subject Descriptors: I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Algorithms, economics, theory

Additional Key Words and Phrases: Game theory, opponent exploitation, multiagent learning

1. INTRODUCTION

In repeated interactions against an opponent, an agent must determine how to balance between *exploitation* (maximally taking advantage of weak opponents) and *exploitability* (making sure that he himself does not perform too poorly against strong opponents). In two-player zero-sum games, an agent can simply play a minimax strategy, which guarantees at least the value of the game in expectation against any opponent. However, doing so could potentially forego significant profits against suboptimal opponents. Thus, an equilibrium strategy has low (zero) exploitability, but achieves low exploitation. On the other end of the spectrum, agents could attempt to learn the opponent's strategy and maximally exploit it; however, doing so runs the risk of being exploited in turn by a deceptive opponent. This is known as the "get taught and exploited problem" [Sandholm 2007]. Such deception is common in games such as poker; for example, a player may play very aggressively initially, then suddenly switch to a more conser-

This material is based upon work supported by the National Science Foundation under grants IIS-0964579, IIS-0905390, and CCF-1101668. We also acknowledge Intel Corporation and IBM for their machine gifts.
Author's addresses: S. Ganzfried and T. Sandholm, Computer Science Department, Carnegie Mellon University.

EC'12, June 4–8, 2012, Valencia, Spain.

vative strategy to capitalize on the fact that the opponent tries to take advantage of his aggressive 'image,' which he now leaves behind. Thus, pure opponent modeling potentially leads to a high level of exploitation, but at the expense of exploitability. Respectively, the game solving community has, by and large, taken two radically different approaches: finding game-theoretic solutions and opponent modeling/exploitation.

In this paper, we are interested in answering a fundamental question that helps shed some light on this tradeoff:

> **Is it possible to play a strategy that is not an equilibrium in the stage game while simultaneously guaranteeing at least the value of the game in expectation in the worst case?**

If the answer is no, then fully safe exploitation is not possible, and we must be willing to accept some increase in worst-case exploitability if we wish to deviate from equilibrium in order to exploit suboptimal opponents. However, if the answer is yes, then safe opponent exploitation would indeed be possible.

Recently it was proposed that safe opponent exploitation is not possible [Ganzfried and Sandholm 2011]. The intuition for that argument was that the opponent could have been playing an equilibrium all along, and when we deviate from equilibrium to attempt to exploit him, then we run the risk of being exploitable ourselves. However, that argument is incorrect. It does not take into account the fact that our opponent may give us a *gift* by playing an identifiably suboptimal strategy, such as one that is strictly dominated.[1] If such gift strategies are present in a game, then it turns out that safe exploitation can be achieved; specifically, we can deviate from equilibrium to exploit the opponent provided that our worst-case exploitability remains below the total amount of profit won through gifts (in expectation).

Is it possible to obtain such gifts that do not correspond to strictly-dominated strategies? What about other forms of dominance, such as weak, iterated, and dominance by mixed strategies? Recently it was conjectured that all non-iteratively-weakly-dominated strategies are best responses to each equilibrium strategy of the other player [Waugh 2009]. This would suggest that such undominated strategies cannot be gifts, and that gift strategies must therefore be dominated according to some form of dominance. We disprove this conjecture and present a game in which a non-iteratively-weakly-dominated strategy is not a best response to an equilibrium strategy of the other player. Safe exploitation is possible in the game by taking advantage of that particular strategy. We define a formal notion of gifts, which is more general than iteratively-weakly-dominated strategies, and show that safe opponent exploitation is possible specifically in games in which such gifts exist.

Next, we provide a full characterization of the set of safe exploitation strategies, and we present several efficient algorithms for converting any opponent modeling algorithm (that is arbitrarily exploitable) into a fully safe opponent exploitation procedure. One of our algorithms is similar to a procedure that guarantees safety in the limit as the number of iterations goes to infinity [McCracken and Bowling 2004]; however, the algorithms in that paper can be arbitrarily exploitable in the finitely-repeated game setting, which is what we are interested in. The main idea of the algorithm is to play an ϵ-safe best response (a best response subject to the constraint of having exploitability at most ϵ) at each time step rather than a full best response, where ϵ is determined by the total amount of gifts obtained thus far from the opponent. Safe best responses have also been studied in the context of Texas Hold'em poker [Johanson et al. 2007], though that work did not use them for real-time opponent exploitation. We also present several other safe algorithms which alternate between playing an equilibrium and a best

[1] We thank Vince Conitzer for pointing this out to us.

response depending on how much has been won so far in expectation. We note that algorithms have been developed which guarantee ϵ-safety against specific classes of opponents (stationary opponents and opponents with bounded memory) [Powers et al. 2007]; by contrast, our algorithms achieve full safety against all opponents.

It turns out that safe opponent exploitation is also possible in sequential games, though we must redefine what strategies constitute gifts and must make pessimistic assumptions about the opponent's play in game states off the path of play. We present efficient algorithms for safe exploitation in games of both perfect and imperfect information, and fully characterize the space of safe strategies in these game models. We also show when safe exploitation can be performed in the middle of a single iteration of a sequential game.

We compare our algorithms experimentally on Kuhn poker [Kuhn 1950], a simplified form of poker which is a canonical problem for testing game-solving algorithms and has been used as a test problem for opponent-exploitation algorithms [Hoehn et al. 2005]. We observe that our algorithms obtain a significant improvement over the best equilibrium strategy, while also guaranteeing safety in the worst case. Thus, in addition to providing theoretical advantages over both minimax and fully-exploitative strategies, safe opponent exploitation can be effective in practice.

2. GAME THEORY BACKGROUND

In this section, we briefly review relevant definitions and prior results from game theory and game solving.

2.1. Strategic-form games

The most basic game representation, and the standard representation for simultaneous-move games, is the *strategic form*. A *strategic-form game* (aka matrix game) consists of a finite set of players N, a space of *pure strategies* S_i for each player, and a utility function $u_i : \times S_i \to \mathbb{R}$ for each player. Here $\times S_i$ denotes the space of *strategy profiles*—vectors of pure strategies, one for each player.

