Joint Action Learners in Competitive Stochastic Games

A thesis presented

by

Ivo Parashkevov

to

Computer Science

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

January 2, 2007

Contents

Abstract

This thesis investigates the design of adaptive utility maximizing software agents for competitive multi-agent settings. The focus is on evaluating the theoretical and empirical performance of Joint Action Learners (JALs) in settings modeled as stochastic games. JALs extend the well-studied Q-learning algorithm. A previously introduced JAL optimizes with respect to stationary or convergent opponents and outperforms various other multi-agent learning algorithms from the literature. However, its deterministic best-response dynamics do not allow it to perform well in settings where non-determinism is required. A new JAL is presented which overcomes this limitation. Non-determinism is achieved through a randomized action selection mechanism discussed in the game theory community.

The analysis of JALs is conducted with respect to a new set of evaluation criteria for self-interested agents. Further research is required before all criteria could be met reliably. In addition, some learning desiderata prove impossible to achieve in settings where the rewards of the opponents are not observable.

Chapter 1

Introduction

In a world economy of self-interested, utility maximizing individuals and organizations, *software agents* can supplement humans as decision makers. They can bid in auctions, conduct negotiations using predefined protocols of communication, and execute trades. To be effective, however, such agents need to be *adaptive* and *learn* from repeated interaction with other agents and the environment. This would allow them to optimize in the presence of unforeseen circumstances or changes in the environment.

This thesis is concerned with the design of adaptive self-interested artificial agents for competitive, multi-agent settings. The main object of investigation is the design of algorithms that allow such agents to perform well against sophisticated adversaries.

Designing software algorithms for multi-agent learning (MAL) poses unique challenges. Traditional reinforcement learning approaches in artificial intelligence inform the design of algorithms for complex, but *static* environments. In multi-agent systems, however, traditional notions of optimality no longer hold, as outcomes are dependent on the behavior of all agents involved. The matter is further complicated by the fact that the other agents may also be learning and adapting.

While MAL is a relatively novel topic within the AI community, it has been investigated by economists since the early days of game theory. Consequently, AI researchers have adopted many game theoretic solution concepts and approaches.

Stochastic games are an example of a game theoretic construct that has been embraced by AI in modeling multi-agent interaction [Bowling and Veloso, 2000]. In a stochastic game the world is assumed to have a finite number of states and agents can take a finite number of actions. This thesis adopts the stochastic game framework.

The intensified exchange between the AI and game theory communities has led to prolific research and numerous innovations in MAL. However, game theorists have traditionally pursued different agendas. For example, they are frequently interested in *describing* and *predicting* the behavior of natural agents such as humans and organizations, as opposed to *designing* artificial ones. There has been a multiplicity of agendas within AI as well. Instead of self-interested agents, some researchers have focused on designing cooperative agents that strive to maximize the utility of the entire system, or of some central mechanism.

Consequently, there has been a wide variety (and a certain lack of clarity) of objectives pursued in the MAL literature. Few of the existing MAL algorithms are applicable or well-motivated for competitive settings. In addition, any attempts to specify evaluation criteria for success in designing self-interested agents have been only partially successful. Fortunately, a debate on the issue was recently begun and is currently picking up momentum[1]

One class of algorithms readily applicable to competitive stochastic games is that of Joint Action Learners (JALs) [Claus and Boutilier, 1998]. JAL algorithms are a multi-agent extension to the widely studied Q-learning [Watkins and Dayan, 1992]. They were originally discussed in coordination settings, but are also well-motivated for the design of self-interested agents. In addition to their broad applicability, JALs are appealing for their simplicity, speed, and low informational requirements.

Investigating JALs is at the focus of this thesis. Theoretical and empirical analysis is conducted with respect to a new set of evaluation criteria informed by previous work in the field. The adopted criteria are: *Rationality* – the ability to learn a best-response against stationary (or convergent) opponents, *Safety* – the ability to obtain the safety value of a game, and *Constant Adaptability* – the ability to remain equally

[1]For a good discussion on this topic see [Shoham *et al.*, 2006]

adaptive to changes in the environment throughout the learning process. This set is certainly not definitive, as there could be more and more stringent requirements for the performance of artificial agents in competitive games. However, it provides for a useful discussion, and helps identify ways to improve current approaches.

In discussing possible evaluation criteria, the thesis presents a desirable property of MAL that is impossible to guarantee for every game. It concerns the reachability of the main solution concept adopted in game theory – Nash equilibrium. More specifically, it is impossible to guarantee that a learner which does not observe the rewards of the opponents will converge on a Nash equilibrium strategy if all opponents have adopted a stationary Nash strategy. Therefore, we cannot guarantee *Stability* of the learning process.

The analysis of JALs begins with a discussion of the theoretical behavior of the JAL algorithm previously implemented in the reinforcement learning literature. This algorithm – FP-Q – can provably meet the *Rationality* criterion. However, it suffers from a major drawback – it cannot learn non-deterministic policies, and therefore cannot exhibit *Safety*. In addition, as any other variant of the widely studied Q-learning, it cannot exhibit *Constant Adaptibility*.

In an attempt to expand the JAL class and address FP-Q's inability to learn non-deterministic policies, the thesis presents a novel algorithm named Smooth FP-Q. Non-determinism is achieved by adopting a randomized action selection mechanism inspired by work done on extending the fictitious play algorithm from the game theory community.

FP-Q and Smooth FP-Q are evaluated empirically against a variety of opponents on different stochastic games. The tests demonstrate that FP-Q can do well in practice, as long as the game being played has a deterministic equilibrium. Other multi-agent extensions of Q-learning, such as WoLF-PHC [Bowling and Veloso, 2002], Nash-Q [Hu and Wellman, 2003], and CE-Q [Greenwald and Hall, 2003], exhibit inferior performance. Smooth FP-Q improves on FP-Q by being able to obtain the safety value of a game with a unique, non-deterministic equilibrium. In addition, it is capable of learning beneficial non-deterministic policies for large stochastic games.

To summarize, the major contributions of this thesis are listed in decreasing order of importance:

- Smooth FP-Q – a new Joint Action Learner capable of playing non-deterministic policies. In empirical tests, Smooth FP-Q is able to obtain the safety value of a zero-sum game against optimal opponents. In addition, it can learn useful mixed policies for large stochastic games.
- A new theoretical result: unless the rewards of all opponents are observable, it is impossible to guarantee convergence to a Nash equilibrium strategy if opponents have adopted a stationary Nash strategy.
- A new set of evaluation criteria for learning algorithms in competitive stochastic games. The criteria are informed by considerations previously put forth in the multi-agent learning literature, as well as the new theoretical result above.
- Empirical and theoretical analysis of FP-Q – the original JAL. In a tournament setting, FP-Q outperforms other multi-agent extensions of Q-learning, such as Nash-Q, CE-Q, and WoLF-PHC.

The thesis is organized as follows. Chapter 2 introduces notation and fundamental concepts relevant to the material at hand. Chapter 3 offers a survey of existing MAL approaches, covering work done in AI and game theory. Particular attention is paid to previous work on Joint Action Learners. Chapter 4 discusses the research agenda of this thesis in more detail. After a survey of evaluation criteria previously put forth in the literature, it presents the criteria adopted in this work. In addition, it demonstrates that not all learning desiderata are achievable with an impossibility result. Chapter 5 discusses Joint Action Learners with respect to the criteria defined previously. It also presents the novel Smooth FP-Q, which overcomes FP-Q's inability to play non-deterministic policies. Chapter 6 offers an extensive empirical investigation of JAL previously implemented in the literature, and demonstrates good performance against a variety of opponents on several stochastic games. Chapter 7 contains empirical tests on Smooth FP-Q and demonstrates its ability to learn useful

mixed policies. Finally, Chapter 8 offers concluding remarks and discusses potential venues of future research.

Chapter 2

Preliminaries

This chapter establishes the notation and terminology from the fields of reinforcement learning and game theory that are utilized in subsequent chapters.

2.1 Markov Decision Processes

The framework traditionally employed for modeling single-agent decision problems – *Markov Decision Process* – is ordinarily formalized as follows:

Definition 1. A Markov Decision Process (MDP) is a tuple $\langle S, A, R, T \rangle$, where:

- S is the set of possible states of the world. S is assumed to be finite.
- A is the set of possible actions available to the agent. A is assumed to be finite.
- $R : S \times A \to \mathbb{R}$ is the reward (utility) function of the agent
- $T : S \times A \to \Delta(S)$ is the transition model, where $\Delta(S)$ is the set of probability distributions over the state space S. $T(s, a, s')$ denotes the probability of going from state s to state s' after taking action a.

The model describes the interaction of an agent with the world. The agent starts at some state $s_0 \in S$, takes an action $a \in A$, receives reward $R(s, a)$, and the world transitions to a new state $s' \in S$, based on the probability distribution $T(s, a)$.

As the name of the model indicates, it makes a fundamental assumption about the world, known as the *Markov* assumption. The Markov assumption is often stated as "The future is independent of the past, given the present." More specifically, the probability of arriving at state s' in time t depends only on the current state s and the action chosen by the agent. The history of states visited and actions taken prior to time t does not matter.

2.1.1 MDP solution concepts

The MDP framework does not specify the objective of the agent, i.e., what the "solution" of the MDP looks like. One commonly adopted objective for the agent is to find a policy π so as to maximize the sum of the total discounted expected rewards,

$$V^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t E(R_t(s,\pi)|s_0 = s), \tag{2.1}$$

where $s \in S$, s_0 is the starting state, $R_t(s,\pi)$ is the reward at state s for playing as prescribed by policy π at time t, and γ is the discount factor. A policy, in this context, is a probability distribution over all actions, defined for each state.

The optimal policy π^* can be found by solving the following Bellman equation [Bellman, 1957]:

$$V^{\pi^*}(s) = \max_a \{R(s,a) + \gamma \sum_s T(s,a,s')V^{\pi^*}(s')\} \tag{2.2}$$

The solution of this equation can be obtained by an iterative search method.

The Bellman equation allows us to define the state-action value function, also known as Q-function (or Q-values). A Q-value for a given state-action pair, $Q^{\pi}(s,a)$, defines the expected reward for choosing action a in state s, assuming the agent follows policy π from that point on. Formally,

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V^{\pi}(s') \tag{2.3}$$

The theory of Markov Decision Processes assumes that the environment is stationary and as such contains no other adaptive agents. The multi-agent extension of an MDP is called a stochastic game. Before delving into stochastic games, however, matrix games, the simplest form of multi-agent environment, are reviewed.

2.2 Matrix Games

Matrix games were developed in the field of game theory to model strategic settings in which the outcome depends on the actions of more than one agent[1]. A *matrix game* [Osborne and Rubinstein, 1994] is a tuple $\langle n, A_{1...n}, R_{1...n} \rangle$, where:

- n is the number of players
- A_i is the set of actions available to player i, and $\mathrm{A} = (A_1 \times A_2 \times \ldots \times A_n)$ is the joint action space
- $R_i : A_1 \times A_2 \times \cdots \times A_n \rightarrow \mathbb{R}$ is the reward function of player i.

Such models are called matrix games because the functions R_i can be written as n-dimensional matrices. In this thesis, the discussion of matrix games will be confined to the two-player case, $n = 2$.

Matrix games are one-shot because all players engage in decision-making only once, simultaneously. They can choose to play a particular action $a_i \in A_i$, or *mixed strategy* $\sigma_i \in \Delta(A_i)$, where $\Delta(A_i)$ is the probability distribution over the actions available to player i. An action a_i is in the *support* of a mixed strategy σ_i if the probability of playing a_i under the distribution defined by σ_i is strictly positive. A *pure strategy* is a strategy with a single action in its support.

Note that in the discussion of matrix games, the term *strategy* was used to refer to a probability distribution over the action space, while the term *policy* was used in discussing MDPs. The two terms are often used interchangeably in the learning literature. However, the distinction is useful. In this thesis, a strategy is defined for

[1]Throughout this thesis, we use the terms "agent" and "player" interchangeably

$$R_1 = \begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix} R_2 = \begin{pmatrix} 3 & 5 \\ 0 & 1 \end{pmatrix} \qquad R_1 = \begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix} R_2 = \begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} R_2 = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

Figure 2.1: Examples of matrix games. To the left: Prisoners' Dilemma. To the right: a coordination game. At the bottom: Rock Paper Scissors.

a single state, while policies can be defined for many states. Thus, a policy can be thought of as a collection of strategies, one for each state.

2.2.1 Examples

Any one-shot strategic interaction with a finite number of players and actions can be modeled as a matrix game. Figure 2.1 provides a few examples. The game in the upper left corner is a version of the Prisoners' Dilemma, which has been extensively studied in the game theory literature. To the right is an example of a *common payoff* game, which has the property that the reward to all players is equal for each joint action, i.e., the reward matrices of the players are identical. Such games are also commonly referred to as *coordination games*, as players are interested in coordinating on which joint action to play so as to maximize their common payoff.

The bottom of the figure depicts a model of the game of Rock Paper Scissors (RPS). RPS is an example of a *zero-sum game*, defined as a game in which the sum of the reward of all players for each joint action is 0.

2.2.2 Matrix game solution concepts

In order to discuss solution concepts in matrix games, it would be useful to introduce some additional notation. The strategy of agent i is denoted by σ_i, and the strategy of all players, called *joint strategy*, is denoted by σ. Additionally, σ_{-i} denotes the joint strategy of all players but agent i. Writing $\langle \sigma_i, \sigma_{-i} \rangle$ signifies player i following

strategy σ_i, and all other players following their respective part of the joint strategy σ_{-i}. By definition, $\sigma = \langle \sigma_i, \sigma_{-i} \rangle$. The definition of the reward function R_i can now be extended over joint strategies as well:

$$R_i(\sigma) = \sum_{a \in A} R_i(a) \prod_{i=1}^{n} \sigma_i(a_i). \tag{2.4}$$

In single agent environments such as MDPs, a solution concept is well-defined for a given reward formulation. In matrix games and multi-agent settings in general, however, no one strategy can be considered optimal, as outcomes depend on the actions of all agents. Thus, optimality can be discussed only with respect to what all other agents are doing. An opponent-dependent notion of optimality is that of the *best-response*.

Definition 2. For a matrix game, the *best-response* of player i to the joint strategy of its opponents, $BR_i(\sigma_{-i})$, is the set of all strategies that are optimal given σ_{-i}. Formally, $\sigma_i^* \in BR_i(\sigma_{-i})$ if and only if

$$\forall \sigma_i \in \Delta(A_i) \qquad R(\langle \sigma_i^*, \sigma_{-i} \rangle) \geq R_i(\langle \sigma_i, \sigma_{-i} \rangle), \tag{2.5}$$

where $\Delta(A_i)$ is the set of all probability distributions over A_i.

The solution concept most frequently adopted in matrix games is that of the *Nash equilibrium* [Nash, 1951], in which all players best-respond to each other.

Definition 3. A *Nash equilibrium* is a joint strategy σ such that

$$\forall i \qquad \sigma_i \in BR_i(\sigma_{-i}) \tag{2.6}$$

One reason why Nash equilibria have been a central object of investigation for game theorists is that they have been proven to exist for all games, as long as mixed strategies are allowed [Nash, 1951]. There are other, stronger notions of equilibria, but they do not exist for all games, and are not covered here.

Unfortunately, the notion of Nash equilibrium is rather problematic. For one, it

gives no intuition as to how a multi-agent system may reach it. From a computer science perspective, Nash equilibria are troublesome because their efficient computation is often quite hard. In fact, the computational complexity of finding a Nash equilibrium for general matrix games was one of the open problems in computer science for a long time. Recent results by Daskalakis, et. al. [2005] and Cheng and Deng [2005a; 2005b] demonstrated that the problem is PPAD-complete.

An additional reason why Nash equilibria are problematic is that there could be more than one of them for a given game. This begs the question of why any player should assume that the others would play a particular Nash equilibrium strategy. Even if all Nash equilibria are known to all players, they would still have to choose which one to play. This is known as the *equilibrium selection* problem [Myerson, 1991].

Another solution concept from game theory is that of *correlated equilibrium* [Foster and Vohra, 1997]. A *correlated strategy profile* is a function h from a finite probability space Γ into $\Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_n$, i.e., $h = (h_1, h_2, \ldots h_n)$ is a random variable whose values are sets of strategies, one for each player.

