Rational and Convergent Learning in Stochastic Games

Michael Bowling mhb@cs.cmu.edu
Manuela Veloso veloso@cs.cmu.edu

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

This paper investigates the problem of policy learning in multiagent environments using the stochastic game framework, which we briefly overview. We introduce two properties as desirable for a learning agent when in the presence of other learning agents, namely rationality and convergence. We examine existing reinforcement learning algorithms according to these two properties and notice that they fail to simultaneously meet both criteria. We then contribute a new learning algorithm, WoLF policy hill-climbing, that is based on a simple principle: "learn quickly while losing, slowly while winning." The algorithm is proven to be rational and we present empirical results for a number of stochastic games showing the algorithm converges.

1 Introduction

The *multiagent learning problem* consists of devising a learning algorithm for *our* single agent to learn a policy in the presence of *other* learning agents that are outside of our control. Since the other agents are also adapting, learning in the presence of multiple learners can be viewed as a problem of a "moving target," where the optimal policy may be changing while we learn. Multiple approaches to multiagent learning have been pursued with different degrees of success (as surveyed in [Weiß and Sen, 1996] and [Stone and Veloso, 2000]). Previous learning algorithms either converge to a policy that is not optimal with respect to the other player's policies, or they may not converge at all. In this paper we contribute an algorithm to overcome these shortcomings.

We examine the multiagent learning problem using the framework of stochastic games. Stochastic games (SGs) are a very natural multiagent extension of Markov decision processes (MDPs), which have been studied extensively as a model of single agent learning. Reinforcement learning [Sutton and Barto, 1998] has been successful at finding optimal control policies in the MDP framework, and has also been examined as the basis for learning in stochastic games [Claus and Boutilier, 1998; Hu and Wellman, 1998; Littman, 1994]. Additionally, SGs have a rich background in game theory, being first introduced in 1953 [Shapley].

In Section 2 we provide a rudimentary review of the necessary game theory concepts: stochastic games, best-responses, and Nash equilibria. In Section 3 we present two desirable properties, rationality and convergence, that help to elucidate the shortcomings of previous algorithms. In Section 4 we contribute a new algorithm toward achieving these properties called WoLF ("Win or Learn Fast") policy hill-climbing, and prove that this algorithm is rational. Finally, in Section 5 we present empirical results of the convergence of this algorithm in a number and variety of domains.

2 Stochastic Games

A *stochastic game* is a tuple $(n, \mathcal{S}, \mathcal{A}_{1\ldots n}, T, R_{1\ldots n})$, where n is the number of agents, $\mathcal{S}$ is a set of states, $\mathcal{A}_i$ is the set of actions available to agent i with $\mathcal{A}$ being the joint action space $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$, T is a transition function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and R_i is a reward function for the ith agent $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$. This is very similar to the MDP framework except we have multiple agents selecting actions and the next state and rewards depend on the joint action of the agents. Also notice that each agent has its own separate reward function. The goal for each agent is to select actions in order to maximize its discounted future reward with discount factor γ.

SGs are a very natural extension of MDPs to multiple agents. They are also an extension of matrix games to multiple states. Two common matrix games are in Figure 1. In these games there are two players; one selects a row and the other selects a column of the matrix. The entry of the matrix they jointly select determines the payoffs. The games in Figure 1 are zero-sum games, where the row player receives the payoff in the matrix, and the column player receives the negative of that payoff. In the general case (general-sum games) each player has a separate matrix that determines its payoff.

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Matching Pennies

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

Rock-Paper-Scissors

Figure 1: Two example matrix games.

Each state in a stochastic game can be viewed as a matrix game with the payoffs for each joint action determined by the matrix entries $R_i(s, a)$. After playing the matrix game and receiving their payoffs the players are transitioned to another state (or matrix game) determined by their joint action. We can see that SGs then contain both MDPs and matrix games as subsets of the framework.

Mixed Policies. Unlike in single-agent settings, deterministic policies in multiagent settings can often be exploited by the other agents. Consider the matching pennies matrix game as shown in Figure 1. If the column player were to play either action deterministically, the row player could win a payoff of one every time. This requires us to consider mixed strategies or policies. A mixed policy, $\rho : \mathcal{S} \rightarrow PD(\mathcal{A}_i)$, is a function that maps states to mixed strategies, which are probability distributions over the player's actions.