The set of *mixed strategies* of player i is the space of probability distributions over his pure strategy space S_i. We will denote this space by Σ_i. Define the *support* of a mixed strategy to be the set of pure strategies played with nonzero probability. If the sum of the payoffs of all players equals zero at every strategy profile, then the game is called *zero sum*. In this paper, we will be primarily concerned with two-player zero-sum games. If the players are following strategy profile σ, we let σ_{-i} denote the strategy taken by player i's opponent, and we let Σ_{-i} denote the opponent's entire mixed strategy space.

2.2. Extensive-form games

An *extensive-form* game is a general model of multiagent decision making with potentially sequential and simultaneous actions and imperfect information. As with perfect-information games, extensive-form games consist primarily of a game tree; each non-terminal node has an associated player (possibly *chance*) that makes the decision at that node, and each terminal node has associated utilities for the players. Additionally, game states are partitioned into *information sets*, where the player whose turn it is to move cannot distinguish among the states in the same information set. Therefore, in any given information set, a player must choose actions with the same distribution at each state contained in the information set. If no player forgets information that he previously knew, we say that the game has *perfect recall*. A (behavioral) *strategy* for player i, $\sigma_i \in \Sigma_i$, is a function that assigns a probability distribution over all actions at each information set belonging to i.

2.3. Nash equilibria

Player i's *best response* to σ_{-i} is any strategy in

$$\arg \max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma_{-i}).$$

A *Nash equilibrium* is a strategy profile σ such that σ_i is a best response to σ_{-i} for all i. An ϵ*-equilibrium* is a strategy profile in which each player achieves a payoff of within ϵ of his best response.

In two player zero-sum games, we have the following result which is known as the *minimax theorem*:

$$v^* = \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2) = \min_{\sigma_2 \in \Sigma_2} \max_{\sigma_1 \in \Sigma_1} u_1(\sigma_1, \sigma_2).$$

We refer to v^* as the *value* of the game to player 1. Sometimes we will write v_i as the value of the game to player i. It is important to note that *any* equilibrium strategy for a player will guarantee an expected payoff of at least the value of the game to that player.

Define the *exploitability* of σ_i to be the difference between the value of the game and the performance of σ_i against its nemesis, formally:

$$\text{expl}(\sigma_i) = v_i - \min_{\sigma_{-i}} u_i(\sigma_i, \sigma_{-i}).$$

For any $\epsilon \geq 0$, define *SAFE(ϵ)* to be the set of strategies with exploitability at most ϵ. Define the ϵ*-safe best response* of player i to σ_{-i} to be

$$\text{argmax}_{\sigma_i \in \text{ SAFE}(\epsilon)} u_i(\sigma_i, \sigma_{-i}).$$

All finite games have at least one Nash equilibrium. In two-player zero-sum strategic-form games, a Nash equilibrium can be found efficiently by linear programming. In the case of zero-sum extensive-form games with perfect recall, there are efficient techniques for finding an equilibrium, such as linear programming [Koller et al. 1994]. An ϵ-equilibrium can be found in even larger games via algorithms such as generalizations of the excessive gap technique [Hoda et al. 2010] and counterfactual regret minimization [Zinkevich et al. 2007]. The latter two algorithms scale to games with approximately 10^{12} game tree states, while the most scalable current general-purpose linear programming technique (CPLEX's barrier method) scales to games with around 10^7 or 10^8 states. By contrast, full best responses can be computed in time linear in the size of the game tree, while the best known techniques for computing ϵ-safe best responses have running times roughly similar to an equilibrium computation [Johanson et al. 2007].

2.4. Repeated games

In repeated games, the *stage game* is repeated for a finite number T of iterations. At each iteration, players can condition their strategies on everything that has been observed so far. In strategic-form games, this generally includes the full mixed strategy of the agent in all previous iterations, as well as all actions of the opponent (though not his full strategy). In extensive-form games, generally only the actions of the opponent along the path of play are observed; in games with imperfect information, the opponent's private information may also be observed in some situations.

3. SAFETY

One desirable property of strategy for a repeated game is that it is *safe*—that it guarantees at least v_i per period in expectation. Clearly playing a minimax strategy at each iteration is safe, since it guarantees at least v_i in each iteration. However, a minimax

strategy may fail to maximally exploit a suboptimal opponent. On the other hand, deviating from stage-game equilibrium in an attempt to exploit a suboptimal opponent could lose the guarantee of safety and may result in an expected payoff below the value of the game against a deceptive opponent (or if the opponent model is incorrect).

3.1. A game in which safe exploitation is not possible

Consider the classic game of Rock-Paper-Scissors (RPS), whose payoff matrix is depicted in Figure 1. The unique equilibrium σ^* is for each player to randomize equally among all three pure strategies.

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Fig. 1. Payoff matrix of Rock-Paper-Scissors.

Now suppose that our opponent has played Rock in each of the first 10 iterations (while we have played according to σ^*). We may be tempted to try to exploit him by playing the pure strategy Paper at the 11th iteration. However, this would not be safe; it is possible that he has in fact been playing his equilibrium strategy all along, and that he just played Rock each time by chance (this will happen with probability $\frac{1}{3^{10}}$). It is also possible that he will play Scissors in the next round (perhaps to exploit the fact that he thinks we are more likely to play Paper having observed his actions). Against such a strategy, we would actually have a negative expected total profit—0 in the first 10 rounds and -1 in the 11th. Thus, our strategy would not be safe. By similar reasoning, it is easy to see that any deviation from σ^* will not be safe, and that safe exploitation is not possible in RPS.

3.2. A game in which safe exploitation is possible

Now consider a variant of RPS in which player 2 has an additional pure strategy T. If he plays T, then we get a payoff of 4 if we play R, and 3 if we play P or S. The payoff matrix of this new game RPST is given in Figure 2. Clearly the unique equilibrium is still for both players to randomize equally between R, P, and S. Now suppose we play our equilibrium strategy in the first game iteration, and the opponent plays T; no matter what action we played, we receive a payoff of at least 3. Now suppose we play the pure strategy R in the second round in an attempt to exploit him (since R is our best response to T). In the worst case, our opponent will exploit us in the second round by playing P, and we will obtain payoff -1. But combined over both time steps, our payoff will be positive no matter what the opponent does at the second iteration. Thus, our strategy constituted a safe deviation from equilibrium. This was possible because of the existence of a 'gift' strategy for the opponent; no such gift strategy is present in standard RPS.