Definition 4. A correlated strategy profile h is called *correlated equilibrium* if

$$\forall g \in \Gamma \quad \forall \Phi : \Sigma_i \to \Sigma_i, \forall i \qquad R(\langle h_i(g), h_{-i}(g)\rangle) \geq R(\langle \Phi(h_i(g)), h_{-i}(g)\rangle) \tag{2.7}$$

To understand the notion of correlated equilibrium, imagine that an umpire announces to all players what Γ and h are. Chance chooses an element $g \in \Gamma$ and hands it to the umpire, who computes $h(g)$, and reveals $h_i(g)$ to player i, and nothing more. Under a correlated equilibrium, a player has no incentive to deviate from the recommendation of the umpire, assuming that the other players will not deviate either. All Nash equilibria are also correlated equilibria, which means that the latter are a more general solution concept. Correlated equilibria have the benefit of being computable using linear programming. However, there can be more than one correlated equilibrium for a given game.[2]

[2]The assumed presence of an umpire obviously solves the multiple equilibria problem, as the umpire selects which equilibrium should be played. However, the problem remains if we try find

In evaluating game equilibria and game outcomes in general, economists often resort to the notion of Pareto-optimality.[3]

Definition 5. An outcome of a game is *Pareto-optimal* if there is no other outcome that makes every player at least as well off and at least one player strictly better off.

The similar idea of dominance is applied in evaluating strategies.

Definition 6. Player i's strategy σ_i is said to strictly dominate strategy σ_i' if

$$\forall \sigma_{-i} \qquad R(\sigma_i, \sigma_{-i}) > R(\sigma_i', \sigma_{-i}). \tag{2.8}$$

Similarly, σ_i weakly dominates σ_i' if

$$\forall \sigma_{-i} \qquad R(\sigma_i, \sigma_{-i}) \geq R(\sigma_i', \sigma_{-i}). \tag{2.9}$$

It is sometimes possible to find Nash equilibria by iterated elimination of strictly dominated strategies. There are games in which a repeated process of elimination of strictly dominated pure strategies leaves only one pure strategy for each player, which must be a Nash equilibrium.

Playing a Nash equilibrium for a given game does not guarantee Pareto-optimal outcomes. An example that illustrates this is the Prisoners' Dilemma game (Figure 2.1). The game models a situation in which the two players (prisoners) can either "cooperate" or "defect" from a cooperative arrangement (betray the opponent). In this game, cooperating is strictly dominated by defecting, and so the only Nash equilibrium of the game is to defect. However, defecting yields $R_1 = R_2 = 1$, while cooperating yields $R_1 = R_2 = 3$ – the Nash equilibrium is Pareto-dominated by another outcome.

correlated equilibria using linear programming.

[3]Named after the Italian economist Vilfredo Pareto. Pareto argued that individual preferences lie at the core of economic decisions and analysis, and only *ordinal*, and not *cardinal* payoffs are important.

2.3 Stochastic Games

A stochastic game is a tuple $\langle n, S, A_{1\ldots n}, T, R_{1\ldots n}\rangle$, where

- n is the number of agents.
- S is the set of states. S is assumed to be finite. In addition, it is assumed here that there is a set $S^T \subseteq S, S^T \neq \emptyset$ of terminal states, i.e., "game over" states. There is also a set $S^0 \subseteq S, S^0 \neq \emptyset$ of starting states, i.e., states in which agents can start playing the game.
- A_i is the set of actions available to player i, and A $= (A_1 \times A_2 \times \ldots \times A_n)$ is the joint action space.
- $T : S \times A \rightarrow \Delta(S)$ is the transition model, where $\Delta(S)$ is the set of probability distributions over the state space S. Note that A here is the set of all joint actions. Consistent with the notation for MDPs, $T(s, \vec{a}, s')$ denotes the probability of going from state s to state s' after the joint action $\vec{a}$.
- $R_i : S \times A_1 \times A_2 \times \cdots \times A_n \rightarrow \mathbb{R}$ is the reward function of player i.

Notice that stochastic games involve multiple states and multiple agents. Thus, they can be thought of as a multi-agent extension to MDPs, or a multi-state extension of matrix games. More specifically, a stochastic game has a matrix game associated with each state. Given a state s, agent i chooses an action a_i based on its policy $\pi_i(s)$, observes reward $R_i(s, \vec{a})$, and the world transitions to a state s' determined by $T(s, \vec{a})$.

Since stochastic games are multi-state, players have *policies* which could be thought of as a collection of *strategies*, one for each state of the world. Formally, $\pi_i \in S \times A_i \rightarrow [0, 1]$, where

$$\forall s \in S \qquad \sum_{a_i \in A_i} \pi(s, a_i) = 1$$

Throughout this thesis, π_i refers to the policy of player i, and π refers to the joint policy of all players. Sometimes $\vec{\pi}$ is used to emphasize a reference to joint policies.

The use of π_{-i} refers to the joint policy adopted by all agents except for agent i. Additionally, Π_i denotes the set of all possible policies of agent i. Finally, $\langle \pi_i, \pi_{-i} \rangle$ describes a joint policy, in which agent i plays π_i, and all other agents play π_{-i}.

As per the definition above, the policies considered in this thesis are *stationary*, as they depend solely on the current state. An example of non-stationary policies are *behavioral policies*, which also depend on the history of play. Such policies are more complex and relatively less well-studied in the stochastic game framework.

When the policy of all opponents is stationary, the stochastic game is equivalent to an MDP. Formally, if T is the stochastic game transition function and $\bar{\pi}_{-i} \in \Pi_{-i}$ is the stationary joint policy of all opponents, then the stochastic game is equivalent to an MDP with transition function

$$\hat{T}(s, a_i, s') = \sum_{a_{-i} \in A_{-i}} \bar{\pi}_{-i}(s, a_{-i}) T(s, \langle a_i, a_{-i}, \rangle, s'). \tag{2.10}$$

2.3.1 Stochastic game solution concepts

As is the case with MDPs, the objective of an agent in a stochastic game is not completely defined by the model itself as the aggregation of reward with respect to time is left unspecified. In this work, the discounted reward framework is adopted. Formally, the value of a joint policy π to agent i at state s, given some discount factor $\gamma \in (0, 1)$, is

$$V_i^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t E\left(R_{i,t}(s, \pi) | s_0 = s\right) \tag{2.11}$$

where $s \in S$, s_0 is the starting state, and $R_{i,t}(s, \pi)$ is the reward to agent i at time t for playing as prescribed by policy π_i in state s.

Since stochastic games are multi-agent environments, the policy of a given agent cannot be evaluated independently from the policies of all other agents. Thus, the solution concepts of matrix games need to be extended to the stochastic games framework.

Definition 7. For a stochastic game, the *best-response* function for player i, $BR(\pi_{-i})$,

is the set of all policies that are optimal, given the other players' joint policy π_{-i}. Formally, $\pi_i^* \in BR(\pi_{-i})$ if and only if

$$\forall \pi_i \in \Pi_i, \forall s \in S \qquad V_i^{\langle \pi_i^*, \pi_{-i} \rangle}(s) \geq V_i^{\langle \pi_i, \pi_{-i} \rangle}(s) \tag{2.12}$$

Definition 8. For a stochastic game, a *Nash equilibrium* is a joint policy π such that

$$\forall i \qquad \pi_i \in BR(\pi_{-i}) \tag{2.13}$$

Fink (1964) demonstrated that every n-player discounted stochastic game possesses at least one stationary policy Nash equilibrium. In addition, the notions of strategy dominance and Pareto-optimality defined for matrix games extend to policies for stochastic games in a straightforward fashion.

Research in game theory is often concerned with a special case of games called *repeated games*. The term usually refers to infinitely repeated, fully observable matrix games. This is partly because, under the *folk theorems* [Osborne and Rubinstein, 1994], playing joint strategies that lead to Pareto-optimal outcomes of the matrix game every time is a Nash equilibrium of the infinitely repeated matrix game for some positive discount factor.[4]

All infinitely repeated games have a single state and a single matrix game associated with it. The repeated game starts in this state, agents act, and the world invariably transitions to the same state again. Since the same matrix game is repeated infinitely many times, there are no terminal states. In this thesis, it is assumed that at least one terminal state is reachable from all starting states. In other words, there is always a way to end a game, regardless of the way it started or the agents' play. Therefore, the class of infinitely repeated games is not discussed here.

[4] In fact, the folk theorems state that all outcomes with payoffs above players' safety values are obtainable in equilibrium in an infinitely repeated setting for some positive discount factor. The safety value for a given player is the value obtained by minimizing the maximum possible loss.

2.4 Learning in Stochastic Games

The idea of learning in game theory began as a means to compute the policies and associated values to all players of a Nash equilibrium in a known game [Brown, 1951]. This work considers stochastic games that are not fully known by a participating agent. More specifically, T, the transition model, and R_i, the reward function of agent i, are unknown, but could be observed through repeated interaction with the agents and the environment. Thus, *learning* will mean the process of finding and identifying beneficial policies for a given agent *through a repeated play of the stochastic game.* Repeating the stochastic game many times requires that each game iteration always ends for some finite time, which is why game terminability is assumed.

The phrase "beneficial policies" above is deliberately vague, as it does not address an obvious question – to whom should the learned policy be beneficial? This thesis considers the design of self-interested, competitive agents. Therefore, desired outcomes of learning processes discussed here are policies that are beneficial to the learning agent itself. Note that there have been other research agendas pursued in the literature. For example, one might be interested in designing agents that can learn policies to maximize the utility of all agents, or lead to other, system-wide properties. Section 4.1 offers a more thorough discussion of multi-agent learning research agendas.

The learner starts playing a game with incomplete information about the reward structure or the strategy of the opponent. It does not have any control over the policies of the other agents or any other aspect of the environment. In the course of game play, it can observe and accumulate some fraction of all relevant information, which it can use to alter its policy so as to maximize expected payoff. Informally, a learning rule (or algorithm) provides a mapping from the information available to the space of policies. A few relevant terms need to be defined before a more rigorous definition can be provided.

Definition 9. A *history* of a game h is a sequence of sets $\tau_t = (s, \vec{\sigma}, \vec{a}, \vec{R})$, where

- t denotes time, i.e. number of joint actions taken,

- s is the state at time t,
- $\vec{\sigma}$ is the joint strategy vector at time t
- $\vec{a}$ is the joint action vector at time t,
- $\vec{R}$ is the reward vector at time t,

and all pairs of consecutive sets, $\tau_t = \langle s, \vec{\sigma}, \vec{a}, \vec{R} \rangle, \tau_{t+1} = \langle s', \vec{\sigma}'\vec{a}', \vec{R}' \rangle$ satisfy one of the following conditions:

- $T(s, \vec{a}, s') > 0$
- if $s \in S^T$ then $s' \in S^0$.

Definition 10. An *iteration* of a game is a history with a single terminal state.

Definition 11. Let $\tau_{t,i}$ be the subset of τ_t that is observable to agent i. An *observable history* h_i is the sequence of $\tau_{t,i}, \forall \tau_t \in h$.

A formal definition of a learning rule is now presented.

Definition 12. A *learning rule* is a function $\Phi : H_i \rightarrow \Pi_i$ that provides a unique mapping from the set of histories observable by agent i to the set of policies available to it.

Note that a learning rule is not equivalent to a behavioral policy. Unlike learning rules, behavioral policies represent mappings from the set of histories observable by agent i *and* the current state s to the set of *strategies* Σ_i available for state s.

2.5 Summary

This chapter defined the notions of Markov Decision Process, matrix game, and stochastic game, and introduced relevant notation. The reward formulation to be used throughout this thesis were specified, in addition to other relevant details of the models. The solution concepts associated with each model were also described.

The next chapter investigates existing ways in which an agent might learn to play a game effectively.

Chapter 3

Multi-Agent Learning in Literature

This chapter reviews some of the more prominent algorithms and results in the multi-agent learning (MAL) literature. This review is not exhaustive and it is not meant to be. The focus is on well-understood, "classical" results, which have largely shaped the direction of the field. From the game theory literature, a few model-based algorithms are presented. From the AI literature, the chapter traces the "Bellman heritage" – the evolution of single-agent reinforcement learning techniques and their adaptation to multi-agent settings.

These two bodies of work will inform the analysis of Joint Action Learners (JALs) throughout the remainder of this thesis. Section 3.3 of this chapter reviews previous investigations of JALs in the literature.

One large body of work that is under-represented here is that of no-regret learning. Section 4.2.2 discusses the notion of no-regret. Foster and Vohra [1999] offer a comprehensive review of the classical results in this literature.

3.1 Model-Based Learning

The model-based approach to MAL was historically developed within the game theory community. As suggested by its name, model-based learning revolves around the idea of modeling the behavior of the opponent, and then calculating an adequate

response. Inherently, this approach involves maintaining *beliefs* about the policy of the opponent, which are to be updated based on observations obtained during the learning process.

More specifically, a model-based learning algorithm involves the following general steps:

1. Initialize a model of the opponents based on any prior knowledge.
2. Compute a best-response policy based on this model.
3. Observe the actions of the opponents and update this models accordingly.
4. Go to step 3.

Below, two specific model-based MAL algorithms are reviewed.

3.1.1 Fictitious play

Fictitious play is probably the earliest and most studied multi-agent learning algorithm. It was introduced by Brown [1951] as an iterative way of computing Nash equilibria for matrix games. The name comes from the idea that a player would "simulate" play of the game in her "mind" and decide on her strategy based on this simulation. Thus, fictitious play was also conceived as a justification of Nash equilibria as solution concepts of games. The simplest, two-player version of the algorithm is discussed here. A more comprehensive review and analysis is provided by Fudenberg and Kreps [1993] and Fudenberg and Levine [1998].

Under fictitious play, the key assumption is that the opponent is stationary. Thus, the model of the opponent is simply the empirical frequencies of actions played in the past. Formally, the assumed probability of playing action a'_{-i} is defined by the equation

$$P(a'_{-i}) = \frac{C(a'_{-i})}{\sum_{a_{-i} \in A_{-i}} C(a_{-i})}, \tag{3.1}$$

where $C(a'_{-i})$ is the number of times the opponent has played action a'. For example, in the Rock Paper Scissors game, if the opponent has played {r, p, r, s, p, s, s, r, s, r} in the past, then the assumed model is {P(r) = 0.4, P(p) = 0.2, P(s) = 0.4}.

In traditional fictitious play, the agent computes and plays a best-response to the opponent's model. The payoff matrix is known, and the computation is simple. There may be more than one best-response, in which case some tie-braking mechanism must be defined.

The propositions below describe the asymptotic behavior of fictitious play in self-play.

Proposition 1. *If σ is a pure-strategy Nash equilibrium, and σ is played at any one time t' in the process of fictitious play, then σ is played at all $t \in (t', \infty)$. That is, pure-strategy Nash equilibria are absorbing for the process of fictitious play.*

Proposition 2. *Any pure-strategy steady state of fictitious play must be a Nash equilibrium*

Proposition 3. *Under fictitious play, if the empirical distributions over each player's actions converge, the strategy profile corresponding to the product of these distributions is a Nash equilibrium.*

Refer to [Fudenberg and Levine, 1998] for formal proofs of these propositions.

Proposition 4. *Under fictitious play the empirical distributions converge if the game has generic payoffs and is 2×2 [Miyasawa, 1961], or zero-sum [Robinson, 1951], or is solvable by iterated elimination of strictly dominated strategies [Nachbar, 1990], or has strategic complements and satisfies another technical condition [Krishna and Sjostrom, 1995].*

The results quoted above concern the convergence of empirical distribution of play. This notion of convergence is problematic for a number of reasons. One may argue that the players themselves never learn about the Nash equilibrium that remains observable only to an outsider. Moreover, play could be correlated in a way that leads to both players consistently receiving rewards much lower than the Nash equilibrium

ones. Fudenberg and Levine [1998] present a concrete example with a certain coordination game in which fictitious play learners always end up playing an undesirable strategy profile. The empirical frequencies of each player converge to those of the Nash equilibrium, but the joint probabilities (probabilities over joint actions) do not.

Unfortunately, even the empirical distributions of play do not always converge. Shapley was the first to explicitly prove this [Shapley, 1964]. He presented a game in which play cycles indefinitely between a number of pure-strategy profiles. The number of consecutive periods that each profile in the sequence is played increases sufficiently quickly that the empirical distributions do not converge, but follow a limit cycle.

3.1.2 Rational learning

Rational learning [Kalai and Lehrer, 1993] is a more sophisticated variant of model-based learning in which learning occurs through Bayesian updating of individual prior. Unlike fictitious play, rational learning is designed to learn equilibria of the *repeated* matrix game, i.e., the stochastic game that has infinitely many states, each of which represents the same matrix game. Thus, the opponent model is a probability distribution over repeated game policies. After each play, the model is updated to be the posterior obtained by Bayesian conditioning of the previous model.

