A

Safe Opponent Exploitation

SAM GANZFRIED, Carnegie Mellon University, Computer Science Department
TUOMAS SANDHOLM, Carnegie Mellon University, Computer Science Department

We consider the problem of playing a 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 (incorrectly) stated 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 assertion 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 extensive-form games of perfect and imperfect information, 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 stage-game equilibrium strategies only 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

ACM Reference Format:
Ganzfried, S., Sandholm, T. 2015. Safe opponent exploitation. *ACM Trans. Econ. Comp.* V, N, Article A (January 2015), 28 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

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 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

A shorter early version of this paper appeared in Proceedings of the ACM Conference on Electronic Commerce (EC), 2012. This material is based on work supported by the National Science Foundation under grant IIS-1320620, as well as XSEDE computing resources provided by the Pittsburgh Supercomputing Center.
Author's addresses: S. Ganzfried and T. Sandholm, Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA.

DOI: http://dx.doi.org/10.1145/0000000.0000000

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 conservative 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 exploitation 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 exploitation.

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

> **Is it possible to exploit the opponent more than any equilibrium strategy of a stage game would, while simultaneously guaranteeing at least the value of the full 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 stated 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 claimed 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 claim 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 exploitation architecture (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 our algorithm is

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

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 online opponent exploitation. We also present several other safe algorithms which alternate between playing an equilibrium and a best response depending on how much has been won so far in expectation. 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 extensive-form 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 an extensive-form game. This may be useful when a mistake is observed early on.

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.

The rest of the paper is organized as follows. In Section 2, we describe several alternative uses of the approach, and its applicability to more general game classes, such as infinitely-repeated, general-sum, and multi-player games. In Section 3, we present game theory background. In Section 4, we define safety and present an example of a game where safe exploitation is not possible, as well as a game where it is possible. In Section 5, we give a full characterization of when safe exploitation is possible, which turns out to coincide with games for which a gift strategy (which we define) exists for the opponent. In Section 6, we present several new algorithms for safely exploiting opponents, and show that prior algorithms are either unsafe or unexploitative. In Sections 7 and 8, we provide a full characterization of safe strategies in strategic-form and extensive-form games (of both perfect and imperfect information), respectively. In Section 9, we present experiments in an extensive-form game of imperfect-information that demonstrate that our algorithms safely exploit suboptimal opponents significantly more than repeatedly playing the best stage-game Nash equilibrium. Finally, we conclude and present future research directions in Section 10.

2. USES, APPLICABILITY, AND GENERALITY OF THE APPROACH

In this section we suggest two alternative uses of the approach, as well as discuss its applicability and generality.

2.1. Two alternative uses of the methodology

We can view safe exploitation as a meta-algorithm that enforces the safety of *any* opponent exploitation procedure by ensuring that it does not risk too much at any point. An opponent exploitation architecture consists of two components: 1) an opponent modeling algorithm, which takes as input the observations of both players' actions (to the extent that they are observable) and constructs a model of the opponent's strategy, and 2) a strategy selection algorithm, which takes the opponent model and the obser-

vations as input and outputs an exploitative strategy. This strategy may not be safe in general.

The first way to use our safe exploitation methodology is to obtain safety by curtailing the strategies that the architecture may propose. This is depicted in Figure 1.

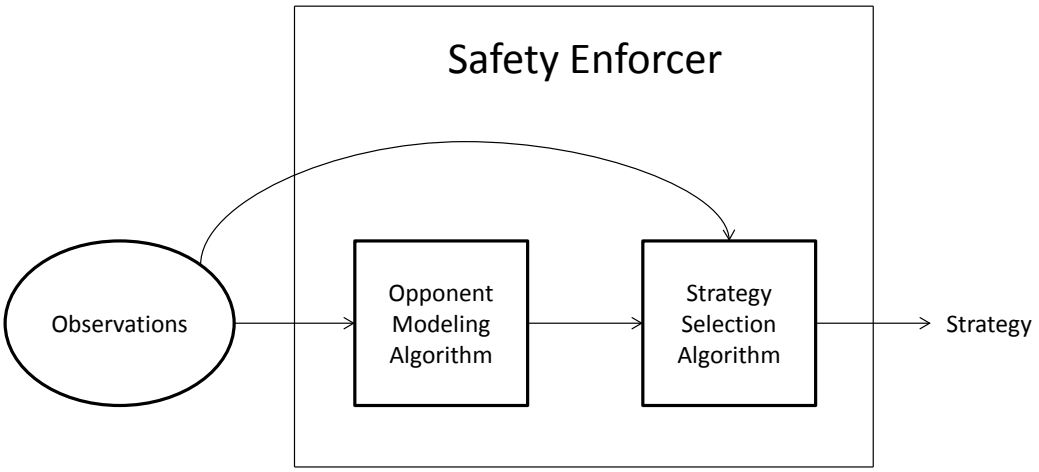

Fig. 1. Our safe exploitation methodology used as a meta-algorithm which makes any opponent exploitation architecture safe. An opponent exploitation architecture consists of two components: an opponent modeling algorithm and a strategy selection algorithm.

The second way to use the methodology is to view our safe exploitation algorithms as alternatives to standard exploitation algorithms within the opponent exploitation paradigm. Our safe algorithms still work with any opponent modeling algorithm to construct an opponent model, but replace a potentially unsafe strategy selection algorithm with a new algorithm that guarantees safety. This is depicted in Figure 2.