	R	P	S	T
R	0	-1	1	4
P	1	0	-1	3
S	-1	1	0	3

Fig. 2. Payoff matrix of RPST.

4. CHARACTERIZING GIFTS

What exactly constitutes a gift? Does it have to be a strictly-dominated pure strategy, like T in the preceding example? What about weakly-dominated strategies? What about iterated dominance, or dominated mixed strategies? In this section we first provide some negative results which show that several natural candidate definitions of gifts strategies are not appropriate. Then we provide a formal definition of gifts and show that safe exploitation is possible if and only if such gift strategies exist.

Recent work has conjectured the following:

CONJECTURE 4.1. *[Waugh 2009] An equilibrium strategy makes an opponent indifferent to all non-[weakly]-iteratively-dominated strategies. That is, to tie an equilibrium strategy in expectation, all one must do is play a non-[weakly]-iteratively-dominated strategy.*

This conjecture would seem to imply that gifts correspond to strategies that put weight on pure strategies that are weakly iteratively dominated. However, consider the game shown in Figure 3.

	L	M	R
U	3	2	10
D	2	3	0

Fig. 3. A game with a gift strategy that is not weakly iteratively dominated.

It can easily be shown that this game has a unique equilibrium, in which P1 plays U and D with probability $\frac{1}{2}$, and P2 plays L and M with probability $\frac{1}{2}$. The value of the game to player 1 is 2.5. If player 1 plays his equilibrium strategy and player 2 plays R, player 1 gets expected payoff of 5, which exceeds his equilibrium payoff; thus R constitutes a gift, and player 1 can safely deviate from equilibrium to try to exploit him. But note that R is not dominated under any form of dominance. This disproves the conjecture, and causes us to rethink our notion of gifts.

PROPOSITION 4.2. *It is possible for a strategy that survives iterated weak dominance to obtain expected payoff worse than the value of the game against an equilibrium strategy.*

We might now be tempted to define a gift as a strategy that is not in the support of any equilibrium strategy.

	L	R
U	0	0
D	-2	1

Fig. 4. Strategy R is not in the support of an equilibrium for player 2, but is also not a gift.

However, the game in Figure 4 shows that it is possible for a strategy to not be in the support of an equilibrium and also not be a gift (since if P1 plays his only equilibrium strategy U, he obtains 0 against R, which is the value of the game).

Now that we have ruled out several candidate definitions of gift strategies, we now present our new definition, which we relate formally to safe exploitation in Proposition 4.4.

Definition 4.3. A strategy σ_{-i} is a *gift strategy* if there exists an equilibrium strategy σ_i^* for the other player such that σ_{-i} is not a best response to σ_i^*.

PROPOSITION 4.4. *Assuming we are not in a trivial game in which all of player i's strategies are minimax strategies, then non-stage-game-equilibrium safe strategies exist if and only if there exists at least one gift strategy for the opponent.*

PROOF. Suppose some gift strategy σ_{-i} exists for the opponent. Then there exists an equilibrium strategy σ_i^* such that $u_i(\sigma_i^*, \sigma_{-i}) > v_i$. Let $\epsilon = u_i(\sigma_i^*, \sigma_{-i}) - v_i$. Let s_i' be a non-equilibrium strategy for player i. Suppose player i plays σ_i^* in the first round, and in the second round does the following: if the opponent did not play σ_{-i} in the first round, he plays σ_i^* in all subsequent rounds. If the opponent did play σ_{-i} in the first round, then in the second round he plays $\hat{\sigma}_i$, where $\hat{\sigma}_i$ is a mixture between s_i' and σ_i^* that has exploitability in $(0, \epsilon)$ (we can always obtain such a mixture by putting sufficiently much weight on σ_i^*), and he plays σ_i^* in all subsequent rounds. Such a strategy constitutes a safe strategy that deviates from stage-game equilibrium.

Now suppose no gift strategy exists for the opponent, and suppose we deviate from equilibrium for the first time in some iteration t'. Suppose the opponent plays his nemesis strategy at time step t', and plays an equilibrium strategy at all future time steps. Then we will win less than v^* in expectation against his strategy. Therefore, we cannot safely deviate from equilibrium. □

5. SAFETY ANALYSIS OF SOME NATURAL EXPLOITATION ALGORITHMS

Now that we know it is possible to safely deviate from equilibrium in certain games, can we construct efficient procedures for implementing such safe exploitative strategies? In this section we analyze the safety of several natural exploitation algorithms. Some of the algorithms—specifically RWYWE, BEFFE, and BEFEWP—are new contributions, while the other algorithms are presented for purposes of comparison. In short, we will show that all prior algorithms and natural other candidates are all either unsafe or unexploitative; we present algorithms that are safe and exploitative.

5.1. Risk What You've Won (RWYW)

The 'Risk What You've Won' algorithm (RWYW) is quite simple and natural; essentially, at each iteration it risks only the amount of profit won so far. More specifically, at each iteration t, RWYW plays an ϵ-safe best response to a model of the opponent's strategy (according to some opponent modeling algorithm M), where ϵ is our current cumulative payoff minus $(t-1)v^*$. Pseudocode is given in Algorithm 1.

Algorithm 1 Risk What You've Won (RWYW)

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(k^t)} M(\pi)$
  Play action a_i^t according to π^t
  Update M with opponent's actions, a_{-i}^t
  $k^{t+1} \leftarrow k^t + u_i(a_i^t, a_{-i}^t) - v^*$
end for

PROPOSITION 5.1. *RWYW is not safe.*

PROOF. Consider RPS, and assume our opponent modeling algorithm M says that the opponent will play according to his distribution of actions observed so far. Since initially $k^1 = 0$, we must play our equilibrium strategy σ^* at the first iteration, since it is the only strategy with exploitability of 0. Without loss of generality, assume the

opponent plays R in the first iteration. Our expected payoff in the first iteration is 0, since σ^* has expected payoff of 0 against R (or any strategy). Suppose we had played R ourselves in the first iteration. Then we would have obtained an actual payoff of 0, and would set $k^2 = 0$. Thus we will be forced to play σ^* at the second iteration as well. If we had played P in the first round, we would have obtained a payoff of 1, and set $k^2 = 1$. We would then set π^2 to be the pure strategy P, since our opponent model dictates the opponent will play R again, and P is the unique k^2-safe best response to R. Finally, if we had played S in the first round, we would have obtained an actual payoff of -1, and would set $k^2 = -1$; this would require us to set π^2 equal to σ^*.

