Learning in games with more than two players

Thuc Vu[*], Rob Powers, Yoav Shoham
Stanford University
{thucvu, powers, shoham}@cs.stanford.edu

ABSTRACT

We address the problem of learning in repeated N-player (as opposed to 2-player) general-sum games. We describe an extension to existing criteria focusing explicitly on such settings. While there have been several criteria proposed recently for evaluating learning algorithms in multi-agent systems, most of this work has focused on the two-player setting. Relatively little work has addressed situations in which there are a mixture of several agents using the algorithm in consideration against opponents using other algorithms. Roughly speaking, our proposed criteria require that the agents employing the particular learning algorithm work together to achieve a joint best-response against a target class of opponents, while guaranteeing they each achieve at least their individual security-level payoff against any possible set of opponents outside this target class. We then provide algorithms that provably meet these criteria for two target classes: stationary strategies and adaptive strategies with a bounded memory. We also demonstrate that the algorithm for stationary strategies outperforms existing algorithms in tests spanning a wide variety of repeated games with more than two players.

1. INTRODUCTION

Recently there have been several proposals for criteria with which to evaluate learning algorithms in multi-agent environments and corresponding algorithms achieving these criteria [2, 7, 22, 21]. However, these approaches focus primarily on settings with two agents and do not adequately address the different situations arising from multiple opponents. While other research has focused on situations with more than two agents, it has for the most part either concentrated on games with common payoffs or assumed no cooperation is possible and focused on no-regret payoff guarantees for each agent. In this paper we will try to address how an agent should behave in general-sum games in which the

---

[*] Currently a student

opponents can be either cooperative or adversarial. Following the approach of [22], we propose new criteria focusing on payoff guarantees for the agent against various classes of opponents:

Targeted Group Optimality: *The payoffs achieved by all the agents using this algorithm are at least within ϵ of being Pareto-efficient over the set of possible outcomes given the actual strategies of any agents that are members of the target set.*

Safety: *Against any combination of opponents, the algorithm always receives at least within ϵ of its security value for the game.*

In the rest of this paper we will first start with a brief review of related work on learning in multi-agent systems, focusing on the limitations of current approaches. Building on this we detail our proposal for a set of desirable criteria for agents in n-player repeated games and confront some of the thorny issues that arise when playing against a mix of different opponent types during a single game. Next, we outline a novel algorithm achieving these criteria and show encouraging empirical results against a number of existing algorithms in a wide range of multi-agent environments. In the final section, we'll point out some of the successes and limitations of our approach and pose some open questions for future work in this domain.

Throughout this paper, we will focus our attention on the class of repeated games with average reward. In this setting the players repeatedly play a simultaneous move normal form game, represented as a tuple, $G = (n, A, R_{1...n})$, where n is the number of players, $A = A_1 \times ... \times A_n$, where A_i is the set of actions for player i, and $R_i : A \to \Re$ is the reward function for agent i. We also use m to denote the maximum number of actions for an agent in G, and $|A|$ the number of stage-game outcomes in G. After each round, the agents accumulate their reward from the joint outcome and get to observe the prior actions of the other agents. Each agent is assumed to be trying to maximize its average reward for games with finite repetitions or the limit average for infinitely repeated games. For our purposes we assume that the full game structure and payoffs are known to all agents from the start of the game and the payoffs in the game are bounded.

2. RELATED WORK

One of our main objectives in this paper is to develop a set of criteria for learning algorithms in repeated games with an arbitrary number of players. In reviewing existing work, we start with work focusing explicitly on games with more than

two players. Following this we consider proposals of criteria for learning algorithms and see how well they generalize to games with more than two players.

Many researchers have focused on the problem of coordinating multiple agents to achieve mutually beneficial outcomes, but have for the most part restricted their attention to team games [16, 15, 24, 5]. In a team game, all the agents get identical payoffs for each action, so the challenge is focused entirely on how the agents can independently coordinate on an optimal equilibrium. Additional related work has been carried out in the areas of multi-robot planning and multi-agent pursuit games, but most of these approaches assume explicit communication or sharing of information between the cooperative agents.

Bowling and Veloso[2] were among the first to propose specific requirements for effective learning in multi-agent systems with their two criteria of rationality and convergence:

Rationality: *If the other players' policies converge to stationary policies then the learning algorithm will converge to a stationary policy that is a best-response (in the stage game) to the other players' policies.*

Convergence: *The learner will necessarily converge to a stationary policy.*

While these criteria are suitably general to apply to games with any number of players, the WoLF algorithm they proposed is only guaranteed to achieve these criteria in two player games. Conitzer and Sandholm[7] then proposed a new algorithm that achieved the above criteria for arbitrary repeated games, but only when all the opponents were of the same type (either all stationary or all self-play). Their work provides no guarantees for other cases such as when two agents use their algorithm and a third agent is stationary.

Moreover, as we pointed out in [22], there are limitations to these criteria. The first limitation is that the property of convergence cannot be applied unconditionally, since one cannot ensure that a learning procedure converges against all possible opponents in finite time without sacrificing rationality. So implicit in that requirement is some restrictions on the class of opponents. And indeed both [3] and [7] acknowledge this and choose to concentrate on the case of self-play, in which the opponents are identical to the agent in question. There are no requirements for the learning algorithm when there is a mixture of self-agents and the others.