2.2. Bounds suffice for using the methodology

We expect our algorithms to be useful in practice in many real-world domains, for example, in (cyber)security games. It has been observed that human adversaries in such domains often behave irrationally, and there can be significant benefits to exploiting their mistakes [Blythe et al. 2011; Pita et al. 2010, 2012]. However, the cost of making a mistake ourselves is extremely high in such domains, for example, since human lives could be at stake. Algorithms that can exploit irrational opponents while still guaranteeing safety would be very desirable.

Furthermore, perhaps the main criticism of security games to date is that the numeric payoffs for the attacker and defender are questionable. Our approach does not require an exact model of the game. We only need a lower bound on the gifts (mistakes) that the opponent has given us and an upper bound on the loss from our exploitation. This would be especially useful in security games, since it guarantees robustness even when the game models are not accurate. (Another advantage is that our approach applies also to multi-step games, which are a richer, more powerful framework than the security game models used to date—Stackelberg games—where the defender moves once and then the attacker moves once.)

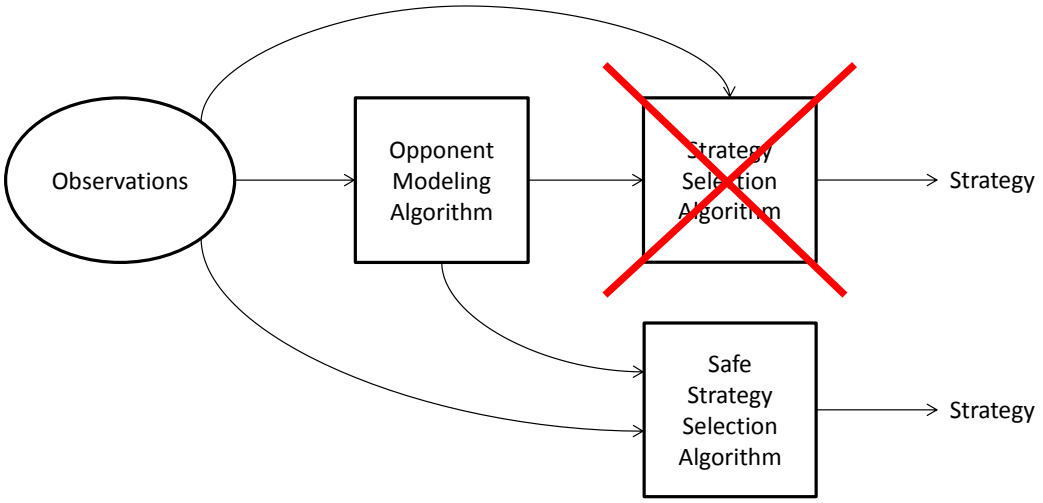

Fig. 2. Our safe exploitation methodology used to replace the strategy selection component while retaining the opponent modeling component of any opponent exploitation architecture.

2.3. The methodology also applies to infinitely repeated, general-sum, and multiplayer games

Our methodology also applies straightforwardly to two-player zero-sum infinitely repeated games. While some of our algorithms specifically depend on the finite time horizon and will not extend to the infinite setting, several of them do not, and will apply straightforwardly. In particular, the algorithm that is most aggressive (among safe algorithms) and performed best in the experiments does not rely on a finite horizon.

For general-sum and multiplayer games, our methodology applies straightforwardly if we replace the minimax value with the *maximin value* (i.e., maximizing our expected payoff minimized over the other's strategies) in our algorithms. In two-player zero-sum games, these two values coincide, and any equilibrium strategy guarantees at least this value in expectation in the worst case. In general-sum and multiplayer games, these properties do not hold; however, in many settings it could be very desirable to exploit opponents' mistakes while still guaranteeing the maximin value. For example, this could be extremely useful in security domains, which are often modeled as non-zero-sum games [Korzhyk et al. 2011]—since safety is of high importance.

2.4. Safe exploitation can be viewed as selection among equilibria of the repeated game

As we discuss in Section 4, in repeated games, the set of safe strategies is the same as the set of maximin strategies *in the repeated game* (and therefore, the set of Nash equilibria in the case where the repeated game is a two-player zero-sum game). Thus, one can view our safe exploitation algorithms as procedures for selecting among equilibria *of the repeated game*. In the context of non-repeated games, our work can be viewed as equilibrium selection in the non-repeated game. However, in both repeated and non-repeated games, as we will discuss in Section 8.3, our equilibrium refinement differs from subgame perfection [Selten 1965], and thus also from all the usual equilibrium refinements, which are further refinements of subgame perfection.

3. GAME THEORY BACKGROUND

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

3.1. 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 aka normal-form game) consists of a finite set of players N, a space of *pure strategies* S_i for each player, and a utility function $u_i : \times S_i \rightarrow \mathbb{R}$ 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. Two-player zero-sum strategic-form games are often represented as a matrix, where the element in row m column n corresponds to player 1's payoff when he plays his m-th pure strategy and player 2 plays his n-th pure strategy.

3.2. 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 nonterminal 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.

3.3. 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* [von Neumann 1928; Osborne and Rubinstein 1994]:

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

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. *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:

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

Since there always exists a nemesis that is a pure strategy, this expression is equal to $v_i - \min_{s_{-i} \in S_{-i}} u_i(\sigma_i, s_{-i})$. For any $\epsilon \geq 0$, define $\text{SAFE}(\epsilon) \subseteq \Sigma_i$ to be the set of strategies with exploitability at most ϵ. The set $\text{SAFE}(\epsilon)$ is defined by linear constraints: $\sigma_i \in \text{SAFE}(\epsilon)$ if and only if $u_i(\sigma_i, s_{-i}) \geq v_i - \epsilon$ for all $s_{-i} \in S_{-i}$. Define an *ϵ-safe best response* of player i to σ_{-i} to be any strategy in

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

All finite games have at least one Nash equilibrium [Nash 1951]. 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 10^8 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].