Nash Equilibria. Even with the concept of mixed strategies there are still no optimal strategies that are independent of the other players' strategies. We can, though, define a notion of best-response. A strategy is a *best-response* to the other players' strategies if it is optimal given their strategies. The major advancement that has driven much of the development of matrix games, game theory, and even stochastic games is the notion of a best-response equilibrium, or *Nash equilibrium* [Nash, Jr., 1950].

A Nash equilibrium is a collection of strategies for each of the players such that each player's strategy is a best-response to the other players' strategies. So, no player can get a higher payoff by changing strategies given that the other players also don't change strategies. What makes the notion of equilibrium compelling is that all matrix games have such an equilibrium, possibly having multiple equilibria. In the zero-sum examples in Figure 1, both games have an equilibrium consisting of each player playing the mixed strategy where all the actions have equal probability.

The concept of equilibria also extends to stochastic games. This is a non-trivial result, proven by Shapley [1953] for zero-sum stochastic games and by Fink [1964] for general-sum stochastic games.

3 Motivation

The multiagent learning problem is one of a "moving target." The best-response policy changes as the other players, which are outside of our control, change their policies. Equilibrium solutions do not solve this problem since the agent does not know which equilibrium the other players will play, or even if they will tend to an equilibrium at all.

Devising a learning algorithm for our agent is also challenging because we don't know which *learning algorithms* the other learning agents are using. Assuming a general case where other players may be changing their policies in a completely arbitrary manner is neither useful nor practical. On the other hand, making restrictive assumptions on the other players' specific methods of adaptation is not acceptable, as the other learners are outside of our control and therefore we don't know which restrictions to assume.

We address this multiagent learning problem by defining two properties of a learner that make requirements on its behavior in concrete situations. After presenting these properties we examine previous multiagent reinforcement learning techniques showing that they fail to simultaneously achieve these properties.

3.1 Properties

We contribute two desirable properties of multiagent learning algorithms: rationality and convergence.

Property 1 *(Rationality) If the other players' policies converge to stationary policies then the learning algorithm will converge to a policy that is a best-response to their policies.*

This is a fairly basic property requiring the player to behave optimally when the other players play stationary strategies. This requires the player to learn a best-response policy in this case where one indeed exists. Algorithms that are not rational often opt to learn some policy independent of the other players' policies, such as their part of some equilibrium solution. This completely fails in games with multiple equilibria where the agents cannot *independently select* and play an equilibrium.

Property 2 *(Convergence) The learner will necessarily converge to a stationary policy. This property will usually be conditioned on the other agents using an algorithm from some class of learning algorithms.*

The second property requires that, against some class of other players' learning algorithms (ideally a class encompassing most "useful" algorithms), the learner's policy will converge. For example, one might refer to convergence with respect to players with stationary policies, or convergence with respect to rational players.

In this paper, we focus on convergence in the case of self-play. That is, if all the players use the same learning algorithm do the players' policies converge? This is a crucial and difficult step towards convergence against more general classes of players. In addition, ignoring the possibility of self-play makes the naive assumption that other players are inferior since they cannot be using an identical algorithm.

In combination, these two properties guarantee that the learner will converge to a stationary strategy that is optimal given the play of the other players. There is also a connection between these properties and Nash equilibria. When all players are rational, if they converge, then they must have converged to a Nash equilibrium. Since all players converge to a stationary policy, each player, being rational, must converge to a best response to their policies. Since this is true of each player, then their policies by definition must be an equilibrium. In addition, if all players are rational and convergent with respect to the other players' algorithms, then convergence to a Nash equilibrium is guaranteed.

3.2 Other Reinforcement Learners

There are few RL techniques that directly address learning in a multiagent system. We examine three RL techniques: single-agent learners, joint-action learners (JALs), and minimax-Q.

Single-Agent Learners. Although not truly a multiagent learning algorithm, one of the most common approaches is to apply a single-agent learning algorithm (e.g. Q-learning, TD(λ), prioritized sweeping, etc.) to a multi-agent domain.