The second limitation is that the requirement of convergence to a stationary strategy is particularly hard to justify. When combined with the requirement to play a best response to any stationary opponent, this requires the agents to converge to playing a Nash equilibrium of the stage game. While at first glance this may seem desirable, consider the game of Prisoner's Dilemma. Any algorithm satisfying the above criteria will be forced to Defect at each period in order to arrive at the unique Nash equilibrium. In the repeated game, however, two agents could instead use a strategy such as Tit-for-Tat, to achieve a much higher reward for each agent without providing the opponent with an incentive to deviate. (Tit-for-Tat starts by cooperating and thereafter repeats whatever action the opponent played last.)

Brafman and Tennenholtz addressed this problem directly in [4] and made a counter-proposal for how to consider equilibria in repeated games. They require that the learning algorithms form an Efficient Learning Equilibrium (ELE) in which any agent deviating from its algorithm will suffer a net loss of payoff within a polynomial number of stage games. They also propose an ELE algorithm based on the folk theorem that satisfies this requirement for 2-player repeated game in a perfect monitoring setting. However, once generalized to games with more than two players, the ELE algorithm requires a communication mechanism outside the game or is only applicable for a restricted number of games.

Game theory also addressed the issue of reasonable criteria for learning in multi-agent systems at numerous times with the proposals of universal consistency, no-regret learning, and the Bayes envelope dating back to at least [11] (see [9] for an overview of this history). There is a fundamental similarity in approach throughout, and we will take the approach of Fudenberg and Levine in [10] as being representative. They first proposed two criteria:

Safety: *The learning rule must guarantee at least the minimax payoff of the game.*

Consistency: *The learning rule must guarantee that it does at least as well as the best response (in the stage game) to the empirical distribution of play when playing against an opponent whose play is governed by independent draws from any fixed distribution.*

They then defined *universal consistency* as the requirement that a learning rule do at least as well as the best response to the empirical distribution of play regardless of the actual strategy the opponent is employing (this implies both safety and consistency) and propose an algorithm that achieves this requirement. Recently, these ideas have also been adopted by researchers in the artificial intelligence community (see [12], [13] and [26] for examples in both game theory and AI). In recent work [1], Bowling attempted to combine these criteria by proposing that an agent should both guarantee a no-regret payoff and achieve convergence in self-play. He then put forth, GIGA-WoLF, a no-regret algorithm that provably achieves convergence in self-play for games with two players and two actions per player.

A limitation common to all these approaches is that the game theoretic basis they're derived from was initially focused on large-population games and therefore ignores the effect of the agent's play on the future play of the opponent. This can pose problems in smaller games. Let us again consider the game of Prisoner's Dilemma with a Tit-for-Tat opponent. The only universally consistent strategy would be to defect at every time step, ruling out the higher payoff achievable by cooperating. Clearly, a universally consistent (or no-regret) policy is not the best response in this richer strategy space.

In response to the exiting work, we proposed a new set of criteria in [22]:

Targeted Optimality: *Against any member of the target set of opponents, the algorithm achieves within ϵ of the expected value of the best response to the actual opponent.*

Auto-Compatibility: *During self-play, the algorithm achieves at least within ϵ of the payoff of a Nash equilibrium that is not Pareto dominated by another Nash equilibrium.*

Safety: *Against any opponent, the algorithm always receives at least within ϵ of the security value for the game.*

In addition these requirements were required to hold with probability at least $1 - \delta$ after an initial polynomial period of time. Our first paper provided an algorithm that considered only stationary opponents, but a new algorithm meet-

ing these criteria against memory-bounded opponents was presented in [21]. Unfortunately, neither of the algorithms can easily be generalized to situations with more than two agents. A critical component of the design of the two algorithms is a teaching component based on the Bully and Godfather algorithms proposed by Littman and Stone in [17]. This component relies on having a single opponent in order to calculate the best action or sequence of actions to play given that the opponent will play a best response.

3. NEW CRITERIA

In our work we seek to pull together the advances made in existing work and encourage the development of algorithms that can both cooperate with one another and achieve strong guarantees on payoff against a variety of opponents.

In the criteria of [22] we had three categories of opponents we cared about: members of the target class, identical (self-play) agents, and other (unconstrained) agents. When only considering games for two agents, these three categories are sufficient for all the scenarios. However, once we extend to games with more than two agents, we need to consider mixes of agent types. Since each agent can be any of the three types, there are seven different possible sets of opponent types. We divide those seven sets into two scenarios:

- Each of the opponents is either a self-agent or in the target class.
- At least one opponent is of the Unconstrained type.

We also need to consider the issues of coordination between the agents when selecting actions. Although other researchers may find different assumptions appropriate for particular settings, we have chosen to focus on the most pessimistic/conservative assumptions in which the self-agents have no communication mechanism outside the game. Thus they must choose independently if they are selecting actions according to a probability distribution.