3.4. 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.

4. SAFETY

One desirable property of a strategy for a repeated game is that it is *safe*:

Definition 4.1. A *safe* strategy for a repeated game is a strategy that guarantees a worst-case payoff of at least v_i per period in expectation.

The set of safe strategies is the same as the set of minimax strategies in the full repeated game. Clearly playing a (stage-game) 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). Thus, a natural question to consider is whether there exist strategies that are safe, yet deviate from stage-game equilibrium strategies (in order to exploit an opponent's mistakes).

4.1. 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 3. The unique equilibrium σ^* is for each player to randomize equally among all three pure strategies.

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;

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

Fig. 3. Payoff matrix of Rock-Paper-Scissors.

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.

4.2. 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 4. 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. 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. 4. Payoff matrix of RPST.

5. 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 asserted the following:[2]

ASSERTION 1. *[Waugh 2009] An equilibrium strategy makes an opponent indifferent to all non-[weakly]-iteratively-dominated strategies. That is, to tie an equilibrium*

[2]This is made as a statement of fact in prior work [Waugh 2009], and not in the form of an assertion.

strategy in expectation, all one must do is play a non-[weakly]-iteratively-dominated strategy.

This assertion 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 5.

	L	M	R
U	3	2	10
D	2	3	0

Fig. 5. 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 R is not dominated under any form of dominance. This disproves the assertion, and causes us to rethink our notion of gifts.

PROPOSITION 5.1. *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. 6. Strategy R is not in the support of an equilibrium for player 2, but is also not a gift.

However, the game in Figure 6 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 5.3.

Definition 5.2. 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^*.[3]

[3]This definition of gift strategies coincides with the strategies for the opponent specified by the third step of a procedure for selecting a particular equilibrium of a (one-shot) two-player zero-sum game, known as Dresher's procedure [Dresher 1961; van Damme 1987]. The procedure assumes the opponent will make a mistake (i.e., by playing a gift strategy), then selects a strategy that maximizes the minimum gain resulting from a possible mistake of the opponent. It has been shown that the strategies selected by this procedure coincide with the *proper equilibria* of the game [van Damme 1987], an equilibrium refinement concept defined by Myerson [1978]. Thus, proper equilibrium strategies exploit all gift strategies, and one could equivalently define gift strategies as strategies that are not a best response to a proper equilibrium strategy of the opponent. One could view proper equilibria, as well as some other equilibrium refinement concepts (e.g., trembling-hand perfect equilibrium) as approaches for exploiting mistakes of the opponent in (non-repeated) games—although they are typically thought of as means to prescribe action probabilities

When such a strategy σ_{-i} exists, player i can win an immediate profit beyond v_i against an opponent who plays σ_{-i} by simply playing the safe strategy σ_i^*; then he can play a potentially unsafe strategy (that has exploitability below some limit) in future iterations in an attempt to exploit perceived weaknesses of the opponent. Using this definition, RPS and the game depicted in Figure 6 have no gift strategies for either player, while T is a gift for player 2 in RPST, and R is a gift for player 2 in the game depicted in Figure 5.

PROPOSITION 5.3. *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 a nemesis strategy at time step t' (to the strategy we are playing 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. □

The following procedure gives an efficient algorithm, consisting of solving two linear programs (LPs), to determine whether a gift strategy for the opponent exists in a two-player zero-sum strategic-form game (and therefore whether safe exploitation is possible).

(1) Compute an equilibrium by solving the LP; this determines the value of the game to player i, v_i.
(2) Solve the LP that maximizes the expected payoff of player i against the uniform random strategy of the opponent, subject to the constraints that player i's strategy is an equilibrium (these constraints will use v_i). Let $\hat{v}$ denote the optimal objective value of this LP.

---

for information sets that are reached with zero probability in equilibrium. In contrast, our main focus is on repeated games, although our techniques apply to single-shot games as well. Furthermore, we will show in Section 8.3 that even in single-shot games, our safe exploitative strategies differ from the strategies prescribed by subgame perfection [Selten 1965], and thus our approach differs from all prior refinements that are further refinements of subgame perfection. So, our work can be viewed as providing novel equilibrium selection concepts and procedures. In broad strokes, at every point in the game, prior refinements try to play as well as possible against an (almost) rational opponent (e.g., one who "trembles" with small probability), while ours exploits an opponent model (which does not have to be rational in any way) as much as possible subject to safety. So, our approach can exploit the opponent significantly more than prior equilibrium refinements. (Some of the prior refinements also assume that we will "tremble" with small probability ourselves; this is not motivated by exploitation, but rather so that we know how to respond to actions further down the tree at information sets that would otherwise be reached with probability zero.) Another difference is that in our technique, a safe, maximally exploitative strategy can be computed in polynomial time both in theory and practice. In contrast, while proper equilibrium strategies can be computed in polynomial time in theory for both strategic-form and extensive-form games, those polynomial-time algorithms are numerically unstable in practice [Miltersen and Sørensen 2006, 2008].

(3) If $\hat{v} > v_i$, then at least one gift strategy for the opponent exists; otherwise no gift strategies exist.