They, of course, ignore the existence of other agents, assuming their rewards and the transitions are Markovian. They essentially treat other agents as part of the environment.

This naive approach does satisfy one of the two properties. If the other agents play, or converge to, stationary strategies then their Markovian assumption holds and they converge to an optimal response. So, single agent learning is rational. On the other hand, it is not generally convergent in self-play. This is obvious to see for algorithms that learn only deterministic policies. Since they are rational, if they converge it must be to a Nash equilibrium. In games where the only equilibria are mixed equilibria (e.g. Matching Pennies), they could not converge. There are single-agent learning algorithms capable of playing stochastic policies [Jaakkola *et al.*, 1994; Baird and Moore, 1999]. In general though just the ability to play stochastic policies is not sufficient for convergence, as will be shown in Section 4.

Joint Action Learners. JALs [Claus and Boutilier, 1998] observe the actions of the other agents. They assume the other players are selecting actions based on a stationary policy, which they estimate. They then play optimally with respect to this learned estimate. Like single-agent learners they are rational but not convergent, since they also cannot converge to mixed equilibria in self-play.

Minimax-Q. Minimax-Q [Littman, 1994] and Hu & Wellman's extension of it to general-sum SGs [1998] take a different approach. These algorithms observe both the actions and rewards of the other players and try to learn a Nash equilibrium explicitly. The algorithms learn and play the equilibrium independent of the behavior of other players. These algorithms are convergent, since they always converge to a stationary policy. However, these algorithms are not rational. This is most obvious when considering a game of Rock-Paper-Scissors against an opponent that almost always plays "Rock". Minimax-Q will still converge to the equilibrium solution, which is not optimal given the opponent's policy.

In this work we are looking for a learning technique that is rational, and therefore plays a best-response in the obvious case where one exists. Yet, its policy should still converge. We want the rational behavior of single-agent learners and JALs, and the convergent behavior of minimax-Q.

4 A New Algorithm

In this section we contribute an algorithm towards the goal of a rational and convergent learner. We first introduce an algorithm that is rational and capable of playing mixed policies, but does not converge in experiments. We then introduce a modification to this algorithm that results in a rational learner that does in experiments converge to mixed policies.

4.1 Policy Hill Climbing

A simple extension of Q-learning to play mixed strategies is policy hill-climbing (PHC) as shown in Table 1. The algorithm, in essence, performs hill-climbing in the space of mixed policies. Q-values are maintained just as in normal Q-learning. In addition the algorithm maintains the current mixed policy. The policy is improved by increasing the probability that it selects the highest valued action according to a learning rate $\delta \in (0,1]$. Notice that when $\delta = 1$ the algorithm is equivalent to Q-learning, since with each step the policy moves to the greedy policy executing the highest valued action with probability 1 (modulo exploration).

1. Let α and δ be learning rates. Initialize,
$$Q(s,a) \leftarrow 0, \qquad \pi(s,a) \leftarrow \frac{1}{|\mathcal{A}_i|}.$$
2. Repeat,
   (a) From state s select action a with probability $\pi(s,a)$ with some exploration.
   (b) Observing reward r and next state s',
   $$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a')\right).$$
   (c) Update $\pi(s,a)$ and constrain it to a legal probability distribution,
   $$\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \delta & \text{if } a = \operatorname{argmax}_{a'} Q(s,a') \\ \frac{-\delta}{|A_i|-1} & \text{otherwise} \end{cases}.$$

Table 1: Policy hill-climbing algorithm (PHC) for player i.

This technique, like Q-learning, is rational and will converge to an optimal policy if the other players are playing stationary strategies. The proof follows from the proof of Q-learning, which guarantees the Q values will converge to Q^* with a suitable exploration policy.[1] Similarly, π will converge to a policy that is greedy according to Q, which is converging to Q^*, the optimal response Q-values. Despite the fact that it is rational and can play mixed policies, it still doesn't show any promise of being convergent. We show examples of its convergence failures in Section 5.