Now, suppose the opponent had actually played according to his equilibrium strategy in iteration 1, plays the pure strategy S in the second round, then plays the equilibrium in all subsequent rounds. As discussed above, our expected payoff at the first iteration is zero. Against this strategy, we will actually obtain an expected payoff of -1 in the second iteration if the opponent happened to play R in the first round, while we will obtain an expected of 0 in the second round otherwise. So our expected payoff in the second round will be $\frac{1}{3} \cdot (-1) + \frac{2}{3} \cdot 0 = -\frac{1}{3}$. In all subsequent rounds our expected payoff will be zero. Thus our overall expected payoff will be $-\frac{1}{3}$, which is less than the value of the game; so RWYW is not safe. □

RWYW is not safe because it does not adequately differentiate between whether profits were due to skill (i.e., from gifts) or to luck.

5.2. Risk What You've Won in Expectation (RWYWE)

A better approach than RWYW would be to risk the amount won so far *in expectation*. Ideally we would like to do the expectation over both our randomization and our opponent's, but this is not possible in general since we only observe the opponent's action, not his full strategy. However, it would be possible to do the expectation only over our randomization. It turns out that we can indeed achieve safety using this procedure, which we call RWYWE. Pseudocode is given in Algorithm 2. Here $u_i(\pi_i^t, a_{-i}^t)$ denotes our expected payoff of playing our mixed strategy π_i^t against the opponent's observed action a_{-i}^t.

Algorithm 2 Risk What You've Won in Expectation (RWYWE)

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(k^t)} M(\pi)$
  Play action a_i^t according to π^t
  The opponent plays action a_{-i}^t according to unobserved distribution π_{-i}^t
  Update M with opponent's actions, a_{-i}^t
  $k^{t+1} \leftarrow k^t + u_i(\pi_i^t, a_{-i}^t) - v^*$
end for

LEMMA 5.2. *Let π be updated according to RWYWE, and suppose the opponent plays according to π_{-i}. Then for all $n \geq 0$,*

$$E[k^{n+1}] = \sum_{t=1}^{n} u_i(\pi_i^t, \pi_{-i}^t) - nv^*.$$

PROOF. Since $k^1 = 0$, the statement holds for $n = 0$. Now suppose the statement holds for all $t \leq n$, for some $n \geq 0$. Then

$$\begin{aligned} E[k^{n+2}] &= E[k^{n+1} + u_i(\pi_i^{n+1}, a_{-i}^{n+1}) - v^*] && (1)\\ &= E[k^{n+1}] + E[u_i(\pi_i^{n+1}, a_{-i}^{n+1})] - E[v^*] && (2)\\ &= \left[\sum_{t=1}^{n} u_i(\pi_i^t, \pi_{-i}^t) - nv^*\right] + E[u_i(\pi_i^{n+1}, a_{-i}^{n+1})] - v^* && (3)\\ &= \left[\sum_{t=1}^{n} u_i(\pi_i^t, \pi_{-i}^t) - nv^*\right] + u_i(\pi_i^{n+1}, \pi_{-i}^{n+1}) - v^* && (4)\\ &= \sum_{t=1}^{n+1} u_i(\pi_i^t, \pi_{-i}^t) - (n+1)v^* && (5) \end{aligned}$$

□

LEMMA 5.3. *Let π be updated according to RWYWE. Then for all $t \geq 1$, $k^t \geq 0$.*

PROOF. By definition, $k^1 = 0$. Now suppose $k^t \geq 0$ for some $t \geq 1$. By construction, π^t has exploitability at most k^t. Thus, we must have

$$u_i(\pi_i^t, a_{-i}^t) \geq v^* - k^t.$$

Thus $k^{t+1} \geq 0$ and we are done. □

PROPOSITION 5.4. *RWYWE is safe.*

PROOF. By Lemma 5.2,

$$\sum_{t=1}^{T} u_i(\pi_i^t, \pi_{-i}^t) = E[k^{T+1}] + Tv^*.$$

By Lemma 5.3, $k^{T+1} \geq 0$, and therefore $E[k^{T+1}] \geq 0$. So

$$\sum_{t=1}^{T} u_i(\pi_i^t, \pi_{-i}^t) \geq Tv^*,$$

and RWYWE is safe. □

RWYWE is similar to the Safe Policy Selection Algorithm (SPS), proposed in [McCracken and Bowling 2004]. The main difference is that SPS uses an additional decay function $f : \mathbf{N} \to \mathbf{R}$ setting $k^1 \leftarrow f(1)$ and using the update step

$$k^{t+1} \leftarrow k^t + f(t+1) + u_i(\pi^t, a_{-i}^t) - v^*.$$

They require f to satisfy the following properties

(1) $f(t) > 0$ for all t

(2) $\lim_{T\to\infty} \frac{\sum_{t=1}^{T} f(t)}{T} = 0$

In particular, they obtained good experimental results using $f(t) = \frac{\beta}{t}$. They are able to show that SPS is safe in the limit as $T \to \infty$;[2] however SPS is arbitrarily exploitable in finitely repeated games. Furthermore, even in infinitely repeated games, SPS can

[2]We recently discovered a mistake in their proof of safety in the limit; however, the result is still correct.

lose a significant amount; it is merely the average loss that approaches zero. We can think of RWYWE as SPS but using $f(t) = 0$ for all t.

5.3. Best equilibrium strategy

Given an opponent modeling algorithm M, we could play the best Nash equilibrium according to M at each time step:

$$\pi^t = \text{argmax}_{\pi \in \text{ SAFE}(0)} M(\pi).$$

This would clearly be safe, but can only exploit the opponent as much as the best equilibrium can, and potentially leaves a lot of exploitation on the table.