Kalai and Lehrer demonstrate that if individual beliefs are compatible with actual play then a best-response to beliefs about the opponent (the model) leads to accurate predictions, and play converges to Nash equilibrium play. "Compatible beliefs" in this context means that players do not assign zero probability to events that can occur in the play of the game. This result rests on the assumption that the other players' actions *and* beliefs are independent from each other.

3.2 Model-Free Learning

The model-free approach has been pursued primarily in Artificial Intelligence under the more general heading of *reinforcement learning* [Kaelbling *et al.*, 1996]. Instead of building an explicit model, a model-free learner tries to learn the value of taking each action in each state. This term is well-established in psychology. The premise there is that natural agents (animals, humans) can learn from the rewards and punishments provided by the environment. A natural agent would learn to repeat an action that leads to positive rewards and avoid actions that result in punishment.

Just like any other learning process, reinforcement learning occurs in repeated interaction with the environment. In general multi-state stochastic games we encounter the problem of determining which particular action we took in the past is responsible for the reward (or punishment) we receive at the end of the game. For example, in the game of chess, a player receives some form of feedback with regards to its performance during game play, but the winner is not determined until the very end, and it is not obvious which moves were responsible for the final outcome. This is known as the *credit assignment problem*.

Another issue that arises in the process of learning is the *exploration-exploitation tradeoff*. The problem is one of determining whether it is better to exploit what is already known about the environment and the other players and best-respond to it, or explore by taking an action that may be suboptimal with the hope that more useful information we will be obtained.

Reinforcement learning in AI has its roots at the Bellman equation (Equation 2.2). Recall that the optimal policy in an MDP can be found by iteratively updating this equation, assuming the reward and transition models are known. When these models are unknown, however, we ordinarily resort to working with the Q-values (Equation 2.3), which are the values for taking an action in a given state. The widely studied single-agent Q-learning and its various multi-agent extensions are examined below.

3.2.1 Single agent Q-learning

Recall that the Q-value function (Equation 2.3) incorporates information about both the immediate reward for taking an action a at state s and the value of the next state reached. Furthermore, the Q-function also determines a policy π_Q and a value function V_Q.

$$\forall s \in S \qquad \pi_Q(s) = \operatorname*{argmax}_a Q(s,a), \tag{3.2}$$

$$\forall s \in S \qquad V_Q(s) = \max_a Q(s,a). \tag{3.3}$$

Therefore, to obtain the optimal policy in an MDP, it would suffice to learn the right Q-values; the transition and reward models do not need to be learned explicitly. This is precisely the idea behind the Q-learning algorithm.

Expanding the Q-function formula demonstrates how one can go about learning the right Q-values.[1]

$$\begin{aligned} Q(s,a) &= R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^{\pi}(s') \\ &= R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_{a'} Q(s',a') \\ &= \sum_{s' \in S} T(s,a,s') \left[R(s,a) + \gamma \max_{a'} Q(s',a') \right] \end{aligned}$$

The last line in this expansion follows because $\sum_{s' \in S} T(s,a,s') = 1$ by definition of the transition model (see Section 2.1). The reward received in each state is observed. The goal is then to learn the Q-values without learning the transition model explicitly. Note that the right hand side of the last equation is the expectation over the distribution defined by $T(s,a,s')$. But every time action a is taken in state s, a sample for estimating this expectation is retrieved. This sample can then be used to update an old estimate for $Q(s,a)$. More specifically, the following update rule can be used:

$$Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \left[R(s,a) + \gamma \max_{a'} Q(s',a') \right] \tag{3.4}$$

[1] Many thanks to Avi Pfeffer for making this clear in his lecture notes.

The parameter $\alpha \in [0, 1]$ is the *learning rate* which determines the weight put on the last sample. The term can be confusing – when $\alpha = 1$, the learner is forgetting all information we previously obtained about a given Q-value, and relies solely on newly arrived information.

Q-learning owes its popularity to the fact that, under certain assumptions, the algorithm converges to the optimal policy [Watkins and Dayan, 1992]. The assumptions are:

i. Every state-action pair is visited infinitely many times

ii. The learning rate α is decayed over time. In particular, if α_t is the learning rate used at time t, then it must satisfy

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \qquad \sum_{t=0}^{\infty} \alpha_t^2 < \infty. \tag{3.5}$$

In practice, the learner cannot visit each state-action pair infinitely many times. The question of how much training suffices is empirical, and depends on the properties of the MDP and the goals of the experimenter.

One issue that needs to be addressed in order to ensure that each state-action pair is visited frequently enough is the exploration-exploitation tradeoff. In particular, the conditions under which the learner will optimize with respect to its current Q-values must be decided upon, or a suboptimal action must be taken in order to obtain new information. Aside from pure randomization, the method used to explore can also be tweaked so that the learner chooses an action that has been tried least often in the current state, or one that leads to states that have not been explored. One frequently adopted exploration scheme ties the probability $P(a|s)$ of playing an action a at state s to the expected value of taking a in s. Such is the Boltzman exploration scheme [Kaelbling *et al.*, 1996], which, in the case of Q-learning, takes the form:

$$P(a|s) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{a' \in A} e^{\frac{Q(s,a')}{T}}}, \tag{3.6}$$

where T is a temperature parameter that depends on time, and can be cooled down.

If the parameter T is cooled appropriately, Boltzman exploration is an example of an exploration policy that is Greedy in the Limit of Infinite Exploration (GLIE). GLIE policies always converge to the optimal solution [Singh *et al.*, 2000], but must satisfy the following conditions:

- Each action is executed infinitely often in every state that is visited infinitely often
- In the limit, the learning policy is greedy with respect to the Q-value function with probability 1.

Another frequently employed exploration policy is that of ϵ-greedy exploration, which at time t in state s picks a random exploration action with probability $\epsilon_t(s)$ and the greedy action with probability $1 - \epsilon_t(s)$. ϵ-greedy exploration is GLIE if $\epsilon_t(s) = c/n_t(s)$, where $c \in (0, 1)$ is a constant, and $n_t(s)$ is the number of times state s has been visited.

Q-learning solves the credit assignment problem by having high rewards propagate back to the Q-values for state-action pairs that originated a good outcome. However, the algorithm is not perfect in this regard, as low rewards do not propagate back as easily.

3.2.2 Q-Learning in multi-agent settings

Q-learning has been studied extensively and enjoys a wide popularity among AI researchers. However, it is intrinsically a single-agent algorithm. It was designed specifically to find optimal policies in MDPs; extending it to multi-agent settings is not straightforward.

A naïve extension to a multi-agent stochastic game setting is to simply ignore the other agents and pretend the game is an MDP. The following describes the Q-update

rule for agent i.

$$Q_i(s, a_i) \leftarrow (1-\alpha)Q_i(s, a_i) + \alpha \left[R_i(s, \vec{a}) + \gamma \max_{a_i'} Q_i(s', a_i') \right] \tag{3.7}$$

Several authors have attempted this approach in multi-agent settings, and not without success(e.g., [Sen *et al.*, 1994; Claus and Boutilier, 1998]).

The approach is not entirely unjustifiable in multi-agent settings, due to the connection between stochastic games and MDPs. Recall that if all opponents adopt a stationary policy, then the stochastic game is reduced to an MDP and Q-learning will necessarily converge to the best-response. However, the approach does not adequately address situations with adaptive opponents.

The naïve extension ignores valuable information about the actions taken by the opponents, which is assumed to be observable. A natural way to address this limitation is to redefine the Q-value function to be the value of playing a certain *joint action* at a given state. If this is done, however, the traditional state-value function (Equation 3.3) will no longer be well-motivated. The reason is that the agent cannot assume control over which joint action is actually taken and therefore cannot assume that it can choose the one that maximizes its expected value. Thus, the Q-update rule becomes

$$Q_i(s, \vec{a}) \leftarrow (1-\alpha)Q_i(s, \vec{a}) + \alpha \left[R_i(s, \vec{a}) + \gamma V_i(s') \right], \tag{3.8}$$

and the way to compute $V_i(s')$ remains an open question.

The following section reviews several specific instantiations of this idea. The algorithms discussed differ from one another in the way they compute $V_i(s')$, as well as the policy that is inherently associated with this computation. One commonality between them is that they all compute an equilibrium of the matrix game defined by the Q-values at state s.

3.2.3 Model-Free Equilibrium Learners

Minimax-Q

In two-player zero-sum matrix games, the value of all Nash equilibria to agent i can be found by solving the following linear program:

$$\begin{aligned} \text{Maximize:} \quad & \min_{a_2 \in A_2} \sum_{a_i \in A_i} \sigma(a_i) R_i(a_i, a_{-i}) \\ \text{Subject to:} \quad & \sigma(a_i) \geq 0 \quad \forall a_i \in A_i \\ & \sum_{a_i \in A_i} \sigma(a_i) = i \end{aligned}$$

Littman [1994] used this idea and extended it to two-player zero-sum stochastic games. He employed the general framework of multi-agent Q-learning reviewed in Section 3.2.2 and computed the value of state s to agent 1, $V_1(s)$, by solving

$$V_1(s) = \max_{\sigma_1 \in \Sigma_1(s)} \min_{a_2 \in A_2(s)} Q_1(s, \langle \sigma_1, a_2 \rangle) \tag{3.9}$$

where $Q_1(s, \langle \sigma_1, a_2 \rangle) = \sum_{a_1 \in A_1} \sigma_1(a_1) Q_1(s, \langle a_1, a_2 \rangle)$. This algorithm was shown to converge to the equilibrium of the stochastic game, assuming the other agent executes all of its actions infinitely many times [Littman and Szepesvari, 1996]. The result holds even if the other player does not converge to the equilibrium.

The time complexity of Minimax-Q is dominated by the complexity of the linear program solver. During the last couple of years there has been a considerable effort in improving existing LP solving techniques, which has led to a large variety of algorithms. Implementations that employ the widespread Simplex algorithm have exponential worst-case behavior, but are quite fast in practice.

Nash-Q

Minimax-Q could be applied to general-sum stochastic games, but would no longer be well-motivated. The reason is that the solution to the linear program described

above does not necessarily represent the value of a Nash equilibrium in general-sum matrix games. However, it appears that the problem could be solved if the minimax linear program were to be replaced by a general Nash solver for matrix games. This is precisely the idea of the Nash-Q algorithm presented by Hu and Wellman [2003]. In their version of multi-agent Q-learning, the value function $V_i(s)$ is

$$V_i(s) \in \text{NASH}_i(\vec{Q}(s, \vec{a})), \tag{3.10}$$

where $\text{NASH}_i(\vec{Q}(s, \vec{a}))$ is the set of values of the Nash equilibria of the matrix games defined by the Q-values of all agents at a given state s.

The approach has proven problematic for a number of reasons. One issue is that there may be multiple Nash equilibria of a general matrix game, and therefore the value function is no longer well defined. To get around this problem, one needs to make sure that all agents compute the value of the same Nash equilibrium (in self-play). To address this, Hu and Wellman use the Lemke-Howson method for computing Nash equilibria [Cottle *et al.*, 1992]. The details of this method are beyond the scope of this work. Most importantly, it can generate Nash equilibria in a fixed order, but is limited to two-player games.

Another problem of the approach is that in order to compute the Nash equilibria of a game, an agent needs to observe the rewards of all players. This allows Nash-Q to simulate the Q-update process (Equation 3.8) for all players and compute Nash equilibria on the matrix games they define. Full observability of rewards is a particularly strong assumption. Minimax-Q does not impose this requirement, but is limited to zero-sum games where rewards of the opponent are deducible from own rewards.

Given these limitations, it is somewhat disappointing that convergence results for Nash-Q have been presented only for a limited class of games. In particular, the learning process in self-play has been proven to converge to Nash Q-values if *every* matrix game defined by interim Q-values that arises during learning satisfies one of the following two conditions:

i. It has a global optimum point equilibrium, defined as the joint action that maximizes each agent's payoff.

ii. It has a global saddle point equilibrium, defined as a point at which if a player deviates, it only gets a lower payoff (equilibrium condition), but the other player necessarily gets a higher, or equal, payoff.

As Shoham, et al. point out [Shoham *et al.*, 2003], the existence of a globally optimal Nash equilibrium is guaranteed in but not limited to common-payoff games, and the existence of a saddle equilibrium point is guaranteed in but not limited to zero-sum games. However, it is hard to find instances of games outside the special cases in which one of the two conditions holds.

Friend-or-Foe-Q

Since the practical applicability of Nash-Q is essentially limited to zero-sum and common-payoff games, Littman reinterpreted it as the Friend-or-Foe-Q (FF-Q) learning framework [Littman, 2001]. The FF-Q extension is best thought of as two algorithms. Friend-Q is suited to common-payoff games, in which Nash equilibria are uniquely-valued:

$$V_i(s) = \max_{\vec{a} \in A(s)} Q_i(s, \vec{a}) \tag{3.11}$$

Foe-Q is suited to zero-sum games, and is equivalent to Minimax-Q.

The framework is justified as follows. Assume that there are two competing teams of agents in a stochastic game. For each agent all other agents are either friends (on the same team) or foes (on the other team), hence the name of the framework. Each agent can infer the labeling of other agents from the rewards it observes, and can alter between the Friend-Q and Foe-Q value computation.

One problem with FF-Q is that in common-payoff games there can be more than one equilibria, in which case the algorithm needs an arbiter to determine which equilibrium should be played.

Correlated-Q (CE-Q)

The algorithms described so far in this section all rely on some way of computing the Nash equilibrium for the matrix game defined by Q values of all players at each state. The value for each player of a mutually agreed-upon equilibrium is the value function used in the Q update process. Instead of computing Nash equilibria of Q stage games, the agent can compute any other solution concept. One alternative is to compute the *correlated equilibrium* (Definition 4). This is the technique used by Greenwald and Hall in the unambiguously named Correlated-Q (CE-Q) algorithm [Greenwald and Hall, 2003]. Under CE-Q, the value function is

$$V_i(s) \in \mathrm{CE}_i(\vec{Q}(s, \vec{a})), \tag{3.12}$$

where $\mathrm{CE}_i((Q(s, \vec{a}))$ is the set of values of the correlated equilibria of the matrix game defined by the Q-values of all agents at a given state s. As with Nash-Q, the value function is not well defined, because the set of all correlated equilibria of a matrix game is not necessarily a singleton. CE-Q also requires full observability of all rewards, so that the Q-update process can be simulated for all agents.

To compute equilibria of the Q-value matrix games that arises at each state of the stochastic game, CE-Q constructs and solves a linear program. The variables are the probabilities of each joint action. Let $P(a_i, a_{-i})$ denote the probability of the joint action $\langle a_i, a_{-i} \rangle$. Then, the solution to the following linear program is a correlated equilibrium strategy for the matrix game defined by the Q-values at state s:

$$\begin{aligned}
\text{Maximize:} \quad & \sum_i \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} P(a_i, a_{-i}) Q_i(s, a_i, a_{-i}) \\
\text{Subject to:} \quad & \sum_i \sum_{a_{-i} \in A_{-i}} \left[Q_i(s, a_i, a_{-i}) - Q_i(s, a'_i, a_{-i}) \right] P(a_i, a_{-i}) \geq 0 \quad \forall a_i, a'_i \in A_i \\
& \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} P(a_i, a_{-i}) = 1 \\
& P(a_i, a_{-i}) \geq 0 \quad \forall a_i \in A_i, \forall a_{-i} \in A_{-i}
\end{aligned}$$

In fact, the objective function is not mandatory, as any set of joint action proba-

bilities that satisfy the constraints will be a correlated equilibrium. The introduction of this objective function ameliorates the equilibrium selection problem by restricting the set of reachable equilibria to the ones that satisfy it. Since this function maximizes the sum of the players' rewards, Greenwald and Hall call this particular instantiation of CE-Q *utilitarian*, or uCE-Q. There can be a number of variants to CE-Q, depending on the objective function that best addresses the goals of the designer.

Greenwald and Hall do not offer any theoretical results. However, they demonstrate empirical convergence of their algorithm in multi-state stochastic games in self-play.

3.3 Joint Action Learners

The term "Joint Action Learners" (JALs) has been used to refer to a particular class of multi-agent learning algorithms. JALs are among the very few algorithms that are readily applicable and well-motivated for competitive settings. In addition, they are appealing for their simplicity and speed. It is somewhat surprising that the theoretical and empirical analysis of such algorithms has been relatively limited in the literature.

The main objective of this thesis is to investigate JALs in detail and measure their performance in competitive stochastic games with respect to clearly defined criteria. This objective is pursued throughout subsequent chapters. This section is confined to a review of existing work on JALs in the literature.