4.2 WoLF Policy Hill-Climbing

We now introduce the main contribution of this paper. The contribution is two-fold: using a *variable learning rate*, and the *WoLF principle*. We demonstrate these ideas as a modification to the naive policy hill-climbing algorithm.

The basic idea is to vary the learning rate used by the algorithm in such a way as to encourage convergence, without sacrificing rationality. We propose the WoLF principle as an appropriate method. The principle has a simple intuition, learn quickly while losing and slowly while winning. The specific method for determining when the agent is winning is by comparing the current policy's expected payoff with that of the average policy over time. This principle aids in convergence by giving more time for the other players to adapt to changes in the player's strategy that at first appear beneficial, while allowing the player to adapt more quickly to other players' strategy changes when they are harmful.

The required changes for WoLF policy hill-climbing are shown in Table 2. Practically, the algorithm requires two

[1] The issue of exploration is not critical to this work. See [Singh *et al.*, 2000a] for suitable exploration policies for online learning.

1. Let α, $\delta_l > \delta_w$ be learning rates. Initialize,

$$Q(s,a) \leftarrow 0, \qquad \pi(s,a) \leftarrow \frac{1}{|\mathcal{A}_i|}, \qquad C(s) \leftarrow 0.$$

2. Repeat,

(a,b) Same as PHC in Table 1

(c) Update estimate of average policy, $\bar{\pi}$,

$$\begin{array}{rcl} C(s) & \leftarrow & C(s)+1 \\ \forall a' \in \mathcal{A}_i \quad \bar{\pi}(s,a') & \leftarrow & \bar{\pi}(s,a') + \\ & & \frac{1}{C(s)}\left(\pi(s,a') - \bar{\pi}(s,a')\right). \end{array}$$

(d) Update $\pi(s,a)$ and constrain it to a legal probability distribution,

$$\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \delta & \text{if } a = \operatorname{argmax}_{a'} Q(s,a') \\ \frac{-\delta}{|A_i|-1} & \text{otherwise} \end{cases},$$

where,

$$\delta = \begin{cases} \delta_w & \text{if } \sum_a \pi(s,a)Q(s,a) > \sum_a \bar{\pi}(s,a)Q(s,a) \\ \delta_l & \text{otherwise} \end{cases}.$$

Table 2: WoLF policy hill-climbing algorithm for player i.

learning learning rate parameters, δ_l and δ_w, with $\delta_l > \delta_w$. The learning rate that is used to update the policy depends on whether the agent is currently winning (δ_w) or losing (δ_l). This is determined by comparing the expected value, using the current Q-value estimates, of following the current policy π in the current state with that of following the average policy $\bar{\pi}$. If the expectation of the current policy is smaller (i.e. the agent is "losing") then the larger learning rate, δ_l is used.

WoLF policy hill-climbing remains rational, since only the speed of learning is altered. In fact, any bounded variation of the learning rate would retain rationality. Its convergence properties, though, are quite different. In the next section we show empirical results that this technique converges to rational policies for a number and variety of stochastic games. The WoLF principle also has theoretical justification for a restricted class of games. For two-player, two-action, iterated matrix games, gradient ascent (which is known not to converge [Singh *et al.*, 2000b]) when using a WoLF varied learning rate is guaranteed to converge to a Nash equilibrium in self-play [Bowling and Veloso, 2001].

Something similar to the WoLF principle has also been studied in some form in other areas, notably when considering an adversary. In evolutionary game theory the *adjusted replicator dynamics* [Weibull, 1995] scales the individuals' growth rate by the inverse of the overall success of the population. This will cause the population's composition to change more quickly when the population as a whole is performing poorly. A form of this also appears as a modification to the *randomized weighted majority* algorithm [Blum and Burch, 1997]. In this algorithm, when an expert makes a mistake, a portion of its weight loss is redistributed among the other experts. If the algorithm is placing large weights on mistaken experts (i.e. the algorithm is "losing"), then a larger portion of the weights are redistributed (i.e. the algorithm adapts more quickly.) Neither research lines recognized their modification as essentially involving a variable learning rate, nor has such an approach been applied to learning in stochastic games.