PROPOSITION 5.4. *The above procedure determines in polynomial time whether a gift strategy for the opponent exists in a given two-player zero-sum game.*

PROOF. Suppose a gift strategy s_{-i} for the opponent exists. Then there exists an equilibrium strategy σ_i^* such that $u_i(\sigma_i^*, s_{-i}) > v_i$. For every other strategy t_{-i} for the opponent, we have $u_i(\sigma_i^*, t_{-i}) \geq v_i$. Thus, player i's expected payoff of playing σ_i^* against the uniform random strategy will strictly exceed v_i, and $\hat{v} > v_i$.

Now suppose no gift strategies exist. Then for all equilibrium strategies σ_i^* and all strategies s_{-i} for the opponent, we have $u_i(\sigma_i^*, s_{-i}) = v_i$. Thus, all equilibrium strategies will obtain expected payoff v_i against the uniform random strategy, and we have $\hat{v} = v_i$.

The procedure is polynomial time since it consists of solving LPs of polynomial size (the LP formulations for computing a best response as well as the equilibrium constraints are described by, for example, Koller et al. [1994]). □

6. 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. In short, we will show that all prior algorithms and natural other candidate algorithms are either unsafe or unexploitative. We introduce algorithms that are safe and exploitative.

6.1. 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**
  $\pi^t \leftarrow \text{argmax}_{\pi \in \text{SAFE}(\max\{k^t, 0\})} M(\pi)$
  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 6.1. *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.

6.2. 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. For example, suppose we play according to the equilibrium σ^* at one iteration of RPS, and end up selecting action R, while the opponent selects action P; then our actual payoff is -1, but our expected payoff (over our own randomization) is 0. 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. The difference between RWYWE and RWYW is in the step for updating k^t: RWYW uses $u_i(a_i^t, a_{-i}^t)$ while RWYWE uses $u_i(\pi_i^t, a_{-i}^t)$.

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 6.2. *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

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

□

LEMMA 6.3. *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 6.4. *RWYWE is safe.*

PROOF. By Lemma 6.2,

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

By Lemma 6.3, $k^{T+1} \geq 0$, 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) [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 $T \to \infty$;[4] however SPS is arbitrarily exploitable

[4]We recently discovered a mistake in their proof of safety in the limit; however, the result is still correct. A corrected proof is available at http://webdocs.cs.ualberta.ca/~bowling/papers/04aaai-fallsymp-errata.pdf.

in finitely repeated games. Furthermore, even in infinitely repeated games, SPS can 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.

6.3. 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.

6.4. Regret minimization between an equilibrium and an opponent exploitation algorithm

We could use a no-regret algorithm (e.g., Exp3 [Auer et al. 2002]) to select between an equilibrium and an (unsafe) opponent exploitation algorithm at each iteration. As prior work has pointed out [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. 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.

6.5. 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].

6.6. 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). Specifically, we play a full best response at time step t if the amount of gifts we have accumulated, k^t, is at least $(T-t+1)(v^* - \epsilon)$, where ϵ is the exploitability of a full best response. Otherwise, we play the best equilibrium. Pseudocode is given in Algorithm 3.

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 6.5. *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

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**
  $\pi^t_{BR} \leftarrow \text{argmax}_\pi M(\pi)$
  $\epsilon \leftarrow v^* - \text{min}_{\pi_{-i}} u_i(\pi^t_{BR}, \pi_{-i})$
  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

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].

6.7. 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 6.6. *BEFEWP is safe.*

7. 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. 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.[5]

Definition 7.1. 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 7.2. *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 6.4.

Now suppose π is safe, but at some iteration t' selects $\pi^{t'}$ with exploitability exceeding $k^{t'}$, as defined in Definition 7.1 (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

$$\begin{aligned}
&\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - e' \\
<\ &\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - k^{t'} \\
=\ &\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) + v^* - \left(\sum_{t=1}^{t'-1} u_i(\pi^t, a^t_{-i}) - (t'-1)v^* \right) \qquad (1) \\
=\ &t'v^*.
\end{aligned}$$

In Equation 1, we use Lemma 6.2 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

[5]We could generalize the approaches to play strategies in SAFE($f(k^t)$) at each time step rather than SAFE(k^t), where $f(k^t) \leq k^t$ is an arbitrary function that is a potentially lower upper bound on the exploitability. This would result in a larger worst-case payoff guarantee when $f(k^t) < k^t$, but potentially at the expense of exploitation (since we are now restricting our space of strategies to a smaller set). In the opposite direction, we could also select strategies in SAFE($k^t + \delta$) for $\delta > 0$; this would lead to strategies that are approximately safe (within an additive factor δ), and potentially achieve higher levels of exploitation.

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. □

8. SAFE EXPLOITATION IN EXTENSIVE-FORM GAMES

In extensive-form 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).

8.1. Extensive-form games of perfect information

In extensive-form 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. We present an extensive-form version of RWYWE below as Algorithm 5. As in the strategic-form case, the time step t refers to the iteration of the repeated game (not to the depth of the tree within a single iteration); the strategies refer to behavioral strategies for a single iteration of the full extensive-form game.