JALs were first presented by Claus and Boutilier [1998] as a possible multi-agent extension to Q-learning. Recall from Section 3.2.1 that Q-learning is a widely studied and established algorithm for single-agent reinforcement learning. Section 3.2.2 reviewed the challenges of extending Q-learning for multi-agent settings and concluded that a natural first step towards addressing these challenges is to redefine the Q-value function to be the value of playing a certain *joint action* at a given state. Thus, the Q-update function for agent i can be generalized as:

$$Q_i(s, \vec{a}) \leftarrow (1 - \alpha)Q_i(s, \vec{a}) + \alpha \left[R_i(s, \vec{a}) + \gamma V_i(s')\right], \tag{3.13}$$

where $V_i(s')$ is the state value function, i.e., the value to agent i for being at state s'.

Section 3.2.3 discussed a number of multi-agent Q-learning extensions that use this same Q-update rule, but differ in the way they compute $V_i(s')$. Joint Action Learners are another class of algorithms that belongs to this family. To compute the value function $V_i(s')$, JALs maintain an explicit model of the opponent(s) for each state. In the instantiation by Claus and Boutilier [1998], this is achieved through the mechanism of *fictitious play* (see Section 3.1.1). The JAL player i assumes that its opponents are stationary and keeps a count $C(s, a_{-i}), \forall a_{-i} \in A_{-i}, \forall s \in S$. The assumed model of the opponent is the empirical frequencies of play, where the probability of the opponent playing action a_{-i} in state s is

$$P(a_{-i}|s) = \sum_{a_{-i}} \frac{C(s, a_{-i})}{n(s)}, \tag{3.14}$$

where $n(s)$ is the number of times state s has been visited. This model allows the agent i to compute the state-value function as

$$V_i(s') = \max_{a_i} P(a_{-i}|s')Q(s, \langle a_i, a_{-i}\rangle) \tag{3.15}$$

Claus and Boutilier [1998] point out that fictitious play is not the only way a Joint Action Learner can maintain a model of the opponent. Section 3.1.2 presented an alternative by Kalai and Lehrer [Kalai and Lehrer, 1993], known as *rational learning*. To avoid confusion between the specific algorithm presented by Claus and Boutilier and the class of Joint Action Learners in general, this thesis will refer to the former as Fictitious Play Q-learning, or FP-Q. The full algorithm can be found in Table 3.1.

Note that the term "Joint Action Learner" could also be confusing. The reason is that all algorithms discussed in Section 3.2.3 maintain Q-value functions for joint actions and could also be referred to "joint action learners." Unlike these algorithms, however, algorithms that fit within the JAL framework learn an explicit model of joint play. In summary, algorithms within the JAL framework exhibit both of the following properties:

- Q-values are maintained for all possible joint actions at a given state

- The joint play of all opponents is modeled explicitly

(1) Let $\alpha_0 \in (0, 1]$ be the initial learning rate, and ϵ be the initial exploration rate. Initialize $Q(s, \vec{a})$ arbitrarily, $C(s, a_{-i}) \leftarrow 0 \forall s \in S, \forall a_{-i} \in A_{-i}$, $n(s) \leftarrow 0 \forall s \in S$.

(2) Repeat,

(a) Observe state s, $n(s) \leftarrow n(s) + 1$

(b) From state s select action a_i with probability $(1 - \epsilon)$ by solving

$$\underset{a_i}{\text{argmax}} \sum_{a_{-i}} \frac{C(s, a_{-i})}{n(s)} Q(s, \langle a_i, a_{-i} \rangle),$$

and a random action with probability ϵ.

(c) Observing the opponent's action a_{-i}, the reward $R(s, a_i)$, and the next state s',

$$\begin{aligned} Q(s, \langle a_i, a_{-i} \rangle) &\leftarrow (1 - \alpha) Q(s, \langle a_i, a_{-i} \rangle) + \alpha (R(s, a_i) + \gamma V(s') \\ C(s, a_{-i}) &\leftarrow C(s, a_{-i}) + 1 \end{aligned}$$

where

$$V(s') = \max_{a_i} \sum_{a_{-i}} \frac{C(s', a_{-i})}{n(s')} Q(s', \langle a'_i, a_{-i} \rangle))$$

(d) Decay α and ϵ as per Q-learning.

Table 3.1: The FP-Q algorithm.

While FP-Q (and JALs as a whole) can be applied to any general stochastic game, Claus and Boutilier [1998] only investigate its performance in the restricted set of coordination matrix games. They test FP-Q empirically and compare it with traditional single agent Q-learning. FP-Q proves superior in two ways - (1) converges faster than Q-learning, and (2) its opponent modeling component makes it amenable to the design of various exploration strategies that bias play towards Pareto-optimal equilibria. This second point is particularly important in coordination games.

FP-Q was also investigated by Uther and Veloso under the name of Opponent

Modeling [Uther and Veloso, 2003].[2] The evaluation was again empirical, but on the more comprehensive zero-sum stochastic game of Hexcer. Interestingly, experiments were not conducted in self-play, but against single-agent Q-learning and Minimax-Q. FP-Q proved overall superior to both algorithms. In particular, it exhibited slower learning but better final performance against Q-learning, and faster learning and better final performance against Minimax-Q. Uther and Veloso attribute the effectiveness of FP-Q to the opponent modeling component. They note that on the surface, FP-Q is similar to Q-learning, except that the latter's stochastic updates model the environment *and* the opponent at the same time. However, changes in the opponent action probabilities can occur much faster than Q-learning can update its Q-tables. Unlike Q-learning, FP-Q is capable of detecting this directly through its explicit model of the opponent, which allows for a faster back propagation of Q-values than the learning rate would allow.

Beyond these two papers, JALs are ordinarily mentioned in literature surveys, but are not investigated in any detail. The main reason is the belief that JALs are incapable of playing mixed policies. As Bowling and Veloso point out [Bowling and Veloso, 2002], this is due to the fact that existing JAL algorithms such as FP-Q best-respond in each state.

The inability to play mixed policies is a serious disadvantage to any learning algorithm. For example, such an algorithm would not converge in self-play on a game with a unique mixed Nash equilibrium such as Rock Paper Scissors (Figure 2.1). Playing the unique Nash policy of this game may be important in a competitive setting against powerful opponents as it renders the agent unsusceptible to exploitation.

One of the major contributions of this thesis is the introduction of Smooth FP-Q – a Joint Action Learner that is capable of playing mixed policies. The algorithm is described in detail in Chapter 5. An empirical investigation of its behavior is offered in Chapter 7.

[2]Curiously, this paper was written in 1997, before the work by Claus and Boutilier. It remained unpublished, but was placed on the web, and people started referencing it. It was finally published by popular demand, as a CMU Technical Report in 2003.

3.4 Iterated Policy Adjustment

This section briefly reviews a rather different approach to multi-agent learning, one that will be called iterated policy adjustment. The general premise is that a player adjusts its policy after each iteration of play so as to increase its expected payoff. Work done in this field is often neither model-based nor model-free, as the policy of the opponent is sometimes assumed to be fully observable. Some of the early work in the domain is quite impractical, as it concerns very restricted domains and rests on strong assumptions. However, it has recently led to algorithms of broad applicability.

3.4.1 Infinitesimal gradient ascent and WoLF-IGA

Early work in this domain examined gradient ascent as a technique for learning in simple two-player, two-action, general-sum matrix games. One example is the Infinitesimal Gradient Ascent algorithm (IGA) [Signh *et al.*, 2000]. A player moves its strategy in the direction of the current gradient with some step size, η. To do so, it needs to compute the partial derivative of its expected payoff with respect to its strategy, which can only be done if the actual strategy of the opponent is fully observable. In IGA, the step size is infinitesimal, i.e., $\eta \to 0$.

Singh, et al. showed that if both players use Infinitesimal Gradient Ascent, then their strategies will converge to a Nash equilibrium or the average payoffs over time will converge in the limit to the expected payoffs of a Nash equilibrium.

In subsequent work, Bowling and Veloso augmented IGA with the Win or Learn Fast principle (WoLF), and presented WoLF-IGA [Bowling and Veloso, 2002]. The essence of WoLF is to learn quickly when losing, and cautiously when winning. In the case of IGA, this idea translates into a different step size η - relatively large one when the agent is under-performing, and a small one when thins are going better than expected.

The natural question that arises is how to determine when the algorithm is "winning" or "losing". In WoLF-IGA, the yardstick is the payoff under a certain Nash equilibrium of the game. Assuming such equilibrium is known, Bowling and Veloso

proved that in a two-player, two-action, general sum matrix game, WoLF-IGA always converges to some Nash equilibrium in self-play.

3.4.2 WoLF-PHC

The convergence result of WoLF-IGA is quite perplexing. If a Nash equilibrium is already known, why would the agent go through the trouble of learning one? Bowling and Veloso acknowledge that WoLF-IGA is not practical. However, their analysis was a proof of concept for the WoLF principle as a means towards improving the convergence properties of iterated policy adjustment algorithms. This allowed them to design WoLF-PHC, a practical, general-purpose multi-agent learning algorithm, based on Policy Hill Climbing (PHC) [Bowling and Veloso, 2002].

WoLF-PHC, presented in full in Table 3.2, is based on Q-learning (Section 3.2.1). As the name suggests, the algorithm performs hill-climbing in the space of mixed policies. Q-values are maintained as in standard single-agent Q-learning. However, the algorithm also maintains the current mixed policy π, which is improved by increasing the probability that it selects the highest valued action according to a learning rate δ. When $\delta = 1$, the algorithm is equivalent to Q learning. In order to apply the WoLF principle, the algorithm also maintains the average policy $\bar{\pi}$. Determination of "winning" and "losing" is now done by comparing the expected value of the current policy to that of the average policy.

Bowling and Veloso test WoLF-PHC in a wide variety of matrix and general stochastic games, and demonstrate that the algorithm often converges to some Nash equilibrium policy profile in self-play. In addition, it behaves quite well against other opponents.

3.5 Summary and Discussion

This chapter reviewed some of the more prominent multi-agent learning algorithms in the literature. Among the algorithms reviewed from the game theory community were

classical fictitious play and rational learning. From the reinforcement learning community, the chapter reviewed Q-learning and some of its multi-agent modifications. Particular attention was paid to work done on the class of Joint Action Learners, as this class remains under scrutiny throughout this thesis. Iterated policy adjustment algorithms and the novel WoLF-PHC were also briefly described.

Much of the work done to date revolves around theoretical or empirical convergence to Nash equilibrium policies. This is not surprising considering the centrality of Nash equilibria in game theory. However, the investigation of equilibrium convergence properties in self-play appears to be of limited utility, as it is not immediately clear that convergent learners will play well against any opponent.

In order to design and evaluate multi-agent learning algorithms, we first need to specify our objectives. A first step in that direction is to pin-point desiderata and criteria for multi-agent learning. This is pursued in the next chapter.

(1) Let $\alpha \in (0,1]$, $\delta_l > \delta_w \in (0,1]$ be learning rates. Initialize,

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

(2) Repeat,

(a) From state s select action a according to policy $\pi(s)$ with suitable exploration.

(b) Observing reward $R(s,a)$ and next state s',

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s',a')).$$

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

$$\begin{aligned} C(s) &\leftarrow C(s)+1 \\ \forall a' \in A_i \qquad \bar{\pi}(s,a') &\leftarrow \bar{\pi}(s,a') + \frac{1}{C(s)}(\pi(s,a') - \bar{\pi}(s,a')). \end{aligned}$$

(d) Step π closer to the optimal policy w.r.t. Q,

$$\pi(s,a) \leftarrow \pi(s,a) + \Delta_{sa},$$

while constrained to a legal probability distribution,

$$\Delta_{sa} = \begin{cases} -\delta_{sa} & \text{if} \quad a \neq \text{argmax}_{a'} Q(s,a') \\ \sum_{a' \neq a} \delta_{sa'} & \text{otherwise} \end{cases}$$

$$\delta_{sa} = \min\left(\pi(s,a), \frac{\delta}{|A_i|-1}\right),$$

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

Table 3.2: The WoLF-PHC algorithm.

Chapter 4

Evaluation Criteria

Traditionally, research in AI has been focused on designing algorithms that lead to the adoption of optimal behavior by an agent, given some ordinarily restricted environment. In multi-agent learning (MAL) settings, however, traditional notions of optimality no longer hold, as outcomes are contingent on the behavior of all agents involved. Not surprisingly, it has been relatively difficult to pinpoint a set of criteria that should be used in the evaluation of learning algorithms for such settings.

Additionally, MAL is not entirely an AI endeavor. On the contrary, AI researchers have joined the investigation of MAL relatively recently, as it was economists (game theorists) who first examined the issue. The two communities pursue different agendas and consider different constraints. However, they also inform and influence each other. This makes for a variety of objectives and approaches that is both fascinating and confusing.

This chapter examines this variety and clarify the research agenda pursued here. It proceeds by investigating the algorithm evaluation criteria previously put forth in the literature, and then present a new set of criteria. In addition, it presents an impossibility result, demonstrating that not all multi-agent learning desiderata are achievable in settings of restricted observability of opponents' rewards.

4.1 Research Agenda

The multiplicity of objectives in MAL research can be quite perplexing. In a recent article [Shoham *et al.*, 2006], Shoham, Powers and Grenager (SPG) from Stanford University provide a broad overview of MAL research to date, and identify five distinct agendas that have been pursued persistently. This section reviews this work and specifies the research agenda pursued in this thesis.

The first agenda outlined by SPG is computational in nature. Researchers who adopt it regard MAL as an iterative way of computing Nash equilibria or other solution concepts of a game (e.g., correlated equilibria, dominant strategy equilibria, etc.). For example, fictitious play [Brown, 1951] was originally proposed as a way of computing a sample Nash equilibrium in zero-sum games. Ordinarily, adaptive mechanisms are not the fastest way to compute an equilibrium, but have the advantage of being easily understood and implemented.

The second agenda is descriptive - it asks how natural agents (e.g., individuals, firms, government agencies) learn to behave in the presence of other, presumably adaptive natural agents. The goal within this agenda is to come up with learning models that best describe observable and measurable behavior of natural agents. As Fudenberg and Levine point out in one still unpublished paper [Fudenberg and Levine, 2006], this agenda is central to economists and other social scientists. However, the goal of describing behavior is not to be decoupled from the related goal of *predicting* it. Predicting in this context refers not only to matching data in experiments, but to the broader and more important issue of when and whether we should expect play by natural agents to resemble an equilibrium.

The third agenda outlined by SPG is called "normative," and can best be defined as the study of whether learning rules are in equilibrium with each other. It appears that this agenda is restricted to repeated matrix games. To follow SPG's example, one might wonder if fictitious play and Q-learning are in equilibrium on the Prisoners Dilemma if they are properly initialized. Economists question the legitimacy of this agenda. It is explicitly rejected by Fudenberg and Kreps [1993]. Fudenberg and Levine [2006] elaborate on the critique against it, by pointing out that there is no

reason to believe that learners are initialized in any particular way. In other words, it is unjustifiable to assume that learning rules start a game in any kind of "learning equilibrium."[1]

The two remaining agendas are prescriptive, as they investigate how agents *should* learn in different contexts. The first concerns dynamic cooperative settings, in which one wishes to obtain distributed control. The goal here is to design autonomous agents that adapt and behave in a way that maximizes the payoff obtained by the entire system. Such settings are ordinarily modeled as common-payoff games. Distributed control is desirable in order to relieve the burden on a central mechanism as well as the system's dependency on it (single point of failure).

The fifth and final agenda is concerned with the way agents should learn in non-cooperative settings. The question asked here is this: How should an agent behave and adapt in a dynamic environment in order to maximize its expected payoff? SPG believe that this agenda is best aligned with the design stance of AI, as it investigates the design of optimal effective agent strategies for different environments. An effective strategy is one which procures the highest possible reward to the agent.

In line with the fifth, "AI" research agenda outlined by SPG, this thesis investigates the design of agents that maximize their expected rewards. To that end, the right learning algorithm is one which allows the agent to procure rewards that are "good enough," given the limitations of the environment and the behavior of all other agents.

4.2 Criteria Previously Presented in Literature

This section reviews the evaluation criteria put forth by MAL research to date. Most of the work reviewed is very recent, and the debate on which criteria are most adequate and desirable will likely pick up momentum in the future.

[1] While this rejection is reasonable in the case of natural agents, it may not be so for artificial systems. Brafman and Tennenholtz [2004] present a compelling argument in support of the normative approach to MAL. This is a fascinating topic, but beyond the scope of this work.

4.2.1 Convergence to equilibrium

Although not explicitly stated, the goal of many researchers in the AI community has been to design algorithms that learn to play some kind of an equilibrium strategy in self-play. For the Minimax-Q [Littman, 1994] and Nash-Q [Hu and Wellman, 2003] algorithms this has been the Nash equilibrium, and for the CE-Q algorithm [Greenwald and Hall, 2003] this has been the correlated equilibrium.