5 Results

In this section we show results of applying policy hill-climbing and WoLF policy hill-climbing to a number of different games, from the multiagent reinforcement learning literature. The domains include two matrix games that help to show how the algorithms work and the effect of the WoLF principle on convergence. The algorithms were also applied to two multi-state SGs. One is a general-sum grid world domain used by Hu & Wellman [1998]. The other is a zero-sum soccer game introduced by Littman [1994].

The experiments involve training the players using the same learning algorithm. Since PHC and WoLF-PHC are rational, we know that if they converge against themselves, then they must have converged to a Nash equilibrium. For the matrix game experiments $\delta_l/\delta_w = 2$, but for the other results a more aggressive $\delta_l/\delta_w = 4$ was used. In all cases both the δ and α were decreased proportionately to $1/C(s)$, although the exact proportion varied between domains.

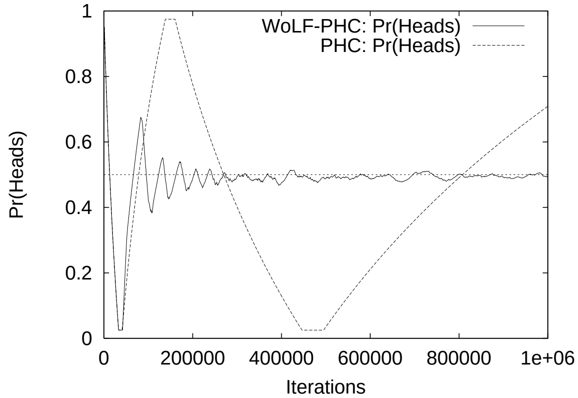
5.1 Matrix Games

The algorithms were applied to the two matrix games, from Figure 1. In both games, the Nash equilibrium is a mixed policy consisting of executing the actions with equal probability. The large number of trials and small ratio of the learning rates were used for the purpose of illustrating how the algorithm learns and converges.

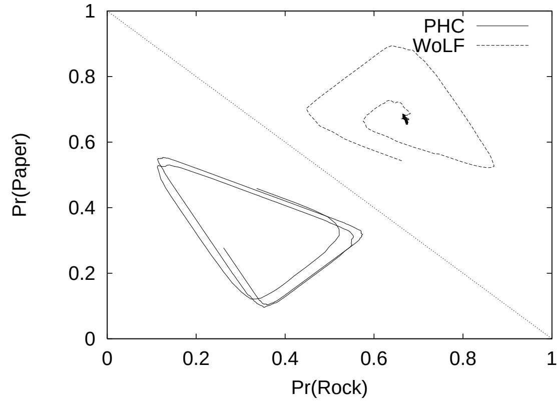
The results of applying both policy hill-climbing and WoLF policy hill-climbing to the matching pennies game is shown in Figure 2(a). WoLF-PHC quickly begins to oscillate around the equilibrium, with ever decreasing amplitude. On the other hand, PHC oscillates around the equilibrium but without any appearance of converging. This is even more obvious in the game of rock-paper-scissors. The results are shown in Figure 2(b), and show trajectories of the players' strategies in policy space through one million steps. Policy hill-climbing circles the equilibrium policy without any hint of converging, while WoLF policy hill-climbing very nicely spirals towards the equilibrium.

5.2 Gridworld

We also examined a gridworld domain introduced by Hu and Wellman [1998] to demonstrate their extension of Minimax-Q to general-sum games. The game consists of a small grid shown in Figure 3(a). The agents start in two corners and are trying to reach the goal square on the opposite wall. The players have the four compass actions (i.e. N, S, E, and W), which are in most cases deterministic. If the two players attempt to move to the same square, both moves fail. To make the game interesting and force the players to interact, while in the initial starting position the North action is uncertain, and is only executed with probability 0.5. The optimal path for each agent is to move laterally on the first move and then move North to the goal, but if both players move laterally then the actions will fail. There are two Nash equilibria for this game. They involve one player taking the lateral move



(a) Matching Pennies Game



(b) Rock-Paper-Scissors Game

Figure 2: (a) Results for matching pennies: the policy for one of the players as a probability distribution while learning with PHC and WoLF-PHC. The other player's policy looks similar. (b) Results for rock-paper-scissors: trajectories of one player's policy. The bottom-left shows PHC in self-play, and the upper-right shows WoLF-PHC in self-play.

and the other trying to move North. Hence the game requires that the players coordinate their behaviors.