5.4. Regret minimization between an equilibrium and an opponent modeling algorithm

We could use a no-regret algorithm (e.g., [Auer et al. 2002]) to select between an equilibrium and opponent modeling algorithm M at each iteration. As pointed out in [McCracken and Bowling 2004], this would be safe in the limit as $T \to \infty$. However, this would not be safe in finitely-repeated games. Note that even in the infinitely-repeated case, no-regret algorithms only guarantee that average regret goes to 0 in the limit; in fact, total regret can still grow arbitrarily large.

5.5. Regret minimization in the space of equilibria

Regret minimization in the space of equilibria is safe, but again would potentially miss out on a lot of exploitation against suboptimal opponents. This procedure was previously used to exploit opponents in Kuhn poker [Hoehn et al. 2005].

5.6. Best equilibrium followed by full exploitation (BEFFE)

The BEFFE algorithm works as follows. We start off playing the best equilibrium strategy according to some opponent model M. Then we switch to playing a full best response for all future iterations if we know that doing so will keep our strategy safe in the full game (in other words, if we know we have accrued enough gifts to support full exploitation in the remaining iterations). Pseudocode is given in Algorithm 3.

Algorithm 3 Best Equilibrium Followed by Full Exploitation (BEFFE)

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t_{BR} \leftarrow \text{argmax}_\pi M(\pi)$
  $\epsilon \leftarrow v^* - \min_{\pi_{-i}} u_i(\pi^t_{BR}, \pi_{-i})$
  if $k^t >= (T - t + 1)(v^* - \epsilon)$ **then**
    $\pi^t \leftarrow \pi^t_{BR}$
  else
    $\pi^t \leftarrow \text{argmax}_{\pi \in \text{ SAFE}(0)} M(\pi)$
  end if
  Play action a^t_i according to π^t
  The opponent plays action a^t_{-i} according to unobserved distribution π^t_{-i}
  Update M with opponent's actions, a^t_{-i}
  $k^{t+1} \leftarrow k^t + u_i(\pi^t_i, a^t_{-i}) - v^*$
end for

This algorithm is similar to the DBBR algorithm [Ganzfried and Sandholm 2011], which plays an equilibrium for some fixed number of iterations, then switches to full

exploitation. However, BEFFE automatically detects when this switch should occur, which has several advantages. First, it is one fewer parameter required by the algorithm. More importantly, it enables the algorithm to guarantee safety.

PROPOSITION 5.5. *BEFFE is safe.*

PROOF. Follows by same reasoning as proof of safety of RWYWE, since we are playing a strategy with exploitability at most k^t at each iteration. □

One possible advantage of BEFFE over RWYWE is that it potentially saves up exploitability until the end of the game, when it has the most accurate information on the opponent's strategy (while RWYWE does exploitation from the start when the opponent model has noisier data). On the other hand, BEFFE possibly misses out on additional rounds of exploitation by waiting until the end, since it may accumulate additional gifts in the exploitation phase that it did not take into account. Furthermore, by waiting longer before turning on exploitation, one's experience of the opponent can be from the wrong part of the space; that is, the space that is reached when playing equilibrium but not when exploiting. Consequently, the exploitation might not be as effective because it may be based on less data about the opponent in the pertinent part of the space. This issue has been observed in opponent exploitation in Heads-Up Texas Hold'em poker [Ganzfried and Sandholm 2011].

5.7. Best equilibrium and full exploitation when possible (BEFEWP)

BEFEWP is similar to BEFFE, but rather than waiting until the end of the game, we play a full best response at each iteration where its exploitability is below k^t; otherwise we play the best equilibrium. Pseudocode is given in Algorithm 4.

Algorithm 4 Best Equilibrium and Full Exploitation When Possible (BEFEWP)

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t_{BR} \leftarrow \text{argmax}_\pi M(\pi)$
  $\epsilon \leftarrow v^* - \text{min}_{\pi_{-i}} u_i(\pi^t_{BR}, \pi_{-i})$
  if $\epsilon <= k^t$ **then**
    $\pi^t \leftarrow \pi^t_{BR}$
  else
    $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(0)} M(\pi)$
  end if
  Play action a^t_i according to π^t
  The opponent plays action a^t_{-i} according to unobserved distribution π^t_{-i}
  Update M with opponent's actions, a^t_{-i}
  $k^{t+1} \leftarrow k^t + u_i(\pi^t_i, a^t_{-i}) - v^*$
end for

Like RWYWE, BEFEWP will continue to exploit a suboptimal opponent throughout the match provided the opponent keeps giving us gifts. It also guarantees safety, since we are still playing a strategy with exploitability at most k^t at each iteration. However, playing a full best response rather than a safe best response early in the match may not be the greatest idea, since our data on the opponent is still quite noisy.

PROPOSITION 5.6. *BEFEWP is safe.*

6. A FULL CHARACTERIZATION OF SAFE STRATEGIES IN STRATEGIC-FORM GAMES

In the previous section we saw a variety of opponent exploitation algorithms, some which are safe and some which are unsafe. In this section, we fully characterize the space of safe algorithms. Informally, it turns out that an algorithm will be safe if at each time step it selects a strategy with exploitability at most k^t, where k is updated according to the RWYWE procedure. Note that this does not mean that RWYWE is the only safe algorithm, or that safe algorithms must explicitly use the given update rule for k^t; it just means that the exploitability at each time step must be bounded by the particular value k^t, assuming that k had hypothetically been updated according to the RWYWE rule.

Definition 6.1. An algorithm for selecting strategies is *expected-profit-safe* if it satisfies the rule

$$\pi^t \in \text{SAFE}(k^t)$$

at each time step t from 1 to T, where initially $k^1 = 0$ and k is updated using the rule

$$k^{t+1} \leftarrow k^t + u_i(\pi^t, a^t_{-i}) - v^*.$$

PROPOSITION 6.2. *A strategy π (for the full game, not the stage game) is safe if and only if it is expected-profit-safe.*

PROOF. If π is expected-profit-safe, then it follows that π is safe by similar reasoning to the proof of Proposition 5.4.