Algorithm 5 Extensive-Form 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

LEMMA 8.1. *Let π be updated according to Extensive-Form RWYWE, and suppose the opponent plays according to π_{-i}. Then for all $n \geq 0$,*

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

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

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

$$\leq \left[\sum_{t=1}^{n} u_i(\pi_i^t, \pi_{-i}^t) - nv^*\right] + u_i(\pi_i^{n+1}, \pi_{-i}^{n+1}) - v^*$$
$$= \sum_{t=1}^{n+1} u_i(\pi_i^t, \pi_{-i}^t) - (n+1)v^*$$

□

LEMMA 8.2. *Let π be updated according to Extensive-Form 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

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

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

PROPOSITION 8.3. *Extensive-Form RWYWE is safe.*

PROOF. By Lemma 8.1,

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

By Lemma 8.2, $k^{T+1} \geq 0$, and therefore $E[k^{T+1}] \geq 0$. So

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

and Extensive-Form RWYWE is safe. □

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

Definition 8.4. An algorithm for selecting strategies in extensive-form 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 Extensive-Form RWYWE.

LEMMA 8.5. *Let π be updated according to Extensive-Form RWYWE, and suppose the opponent plays according to $\pi_{-i} = \tau_{-i}$, where τ_{-i} is defined in Algorithm 5. 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 $t = 0$. Now suppose the statement holds for all $t \leq n$, for some $n \geq 0$. Then

$$E[k^{n+2}] = E[k^{n+1} + u_i(\pi_i^{n+1}, \tau_{-i}^{n+1}) - v^*]$$
$$= E[k^{n+1}] + E[u_i(\pi_i^{n+1}, \tau_{-i}^{n+1})] - E[v^*]$$

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

□

PROPOSITION 8.6. *A strategy π in an extensive-form game of perfect information 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 8.3.

Now suppose π is safe, but at some iteration t' selects $\pi^{t'}$ with exploitability exceeding $k^{t'}$, as defined in Definition 8.4; let e' denote the exploitability of $\pi^{t'}$. Suppose the opponent had been playing the pure strategy that selects action a_{-i}^t with probability 1 at each iteration t for all $t < t'$, and suppose he plays his nemesis strategy at time step t' (and follows a minimax strategy at all future iterations). Then our expected payoff is

$$
\begin{aligned}
&\sum_{t=1}^{t'-1} u_i(\pi^t, a_{-i}^t) + v^* - e' \\
&< \sum_{t=1}^{t'-1} u_i(\pi^t, a_{-i}^t) + v^* - k^{t'} \\
&= \sum_{t=1}^{t'-1} u_i(\pi^t, a_{-i}^t) + v^* - \left(\sum_{t=1}^{t'-1} u_i(\pi^t, a_{-i}^t) - (t'-1)v^*\right) \\
&= t'v^*.
\end{aligned}
$$

In Equation 2, we use Lemma 8.5 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 profit-safe, and we are done. □

8.2. Extensive-form games of imperfect information

In extensive-form 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. For example, in an auction we may not see the opponent's valuation, and in a poker hand we will not see the opponent's private cards if he folds (while we will see them if neither player folds during the hand). The extent to which his private information is revealed will in general depend on the rules and information structure of the game. We consider the two cases—when his private information is observed and unobserved—separately.

8.2.1. Setting where the 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 Extensive-Form 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 extensive-form 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 8.7. *Algorithm 6 is safe.*

PROOF. Follows by identical reasoning to the proof of Proposition 8.3, using the new definition of τ. □

Definition 8.8. An algorithm for selecting strategies in extensive-form 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 8.9. *A strategy π in an extensive-form game of imperfect information is safe if and only if it is expected-profit-safe.*

PROOF. Follows by similar reasoning to the proof of Proposition 8.6, using the new definition of τ. □

8.2.2. Setting where the 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.

8.3. Gift detection and exploitation within a game iteration

In some situations, we can detect gift actions early in the game that enable us to do safe exploitation even in the middle of a single game iteration. For example, an opponent may make a bet size known to be suboptimal early in a poker hand.

As a second, concrete example, consider the extensive-form game where players play the game depicted in Figure 7 followed by the game depicted in Figure 8. (The second game is the first game with all payoffs doubled, so the extensive-form game is not quite the same as just repeating the first stage game twice.) The unique stage-game equilibrium for both rounds is for P1 to play up (U and u) and for P2 to play left (L and ℓ). Down is strictly dominated for P1, and is therefore a gift. P2 can exploit this gift by playing r in the second round if he observes that player 1 has played D in the first round (since r outperforms ℓ against d). If P1 does in fact play D in the first round, P2 gains at least 3, and P2 will risk at most 2 by playing r in the second round; so this exploitation would be safe. The extensive-form representation is given in Figure 9. All subgame perfect equilibrium strategies [Selten 1965] for P2 involve him playing ℓ (ℓ1/ℓ2/ℓ3/ℓ4), while there exist safe exploitative strategies that put positive weight on r3 and r4. Since subgame perfect equilibrium is the coarsest of the traditional equilibrium refinements, this example demonstrates that our approach provides a new equilibrium refinement that differs from all the traditional ones.

	L	R
U	4	5
D	1	0

Fig. 7. Payoff matrix for first stage game of extensive-form game where we can detect and exploit a gift within a game iteration.

	ℓ	r
u	8	10
d	2	0

Fig. 8. Payoff matrix for second stage game of extensive-form game where we can detect and exploit a gift within a game iteration.

In general, one can use a variant of the Extensive-Form 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 allows us to safely deviate from equilibrium to exploit him even during a game iteration.

9. EXPERIMENTS

We ran experiments using the extensive-form imperfect-information variants of several of the safe algorithms presented in Section 6. 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].