WoLF policy hill-climbing successfully converges to one of these equilibria. Figure 3(a) shows an example trajectory of the players' strategies for the initial state while learning over 100,000 steps. In this example the players converged to the equilibrium where player one moves East and player two moves North from the initial state. This is evidence that WoLF policy hill-climbing can learn an equilibrium even in a general-sum game with multiple equilibria.

5.3 Soccer

The final domain is a comparatively large zero-sum soccer game introduced by Littman [1994] to demonstrate Minimax-Q. An example of an initial state in this game is shown in Figure 3(b), where player 'B' has possession of the ball. The goal is for the players to carry the ball into the goal on the opposite side of the field. The actions available are the four compass directions and the option to not move. The players select actions simultaneously but they are executed in a random order, which adds non-determinism to their actions. If a player attempts to move to the square occupied by its opponent, the stationary player gets possession of the ball, and the move fails. Unlike the grid world domain, the Nash equilibrium for this game requires a mixed policy. In fact any deterministic policy (therefore anything learned by an single-agent learner or JAL) can always be defeated [Littman, 1994].

Our experimental setup resembles that used by Littman in order to compare with his results for Minimax-Q. Each player was trained for one million steps. After training, its policy was fixed and a challenger using Q-learning was trained against the player. This determines the learned policy's worst-case performance, and gives an idea of how close the player was to the equilibrium policy, which would perform no worse than losing half its games to its challenger. Unlike Minimax-Q, WoLF-PHC and PHC generally oscillate around the target solution. In order to account for this in the results, training was continued for another 250,000 steps and evaluated after every 50,000 steps. The *worst performing policy* was then used for the value of that learning run.

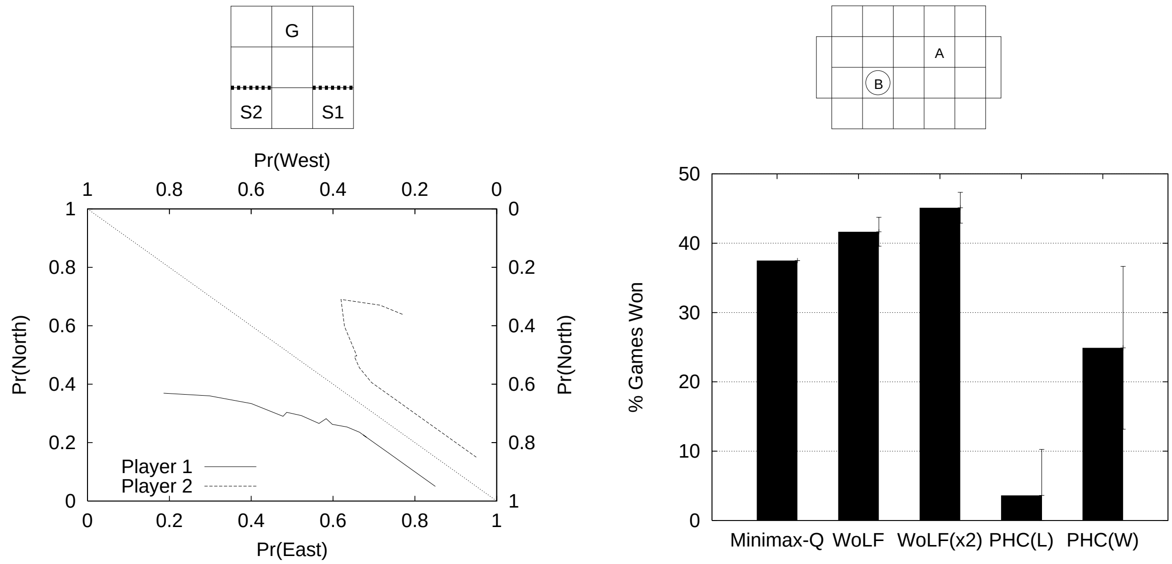
Figure 3(b) shows the percentage of games won by the different players when playing their challengers. "Minimax-Q" represents Minimax-Q when learning against itself (the results were taken from Littman's original paper.) "WoLF" represents WoLF policy hill-climbing learning against itself. "PHC(L)" and "PHC(W)" represents policy hill-climbing with $\delta = \delta_l$ and $\delta = \delta_w$, respectively. "WoLF(2x)" represents WoLF policy hill-climbing learning with twice the training (i.e. two million steps). The performance of the policies were averaged over fifty training runs and the standard deviations are shown by the lines beside the bars. The relative ordering by performance is statistically significant.