Convergence to equilibrium is an obvious yardstick for measurement of success when pursuing the equilibrium research agenda. It is also relevant for the design of autonomous agents for cooperative settings – such was the motivation for the CE-Q algorithms. Given certain conditions, the approach could be beneficial in pursuing our research agenda as well. For example, playing a Nash strategy in a zero-sum game would guarantee a self-interested agent at least 0 reward on average, which is also the safety value of the game.

However, designing algorithms with nothing but convergence to an equilibrium in mind is still relatively impractical when designing self-interested agents. Learning to play an equilibrium strategy is of no use in competitive settings unless all opponents learn to play the exact same equilibrium. In order to achieve this in games with multiple equilibria, one would need an oracle or some coordination device. Even in games with unique equilibria, an equilibrium learner would not learn to exploit suboptimal behavior on the part of its opponents.

In addition, pursuing this criterion often means that learners have to compute Nash equilibria explicitly, which is generally hard. The computational complexity of finding equilibria in matrix games was recently shown to be PPAD-complete [Daskalakis *et al.*, 2005; Chen and Deng, 2005a; Chen and Deng, 2005b]. Commonly used algorithms for 2-player games have exponential worst case behavior, and we often need to resort to approximation for computing equilibria of n-player games [McKelvey and McLennan, 1996].

A final criticism towards the equilibrium approach in learning is based on two impossibility results. As demonstrated by Hart and Mas-Colell [2003], convergence to a Nash equilibrium in general is impossible in uncoupled dynamics, i.e. when

the agents do not observe the rewards of their opponents. It comes as no surprise that many of the equilibrium learners have been implemented for settings in which opponents' rewards can be inferred (e.g. zero-sum games, coordination games) or are fully observable. However, it is commonly believed that observability of all rewards is a strong and rather impractical assumption.

Another impossibility result is offered in [Zinkevich *et al.*, 2005]. The authors demonstrate that any algorithm that relies on Q-values for the derivation of policies cannot learn equilibrium policies for a certain class of games. Even if such an algorithm converges on the equilibrium Q-values, these values contain insufficient information for reconstructing the equilibrium policy. Therefore, any variant of Q-learning can only learn equilibrium policies for limited classes of games. One interpretation of this result could be that relying on Q-values is simply not the right approach for MAL, and the general Q-update rule (Equation 3.8) should be revisited. Unfortunately, most existing algorithms rely on this rule.

Some of these criticisms of adopting convergence to equilibrium as a learning criterion have been repeatedly raised in the literature. One example of a comprehensive (and provocative) discussion of the problem is work done by Shoham, Powers, and Grenager [Shoham *et al.*, 2006].

4.2.2 Regret minimization

Regret minimization is one of the oldest criteria used in multi-agent learning, as it dates back to the early days of game theory [Blackwell, 1956; Hannan, 1957]. The basic idea has been rediscovered repeatedly in game theory, AI, and operations research. It has been referred to using a variety of names - "universal consistency," "no-regret," "Bayes envelope," etc. A comprehensive review of this literature is beyond the scope of this work, but I refer the reader to a summary by Foster and Vohra [1999].

Informally, regret is what an agent "feels" after having played a suboptimal strategy. Ideally, we would like to minimize regret with respect to playing the mixed strategy that is a best-response to the strategies employed by all other agents. In general, it is unclear what this best response is (otherwise we would program the

agent to play it!), which is why researchers have had to offer other, relaxed notions of regret. One notion that is adopted regularly is the regret with respect to pure strategies. More specifically, the regret $r_i^t(a_j, s_i)$ at time t of agent i for playing the sequence of actions s_i instead of playing action a_j, given that the opponents played the sequence s_{-i}, is defined as follows:

$$r_i^t(a_j, s_i|s_{-i}) = \sum_{k=1}^{t} R(a_j, s_{-i}^k) - R(s_i^k, s_{-i}^k) \tag{4.1}$$

The goal is to design agents that achieve at most zero total regret.

A no-regret property provides relatively strong guarantees about the performance of a learning algorithm. Naturally, it would be better if an even stronger lower bound on performance could be provided by designing algorithms which exhibit no-regret with respect to a richer class of policies.

It appears that most of the regret minimization literature has focused on repeated matrix games. In recent work, Bowling [2005] combined the no-regret requirement with that of convergence. He presented GIGA-WoLF – a no-regret algorithm that provably converges to Nash equilibrium in self-play in matrix games with two players and two actions per player. Mannor and Shimkin [2003] discuss some difficulties in extending the approach for general stochastic games and point out that the no-regret property may not be attainable in general.

It is worth pointing out that much of the work in this space has been descriptive, rather than prescriptive. Often the focus is on establishing the existence (or non-existence) of no-regret algorithms for different settings, but the results do not inform the construction of such algorithms. A prominent exception is the work done by Fudenberg and Levine [1995].

4.2.3 Rationality and Convergence

Bowling and Veloso [2002] were the first in the AI community to depart from the ideas above and put forth other specific criteria for evaluating multi-agent learning

algorithms. They demand that an algorithm exhibits the following two properties:

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 the other players' policies.

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

Bowling and Veloso further clarify that the second property depends on the other agents' using an algorithm from some class of learning algorithms. Conditioning the second criterion on the type of opponents appears inevitable. This is why in their work the two authors focus on convergence in self-play.

It is important to clarify that "convergence" in this context means convergence of policies. There are other, weaker notions of convergence: of average reward, empirical distribution of actions, etc.

Relaxing the requirement of convergence to a Nash equilibrium and breaking it down into the two criteria above allows for more flexibility and better expected performance than the one an equilibrium learner could exhibit. As previously noted, equilibrium learners guarantee convergence to an equilibrium policy regardless of the policy actually adopted by the opponents. Thus, an equilibrium learner may not learn to play best-response against opponents who play non-equilibrium strategies. The property of rationality, on the other hand, means that an algorithm will learn to play best-response to equilibrium as well as non-equilibrium strategies of its opponents. This will allow for an improved (optimal) performance against opponents with bounded computational or physical capabilities.

Note that there is no mention of equilibria in the two properties. However, when all players are rational and convergent, they will converge to a Nash equilibrium. In particular, an algorithm that satisfies the two properties will always converge to a Nash equilibrium in self-play. Therefore, it appears that rational and convergent algorithms fall under the impossibility results discussed in the previous section.

In addition, the two criteria are designed with a strong assumption in mind - all opponents employ stationary policies or learning algorithms that converge to a stationary policy. What would happen to an algorithm that satisfies both requirements

when it faces a non-stationary opponent? Consider an adaptive opponent which can sustain a stationary strategy for an indefinite amount of time, which but can also switch to a new policy at any one time. Assume that the opponent sustains its strategy long enough to allow for the rational and convergent agent to converge to a best response. Given the payoffs under this agent's best response, the opponent may decide it would be better off if it switched to another policy. This could render the strategy employed by the agent suboptimal.

Chang and Kaelbling [2002] demonstrate what happens to an empirically rational and convergent algorithm when pitted against a cunning adaptive opponent. They take Bowling and Veloso's WoLF-PHC [Bowling and Veloso, 2002] and design a dynamic opponent unambiguously named PHC-Exploiter. They demonstrate that in a zero-sum setting where the only Nash equilibrium requires the use of mixed policies, PHC-Exploiter is able to achieve unbounded rewards (as time proggresses) against any PHC opponent. PHC-Exploiter begins the game by consistently playing some policy π_1. It keeps this policy stationary long enough for WoLF-PHC to learn some $\pi_2 \in BR(\pi_1)$, but then switches to some π_1' such that the reward for WoLF-PHC is much lower. WoLF-PHC, in turn, slowly adjusts its policy, which leads to a cyclical nature of play. However, Chang and Kaelbling demonstrate that it is possible to tweak PHC-Exploiter so that it receives higher total rewards every cycle. Note that PHC-Exploiter is not an unreasonable algorithm in general, as it performs well against a variety of other opponents (including Nash-Q), as well as in self-play.

4.2.4 Ability to beat Fair opponents

Chang and Kaelbling explain their results by pointing out that their algorithm is inherently superior to all PHC variants. To make this distinction, they propose a classification of learning algorithms based on the amount of history they can maintain in memory (explicitly or implicitly), as well as the beliefs about the amount of memory maintained by the opponent. An agent is classified by the cross-product of two parameters - H, which ranges from H_0 for memoryless agents to H_∞ for agents with unbounded memory, and B, which ranges from B_0 for agents that believe their

opponent is memoryless to B_∞ for agents that believe their opponent has unbounded memory. By this classification, (WoLF-)PHC is an $H_\infty \times B_0$ algorithm, as it implicitly stores information about the entire history of play, but assumes a stationary opponent. PHC-Exploiter, on the other hand, is in the $H_\infty \times B_\infty$ space, and is therefore superior. Chang and Kaelbling discuss many of the more prominent learning algorithms in the literature with regards to this classification.

Based on their classification, the two researchers define the notion of a *fair* opponent which allows them to propose a new criterion for the evaluation of learning algorithms: a good learning algorithm should be able to "beat" any fair opponent.

Fair opponent. A fair opponent for a player in class $H_s \times B_t$ is any player from a class $H_{s'} \times B_{t'}$, where $s' \leq s$ and $t' \leq t$.

Ideally, one would like to design agents in the $H_\infty \times B_\infty$ space which can beat all fair opponents. As Chang and Kelbling point out, however, this is impossible. They refer to a rather technical paper by Nachbar and Zame [1996] from the game theory community. The crux of this analysis is that it is impossible to design an agent that will learn to predict the opponent's future strategy *and* optimize over those beliefs at the same time. Therefore, we can only hope to design agents that are capable of "beating" opponents from inferior classes.

Unfortunately, Chang and Kaelbling do not elaborate on their understanding of the word "beat." Does "beating" imply that an agent should always get higher rewards in a game that allows for that possibility? For example, would playing a best response to the opponent's strategy constitute "beating" it if the resulting payoff is lower than the one of the opponent?

4.2.5 Targeted Optimality, Compatibility, and Safety

In very recent work, Powers and Shoham [2005] reviewed some of the criteria above and proposed a new set that builds on them:

Targeted Optimality. When the opponent is a member of the selected set of opponents, the average payoff is at least $V_{BR} - \epsilon$, where V_{BR} is the expected value of the

best response in terms of average payoff to the actual opponent.

Compatibility. During self-play, the average payoff is at least $V_{selfPlay} - \epsilon$, where $V_{selfPlay}$ is defined as the minimum value achieved by any Nash equilibrium that is not Pareto dominated by another Nash equilibrium.

Safety. Against any opponent, the average payoff is at least $V_{security} - \epsilon$, with $V_{security}$ defined as $max_{\pi_1 \in \Pi_1} min_{\pi_2 \in \Pi_2} EV(\pi_1, \pi_2)$ [where $EV(\pi_1, \pi_2)$ is the expected payoff to a player for playing strategy π_1 against an opponent playing π_2].

Powers and Shoham present these criteria in the context of known, fully observable two-player repeated matrix games, with average awards. However, their ideas can be extended to general stochastic games.

4.3 New Set of Learning Desiderata

As elaborated in Section 4.1, the goal of this research is to design agents capable of procuring rewards that are "good enough" in a competitive setting, given the limitations of the environment and the behavior of all other agents. Specifying evaluation criteria is in effect defining what constitutes "good enough" rewards. When is an algorithm considered to have achieved such rewards? How can we be reasonably sure that an algorithm will maintain high performance against adaptive opponents?

The following is a set of specific learning desiderata that begins to address these questions. Note that this set is certainly not definitive. As previously mentioned, the discussion on evaluation criteria is a relatively recent phenomenon for the AI community, and will likely intensify in the future.

The properties are defined for two-player settings, but the extension to n-player settings is straightforward.

Desideratum 1 (Rationality)**.** Let π_{-i} be a stationary policy adopted by the opponent of agent i.[2] Let $P^{s_0}_{\pi_i, \pi_{-i}}$ be the probability distribution induced by policy π_i and

[2]Alternatively, a policy that converges with time

π_{-i} and starting state s_0, and let $E^{s_0}_{\pi_i,\pi_{-i}}$ be the corresponding expectation operator. Then, for all ϵ there is a time $\bar{t}(\epsilon)$ such that the reward to agent i for playing π_i at time $t \in (\bar{t}(\epsilon), \infty)$, $R_t(\pi_i, \pi_{-i})$ should satisfy the equation

$$E^{s_0}_{\pi_i,\pi_{-i}} R_t(\pi_i, \pi_{-i}) - E^{s_0}_{\pi_i,\pi_{-i}} R^{BR}(\pi_{-i}) \leq \epsilon, \tag{4.2}$$

where $R^{BR}(\pi_{-i})$ is the reward obtained by the best-response to π_{-i}.

Desideratum 2 (Safety)**.** Against any opponent and for all ϵ there is a time $\bar{t}(\epsilon)$ such that for all $t \in (\bar{t}(\epsilon), \infty)$ agent i is able to obtain at least

$$\max_{\pi_i \in \Pi_i} \min_{\pi_{-i} \in \Pi_{-i}} EV_i(\pi_i, \pi_{-i}) - \epsilon \tag{4.3}$$

Desideratum 3 (Constant Adaptability)**.** Let π_{-i} be a stationary policy adopted by the opponent of agent i at time t_0. Then, the reward to agent i for playing π_i at time t should satisfy Equation 4.2 for all $t \in (\bar{t} + t_0, \infty)$, where $\bar{t}$ is as defined per Desideratum 1.

Desideratum 4 (Stability)**.** Let π_{-i} be the stationary policy adopted by the opponent of agent i. Let there exist some $\pi'_i \in \Pi_i$ such that $\pi_{-i} \in BR(\pi'_i)$ and $\pi' \in BR(\pi_{-i})$, i.e. let π_{-i} be a Nash equilibrium policy. Then, for all ϵ there is a time $\bar{t}(\epsilon)$ such that the strategy adopted by agent i at time $t \in (\bar{t}(\epsilon), \infty)$, $\pi_{i,t}$ should satisfy the equation

$$BR(\pi_{i,t}) = \pi_{-i}. \tag{4.4}$$

The first desideratum is similar to Bowling and Veloso's *Rationality*. It guarantees that the agent can learn to obtain the expected reward of the best-response to a stationary policy adopted by the opponent. Note that when the opponents are stationary, the stochastic game is in fact reduced to an MDP (see Section 2.3). A multi-agent learning algorithm should be able to learn optimal behavior in such settings, as existing single-agent learning techniques provably do so (e.g., Q-learning).

The second desideratum follows Powers and Shoham's *Safety* requirement. A learning agent should be able to learn to secure the safety value of any game against infinitely powerful opponents.

The third desideratum addresses the issue of adaptability. As discussed in Section 4.2.4, Chang and Kaelbling [2002] demonstrate the importance of adaptability and illustrate how a cunning adaptive opponent can exploit a rational and convergent learner. To address the issue, they offer a parametrization of learning algorithms that allows them to define the notion of *fair* opponents, and then suggest that a learner should be able to "beat" all fair opponents. As they themselves point out, however, this is impossible for some games.

The formulation of adaptability in Desideratum 3 takes a different approach – it relies on the previously defined *Rationality*. It requires that a learner's ability to satisfy these properties does not diminish with time. In particular, the amount of time needed to adopt a best-response to a stationary opponent should not depend on the time at which the opponent adopts a stationary policy. This time can either be at the beginning of the learning process or later on.

The fourth and final desideratum posits the requirement that if the opponent plays any Nash policy, the learning agent should not only play a best-response as per Desideratum 1, but should also converge to the Nash policy that is "the other side" of the equilibrium played by the opponent. Note that this is the only property that requires convergence of policy, as opposed to convergence of expected reward. How does this desideratum fit into the agenda pursued here? Why does it help define the notion of "good enough" rewards? This desideratum is put forth with stability in mind - if met, the opponent will have no incentive to deviate from its policy (by definition of Nash equilibrium), and learning may be considered completed. This property is desirable for two reasons. First, it makes sure that there will be no more periods in which the learning agent receives suboptimal rewards (assuming that learning some best-response to the policy of the opponent takes any positive amount of time). Second, it relieves the agent from any cost associated with the learning process.

The issue of cost of learning is very important and it is relatively surprising that, it has not been given due attention in the literature. Ordinarily, learning has some positive cost associated with it. For example, there is always some computational cost of the algorithm. Ideally, we would like to model the cost of learning in learning

$$R_1 = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} R_2 = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Figure 4.1: A modified Rock Paper Scissors game

itself and have the agent optimize with respect to the environment, its opponents, *and* the value of the information it expects to obtain as a result of a given amount of subsequent computation. This is very much in line with the general concept of *bounded rationality* that has been pervasive in the AI literature as of late (c.f. [Russel and Wefald, 1991]). The issue of learning cost is not explored any further in this thesis, but is a potential venue for future research. Here, it is simply acknowledged that it is desirable for the learning process to end in some finite amount of time (but not to the disadvantage of our learner).