Now suppose π is safe, but at some iteration t' selects $\pi^{t'}$ with exploitability exceeding $k^{t'}$, as defined in Definition 6.1 (assume t' is the first such iteration); let e' denote the exploitability of $\pi^{t'}$. Suppose the opponent had been playing the pure strategy that selects action a^t_{-i} with probability 1 at each iteration t for all $t < t'$, and suppose he plays his nemesis strategy to $\pi^{t'}$ at time step t' (and follows a minimax strategy at all future iterations). Then our expected payoff in the first t' iterations is

$$\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - e' \tag{6}$$

$$< \sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - k^{t'} \tag{7}$$

$$= \sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - \left(\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) - (t'-1)v^* \right) \tag{8}$$

$$= t'v^*. \tag{9}$$

Note that in Equation 8, we use Lemma 5.2 and the fact that $E[k^{t'}] = k^{t'}$, since the opponent played a deterministic strategy in the first $t'-1$ rounds. We will obtain payoff at most v^* at each future iteration, since the opponent is playing a minimax strategy. So π is not safe and we have a contradiction; therefore π must be expected-profit-safe, and we are done. □

7. SAFE EXPLOITATION IN SEQUENTIAL GAMES

In sequential games, we cannot immediately apply RWYWE (or the other safe algorithms that deviate from equilibrium), since we do not know what the opponent would

have done at game states off the path of play (and thus cannot evaluate the expected payoff of our mixed strategy).

7.1. Sequential games of perfect information

In sequential games of perfect information, it turns out that to guarantee safety we must assume pessimistically that the opponent is playing a nemesis off the path of play (while playing his observed action on the path of play). This pessimism potentially limits our amount of exploitation when the opponent is not playing a nemesis, but is needed to guarantee safety.

Algorithm 5 Sequential RWYWE

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(k^t)} M(\pi)$
  Play action a_i^t according to π^t
  The opponent plays action a_{-i}^t according to unobserved distribution π_{-i}^t
  Update M with opponent's actions, a_{-i}^t
  $\tau_{-i}^t \leftarrow$ strategy for the opponent that plays a_{-i}^t on the path of play, and plays a best response to π^t off the path of play
  $k^{t+1} \leftarrow k^t + u_i(\pi_i^t, \tau_{-i}^t) - v^*$
end for

PROPOSITION 7.1. *Sequential RWYWE is safe.*

PROOF. Similar to proof of Proposition 5.4. Due to space constraints, we have had to omit several of our proofs. □

We now provide a full characterization of safe exploitation algorithms in sequential games—similarly to what we did for strategic-form games earlier in the paper.

Definition 7.2. An algorithm for selecting strategies in sequential games of perfect information is *expected-profit-safe* if it satisfies the rule

$$\pi^t \in \text{SAFE}(k^t)$$

at each time step t from 1 to T, where initially $k^1 = 0$ and k is updated using the same rule as Sequential RWYWE.

PROPOSITION 7.3. *A strategy π in a sequential game of perfect information is safe if and only if it is expected-profit-safe.*

7.2. Sequential games of imperfect information

In sequential games of imperfect information, not only do we not see the opponent's action off of the path of play, but sometimes we do not even see his private information. We consider the two cases—when his private information is observed and unobserved—separately.

7.2.1. Opponent's private information is observed at the end of the game. When the opponent's private information is observed at the end of each game iteration, we can play a procedure similar to Sequential RWYWE. Here, we must pessimistically assume that the opponent would have played a nemesis at every information set off of the path of play (though we do not make any assumptions regarding his play along the path of play

other than that he played action a^t_{-i} with observed private information θ^t_{-i}). Pseudocode for this procedure is given in Algorithm 6.

Algorithm 6 Safe exploitation algorithm for sequential games of imperfect information where opponent's private information is observed at the end of the game

$v^* \leftarrow$ value of the game to player i
$k^1 \leftarrow 0$
for $t = 1$ to T **do**
  $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(k^t)} M(\pi)$
  Play action a^t_i according to π^t
  The opponent plays action a^t_{-i} with observed private information θ^t_{-i}, according to unobserved distribution π^t_{-i}
  Update M with opponent's actions, a^t_{-i}, and his private information, θ^t_{-i}
  $\tau^t_{-i} \leftarrow$ strategy for the opponent that plays a best response to π^t subject to the constraint that it plays a^t_{-i} on the path of play with private information θ^t_{-i}
  $k^{t+1} \leftarrow k^t + u_i(\pi^t_i, \tau^t_{-i}) - v^*$
end for

PROPOSITION 7.4. *Algorithm 6 is safe.*

Definition 7.5. An algorithm for selecting strategies in sequential games of imperfect information is *expected-profit-safe* if it satisfies the rule

$$\pi^t \in \text{SAFE}(k^t)$$

at each time step t from 1 to T, where initially $k^1 = 0$ and k is updated using the same rule as Algorithm 6.

PROPOSITION 7.6. *A strategy π in a sequential game of imperfect information is safe if and only if it is expected-profit-safe.*

7.2.2. Opponent's private information is not observed. Unfortunately we must be extremely pessimistic if the opponent's private information is not observed, though it can still be possible to detect gifts in some cases. We can only be sure we have received a gift if the opponent's observed action would have been a gift for any possible private information he may have. Thus we can run an algorithm similar to Algorithm 6, where we redefine τ^t_{-i} to be the opponent's best response subject to the constraint that he plays a^t_{-i} with *some* private information.

The approaches from this subsection and the previous subsection can be combined if we observe some of the opponent's private information afterwards but not all. Again, we must be pessimistic an assume he plays a nemesis subject to the restriction that we plays the observed actions with the observed part of his private information.

7.3. Detecting gifts within a game iteration

In some situations, we could detect gift actions early in the game that would allow us to risk trying to exploit him even in the middle of a single game iteration. We can use a variant of the Sequential RWYWE update rule to detect gifts during a game iteration, where we redefine τ^t_{-i} to be the opponent's best response to π^t_i subject to the constraint that he has taken the observed actions along the path of play thus far. This would allow us to safely deviate from equilibrium to exploit him even during a game iteration.