WoLF-PHC does extremely well, performing equivalently to Minimax-Q with the same amount of training[2] and continues to improve with more training. The exact effect of the WoLF principle can be seen by its out-performance of PHC, using either the larger or smaller learning rate. This shows that the success of WoLF-PHC is not simply due to changing learning rates, but rather to changing the learning rate at the appropriate time to encourage convergence.

6 Conclusion

In this paper we present two properties, rationality and convergence, that are desirable for a multiagent learning algorithm. We present a new algorithm that uses a variable learning rate based on the WoLF ("Win or Learn Fast") principle. We then showed how this algorithm takes large steps towards achieving these properties on a number and variety of stochastic games. The algorithm is rational and is shown empirically to converge in self-play to an equilibrium even in games with multiple or mixed policy equilibria, which previous multiagent reinforcement learners have not achieved.

[2]The results are not directly comparable due to the use of a different decay of the learning rate. Minimax-Q uses an exponential decay that decreases too quickly for use with WoLF-PHC.



(a) Gridworld Game

(b) Soccer Game

Figure 3: (a) Gridworld game. The dashed walls represent the actions that are uncertain. The results show trajectories of two players' policies while learning with WoLF-PHC. (b) Soccer game. The results show the percentage of games won against a specifically trained worst-case opponent after one million steps of training.

Acknowledgements. Thanks to Will Uther for ideas and discussions. This research was sponsored by the United States Air Force under Grants Nos F30602-00-2-0549 and F30602-98-2-0135. The content of this publication does not necessarily reflect the position or the policy of the sponsors and no official endorsement should be inferred.

References

[Baird and Moore, 1999] L. C. Baird and A. W. Moore. Gradient descent for general reinforcement learning. In *Advances in Neural Information Processing Systems 11*. The MIT Press, 1999.

[Blum and Burch, 1997] A. Blum and C. Burch. On-line learning and the metrical task system problem. In *Tenth Annual Conference on Computational Learning Theory*, 1997.

[Bowling and Veloso, 2001] M. Bowling and M. Veloso. Convergence of gradient dynamics with a variable learning rate. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001. To Appear.

[Claus and Boutilier, 1998] C. Claus and C. Boutilier. The dyanmics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press, 1998.

[Fink, 1964] A. M. Fink. Equilibrium in a stochastic n-person game. *Journal of Science in Hiroshima University, Series A-I*, 28:89–93, 1964.

[Hu and Wellman, 1998] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 242–250, 1998.

[Jaakkola *et al.*, 1994] T. Jaakkola, S. P. Singh, and M. I. Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Advances in Neural Information Processing Systems 6*, 1994.

[Littman, 1994] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163, 1994.

[Nash, Jr., 1950] J. F. Nash, Jr. Equilibrium points in n-person games. *PNAS*, 36:48–49, 1950.

[Shapley, 1953] L. S. Shapley. Stochastic games. *PNAS*, 39:1095–1100, 1953.

[Singh *et al.*, 2000a] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 2000.

[Singh *et al.*, 2000b] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 541–548, 2000.

[Stone and Veloso, 2000] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 2000.

[Sutton and Barto, 1998] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. The MIT Press, 1998.

[Weibull, 1995] J. W. Weibull. *Evolutionary Game Theory*. The MIT Press, 1995.

[Weiß and Sen, 1996] G. Weiß and S. Sen, editors. *Adaptation and Learning in Multiagent Systems*. Springer, 1996.