For the first scenario, ideally the agents should achieve a "joint best response" against the given target class. This is clearly defined when there is only one self-agent. The agent just needs to adopt the best response to the joint play of the agents in the target class. However, it remains an interesting issue when there are at least two self-agents. The self-agents will need to distribute the payoffs between themselves since selfishly trying to maximize one's payoff will not help the agent in many cases, such as the Prisoner's Dilemma game. Each of the self-agents should also be able and willing to cooperate with other agents who are using the same or similar algorithms as long as this will help to increase its current payoff. Thus the self-agents should achieve a Pareto-optimal (PO) outcome among themselves, i.e. there is no other joint outcome that could provide a higher payoff for one self-agent without decreasing the payoff of some other self-agent given the strategies of the opponents in the target class.

The PO condition alone, however, is not sufficient. Each agent has a minimum payoff that it can guarantee by itself without the cooperation of any other agents. In the worst case when all other agents are trying to minimize its payoff, this minimum value is $V_{security}$, defined for agent i as $\max_{\pi_i \in \Pi_i} \min_{\pi_{-i} \in \Pi_{-i}} EV_i(\pi_i, \pi_{-i})$. In this formula, Π_i is the set of strategies for agent i, and Π_{-i} is the set of joint strategies for the other agents. Any strategy that achieve this value on expectation is called a security strategy. Note

that if an agent has determined the strategies of the opponents in the target class, it can sometimes guarantee a higher value than $V_{security}$, which we will call V_g. Thus it is only rational for an agent to cooperate in a PO joint outcome if its payoff is at least V_g. Notice that the best response condition for one self-agent is a special case of this criterion, since the PO condition will guarantee that the agent is using an optimal strategy against the target class.

When there are opponents that are neither members of the target class nor self-agents, we instead require that each agent achieve at least V_g. Ideally, multiple self-agents could each exceed this value in some settings by cooperating with each other against the unconstrained agents.

Combining these two scenarios, we put forth the following new criteria:

Let n be the number of players in the game and m the maximum number of actions for a player. We require that for any choice of $\epsilon > 0$ and $\delta > 0$ there exist a τ, polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, n$, and m, such that for any number of rounds $t > \tau$ the algorithm achieves the following payoff guarantees with probability at least $1 - \delta$:

Targeted Group Optimality: *When each of the agents in the game is either a self-agent or in the target class, the payoffs of all the self-agents are at least $V_g - \epsilon$ and within ϵ of an PO outcome, given the actual strategies of agents in the target class.*

Safety: *Against any set of opponents, the agent must achieve at least $V_g - \epsilon$.*

Note that the Target Group Optimality condition combines and generalizes the Targeted Optimality and Auto-Compatibility conditions from [22].

4. CORRSTRATEGY(S): AN ALGORITHM FOR STATIONARY OPPONENTS

Besides proposing the novel criteria, we also want to provide algorithms that provably achieve the criteria for different target sets. We first consider here such an algorithm for stationary opponents which we call CorrStrategy(S).

4.1 Algorithm Description

CorrStrategy(S) is composed of four modules:

- **Learn Best Response**: Using observations about the opponents' play estimate and play a best-response strategy for the agent to the actual strategy of the opponents.
- **Coordinate**: Select a single, common deterministic joint strategy for all the self-play agents from among a set of Pareto-optimal possibilities.
- **Secure Value**: Play a strategy that ensures that the player receives at least the security value against any possible set of opponents.
- **Signal/Explore**: Observe the opponents' play and use explicit signaling to distinguish self-play agents from agents in the target class.

We show how these modules can be put together in Figure 1. The four bolded rectangles represent the modules. Note that in order to preserve clarity, we only show a detailed view of the "Coordinate" module, the most complex one. The agents following the framework will start with the "Signal" module and make transition between the modules based on the payoffs they receive and the observed behavior of the other agents in the environment.
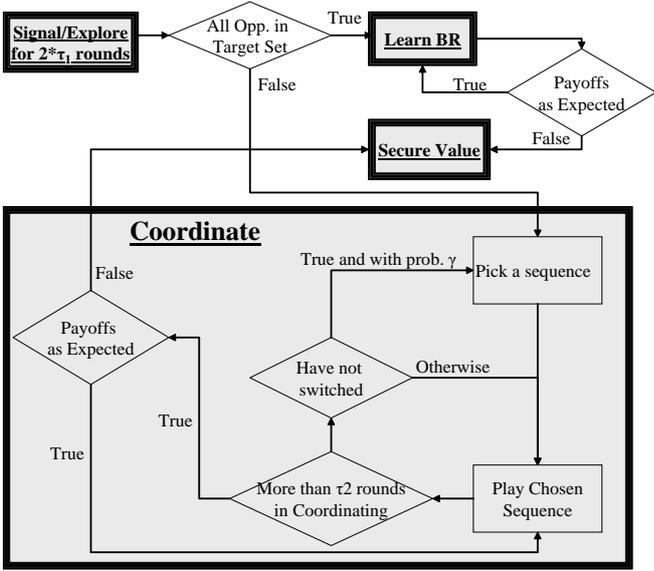
Figure 1: Flow of control for CorrStrategy(S).