4.4 An Impossibility Result

As previously discussed, it is not always possible to obtain desirable properties of a learner, no matter how powerful an algorithm is. Recall the result by Nachbar and Zame [1996] which demonstrates that it is impossible to design an agent that will learn to construct accurate beliefs about the future play of the opponent and optimize over these beliefs at the same time. Another impossibility result [Zinkevich *et al.*, 2005] revealed that Q-values contain insufficient information for reconstructing equilibrium policies, and all Q-learning variants cannot always converge to equilibrium. Also discussed was work done by Hart and Mas-Colell [2003] which demonstrates that learning agents cannot always converge to a Nash equilibrium of a game if they do not observe each other's rewards.

A new impossibility result is presented below, one related to that by Hart and Mas-Colell. Even if one agent is somehow able to compute the equilibrium of a game and plays a stationary Nash strategy, a learning opponent that does not observe the entire reward vector will not always be able to learn its side of the same Nash equilibrium. Thus, achieving Desideratum 4 is impossible in general if the rewards

of the opponents are not observable. This result, like Hart and Mas-Colell's, does not depend on any assumptions about the decision-making dynamics of the learner. It is based solely on the requirement that the learner does not observe the rewards of the opponent.

Theorem 1. *Let π_{-i} be the stationary policy adopted by the opponent of agent i. Let there exist some $\pi_i' \in \Pi_i$ such that $BR(\pi_i') = \pi_i$, i.e., let π_{-i} be Nash equilibrium policy. Then, there is no learning algorithm that does not take as input the entire reward vector and can always converge on π_i such that $BR(\pi_i) = \pi_{-i}$.*

Proof. Consider the matrix game presented in Figure 4.1, heretofore referred to as ModRPS. The reward function of player 1 R_1 is the same as R_1 in Rock Paper Scissors (RPS). R_2, however, is different. Recall that RPS has a unique Nash equilibrium $\langle(\frac{1}{3},\frac{1}{3},\frac{1}{3}),(\frac{1}{3},\frac{1}{3},\frac{1}{3})\rangle$. ModRPS also has a unique Nash $\langle(\frac{1}{4},\frac{1}{4},\frac{1}{2}),(\frac{1}{3},\frac{1}{3},\frac{1}{3})\rangle$, in which the strategy of player 2 is the same as the Nash strategy under RPS, but player 1's strategy is different. Let $\Phi(H_1)$ be a learning rule employed by player 1. H_1 is the set of histories observable by player 1, where $\forall h_1 \in H_1$, h_1 is a sequence of tuples $\tau_{t,1} \subseteq \tau_t \backslash R_2$. In other words, player 1 is able to observe any information during play except for player 2's rewards. Let player 2 play the stationary strategy $\sigma_2 = (\frac{1}{3},\frac{1}{3},\frac{1}{3})$ sharing a pseudo-random number generator with a fixed seed with player 1. Now assume that $\Phi(H_i)$ is a learning rule that can always converge on a Nash strategy, given that its opponent plays a stationary Nash strategy. Since σ_2 is a stationary Nash strategy in RPS, Φ will converge to $\sigma_1 = (\frac{1}{3},\frac{1}{3},\frac{1}{3})$ for any observable history $h_1 \in H_1$ generated by playing against player 2 in RPS. σ_2 is a Nash strategy in ModRPS as well, and Φ must converge to $\sigma_1' = (\frac{1}{4},\frac{1}{4},\frac{1}{2})$ for any observable history generated by playing against player 2 in ModRPS. However, given the fixed random seed and player 2's stationary strategy, any history observable by player 1 induced by its play against player 2 in RPS is identical to the history observable by player 1 induced by the same play against player 2 in ModRPS. Since $\sigma_1 \neq \sigma_1'$, it follows that Φ is not a legitimate learning rule because it is not a function. If we relax the assumption that both players share a pseudo-random number generator with a fixed seed, then Φ will be able to converge on σ_1 or σ_1' in both games as a function of randomness. Thus, there is no learning rule $\Phi(H_i)$ that always converges on a Nash

strategy given a stationary Nash strategy of the opponent. Any such convergence is generally due to chance. □

The repercussions of this result in Economics and social theory in general are far reaching. For example, it appears that stability is not always attainable without some kind of coordination which, in turn, requires negotiations. However, a more thorough investigation of these repercussions is beyond the scope of this work.

Since achieving *Safety* is not always possible, it would be unreasonable to adopt it as a criterion for the evaluation of learning algorithms. In general, one can only hope to achieve the first three desiderata put forth in Section 4.3. These desiderata – *Rationality*, *Safety*, and *Constant Adaptability* – will heretofore be referred to as Criteria, and will be used in the evaluation of Joint Action Learners.

4.4.1 Potential weaknesses

The biggest weakness of these criteria is that they do not adequately address a situation in which a learner is faced with an adaptive opponent who does not converge to a stationary policy. Consider an opponent that changes its policy every iteration of the game. Since the opponent is not stationary, the *Rationality* criterion does not hold. Consequently, *Constant Adaptability* is rendered irrelevant as well. The only remaining criterion for this situation is *Safety*. However, an agent may be able to do better than obtaining the safety value of the game in such setting. Unfortunately, addressing this problem leads to a number of complications. First, we ought to find a way to distinguish between the opponents against which *Safety* suffices, and others, against which a better performance is to be expected. This calls for a parametrization of the space of learning algorithms. Chang and Kaelbling's work suggests one possible approach [Chang and Kaelbling, 2002] (see Section 4.2.4). However, it does not capture the amount of information that the opponent is capable of observing. For example, an opponent in $H_\infty \times B_0$ who can observe the entire reward vector is inherently superior than an agent in the same space who cannot. Even if we are to establish a meaningful and complete parametrization, we still need to determine what constitutes "better" performance against weaker adaptive non-stationary opponents.

Another possible objection to the criteria is that they do not capture the effect that a learner can have on the future play of the opponent(s). Recall the previous assertion that *learning* cannot be decoupled from *teaching*. Assuming that at least one of the opponents is adaptive, an agent could influence the course of the game by teaching it to play a policy that is conducive to more desirable outcomes. Intuitively, the ability to teach adaptive opponents is particularly important in settings with relatively few players. This is the line of reasoning employed by Shoham, Powers and Grenager [Shoham *et al.*, 2006] in their criticism against no-regret learning.

The importance of teaching in MAL is undisputable. However, in order to engage in teaching at any one time, an agent needs to have some knowledge about the game. In the special case of repeated games it would suffice to know the payoff structure of the stage game. In a general stochastic game, however, the agent would need to have knowledge about the payoff structure of all stage games, as well as the transition model. This thesis does not assume that such kowledge is available apriori. On the contrary, it is assumed that obtaining such knowledge (implicitly or explicitly) is among the learner's major goals.

Another potential criticism is that the criteria posit requirements for the behavior of learners in the limit of time and ignore performance during the early stages of learning. Ideally, the criteria would be more stringent in that regard. For example, an algorithm that achieves them in polynomial time should be valued more highly than one which achieves them in exponential time. In order to make headway in this respect, we need to gain a better understanding of the amount of exploration required to visit each state of the game sufficiently many times. Recent studies of the sample complexity of reinforcement learning [Kakade, 2003] might provide valuable insights in this direction.

4.5 Summary

This chapter reviewed some of the criteria employed in the evaluation of multi-agent learning algorithms. After specifying the research agenda pursued here, a new set of

criteria was presented which, albeit limited, goes a long way towards addressing the goal of designing self-interested agents for stochastic games. In particular, a learner should exhibit the properties of *Rationality*, *Self-Compatibility*, *Constant Adaptability*, and *Safety*. Additionally, evidence was shown that not all learning desiderata are achievable. The impossibility result was that a learner cannot in general converge to a Nash policy in response to a stationary Nash policy of the opponent, unless the reward vector is fully observable. Thus, *Stability* of the learning process is not always attainable.

This analysis allows for a more focused and informed discussion of Joint Action Learners, presented in subsequent chapters.

Chapter 5

JALs, Evaluation Criteria and a New Algorithm

One reason for investigating the JAL class is that FP-Q is one of very few algorithms applicable to adversarial stochastic games. In fact, of all multi-agent Q-learning extensions discussed in Chapter 3, only FP-Q and WoLF-PHC are well-motivated for the design of self-interested learning agents. The other algorithms were conceived with different research agendas in mind. Applying them in a competitive setting would make some of the assumptions behind their design highly implausible.

For example, both Nash-Q and CE-Q require full observability of rewards for all agents, as well as some kind of an equilibrium selection mechanism. Minimax-Q and FF-Q were defined only for restricted classes of games in which the rewards could be inferred (e.g., zero-sum and common payoff games). Furthermore, all four of these algorithms require significant computational resources, as they involve solving an LP for every step of the game, or, in the case of Nash-Q, computing a Nash equilibrium. In contrast, FP-Q and JALs in general do not require the observability of opponents' rewards, involve any complex computation throughout game play, or assume any kind of coordination with the opponents.

This chapter begins with an evaluation of FP-Q with respect to the criteria put forth in Chapter 4. The analysis reveals that its inability to learn mixed policies is

a major impediment towards guaranteeing *Safety*. Section 5.2 examines work done in the game theory community on overcoming the problem for the case of fictitious play. Ideas discussed are then incorporated into a new Joint Action Learner, named Smooth FP-Q.

5.1 FP-Q and the Evaluation Criteria

The first evaluation criterion put forth in Chapter 4 was that of *Rationality*. Below it is show that FP-Q satisfies this property, given the standard Q-learning assumptions of infinite exploration and learning rate decay as constrained by Equation 3.5.

Theorem 2. *FP-Q is Rational as defined per Desideratum 1, assuming infinite exploration of the state space and learning rate decay suitable for Q-learning.*

Proof. Recall that when the opponent is playing a stationary policy $\bar{\pi}_{-i}$, the stochastic game is reduced to an MDP with transition model

$$\hat{T}(s, a_i, s') = \sum_{a_{-i} \in A_{-i}} \bar{\pi}_{-i}(s, a_{-i}) T(s, \langle a_i, a_{-i}, \rangle, s').$$

In the limit of time, the opponent modeling component of FP-Q will get arbitrarily close to the actual policy $\bar{\pi}_{-i}$. As in Q learning, FP-Q will also implicitly learn the transition model T through stochastic approximation. Therefore, its Q-values will converge to the Q-values of Q-learning in an MDP with transition model $\hat{T}$. Given a learning rate decay that satisfies Equation 3.5 and an infinite exploration of the state space, Q-learning will learn the optimal policy in such MDP [Watkins and Dayan, 1992; Singh *et al.*, 2000]. Therefore, given the same assumptions, FP-Q will learn a best-response to $\bar{\pi}_{-i}$ in the stochastic game with transition model T. Since it will approach the rewards of the best-response asymptotically, there must be a time T for every ϵ such that the reward under its policy at time T is ϵ away from the reward of the best-response. □

Unfortunately, there cannot be theoretical guarantees that FP-Q can satisfy any of

the other evaluation criteria. In fact, it is immediately clear that FP-Q cannot satisfy the *Safety* criterion because of its inability to learn mixed policies (briefly discussed in Section 3.3). One example of a in which a mixed policy would be required to guarantee *Safety* is Rock Paper Scissors (Figure 2.1). As a zero-sum game, its safety value is 0, and is obtainable by playing the unique mixed Nash strategy of the game: $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$. No matter what the opponent plays, adopting this strategy yields at least 0 expected reward. One might think that playing deterministically every iteration is not harmful on average as long as the empirical distribution of play resembles the Nash strategy. However, as Fudenberg and Levine [1998] demonstrate for the case of fictitious play, empirical convergence of marginal probabilities is not a sufficient guarantee for safety level payoffs – it can lead to correlated play, in which the agent always gets the smallest reward in the game.

Meeting the third evaluation criterion, *Constant Adaptability*, is impossible not only for FP-Q, but for any Q-learning variant because of the decaying learning rate. Recall from Section 3.2.1 that the proof of optimality of Q-learning depends on decaying the learning rate α according to a schedule that meets specific criteria.

There is thus an abundant opportunity for improvement of existing JAL implementations. FP-Q can provably meet only one out of the three evaluation criteria put forth in Chapter 4. Furthermore, meeting this one criterion – *Rationality* – is also achievable by the simple and widely studied Q-learning.

In subsequent sections, it is demonstrated that there are alternatives to the best-response dynamics of FP-Q that would allow a JAL algorithm to learn mixed policies. Smooth FP-Q, a stochastic variant of FP-Q inspired by work done in the game theory community on extensions of fictitious play, is introduced.

5.2 Fictitious Play and Mixed Policies

The traditional process of fictitious play is deterministic; the player always maximizes its expected payoff based on the observed history of actions by the opponent and the reward structure of the game. The only situation in which a fictitious play agent

might randomize is when the expected payoff of two or more actions is exactly the same. Because of this limitation, convergence results in self-play have traditionally concerned the empirical frequencies of actions taken by each player, i.e., the marginal probability distributions over individual players' actions. As we reviewed in Section 3.1.1, this notion of convergence is very weak. It allows for the possibility that the joint distribution of play is correlated, which can lead to payoffs below the safety level of the game.

To address this limitation, Fudenberg and Kreps (1993) and Fudenberg and Levine (1995, 1998) examined alternatives to the best-response dynamics that would allow for mixed policies and therefore convergence of *intended play*. The basic idea behind their approach is to introduce a small degree of randomness in the player's decision. Let $R_i(a_i, \hat{\sigma}_{-i})$ be the reward to agent i for playing $a_i \in A_i$ if the opponent plays the observed empirical frequency $\hat{\sigma}_{-i}$. Under traditional fictitious play, agent i would choose the action a_i that maximizes this expected payoff, i.e., $a_i \in A_i$ s.t. $a_i = \text{argmax}_{a_i} R_i(a_i, \hat{\sigma}_{-i})$. Now suppose that instead of the actual reward vector $R_i(\hat{\sigma}_{-i})$, agent i observes $\bar{R}_i(\hat{\sigma}_{-i}) = R_i(\hat{\sigma}_{-i}) + \eta_i$, where η_i is a random vector drawn from $\Delta\eta_i$ and satisfies a certain technical condition.[1] In other words, assume that there is a small amount of random noise η_i in the observations (or calculations) of agent i. To maximize her expected utility, she must play each action $a_i \in A_i$ with probability

$$P(a_i) = P\left(\underset{a_i}{\text{argmax}}\, R_i(a_i, \hat{\sigma}_{-i}) = \underset{a_i}{\text{argmax}}\, \bar{R}_i(a_i, \hat{\sigma}_{-i})\right), \tag{5.1}$$

which defines her strategy $\hat{\sigma}_i$ as a function of $\hat{\sigma}_{-i}$ and $\Delta\eta_i$. In other words, the agent must pick a strategy $\hat{\sigma}_i$ under which the probability of playing each action a_i reflects the probability that a_i is the true utility maximizing action, given $\bar{R}$ and an estimate of the likelihood and size of the error. Suppose that agent i does not know if its observations or calculations are noisy, but doubts that this may be the case. In this case, it would still play $\hat{\sigma}_{-i}$, and will choose to solve Equation 5.1 with respect to some $\Delta\eta_i$ that best reflects its degree of skepticism.

[1] Due to space constraints, we avoid technical details, but refer the reader to [Fudenberg and Levine, 1998], Chapter 4.

The modification of fictitious play that implements this strategy generation mechanism is called smooth fictitious play [Fudenberg and Levine, 1998]. The name reflects the fact that small changes in observed histories result in smooth shifts of the chosen strategy. In contrast, small changes under traditional fictitious play may result in radical shifts of behavior, which is considered undesirable. Another illustrative interpretation of the name is that the probability of selecting each action smoothly decreases as the distance between its expected utility and the expected utility of the perceived best-response action increases. A smooth fictitious play agent is almost indifferent between actions with similar expected utility.

Assuming some error in the perception of expected rewards means that the agent exhibits cautiousness in the generation of its strategy. This reflects people's demonstrated risk-aversion, which makes smooth fictitious play a particularly attractive model of behavior from an economist's point of view. Research in psychology of threshold perception shows that when people are asked to discriminate between two alternatives, their behavior ranges from random to biased to deterministic as the alternatives become more distinct [Massaro and Friedman, 1990], which adds to the plausibility of smooth fictitious play.