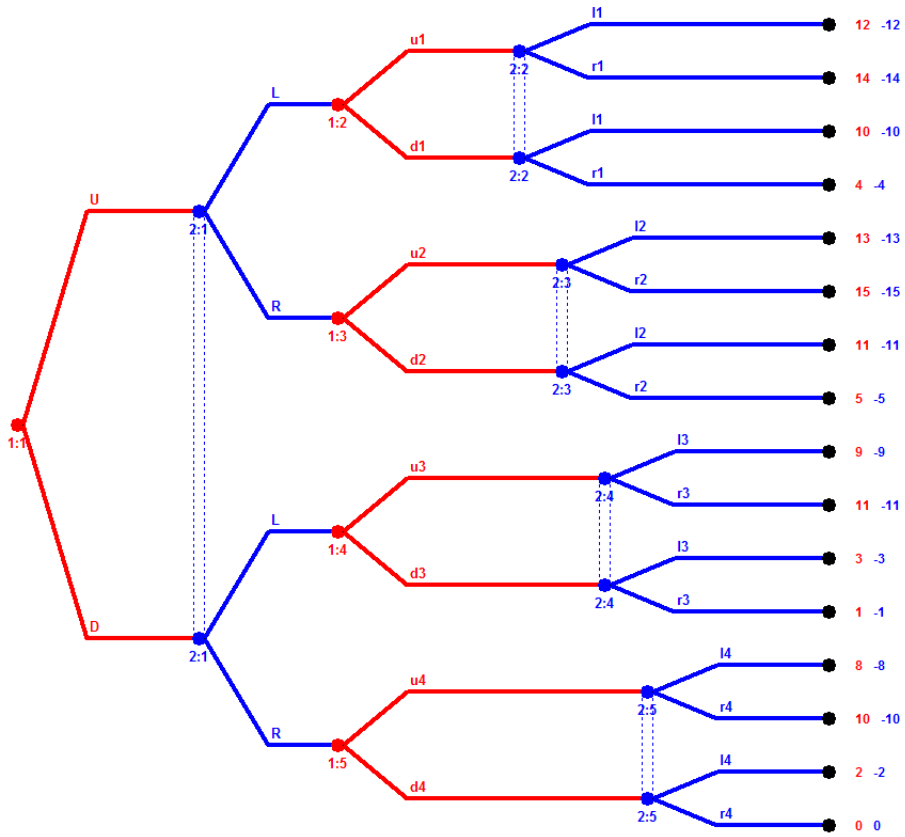

Fig. 9. Extensive-form representation of a game where we can detect and exploit a gift within a game iteration. *X*:*Y* denotes the *Y*th information set of Player *X*. Dotted lines tie together nodes within an information set. At leaves of the game tree, the payoff of Player 1 is listed first, followed by the payoff of Player 2.

9.1. 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 $-\frac{1}{18} \approx -0.0556$. For any $0 \leq \alpha \leq 1$ the following strategy profile is an equilibrium (and these are all the equilibria) [Kuhn 1950].

— P1 bets with a J in the first round with probability $\frac{\alpha}{3}$
— P1 always checks with a Q in the first round
— P1 bets with a K in the first round with probability α
— If P1 bets in the first round, then:
  — P2 always folds with a J
  — P2 calls with a Q with probability $\frac{1}{3}$
  — P2 always calls with a K
— If P1 checks in the first round, then:
  — P2 bets with a J with probability $\frac{1}{3}$
  — P2 always checks with a Q
  — P2 always bets with a K
— If P1 checks and P2 bets, then:
  — P1 always folds with a J
  — P1 calls with a Q with probability $\frac{\alpha}{3} + \frac{1}{3}$
  — P1 always calls with a K

Note that player 2 has a unique equilibrium strategy, while player 1 has infinitely many equilibrium strategies parameterized by a single value (α). In our experiments, we play the role of player 1 while the opponent plays the role of player 2.

Player 2 has four actions that are played with probability zero in his equilibrium strategy. These actions are 1) calling a bet with a J, 2) folding to a bet with a K, 3) checking a K if player 1 checks, and 4) betting a Q if player 1 checks. The first three of these are dominated, while the fourth is iteratively dominated. In this game, it turns out that the gift strategies for player 2 are exactly the strategies that play at least one of these four actions with positive probability.

9.2. Experimental setup

We experimented using several of the safe strategies described in Section 6—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, 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, as prior work in Texas Hold'em has done [Ganzfried and Sandholm 2011].

We adapted all five algorithms to the imperfect-information setting by using the pessimistic update rule described in Algorithm 6. To compute ϵ-safe best responses, which is a subroutine in several of the algorithms, we used the procedure described in Section 9.3. We ran the algorithms against four general classes of opponents.

— The first class of opponent chooses a mixed strategy in advance that selects an action uniformly at random at each information set, then follows this strategy for all

game iterations. (Similar random opponents were used also in prior work when experimenting on Kuhn poker [Hoehn et al. 2005]).
— The second opponent class is also static but more sophisticated. At each information set the opponent selects each action with probability chosen uniformly randomly 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. As in the first class, the strategy is chosen in advance, and played in all iterations.
— The third class of opponents is dynamic. Opponents in this class play the first 100 hands according to a uniform random mixed strategy that is chosen in advance, then play a true best response (i.e., nemesis strategy) to our player's strategy for the remainder of the match. So, after the first 100 hands, we make the opponent more powerful than any real opponent could be in practice, by assuming that the opponent knows our mixed strategy for that iteration.
— Finally, the fourth class is the static unique Nash equilibrium strategy of player 2.

We ran all five algorithms against the same 40,000 opponents from each class. (For the dynamic opponents, this means that we selected 40,000 different choices of the mixed strategy σ' that is played for the initial 100 iterations; for each of these choices, we ran each of the five algorithms against an opponent algorithm that uses σ' for the first 100 iterations, followed by a best response to our strategy for the next 900 iterations.) Each match against a single opponent consisted of 1,000 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. The 95% confidence intervals are reported for all experiments.