It is necessary for the agents to distinguish between self-agents and members of the target class. Otherwise it would be possible for the agents to adopt best-responses assuming the other self-agents are stationary and play a Nash equilibrium with possibly non-PO payoffs. In general, it is not always trivial to distinguish between different types of opponents. One self-agent might appear stationary if the most rational action it needs to take always happens to be the same. This issue can be resolved easily if we allow the self-agents to explicitly signal each other in the "Signal" module.

Since the target class is stationary, the self-agents will signal each other by playing a pure strategy for τ_1 rounds and then switching to a different pure strategy for another τ_1 rounds. By the end of this block, the self-agents will be able to correctly partition all self-agents and stationary agents into two different sets with high probability. Agents whose distribution of actions during the last τ_1 rounds is within ϵ of the distribution from the full history are assumed to be stationary.

Each self-agent can now essentially reduce the current game to a smaller game by removing all stationary opponents and using the expected payoffs for each of the remaining outcomes instead. If there is only one remaining player in the sub-game, it transitions to the "Learn Best Response" module to find the best response to the stationary opponents. Using the sub-game, finding a BR strategy against stationary opponents is straightforward, since the agent can simply choose the action that gives the highest expected payoff. However, the agent needs to keep monitoring its payoff in order to protect itself against an adversarial opponent that pretends to be stationary during the "Signal" phase and changes strategy later on. At any time, if its payoff drops below $V_g - \epsilon$, the agent will switch to the "Secure Value" module.

If there are multiple non-stationary agents left in the sub-game, the self-agents will make the transition to the "Coordinate" module instead. In this module, they will try synchronize with each other in order to achieve the Targeted Group Optimality criterion. Each self-agent will first

solve a linear program with size polynomial in the number of stage-game outcomes, $|A|$, to find a PO outcome in the repeated game:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{|A|} \sum_{j=1}^{n} \alpha_i * PO_j(O_i) \\
\text{subject to} \quad & \forall j \, V_g(j) - \sum_{i=1}^{|A|} \alpha_i * PO_j(O_i) \leq \epsilon \\
& \sum_{i=1}^{|A|} \alpha_i = 1
\end{aligned}$$

In the equations above, $PO_j(O_i)$ is the payoff of agent j in the stage game outcome O_i and $V_g(j)$ is the guaranteed value of agent j. We denote this linear program LP^*.

The solution to LP^* is a distribution over joint outcomes in the stage game that maximizes the sum of the payoffs for the self-agents in the repeated game. In this distribution, α_i is the frequency with which the stage-game outcome O_i should be played by the self-agents to achieve a PO repeated game outcome. The solution always exists because if every self-agent, j, plays its security strategy they are all guaranteed to receive at least $V_g(j)$. Since the self-agents are not allowed to communicate outside the game, they can't mix over joint outcomes so will instead have to approximate this distribution by repeating a deterministic sequence S of joint actions. S will specify which action each agent needs to choose at each step. By using Hoeffding's inequality, it can be proved that self-agents can have an arbitrarily close approximation to the PO outcome in the repeated game with S of length polynomial in $m, n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$.

As there are possibly many such sequences S, the self-agents also need to coordinate with each other to converge to the same one. Each self-agent will pick one such sequence at the beginning and then with probability γ on each round will switch to a different sequence. Each agent only switches once. Let C be the group of agents who have already switched. The self-agents will try to ensure that all the agents in C are always using a joint sequence that maximizes the sum of payoffs. Whenever an agent attempts to pick a different sequence, i.e. joining C, it will pick one that preserves this property.

Since the agents in C are approximating some distribution over joint outcomes, the switching agent just needs to find this distribution and then find a matching deterministic sequence over joint outcomes that will approximate this distribution with the agents in C. To achieve this, the switching agent can re-solve LP^* with additional constraints to guarantee that in the new solution, the action distributions of the agents in C match their current observed distributions.

$$\forall j \in C, k = 1..|A_j| : \sum_{i=1}^{|A|} f_j(i,k) * \alpha_i = \pi_j(k)$$

In the above equation, $\pi_j(k)$ is the observed action distribution of action k for player j, and $f_j(i,k) = 1$ if agent j plays action k in outcome O_i and $f_j(i,k) = 0$ otherwise. If no agents have switched sequences, the agent will simply pick a different sequence that also approximates the solution for current LP^*.

At the end of the process, there are two possibilities that occur with high probability: either all the self-agents are playing the same deterministic sequence of joint actions or there exist unconstrained agents. In the first case, the payoff profile for the self-agents is at most ϵ away from the payoff profile in the PO outcome they are trying to approximate. However, they still need to monitor their payoffs to avoid the case in which there are unconstrained agents that only pretend to cooperate with the self-agents. Once the payoff to any agent drops more than ϵ below their target

payoff in LP^*, the agents will switch to the "Secure Value" module. This will also handle the second case where the unconstrained agents prevented them from converging to a common sequence. Note that in the case the unconstrained agents fully cooperate with the self-agents in an PO outcome, the self-agents will implicitly achieve the Safety requirement, and therefore can safely treat the unconstrained ones as self-agents.