8. EXPERIMENTS

We ran experiments using the sequential imperfect-information variants of several of the safe algorithms presented in Section 5. The domain we consider is Kuhn poker [Kuhn 1950], a simplified form of poker which has been frequently used as a test problem for game-theoretic algorithms [Ganzfried and Sandholm 2010; Gordon 2005; Hawkin et al. 2011; Hoehn et al. 2005; Koller and Pfeffer 1997].

8.1. Kuhn poker

Kuhn poker is a two-person zero-sum poker game, consisting of a three-card deck and a single round of betting. Here are the full rules:

— Two players: P1 and P2
— Both players ante $1
— Deck containing three cards: K, Q, and J
— Each player is dealt one card uniformly at random
— P1 acts first and can either bet $1 or check
  — If P1 bets, P2 can call or fold
    — If P1 bets and P2 calls, then whoever has the higher card wins the $4 pot
    — If P1 bets and P2 folds, then P1 wins the entire $3 pot
  — If P1 checks, P2 can bet $1 or check.
    — If P1 checks and P2 bets, then P1 can call or fold.
      — If P1 checks, P2 bets, and P1 calls, then whoever has the higher card wins the $4 pot
      — If P1 checks, P2 bets, and P1 folds, then B wins the $3 pot
    — If P1 checks and P2 checks, then whoever has the higher card wins the $2 pot

The value of the game to player 1 is $-\frac{1}{18} \approx -0.0556$. Player 2 has a unique equilibrium strategy, while player 1 has infinitely many equilibrium strategies parameterized by a single value.

8.2. Experimental setup

We experimented using several of the safe strategies described in Section 5—RWYWE, Best equilibrium, BEFFE, and BEFEWP. For all algorithms, we used a natural opponent modeling algorithm similar to prior work [Ganzfried and Sandholm 2011; Hoehn et al. 2005]. We also compare our algorithms to a full best response using the same opponent modeling algorithm. This strategy is not safe and is highly exploitable in the worst case; but it provides a useful metric for comparison.

Our opponent model assumes the opponent plays according to his observed frequencies so far in the game, where we assume that we observe his hand at the end of each game iteration as prior work on exploitation in Kuhn poker has done [Hoehn et al. 2005]. We initialize our model by assuming a Dirichlet prior of 5 fictitious hands at each information set at which the opponent has played according to his unique equilibrium strategy.

We adapted all five algorithms to the imperfect-information setting by using the pessimistic update rule described in Algorithm 6. We ran the algorithms against four general classes of opponents. The first class of opponent selects an action uniformly at random at each information set (random opponents were used previously in Kuhn poker [Hoehn et al. 2005]). The second opponent class is also static but more sophisticated; at each information set it selects each action at each information with probability chosen randomly but within 0.2 of the equilibrium probability (recall that player 2 has a unique equilibrium strategy). Thus, these opponents play relatively close to optimally, and are perhaps more indicative of realistic suboptimal opponents. The third

class of opponents is dynamic. Opponents in this class play the first 100 hands randomly, and then play a true best response (i.e., nemesis strategy) to our player's strategy. So, after the first 100 hands, we make the opponent more powerful than any real opponent could be in practice. Finally, the fourth class is the static unique Nash equilibrium strategy of player 2.

We ran all five algorithms against (the same) opponents from each class—800 random opponents, 3800 sophisticated static opponents, 800 dynamic opponents, and 3700 equilibrium opponents. Each match against a single opponent consisted of 1000 hands, and we assume that the hands for both players were dealt identically for each of the algorithms against a given opponent (to reduce variance). For example, suppose algorithm A1 is dealt a K and opponent O is dealt a Q in the first hand of the match. Then in the runs of all other algorithms A against O, A is dealt a K and O is dealt a Q in the first hand.

8.3. Experimental results

The results from our experiments are given in Table I. Against random opponents, the ordering of the performances of the safe algorithms was RWYWE, BEFEWP, BEFFE, Best Nash (and all of the individual rankings are statistically significant using 95% confidence intervals). Against sophisticated static opponents the rankings of the algorithms' performances were identical, though the only significant result at the 95% level was that RWYWE outperformed Best Nash. (Recall that the value of the game to player 1 is $-\frac{1}{18} \approx -0.0556$, so a negative win rate is not necessarily indicative of losing). In summary, against static opponents, our most aggressive safe exploitation algorithm outperforms the other safe exploitation algorithms that either stay within equilibrium strategies or use exploitation only when enough gifts have been accrued to use full exploitation. Against the dynamic opponents, our algorithms are indeed safe as the theory predicts, while the best response algorithm does very poorly (and much worse than the value of the game). As a sanity check, the experiments show that against the equilibrium opponent, all the algorithms obtain approximately the value of the game as they should.

Table I. Win rate in $/hand of the five algorithms against opponents from each class. The $\pm$ given is the 95% confidence interval.

	Opponent			
	Random	Sophisticated static	Dynamic	Equilibrium
RWYWE	0.363 ± 0.003	-0.0104 $\pm$ 0.0013	-0.021 $\pm$ 0.003	-0.055 $\pm$ 0.001
BEFEWP	0.353 ± 0.003	-0.0111 $\pm$ 0.0013	-0.020 $\pm$ 0.003	-0.054 $\pm$ 0.001
BEFFE	0.199 ± 0.003	-0.0121 $\pm$ 0.0013	-0.041 $\pm$ 0.003	-0.054 $\pm$ 0.001
Best Nash	0.143 ± 0003	-0.0142 $\pm$ 0.0013	-0.035 $\pm$ 0.003	-0.054 $\pm$ 0.001
Best response	**0.470 $\pm$ 0.003**	**0.0545 $\pm$ 0.0014**	**-0.121 $\pm$ 0.003**	**-0.055 $\pm$ 0.001**

In some matches, RWYWE steadily accumulates gifts along the way, and k^t increases throughout the match. An example of the graph of profit and k^t for one such opponent is given in Figure 5. In this situation, the opponent is frequently giving us gifts, and we quickly start playing (and continue to play) a full best response according to our opponent model.