9.3. Algorithm for computing safe best responses in extensive-form games

The following LP [Koller et al. 1994] efficiently computes a best response for player 1 to a given strategy y of player 2 in a two-player zero-sum extensive-form game of imperfect information. This algorithm utilizes the sequence form representation of strategies and runs in polynomial time.

$$\begin{array}{ll} \text{maximize}_x & x^T A y \\ \text{subject to} & x^T E^T = e^T \\ & x \geq 0 \end{array}$$

We modify this procedure as follows to compute an ϵ-safe best response for player 1 to strategy y of player 2, where v_1 is the value of the game to player 1 (and all matrices and vectors are as defined by Koller et al. [1994]). This new formulation is used as a subroutine in several of the algorithms in the experiments.

$$\begin{array}{ll} \text{maximize}_x & x^T A y \\ \text{subject to} & x^T E^T = e^T \\ & x \geq 0 \\ & x^T A \geq -qF \\ & q[0] = \epsilon - v_1 \end{array}$$

9.4. 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 Equilibrium (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, and all results are statistically significant except for the difference between RWYWE and BEFEWP. (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, and furthermore all of our new algorithms outperform Best Equilibrium (which plays the best stage game equilibrium strategy at each iteration). 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	0.3636 ± 0.0004	-0.0110 ± 0.0004	-0.02043 ± 0.00044	-0.0556 ± 0.0004
BEFEWP	0.3553 ± 0.0004	-0.0115 ± 0.0004	-0.02138 ± 0.00045	-0.0556 ± 0.0004
BEFFE	0.1995 ± 0.0004	-0.0131 ± 0.0004	-0.03972 ± 0.00044	-0.0556 ± 0.0004
Best Equilibrium	0.1450 ± 0.0004	-0.0148 ± 0.0004	-0.03522 ± 0.00044	-0.0556 ± 0.0004
Best response	$\mathbf{0.4700 \pm 0.0004}$	$\mathbf{0.0548 \pm 0.0004}$	$\mathbf{-0.12094 \pm 0.00039}$	$\mathbf{-0.0556 \pm 0.0004}$

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

10. CONCLUSIONS AND FUTURE RESEARCH

We showed that safe opponent exploitation is possible in certain games, disproving a recent (incorrect) statement. Specifically, profitable deviations from stage-game equilibrium are possible in games where 'gift' strategies exist for the opponent, which we defined formally and fully characterized. 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 exploitation architecture into a safe one. 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

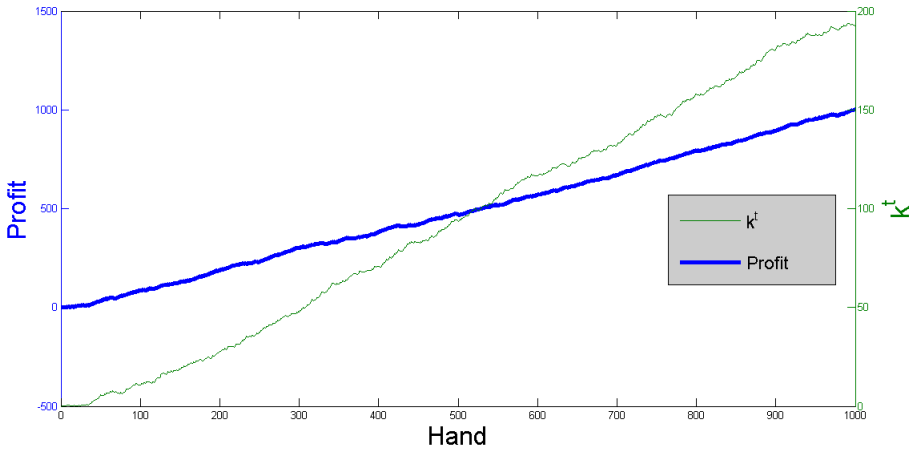

Fig. 10. Profit and k^t over the course of a match of RWYWE against a random opponent. Profits are denoted by the thick blue line using the left Y axis, while k^t is denoted by the thin green line and the right Y axis. Against this opponent, both k^t and profits steadily increase.

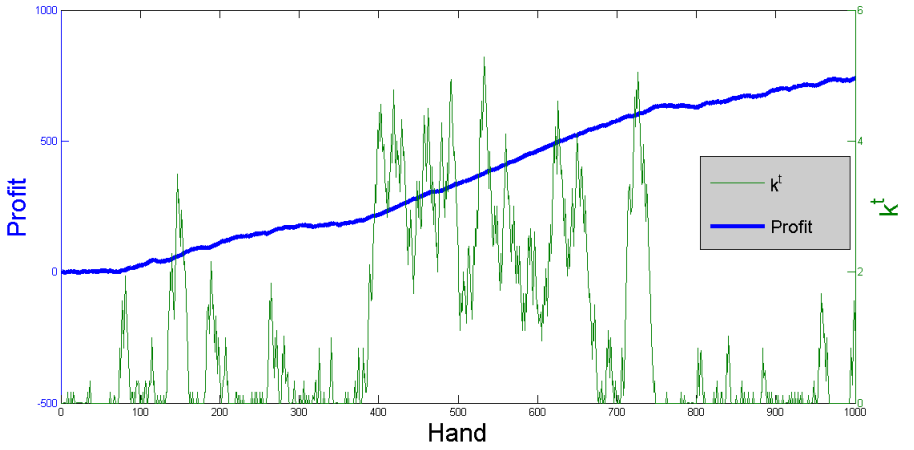

Fig. 11. Profit and k^t over the course of a match of RWYWE against a random opponent. Profits are denoted by the thick blue line using the left Y axis, while k^t is denoted by the thin green line and the right Y axis. Against this opponent, k^t stays relatively close to 0 throughout the match, while profit steadily increases.

algorithms and full characterizations of safe strategies in extensive-form 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).