A self-agent uses the "Secure Value" module to guarantee its payoff ia at least the guaranteed value of $V_g - \epsilon$ in the presence of unconstrained agents. The agent can easily achieve this by calculating and adopting the security strategy for the sub-game obtained by removing all stationary opponents. While using this module, the self-agent still needs to monitor the opponents that are assumed to be stationary. Otherwise, unconstrained agents could impersonate members of the target set at the beginning of the game and then lower the payoff of a self-agent by changing strategies, or more subtly, by using a correlated joint mixed-strategy that still makes them appear stationary. In both cases, any harmful variations can be easily detected with high probability by replacing each stationary agent's actual play with random draws from their observed distribution and detecting if the payoffs to any of the self-agents change by more than ϵ. If such an unconstrained agent is detected, a new V_g can be calculated and a new security strategy played.

THEOREM 1. *CorrStrategy(S) satisfies the Targeted Group Optimality and Safety criteria for the target class of stationary opponents after a number of rounds polynomial in $n, m, \frac{1}{\epsilon}$ and $\frac{1}{\delta}$.*

PROOF. Since we only consider games with bounded payoffs, we can assume, without loss of generality, that all the payoffs are normalized to be between 0 and 1. The proof can be constructed naturally from the following lemmas. The proofs for the lemmas can be constructed using Hoeffding's inequality and are omitted due to space constraints:

LEMMA 1. *For any given $\delta_1 > 0$, $0.5 > \epsilon_1 > 0$, there exists τ_1 polynomial in $\frac{1}{\epsilon_1}, \log(\frac{1}{\delta_1}), \log n$ and $\log m$ such that if an agent uses a full action history of length at least $2\tau_1$, and a recent action history of length τ_1, all self-agents will correctly partition stationary and self-agents into two different sets and the observed action distribution for all stationary opponents will be within ϵ_1 of the true distribution with probability at least $1 - \delta_1$.*

LEMMA 2. *For any given $\delta_2 > 0$, $\epsilon_2 > 0$, there exists a deterministic sequence of joint actions, S, with length polynomial in $\frac{1}{\epsilon_2}, \log(\frac{1}{\delta_2}), \log n$, and m that can approximate within ϵ_2 the distribution over PO outcomes of any solution to LP^*. When there are only self-agents, the difference between the payoffs achieved by the self-agents using S and a PO outcome will be at most $n * m * \epsilon_2$ with probability at least $1 - \delta_2$.*

LEMMA 3. *Within the "Coordinate" block, for any given $\delta_3 > 0$, $T > 0$, and $\gamma \leq 1 - (1 - \delta_3)^{\frac{1}{n^2 T}}$, if each cooperating player attempts to change its distribution of actions on each round with probability γ, there exists a τ_2 polynomial in $\frac{1}{\delta_3}, T$ and n such that with probability at least $1 - \delta_3$, after τ_2 rounds all the self-agents will change their sequence exactly once and no two self-agents will change within T rounds of each other.*

From Lemma 1, by the end of the "Signal" module (after $2\tau_1$ rounds), the self-agents have correctly partitioned stationary players and coop players into two different sets with probability at least $1 - \delta_1$.

Consider first the case in which there are no unconstrained agents. Within the "Coordinate" module, once an agent decides to pick a different sequence, it has to recalculate the optimal solution to LP^*. From Lemma 1, we know that the observed distribution of each stationary opponent is within ϵ_1 of its true distribution. Since the payoffs are bounded between 0 and 1 and there are at most m actions, the payoff for the self-agent can be reduced by at most $m * \epsilon_1$ for each of the up to n stationary opponents. Thus if we choose ϵ_1 to be $\frac{\epsilon}{2mn}$ and ϵ_2 in Lemma 2 to be $\frac{\epsilon}{2mn}$, once all the self-agents converge to the same sequence, the payoff for each agent can only be at most ϵ away from the optimal payoff. From Lemma 3, if we set $T \geq L$, we can choose a γ such that no two agents will switch sequence within a period of L. Thus, each self-agent will always correctly determine the distribution of the agents that have already switched sequences. All self-agents will switch sequence once within a polynomial number of rounds and therefore converge to one sequence. Combining all of these conditions we have that the self-agents will converge to a sequence with payoffs within ϵ_3 of a PO outcome of the repeated game with probability at least $1 - \delta_1 - \delta_2 - \delta_3$. If we then set $\delta_1, \delta_2, \delta_3$ to be $\frac{\delta}{3}$ we satisfy Target Group Optimality. With these values, τ_1 and τ_2 are clearly polynomial in $m, n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$. Moreover, since the losses of the payoffs over the initial period of length $\tau = 2 * \tau_1 + \tau_2$ are bounded by τ, by allowing another $\frac{2\tau}{\epsilon}$ additional rounds to pass, the average payoffs of the agents for the entire game will converge within ϵ of a PO outcome.