Does it pay off to exhibit caution in the decision making? Fudenberg and Levine [1998] demonstrate that, given an appropriately selected $\Delta\eta_i$, a smooth fictitious play agent can guarantee itself the safety value of the game, regardless of the behavior of the opponent. More importantly, smooth fictitious play exhibits a property called ϵ-*universal consistency*, one of the alternative notions of no-regret (see Section 4.2.2). Informally, universal consistency requires that regardless of the opponent's play, a learner almost surely gets at least as much utility as it could have gotten had it known the frequency but not the order of observations in advance. This property provides strong guarantees about the performance of a learning algorithm.

5.3 Smooth FP-Q

FP-Q algorithm could be considered as a mixture of fictitious play and Q-learning. The Q-update mechanism (Equation 3.8) leads to back-propagation of rewards, which allows the agent to consider the future consequences of its actions at the present state. The resulting Q-values reflect the total discounted reward to be expected at the end of the game for taking each respective action, and define a matrix game for each state. Under FP-Q, at each state the agent decides on an action as a fictitious play agent would decide for a matrix game.

A natural extension of FP-Q would be to replace the fictitious play component with its smooth variant. The resulting algorithm is most appropriately called "Smooth FP-Q."

Any implementation of this idea would require explicit solving of Equation 5.1, which entails the introduction of parameters that describe some $\Delta\eta_i$. Fudenberg and Levine [1998] propose a way to solve the equation directly using an implicit estimation of $\Delta\eta_i$ which involves a single parameter λ:

$$P(a_i) = \frac{e^{\frac{1}{\lambda}u_i(a_i,\hat{\sigma}_{-i})}}{\sum_{a_i' \in A_i} e^{\frac{1}{\lambda}u_i(a_i',\hat{\sigma}_{-i})}} \tag{5.2}$$

In a Smooth FP-Q implementation, $u_i(a_i, \hat{\sigma}_{-i})$, the expected utility for playing a_i if the opponent plays the empirical frequency $\hat{\sigma}_{-i}$, would be computed by using the Q-values and the empirical model of the opponent for the current state. The computation becomes:

$$u_i(a_i, \hat{\sigma}_{-i}) = \sum_{a_{-i}} \frac{C(s, a_{-i})}{n(s)} Q(s, \langle a_i, a_{-i} \rangle), \tag{5.3}$$

where s is the current state, $C(s, a_{-i})$ is the number of times the opponent took action a_{-i} in s, and $n(s)$ is the total number of times state s has been visited.

When substituting for u_i in Equation 5.2, the resulting formula is quite similar to what a Boltzman exploration schedule (Equation 3.6) would look like for FP-Q.

The only difference would be that under Smooth FP-Q the parameter λ is to remain fixed, while the equivalent temperature parameter T in the Boltzman formula is to be cooled down. Therefore, it is not unreasonable to get rid of explicit exploration schedules altogether, as Smooth FP-Q has exploration already "built in." If the Q-values are initialized to 0, then behavior will be purely random initially, and for each state it will gradually become biased towards the action that maximizes expected utility with respect to the model of the opponent.[2]

One problem with the direct application of this formula to the FP-Q algorithm is that a different λ might have to be adopted for each state. For larger and more complex stochastic games, this will be impractical at best, and more likely simply impossible. An imperfect way around this problem is to normalize the Q-values for each state. Normalization ensures that the relative probability for playing any two actions at any given state reflects the ratio of their Q-values, regardless of the magnitude of these Q-values. This would allow for a higher likelihood of success with the same λ adopted for each state.

When λ is arbitrarily close to 0, Smooth FP-Q becomes equivalent to FP-Q. On the other hand, for a large enough λ, Smooth FP-Q will select an action at random. Therefore, it is important to fine-tune this parameter to achieve best performance.

Since smooth fictitious play can guarantee the safety level in a matrix game, it would not be unreasonable to expect that Smooth FP-Q could meet the *Safety* criterion by doing the same not only in matrix, but also in multi-state stochastic games. Chapter 7 offers an empirical investigation of the algorithm's performance with respect to that criterion.

Unfortunately, Smooth FP-Q remains unable to meet the third evaluation criterion, *Constant Adaptability*. Possible directions for future research are discussed in Chapter 8.

[2]This is not to say that an explicit exploration cannot or should not be implemented.

5.4 Summary and Discussion

This chapter revealed that FP-Q is provably *Rational*, but is unable to meet the *Safety* criterion due to its inability to learn mixed policies.

In an attempt to address this limitation, Smooth FP-Q, a new Joint Action Learner, was presented. As in FP-Q, Smooth FP-Q maintains a model of the opponent that reflects the empirical frequencies of actions played for every state. However, it employs a randomized action selection mechanism which allows for the convergence on mixed policies. Further investigation is required before it can be asserted that the new mechanism is conducive to achieving *Safety* or learning meaningful mixed policies for multi-state games. These questions are empirically examined in Chapter 7.

Note that both FP-Q and Smooth FP-Q remain unable to exhibit *Constant Adaptability*. The limitation is common to all Q-learning variants. Chapter 8 discusses potential ways to address it in further research.

The fact that FP-Q can only meet one out of the three criteria put forth in Chapter 4 may suggest that it is not viable in competitive settings. The next chapter, however, illustrates the significance of *Rationality*. FP-Q is in fact able to outperform all other multi-agent extensions of Q-learning discussed in Chapter 3.

Chapter 6

Experimental Evaluation of FP-Q

This chapter illustrates the empirical behavior of FP-Q in several stochastic games. Implementations of FP-Q, WoLF-PHC, Nash-Q, CE-Q, naïve Q-learning, and a random player are pitted against each other in an all-vs-all tournament. The tournament is played on three grid-world games already studied in the literature. The results demonstrate that FP-Q converges to an ϵ-best-response much more reliably than any other non-naïve multi-agent extension of Q-learning. This performance reflects FP-Q's provable *Rationality*.

While Smooth FP-Q was not implemented for this tournament, the experiments are revealing for its potential performance. The reason is that Smooth FP-Q and FP-Q are equivalent for $\lambda \to 0$, where λ is the smoothing parameter discussed in Section 5.3.

The chapter begins with a discussion of the grid-world games and their properties. It then elaborates on the implementation details of the learning algorithms used. The end of the chapter offers a presentation and discussion of the results observed.

6.1 Grid Games

Grid-world games, while rather simple, possess all key elements of dynamic strategic interaction. Rewards are postponed in the future, so there is nothing to "lead" the

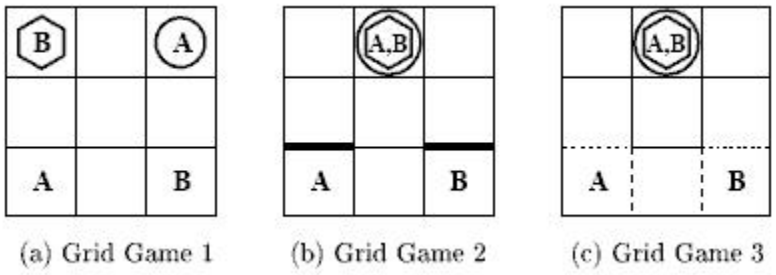

Figure 6.1: Grid Games, initial states. Geometric shapes indicated goals.

agent to the goal state. There are different paths to the goal and the desirability of a given path depends on the behavior of the opponent.

Grid games have been extensively employed in the literature to test multi-agent learning algorithms. The first two grid games used in the tournament – Grid Game 1 and 2 (Figure 6.1[1]) – were introduced by [Hu and Wellman, 2003] in their analysis of Nash-Q, while the third – Grid Game 3 – was described for the first time by [Greenwald and Hall, 2003] in the investigation of CE-Q. Littman used another grid-world game in his examination of Minimax-Q [Littman, 1994]. Sutton and Barto [1998] and Mitchell [1997] also employed grid games to examine their learning algorithms. In all three grid games used for the tournament, player one starts at the bottom left corner, and player two - at the bottom right. In Grid Game 1, player one's goal is located at the upper right corner, and the goal of player two is in the upper left corner. Each player can move up, right, left, or down at any given cell. Actions are deterministic. If a player tries to move outside of the board, it remains in its original cell. If two agents attempt to move to the same cell, they bounce back and receive a reward of -1. The reward for reaching the goal cell is 100. All other actions, including bumping into the walls of the board, yield a reward of 0. The game ends as soon as one of the two players gets to its goal. Since players move simultaneously, it is possible that they both reach their goal at the same time.

Grid Game 2 differs from Grid Game 1 in two ways: 1) the goals for both agents are in the middle cell of the top row, and 2) moving up from any one of the two

[1]Reproduced from Figure 1 in [Greenwald and Hall, 2003].

starting cells is successful only half of the times, i.e. the player who attempts to move up from its starting position remains in it with a probability 0.5. All other actions are deterministic.

The setup for Grid Game 3 is the same as in Grid Game 2, except that all actions are deterministic. In addition, if both players get to the goal from the side, they both get a reward of 120. If one player gets to the goal from the center it receives a reward of 125 and the other agent receives a reward of 100.

Hu and Wellman [2003] and Greenwald and Hall [2003] analyze the equilibrium properties of these games. In all three games there exist *deterministic* Nash and correlated equilibria. In Grid Game 1, there are several equilibria in which the players coordinate their behavior to pass by each other and reach their respective goals. Under Grid Game 2 and 3 there are two symmetric equilibria in which one player moves to the center and then up towards the goal and the other player moves up twice and then towards the goal. Note that these equilibria are asymmetric, as one player gets a higher reward than the other. This is also valid for Grid Game 2 because moving up from the starting state is successful only half of the times which means that the player that plays such a policy would be able to obtain an average total reward of 50, while the player that goes through the center can get an average total reward of 100.[2]

In addition to the deterministic equilibria, Grid Game 2 and 3 also exhibit non-deterministic correlated and Nash equilibria [Greenwald and Hall, 2003]. In Grid Game 2 there is a continuum of equilibria such that for all $p \in [0, 1]$, with probability p one player moves towards the center and the other goes up, and with probability $1 - p$ the roles are reversed. In Grid Game 3, there exist symmetric, non-deterministic, correlated equilibria in which both agents move up with high probability and each of the deterministic equilibria is played with equally low probability.

Grid Game 1 meets the theoretical requirements for convergence of Nash-Q put forth by Hu and Wellman [2003]. The two researchers demonstrated that, as long as there is some way in which opponents can resolve the equilibrium selection problem

[2]These numbers represent average total *non-discounted* reward.

in self-play, Nash-Q always converges on a Nash equilibrium in this game. In addition, they showed that self-play performance in Grid Game 2 can be quite good – convergence to a Nash equilibrium was observed in up to 90% of the cases.

Greenwald and Hall [2003] tested four flavors of CE-Q on all three grid games in self-play. They demonstrated that the utilitarian flavor discussed in Section 3.2.3, as well as two other variants (libertarian and republican), converge on some coordinated equilibrium of each game. The libertarian operator, on the other hand, could lead to repeated collisions and negative rewards in self-play, as there is no equilibrium selection mechanism in place.

Bowling and Veloso [2002] tested WoLF-PHC on Grid Game 2, again in self-play. They demonstrated that this algorithm converges on a Nash policy with time.

6.2 Algorithms and Implementation Details

A number of the multi-agent Q-learning extensions described in Chapter 3 were implemented in order to conduct a comparative analysis of FP-Q. In particular, the naïve extension of single-agent Q-learning (Section 3.2.2, WoLF-PHC (Section 3.4), as well as Nash-Q and CE-Q (Section 3.2.3) were implemented. While these algorithms are already discussed in the literature, there are a number of parameters that could be adjusted. This section describes these parameters in detail.

6.2.1 Exploration, learning rates, and discounting

Recall from Section 3.2.1 that theoretical investigations of Q-learning have led to general criteria for designing learning rate and exploration rate cooling schedules. Unfortunately, choosing specific schedules remains an ad-hoc decision. Most authors approach the problem by simply picking a schedule that seems reasonable and leads to satisfactory results. Tweaking such schedules can be daunting but not impossible as long as a single algorithm is being implemented and tested. However, doing so in a tournament in which every algorithm is pitted against all participating algorithms can

lead to exponentially many combinations of different sets of parameters. Considering the significant computational time required by some of these algorithms, it would not be feasible to do such an extensive test.

As a way around this problem, exploration schedules suggested in the original papers introducing each algorithm were implemented. For the Nash-Q algorithm, Hu and Wellman (2003) suggested an ϵ-greedy exploration. The probability of choosing a random action in state s at time (step) t, $\epsilon_t(s)$, is calculated as follows:

$$\epsilon_t(s) = \frac{1}{1 + n_t(s)}, \tag{6.1}$$

where $n_t(s)$ is the number of times the agent has visited state s up to time t.

Unfortunately, the authors of CE-Q did not put forth any exploration schedules. However, considering that CE-Q is an equilibrium learner very similar to Nash-Q, the implementation of CE-Q used the same exploration schedule.

Bowling and Veloso (2002) also employed an ϵ-greedy exploration policy, but they kept ϵ fixed at 5%. Note that this is not in line with the GLIE guidelines put forth by [Singh *et al.*, 2000]. However, the two authors were able to obtain very good results on a number of games including Grid Game 2, which is why the WoLF-PHC implementation used the same fixed exploration.

When Claus and Boutilier (1998) introduced JALs and specifically FP-Q, they discussed a number of different exploration schedules informed by the Boltzman formula (Equation 3.6). The aim of their research was in fact to derive exploration schedules that lead to better rewards for JALs and naïve Q-learning in matrix coordination games. However, since they did not test their algorithm in larger stochastic games, the FP-Q implementation discussed here deviated from their guidelines and used another ϵ-greedy schedule. In particular, the probability of playing a random action in any state during the ith training game, ϵ_i, was given by the formula

$$\epsilon_i = \epsilon_0 \times e^{-\frac{i^2}{N^2} \ln\left(\frac{\epsilon_0}{\epsilon_N}\right)}, \tag{6.2}$$

where $\epsilon_0 = 0.999$ is the initial exploration rate, $\epsilon_N = 0.001$ is the exploration rate at

the end of the training period, and N is the total number of training games. Note that i stands for an entire game iteration, while t in the formula above refers to the number of actions taken for all games played so far. This formula, graphically represented in Figure 6.2, is inspired by a temperature cooling schedule for simulated annealing. As demonstrated by the graph, the schedule allows for a gradual decay of the exploration probability. The same exploration schedule was used for the naïve extension of Q-learning.

The learning rate, α, also needed a specific cooling schedule. As Bowling and Veloso (2002) point out, WoLF-PHC is particularly sensitive to learning rate adjustments. The reason is that it has another set of learning rates, δ_w and δ_l, which are the policy adjustment step sizes for "winning" and "losing" (see Table 3.2). These rates are mutually dependent; it would be detrimental to take larger steps than could be updated for. In order to avoid having to tweak these rates in an extensive set of preliminary tests, WoLF-PHC used Bowling and Veloso's settings. In particular, the learning rate $\alpha(t)$ at time t was set to

$$\alpha(t) = \frac{1}{1 + \frac{t}{500}}, \tag{6.3}$$

while the policy adjustment learning rates δ_w and δ_l were

$$\delta_w(t) = \frac{1}{1000 + \frac{t}{10}} \qquad \delta_l(t) = 4\delta_w(t). \tag{6.4}$$

Striving for simplicity, the same learning rate (Equation 6.3) was used for all other algorithms.

All algorithms were implemented with a discount factor of $\gamma = 0.9$. Thus, the shortest path to the goal in Grid Game 1 would take 4 steps and yield a total discounted reward of 65.61. In Grid Game 2, going up through the barrier would yield a total average discounted reward of $72.9 \times 0.5 = 36.45$, while going to the goal through the center yields at most 72.9. In Grid Game 3, there are no non-deterministic moves, but the total average discounted reward depends on the path taken by the opponent. If both players enter the goal simultaneously through the sides, each would get 87.48;

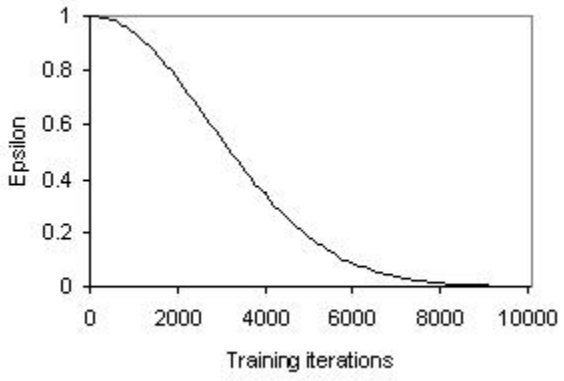

Figure 6.2: Exploration schedule used for FP-Q and Q-learning in the tournament.

if one gets to it from the side and the other through the center, the former would get 72.9, and the latter 91.13.