In other matches, k^t remains very close to 0 throughout the match, despite the fact that profits are steadily increasing; one such example is given in Figure 6. Against this opponent, we are frequently playing an equilibrium or an ϵ-safe best response for some small ϵ, and only occasionally playing a full best response. Note that k^t falling to 0 does not necessarily mean that we are losing or giving gifts to the opponent; it just means

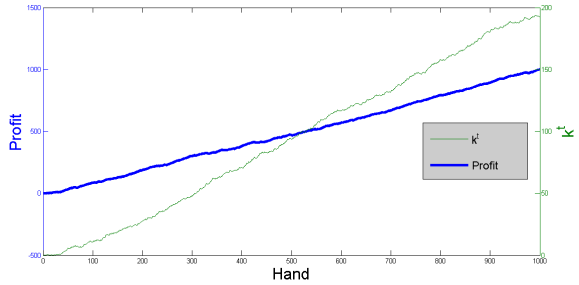

Fig. 5. Profit and k^t over the course of a match of RWYWE against a random opponent. Profits are denoted by the thick blue line using the left Y axis, while k^t is denoted by the thin green line and the right Y axis. Against this opponent, both k^t and profits steadily increase.

that we are not completely sure about our worst-case exploitability, and are erring on the side of caution to ensure safety.

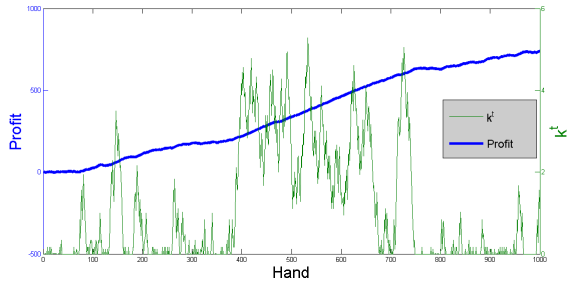

Fig. 6. Profit and k^t over the course of a match of RWYWE against a random opponent. Profits are denoted by the thick blue line using the left Y axis, while k^t is denoted by the thin green line and the right Y axis. Against this opponent, k^t stays relatively close to 0 throughout the match, while profit steadily increases.

9. CONCLUSIONS AND FUTURE RESEARCH

We showed that safe opponent exploitation is possible in certain games, disproving a recent conjecture. Specifically, profitable deviations from stage-game equilibrium are possible in games where 'gift' strategies exist for the opponent, which we define formally and fully characterize. We considered several natural opponent exploitation algorithms and showed that some guarantee safety while others do not; for example, risking the amount of profit won so far is not safe in general, while risking the amount won so far *in expectation* is safe. We described how some of these algorithms can be used to convert any opponent modeling algorithm (that is arbitrarily exploitable) into a fully safe opponent exploitation procedure. Next we provided a full characterization of safe algorithms for strategic-form games, which corresponds to precisely the algorithms that are expected-profit safe. We also provided algorithms and full characterizations of safe strategies in sequential games of perfect and imperfect information. In our experiments against static opponents, several safe exploitation algorithms significantly outperformed an algorithm that selects the best Nash equilibrium strategy; thus we conclude that safe exploitation is feasible and potentially effective in realistic settings. Our most aggressive safe exploitation algorithm outperformed the other safe exploitation algorithms that use exploitation only when enough gifts have been

accrued to use full exploitation. In experiments against an overly strong dynamic opponent that plays a nemesis strategy after 100 iterations, our algorithms are indeed safe as the theory predicts, while the best response algorithm does very poorly (and much worse than the value of the game).

Several challenges must be confronted before applying safe exploitation algorithms to larger sequential games of imperfect information, such as Texas Hold'em poker. First, the best known technique for computing ϵ-safe best responses involves solving a linear program on par with performing a full equilibrium computation; performing such computations in real time, even in a medium-sized abstracted game, is not feasible in Texas Hold'em. Perhaps the approaches of BEFEWP and BWFEE, which alternate between equilibrium and full best response, would be preferable to RWYWE in such games, since full best responses can be computed much more efficiently in practice than ϵ-safe best responses. In addition, perhaps performance can be improved if we integrate our algorithms with lower-variance estimators of our expected profits [Bowling et al. 2008].

REFERENCES

AUER, P., CESA-BIANCHI, N., FREUND, Y., AND SCHAPIRE, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing 32*.

BOWLING, M., JOHANSON, M., BURCH, N., AND SZAFRON, D. 2008. Strategy evaluation in extensive games with importance sampling. *ICML*.

GANZFRIED, S. AND SANDHOLM, T. 2010. Computing equilibria by incorporating qualitative models. *AAMAS*.

GANZFRIED, S. AND SANDHOLM, T. 2011. Game theory-based opponent modeling in large imperfect-information games. *AAMAS*.

GORDON, G. J. 2005. No-regret algorithms for structured prediction problems. Tech. Rep. CMU-CALD-05-112, Carnegie Mellon University.

HAWKIN, J., HOLTE, R., AND SZAFRON, D. 2011. Automated action abstraction of imperfect information extensive-form games. *AAAI*.

HODA, S., GILPIN, A., PEÑA, J., AND SANDHOLM, T. 2010. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research 35*.

HOEHN, B., SOUTHEY, F., HOLTE, R. C., AND BULITKO, V. 2005. Effective short-term opponent exploitation in simplified poker. *AAAI*.

JOHANSON, M., ZINKEVICH, M., AND BOWLING, M. 2007. Computing robust counter-strategies. *NIPS*.

KOLLER, D., MEGIDDO, N., AND VON STENGEL, B. 1994. Fast algorithms for finding randomized strategies in game trees. *STOC*.

KOLLER, D. AND PFEFFER, A. 1997. Representations and solutions for game-theoretic problems. *Artificial Intelligence 94*.

KUHN, H. W. 1950. Simplified two-person poker. *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker.

MCCRACKEN, P. AND BOWLING, M. 2004. Safe strategies for agent modelling in games. *AAAI Fall Symposium on Artificial Multi-agent Learning*.

POWERS, R., SHOHAM, Y., AND VU, T. 2007. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning 67*.

SANDHOLM, T. 2007. Perspectives on multiagent learning. *Artificial Intelligence 171*.

WAUGH, K. 2009. Abstraction in large extensive games. M.S. thesis, University of Alberta.

ZINKEVICH, M., BOWLING, M., JOHANSON, M., AND PICCIONE, C. 2007. Regret minimization in games with incomplete information. *NIPS*.