Let us now consider the case when there is at least one unconstrained agent. Notice that in all the modules the self-agents switch to the Security Value module whenever their payoffs fall below $V_g - \epsilon$. If an agent has correctly divided the stationary and unconstrained agents it can now be guaranteed a payoff at least $V_g - \epsilon$ with probability at least $1 - \delta$ after a period polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, m$, and n by the Hoeffding inequality. If on the other hand an unconstrained agent is simulating a stationary opponent, it will not be able to drive the agent's value below $V_g - \epsilon$ without revealing itself by causing its play to vary from that of its assumed stationary distribution. The agents can then update their estimates of V_g and adopt a new security strategy where the unconstrained agent is categorized correctly. Thus they will always achieve the Safety criteria.

Therefore, there exists a τ polynomial in $m, n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$ such that after an initial experimentation period of length τ, CorrStrategy(S) will satisfy the two criteria in section 3. $\square$

THEOREM 2. *The computational complexity of CorrStrategy(S) for playing one round of the repeated game is polynomial in $|A|$, the size of the game.*

PROOF. To find the worst case complexity for one iteration, we can consider separately the complexity for each module of the algorithm as presented in Figure 1:

- Within the "Signal" module, once all the stationary opponents are recognized, the self-agent can find the sub-game in $O(|A|)$ time.

- Within the "Learn Best Response" module, finding the best response in the sub-game is $O(m)$.
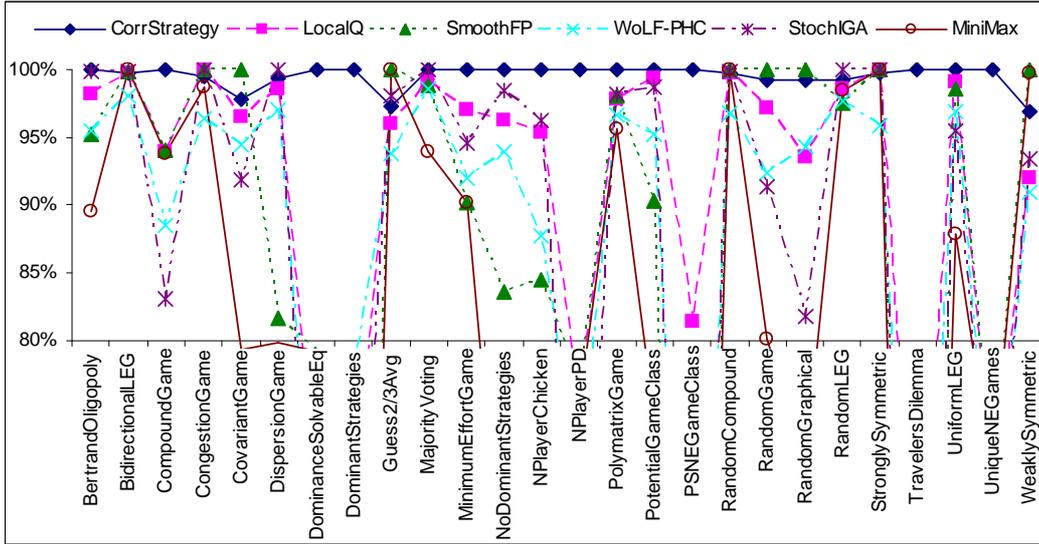
Figure 2: Percent of maximum value for 200K rounds averaged for each agent over all different pairs of opponents for selected games in GAMUT.

- Within the "Secure Value" module, solving the linear program for the security strategy can be done in time polynomial in $|A|$.

- Within the "Coordinate" module, in the worst step the agent has to switch to a different sequence. This process involves solving a linear program with the number of variables and constraints polynomial in the number of outcomes. Thus the complexity for the worst step within this module is also polynomial in $|A|$.

□

4.2 Experimental Results

Even though our algorithm has been proven to meet the proposed criterion, we want to demonstrate empirically that the algorithm performs well against a variety of opponents including those outside the target class. We will use the testing environment first described in [22] by testing against a number of existing approaches from the multi-agent learning literature over a wide variety of repeated games from GAMUT [19]. GAMUT is the result of a project to develop a comprehensive collection of game theoretic matrix games that have been described by researchers in either game theory or artificial intelligence. It contains generators for creating random instances of 34 individual base game classes as well as numerous additional variants and specialized parameter settings (more information and downloads are available at `gamut.stanford.edu`). The existing algorithms we tested against include Local Q-learning[25], a stochastic version of IGA[23], WoLF-PHC[2], JointQ-Max[6], and smooth fictitious play[10]. We also tested all the algorithms against random stationary strategies (Random), the security value strategy (MiniMax), and random strategies that condition their actions on the past outcome (Mem1).