The approach can also be used in settings where we do not have an exact game model—such as in (cyber)security games—because we only need to lower bound the gifts that the opponent has given us and upper bound the maximum expected loss from the exploitative action we are planning to take currently.

Several challenges must be confronted before applying safe exploitation algorithms to larger extensive-form 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 a full best response can be computed much more efficiently in practice than an ϵ-safe best response. In addition, perhaps performance can be improved if we integrate our algorithms with lower-variance estimators of our winnings due to the opponent's mistakes [Bowling et al. 2008; Zinkevich et al. 2006; White and Bowling 2009].

REFERENCES

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal of Computing* 32 (2002), 48–77.

Jim Blythe, Aaron Botello, Joseph Sutton, David Mazzaco, Jerry Lin, Marc Spraragen, and Mike Zyda. 2011. Testing Cyber Security with Simulated Humans. In *Innovative Applications of Artificial Intelligence (IAAI)*. 1622–1627.

Michael Bowling, Michael Johanson, Neil Burch, and Duane Szafron. 2008. Strategy Evaluation in Extensive Games with Importance Sampling. In *Proceedings of the International Conference on Machine Learning (ICML)*. 72–79.

Melvin Dresher. 1961. *Games of Strategy: Theory and Applications*. Prentice Hall.

Sam Ganzfried and Tuomas Sandholm. 2010. Computing equilibria by incorporating qualitative models. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Sam Ganzfried and Tuomas Sandholm. 2011. Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Geoffrey J. Gordon. 2005. *No-regret algorithms for structured prediction problems*. Technical Report CMU-CALD-05-112. Carnegie Mellon University.

John Hawkin, Robert Holte, and Duane Szafron. 2011. Automated action abstraction of imperfect information extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 681–687.

Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. 2010. Smoothing Techniques for Computing Nash Equilibria of Sequential Games. *Mathematics of Operations Research* 35, 2 (2010), 494–512.

Bret Hoehn, Finnegan Southey, Robert C. Holte, and Valeriy Bulitko. 2005. Effective Short-Term Opponent Exploitation in Simplified Poker. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 783–788.

Michael Johanson, Martin Zinkevich, and Michael Bowling. 2007. Computing Robust Counter-Strategies. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 1128–1135.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. 1994. Fast algorithms

for finding randomized strategies in game trees. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*. 750–760.

Daphne Koller and Avi Pfeffer. 1997. Representations and Solutions for Game-Theoretic Problems. *Artificial Intelligence* 94, 1 (July 1997), 167–215.

Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. 2011. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *Journal of Artificial Intelligence Research* 41 (2011), 297–327.

H. W. Kuhn. 1950. A Simplified Two-Person Poker. In *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker (Eds.). Annals of Mathematics Studies, 24, Vol. 1. Princeton University Press, Princeton, New Jersey, 97–103.

Peter McCracken and Michael Bowling. 2004. Safe strategies for agent modelling in games. In *AAAI Fall Symposium on Artificial Multi-agent Learning*.

Peter Bro Miltersen and Troels Bjerre Sørensen. 2006. Computing Proper Equilibria of Zero-Sum Games. In *Computers and Games*. 200–211.

Peter Bro Miltersen and Troels Bjerre Sørensen. 2008. Fast Algorithms for Finding Proper Strategies in Game Ttrees. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 874–883.

Roger B. Myerson. 1978. Refinements of the Nash equilibrium concept. *International Journal of Game Theory* 15 (1978), 133–154.

John Nash. 1951. Non-cooperative games. *Annals of Mathematics* 54 (1951), 289–295.

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

James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. 2010. Robust Solutions to Stackelberg Games: Addressing Bounded Rationality and Limited Observations in Human Cognition. *Artificial Intelligence Journal* 174, 15 (2010), 1142–1171.

James Pita, Richard John, Rajiv Maheswaran, Milind Tambe, and Sarit Kraus. 2012. A Robust Approach to Addressing Human Adversaries in Security Games. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. 660–665.

Rob Powers, Yoav Shoham, and Thuc Vu. 2007. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning* 67, 1-2 (2007), 45–76.

Tuomas Sandholm. 2007. Perspectives on Multiagent Learning. *Artificial Intelligence* 171 (2007), 382–391.

Reinhard Selten. 1965. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft* 12 (1965), 301–324.

Eric van Damme. 1987. *Stability and Perfection of Nash Equilibrium*. Springer-Verlag.

John von Neumann. 1928. Zur Theorie der Gesellschaftsspiele. *Math. Ann.* 100 (1928), 295–320.

Kevin Waugh. 2009. *Abstraction in Large Extensive Games*. Master's thesis. University of Alberta.

Martha White and Michael Bowling. 2009. Learning a Value Analysis Tool For Agent Evaluation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 1976–1981.

Martin Zinkevich, Michael Bowling, Nolan Bard, Morgan Kan, and Darse Billings. 2006. Optimal unbiased estimators for evaluating agent performance. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 573–578.

Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 905–912.