Focusing our attention on settings with more than two players, our first test measured the average performance of a pair of players using the same algorithm playing against another pair of players using identical algorithms. In figure 2, we show the average payoffs of the five most successful

algorithms as well as the performance of the MiniMax algorithm for a representative set of games in GAMUT. The payoffs have been normalized by dividing each algorithm's payoff by the best payoff achieved by any strategy for that game in order to make visual comparisons across games easier. We can see that CorrStrategy(S) performs well across all the games, achieving the highest or close to the highest payoff in nearly every game and unlike other algorithms, CorrStrategy(S) has no pitfalls in which its payoffs are significantly worse than the highest payoffs achieved. This is at least partly due to the fact that agents using CorrStrategy(S) can cooperate with agents using other algorithms as long as they each still achieve their security value.
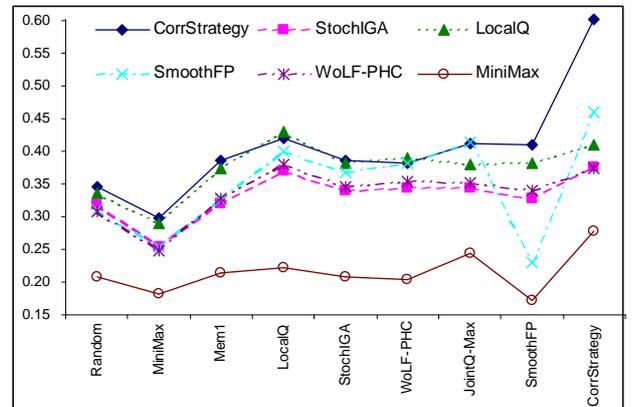

Figure 3: 2v2 By Opp: Value achieved against each pair of opponents averaged over all games.

In Figure 3, we show the results for the same algorithms averaged across games for each possible pair of opponents (note that all payoffs for each game are in the range $[-1, 1]$). We can see that CorrStrategy performs well against each of the different classes of opponents, but truly excels when playing against other CorrStrategy agents. Smooth fictitious play fares similarly well against several of the strategies, but is unable to cooperate in self-play resulting in a payoff near that of the MiniMax algorithm. One of CorrStrat-

egy's largest advantages lies in its ability to cooperate when possible without sacrificing much performance against agents outside its target set.

To further investigate the contribution of this advantage to the strong results achieved by CorrStrategy(S), we ran two alternate experimental settings. In one setting we had each algorithm play by itself against a pair of algorithms using identical algorithms. In the second setting we instead let two versions of the algorithm being tested play against a single instance of an opponent algorithm. In Table 4.2 we can see the average results for these settings compared with the results in the four player setting shown earlier. As we would predict, CorrStrategy(S)'s advantage over the other algorithms increases in settings where it outnumbers its opponents, but its results remain competitive even when it is outnumbered in turn.

	1 Agent vs 2 Opps	2 vs 1	2 vs 2
CorrStrategy	0.45	0.49	0.40
LocalQ	0.42	0.44	0.37
SmoothFP	0.43	0.36	0.35
WoLF-PHC	0.38	0.38	0.34
StochIGA	0.39	0.37	0.33
MiniMax	0.29	0.20	0.21

Table 1: Summary of average payoff by setting.

5. CORRSTRATEGY(A): AN ALGORITHM FOR ADAPTIVE OPPONENTS

As encouraging as the theoretical and experimental results for CorrStrategy(S) may be, we still have yet to address one of our critiques of existing research, which was the tendency to only focus on stationary opponents. Although we addressed this concern explicitly in [21], our algorithm depended critically on the assumption that there was only a single opponent. We are aware of very little additional work to date that deals with adaptive opponents explicitly, although de Farias and Megiddo [8] address it in the design of their experts algorithm and the rational learning approach of Kalai and Lehrer [14] can in principle handle adaptive algorithms of arbitrary complexity as long as they are assigned positive probability in the prior.

In the remainder of this section, we'll outline a way to extend our CorrStrategy algorithm to deal with opponents whose play may be a function of the prior history of the game. We do this by expanding the target set against which we can guarantee a best-response. Note however that we still need to limit the capabilities of the opponents in some way. If we were to consider opponents whose future behavior could depend arbitrarily on the entire history of play, we would lose the ability to learn anything about them in a single repeated game, since we would only ever see a given history once and an opponent's past strategy may bear no relation to their future play.

We therefore adopt the model for bounded memory we used in [21] and assume a limit on each opponent's ability to condition on the history. This model requires that the opponents play a conditional strategy where their distribution over actions can only depend on the most recent k periods of past history, $F_i : o_{-1} \times ... \times o_{-k} \rightarrow \Delta A_i$, where o_{-t} is the outcome of the game t periods ago. Additionally, the opponents have a default past history they assume at the start of the game. Note that even this simple model allows us to capture many methods, such as Tit-for-Tat, that current approaches are unable to properly handle.

Taking the set of conditional strategies with history k as our new target set, we propose the extension CorrStrategy(A). CorrStrategy(A) shares the same basic algorithmic framework as CorrStrategy(S) with the following changes:

- For the "Signal/Explore" module, the self-agents play a uniform mixed strategy for a period long enough to gather sufficient observations to calculate each opponent's distribution of play for each possible history of length k. The self-agents then switch to a pure strategy that is inconsistent with the prior observed distribution of their play. The opponents using the same distribution for each history belong to the target class.

- For the "Learn Best Response" module, we calculate a best response against conditional strategies. This approach maintains counts of the opponent's actions after each history of length k, which it uses to calculate the cycle of agent actions with the highest expected reward out of all possible unique agent action sequences (those that don't contain a length k repeated subsequence). Given sufficient observations, this lets us guarantee that we achieve an ϵ-best response against any members of our target opponent set.[1]

- Note that in order to use this new best-response function in CorrStrategy(A), we need to insure that the algorithm observes each length k history a sufficient number of times. This will be satisfied as long as the initial exploration phase continues for a length of time exponential in k. This exponential exploration period is unavoidable since we need to consider the possibility of opponents that only play a desirable action distribution for a single one of the exponentially many possible histories.

- For the "Coordinate" block, the optimization problem needs to take into account all possible sequences of actions for the adaptive opponents in the target set.

THEOREM 3. *CorrStrategy(A) satisfies the Targeted Group Optimality and Safety criteria for the target class of conditional strategies with bounded memory k.*

The proof of Theorem 3 follows the same framework as the proof of Theorem 1 but we need to take into account a much longer initial period for signaling and observing opponent's distributions and also a longer sequence S to approximate the PO outcome. The initial experimentation period τ required will unfortunately now depend on m^k and $(\frac{1}{\lambda})^{(m^{nk})}$, where λ is the minimum probability the opponent assigns to any action ($\lambda = 1$ for opponents that condition only on the coop player's actions). Note that our worst case time complexity also grows similarly as we may now need to solve an optimization problem with up to m^k variables, although both of these bounds (computational complexity and amount of training) are based on extremely pessimistic assumptions and are likely to prove tractable in practice for larger games with small values of k.

---

[1]This implementation is suitable for conditional strategies that only depend on the self-agents' actions. For general conditional strategies we need to consider the full space of deterministic conditional strategies to find a best response.

6. CONCLUSIONS AND FUTURE WORK

Even though there have been several recent proposals for different criteria to evaluate learning algorithms in multi-agent systems, little attention has been given to two important scenarios: learning in games with more than two players, and learning against adaptive opponents. We have addressed both scenarios by proposing a new set of criteria for learning in games with any number of players that takes a target class as parameter, allowing the designer to choose a class of opponents of interest. This set of criteria encourages learning algorithms to allow cooperation between agents while still guaranteeing a basic security guarantee for each agent. We then designed two algorithms that can achieve the criteria against two target classes: stationary strategies and adaptive strategies with bounded memories. Moreover, our implementation of the algorithm for stationary strategies outperforms a wide range of opponents across the games of a comprehensive test-bed of repeated games.

One interesting problem we will continue to investigate is how to improve the guaranteed payoff of self-agents playing against adversaries outside their target class. In many settings the agents should be able to cooperate with each other in order to reduce the adversarial effects of their opponents and guarantee payoffs above their individual security values. In addition, we are continuing to extend our approach to consider other models of adaptive opponents. A common approach used in the literature on bounded rationality [18, 20] is to assume the players can be modelled by finite automata with k states. It should prove relatively straightforward to extend our CorrStrategy(A) algorithm to handle automata by replacing the best response function and optimization functions once again.

We are also looking at several ways to expand the set of environments these algorithms can be employed within. Of particular concern is looking for ways to weaken the requirement of full prior knowledge about the payoffs of the game. The major challenge seems to lie in creating the capability to cooperate without knowing or being able to observe the space of payoffs available to the other players. An additional area for further consideration would be extending to handle games in which the agents have only partial observability of the previous actions of the other players.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] M. Bowling. Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[2] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[3] M. Bowling and M. M. Veloso. Existence of multiagent equilibria with limited agents. Technical report CMU-CS-02-104, Computer Science Department, Carnegie Mellon University, 2002.

[4] R. Brafman and M. Tennenholtz. Efficient learning equilibrium. In *Advances in Neural Information Processing Systems*, volume 15, Cambridge, Mass., 2002. MIT Press.

[5] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *AAMAS*, 2003.

[6] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

[7] V. Conitzer and T. Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the 20th International Conference on Machine Learning*, pages 83–90, 2003.

[8] D. P. de Farias and N. Megiddo. How to combine expert (or novice) advice when actions impact the environment. In *Advances in Neural Information Processing Systems 16*, 2004.

[9] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–36, 1999.

[10] D. Fudenberg and D. Levine. Universal consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.

[11] J. F. Hannan. Approximation to bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[12] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

[13] A. Jafari, A. Greenwald, D. Gondek, and G. Ercal. On no-regret learning, fictitious play, and nash equilibrium. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 226–223, 2001.

[14] E. Kalai and E. Lehrer. Rational learning leads to nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.

[15] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 326–331, 2002.

[16] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 535–542. Morgan Kaufman, 2000.

[17] M. Littman and P. Stone. Implicit negotiation in repeated games. In *Proceedings of The Eighth International Workshop on Agent Theories, Architectures, and Languages*, pages 393–404, 2001.

[18] A. Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, pages 227–229, 1985.

[19] E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the gamut: A comprehensive approach to evaluating game-theorectic algorithms. *AAMAS*, 2004.

[20] C. H. Papadimitriou and M. Yannakakis. On complexity as bounded rationality. In *STOC-94*, pages 726–733, 1994.

[21] R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 2005.

[22] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[23] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 41–48, Stanford University, 2000. Morgan Kaufman.

[24] X. Wang and T. Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Advances in Neural Information Processing Systems*, volume 15, 2002.

[25] C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8(3/4):279–292, May 1992.

[26] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.