6.2.2 A limited memory for FP-Q

Claus and Boutilier (1998) were the first to suggest using limited *windows* of histories upon which to base beliefs about the opponent of a JAL. As they point out, a key drawback for JALs is that beliefs based on a lot of experience require even more counter-evidence to be overridden. Using an infinite memory of the actions taken by the opponent for each state would result in an ever-decreasing ability to detect changes in its behavior. Setting a limit to the number of opponent actions "remembered" is a way to ensure that the algorithm remains more adaptive.[3]

A limited memory has been considered before in the context of fictitious play [Young, 1993]. When the size of the history window is adjusted appropriately, fictitious play does not lose its convergence properties.

In this implementation, a fixed memory of $M = 100$ was used for each state in

[3]However, its adaptability still decreases with time due to the decreasing learning rate. See Section 5.1

the game. Thus, the action selection rule (Table 3.1, step (2)(b)) becomes:

$$\underset{a_i}{\text{argmax}} \sum_{a_{-i}} \frac{C_M(s, a_{-i})}{\min(n(s), M)} Q(s, \langle a_i, a_{-i} \rangle), \tag{6.5}$$

where M is the size of the memory window, and $C_M(s, a_{-i})$ is the number of times the opponent took action a_{-i} in state s out of the last M actions taken in this state.

Preliminary testing revealed that the performance of FP-Q with a fixed memory window of size $M = 100$ improves dramatically as compared to that of an implementation with infinite memory in each state. Experiments with window sizes within an order of magnitude of the chosen value showed that performance was not affected significantly.

Note that keeping a fixed memory window is certainly not the only way one could tackle the problem of compounding outdated information about the opponent. One alternative is to weight all evidence based on its recency. Keeping a window of size M is the same as assigning a weight of $1/M$ to the last M actions and a weight of 0 to all others. Alternatively, one could try a more gradual decrease of the weights assigned to older observations.

6.2.3 Other implementation details

One last issue that deserves clarification is the equilibrium calculation and selection mechanisms implemented for Nash-Q and CE-Q. As discussed in Section 3.2.3, correlated equilibria of matrix games defined by the Q-values at every state can be solved using linear programming. Recall that modifying the objective function allows for the design of different flavors of CE-Q. Greenwald and Hall (2003) presented four different flavors, named utilitarian, egalitarian, republican, and libertarian. A libertarian version of CE-Q, in which each agent maximizes its expected reward, was implemented here. Let $P(a_i, a_{-i}$ denote the probability for playing the joint action $\langle a_i, a_{-i}$. Then, the objective function of the linear program solved by each agent i

becomes:

$$\sum_{a_{\in}A_i} \sum_{a_{-i} \in A_{-i}} P(a_i, a_{-i}) Q_i(s, a_i, a_{-i}) \tag{6.6}$$

This libertarian operator is in line with our agenda of designing agents who strive to maximize their individual expected reward. Its downside is that self-play is not necessarily coordinated.

Nash-Q allows for more flexibility, as one could implement any algorithm for computing Nash equilibria of matrix games. Hu and Wellman (2003) employed the classical Lemke-Howson method [Cottle *et al.*, 1992], which returns Nash equilibria in fixed order and thus provides for a way to address the equilibrium selection problem in self-play. The Nash-Q implementation used here employed a novel method for computing Nash equilibria which involves the formulation of a mixed integer program (MIP) [Sandholm *et al.*, 2005].

Let $u_i \in (-\infty, \infty)$ be a variable indicating the highest possible expected utility that player i can obtain given the other player's mixed strategy. Let $\sigma_i(a_i) \in [0, 1]$ be the probability that player i will choose action $a_i \in A_i$ under the mixed strategy σ_i. The expected utility for playing a_i is $u_{a_i} \in (-\infty, \infty)$[4], and $r_{a_i} \in [0, \infty)$ indicates the associated regret. In addition, let $b_{a_i} \in \{0, 1\}$ be a boolean variable set to 1 if the probability of playing a_i is 0. Finally, let the constant U_i indicate the maximum difference between two utilities in the game for player i. In Nash-Q, the utilities for playing different actions are defined by the Q-values for the respective state. Thus, for state s,

$$U_i = \max_{a_i^h, a_i^l \in A_i, a_{-i}^h, a_{-i}^l \in A_{-i}} Q_i(s, a_i^h, a_{-i}^h) - Q_i(s, a_i^l, a_{-i}^l). \tag{6.7}$$

In state s, each agent i independently constructs and solves a mixed integer program with the following constraints:

[4] In the original formulation by Sandholm et al. [2005], u_i and u_{a_i} are both defined to be in $(0, \infty)$. However, this may lead to infeasibility of the MIP if there are negative rewards in the game.

$$\forall i \qquad \sum_{a_i \in A_i} \sigma_i(a_i) = 1, \tag{6.8}$$

$$\forall i \quad \forall a_i \in A_i \qquad u_{a_i} = \sum_{a_{-i} \in A_{-i}} \sigma_{-i}(a_{-i}) Q_i(s, a_i, a_{-i}), \tag{6.9}$$

$$\forall i \quad \forall a_i \in A_i \qquad u_i \geq u_{a_i}, \tag{6.10}$$

$$\forall i \quad \forall a_i \in A_i \qquad r_{a_i} = u_i - u_{a_i}, \tag{6.11}$$

$$\forall i \quad \forall a_i \in A_i \qquad \sigma_i(a_i) \leq 1 - b_{a_i}, \tag{6.12}$$

$$\forall i \quad \forall a_i \in A_i \qquad r_{a_i} \leq U_i b_{a_i}. \tag{6.13}$$

Any solution to this set of constraints will yield variables $\sigma_i(a_i) \forall i, \forall a_i \in A_i$ that will be in Nash equilibrium. As with CE-Q, an objective function is not explicitly required. However, in order to allow the Nash-Q player i to find equilibria that maximize its individual expected payoff, the agent was maximizing for u_i. This is equivalent to the libertarian operator used for CE-Q. Note that implementing a libertarian operator for both Nash-Q and CE-Q may preclude them from coordinating in self-play. However, it allows for a more aggressive play, which presumably increases the chance of obtaining the highest possible reward against other algorithms in Grid Game 2 and 3.

The commercial optimization package CPLEX 10.1.[5] was utilized to solve the LPs in CE-Q and MIPs in Nash-Q. CPLEX incorporates a state-of-the-art branch-and-bound solver with numerous optimization techniques. This implementation kept all CPLEX settings to their default values. Interfacing with the package was done through a simplified Java wrapper called JOpt[6], designed by Harvard's EconCS research group.

[5]ILOG Inc., 2006. http://www.ilog.com/products/cplex/

[6]http://www.eecs.harvard.edu/econcs/jopt/

6.3 Results and Discussion

All algorithms were pitted against each other in an all-versus-all tournament on Grid Games 1, 2, and 3 (Figure 6.1). For each match, each pair of players played 10,000 games in training. A game was terminated as soon as the players took 100 steps (actions) or one of the players reached its goal. After the training period, the policy π_i of each player i was fixed. Note that π_i includes information about the *intended* play of player i, as well as its current exploration rate for each state. Thus, π_i reflects player i's actual probability distribution over its actions for each state at the time of fixing. The two policies π_i and p_{-i} were pitted against each other in a testing stage of 1,000 games, keeping track of the average discounted reward each player could obtain against its opponent. In addition, Q-learning was trained against each fixed policy for 10,000 games, its policy was fixed, and then tested for another 1,000 games. This demonstrated average discounted rewards under a best-response against each player. Each match (and subsequent evaluation) was repeated 10 times.

Tables 6.1, 6.2, and 6.3 summarize the results for Grid Game 1, 2, and 3 respectively. The numbers indicate the performance of the row player. There are three columns per opponent - the first indicates the number of times the player obtained a total discounted reward within 1% of the reward under the best-response to the opponent's policy. The second column indicates the number of times the reward obtained was within 5% of the best-response. Finally, the third column gives the average discounted post-training reward against the opponent across all 10 test runs.

The results demonstrate that FP-Q outperforms all non-naïve multi-agent extensions of Q-learning. FP-Q is ordinarily able to obtain rewards close to those of the best-response against its opponents. Its performance is less convincing only in few runs against Nash-Q in Grid Game 2 and in self-play in Grid Game 3. Investigating the policies it obtains in these less successful runs revealed an interesting property. In these runs, FP-Q and its opponent do in fact *play* a Nash equilibrium, although their respective policies are not a Nash policy profile. Therefore, it is possible for FP-Q to change its play in a way that would lead the game to a state off the Nash equilibrium play path, for which its opponent has not learned adequate behavior. Thus, if FP-Q

has converged on the worse side of the asymmetric Nash in Grid Game 2 or 3, it can "confuse" the opponent by taking some action away from the equilibrium path, which would allow it to obtain a higher reward.

During the testing stage, Q-learning is trained against the fixed policy of the opponent, and is able to detect and take advantage of the opportunity for exploitation. However, FP-Q itself is not always able to do so during training. Assuming the policy of the opponent does not change with time, more training iterations should allow FP-Q to eventually detect this through exploration. This also demonstrates the importance of continuous exploration through time, and suggests that a different exploration schedule might have allowed for a more decisive success. FP-Q is not the only algorithm that fails to detect opportunities for exploitation, as the other algorithms suffer from the same disadvantage in Grid Games 2 and 3.

Of the other non-naïve extensions of Q-learning, WoLF-PHC exhibits the best performance. However, it is not able to perform as well as naïve Q-learning and FP-Q. It often obtains rewards close to the best-response ones, but the difference rarely gets smaller than 1%. One possible explanation for this is that WoLF-PHC requires more training to converge to a best-response. As an iterated policy adjustment algorithm, WoLF-PHC starts playing with a policy in which each action is taken with equal probability, and slowly adjusts this policy accordingly. Therefore, its speed of convergence is limited by the policy adjustment step size, δ. Suppose that its Q-values incorporate information about which action is optimal in each state from the very beginning of training. If the optimal policy is deterministic, WoLF-PHC would still need a relatively large number of iterations to converge to this policy, while other learners can play optimally right away.

This observation helps explain why WoLF-PHC's performance is particularly bad against the random player. Since all other learners converge fairly quickly to some deterministic policy, if WoLF-PHC is playing against them, its Q-values quickly incorporate sufficient information about the direction in which the policy should be adjusted at each state. This allows for consistent adjustment in the right direction from an early point of training, and WoLF-PHC ordinarily gets close to a best-response. Against the random player, however, Q-values for a given state are relatively closer to

each other, and their relative magnitude is likely changing during the early stages of play. This means that WoLF-PHC cannot begin to adjust its policy in the right direction early enough. Since adjustments made early are larger than adjustments made later (due to decreasing δ), this translates into a longer period of training required for better performance.

The naïve extension of single-agent Q-learning was able to do surprisingly well in all grid games, which raises questions about the utility of extending it as FP-Q. Investigating the results of direct matches between Q-learning and FP-Q, however, reveals one of the merits of FP-Q's opponent modeling component: FP-Q was consistently able to learn the best path to the goal and thereby secure higher rewards than its opponent. In Grid Game 2, for example, FP-Q always gets the higher rewards of the asymmetric Nash equilibrium – 72.9 – while Q-learning's average total discounted reward is as low as 36.86.

As demonstrated by Claus and Boutilier [1998], maintaining explicit model of the opponent allows for faster convergence. It is possible that in games with asymmetric equilibria such as Grid Game 2 and 3, this translates into higher chances for learning to play the "better" strategy of the equilibrium, leaving any slower opponent with the strategy that yields inferior payoffs.

To illustrate this, consider Grid Game 2, in which it is best to move towards the center from the starting state, as a move up is successful only half of the times. Only one player can move towards the center, as an attempt by both players to do so will be detrimental for both. Initially, no one player "knows" about this, and this better path is available to the learner who learns about it first. A faster learner has a clear advantage here. There will be a time in which the faster learner goes towards the center because this action is reinforced by past experience, while the slower learner does so because of exploration. The negative reward that will be received by both as a result from the collision will be enough to "discourage" the slower learner, but not the faster one, because of the past successes. Consequently, the slower learner will be left with the less appealing side of the equilibrium. This could explain the rewards received by the faster FP-Q against the slower Q-learning in Grid Game 2.

Note that this no longer holds if the difference in speeds of learning is very big. This is illustrated by matches between FP-Q and WoLF-PHC on Grid Game 2 and 3, in which WoLF-PHC ordinarily converges on the better strategy. A possible explanation is that WoLF-PHC is not easily "discouraged" by collisions even if the path through the center is not reinforced. Since it can only adjust its policy so much at every iteration of the game, it randomizes for a long time during the early stages of learning. Thus, the one who actually gets "discouraged" is FP-Q.

The results also revealed the inherent deficiencies of Nash-Q and CE-Q for competitive settings. Despite the fact that they observe the entire reward vector at every step, the two algorithms often disregard the actual behavior of their opponents. Consequently, they demonstrated the worst performance overall. It is also worth pointing out that evaluating their performance took a considerable amount of computational time. The entire tournament ran for several days on an Intel Pentium 4 3.2GHz, but all runs that did not involve Nash-Q or CE-Q were completed in several minutes.

6.4 Summary

This chapter presented empirical evidence that FP-Q can successfully learn to obtain ϵ-best-response rewards and thereby satisfy the *Rationality* criterion against a variety of opponents in games with deterministic Nash equilibria. FP-Q outperformed all implemented non-naïve multi-agent extensions of Q-learning in a tournament setting. Of the other algorithms, WoLF-PHC was ordinarily able to obtain high rewards, but not always sufficiently close to the best-response ones. CE-Q and Nash-Q, two equilibrium learners with high informational requirements and heavy computation costs, were unable to perform nearly as well.

For a few test runs, FP-Q was unable to learn best-response policies for states off the path of actual play against the opponent. It appears that this was due to insufficient exploration of the state space. Better exploration schedules or more training iterations should amend that.

The naïve extension of Q-learning did very well in the tournament. In direct

matches with FP-Q, however, it consistently obtained lower rewards. One hypothesis for this is that FP-Q's explicit model of the opponent allows for faster convergence, which translates into an ability to learn the better strategy of an asymmetric Nash equilibrium. However, a more rigorous investigation of this hypothesis is required.

	FP-Q			Q			WoLF-PHC			Random			CE-Q			Nash-Q		
FP-Q	10	10	65.61	10	10	65.60	10	10	65.38	5	10	64.48	10	10	65.61	9	9	64.95
Q	10	10	65.61	10	10	65.60	10	10	65.29	10	10	65.17	9	9	64.30	10	10	65.60
WoLF-PHC	10	10	65.51	9	10	65.42	0	9	63.15	0	0	57.03	7	7	59.92	8	8	59.77
Random	0	0	0.69	0	0	0.78	0	0	3.08	0	0	12.61	0	0	2.48	0	0	2.55
CE-Q	9	9	59.02	6	6	45.25	2	6	51.92	0	0	52.59	3	3	29.95	3	3	33.58
Nash-Q	9	9	64.90	6	6	44.41	4	7	57.66	0	0	49.06	1	1	20.45	5	5	42.53

Table 6.1: Results on Grid Game 1 for 10 runs. The results for each player (row) are presented with respect to each opponent (column). There are three columns per opponent. The first column gives the number of time the player obtained reward within 1% of the best-response. The second column gives the number of times the player obtained reward within 5% of the best-response. The third column gives the average discounted reward obtained against the opponent.

	FP-Q			Q			WoLF-PHC			Random			CE-Q			Nash-Q		
FP-Q	8	9	58.42	10	10	72.90	6	10	40.54	10	10	69.85	9	10	68.60	7	7	57.83
Q	6	9	36.86	7	10	54.49	7	10	40.47	10	10	69.86	5	10	52.21	9	10	57.54
WoLF-PHC	1	10	70.72	0	10	71.18	0	7	54.59	0	10	67.48	4	8	54.60	8	9	64.54
Random	0	0	0.61	0	0	0.53	0	0	2.40	0	0	16.96	0	0	3.34	0	0	3.34
CE-Q	2	6	37.25	6	7	55.61	3	3	48.91	4	4	63.96	0	0	49.97	3	3	35.97
Nash-Q	1	1	34.95	2	4	42.21	0	0	38.47	3	3	58.20	2	2	37.30	1	3	35.86

Table 6.2: Results on Grid Game 2 for 10 runs. The results for each player (row) are presented with respect to each opponent (column). There are three columns per opponent. The first column gives the number of time the player obtained reward within 1% of the best-response. The second column gives the number of times the player obtained reward within 5% of the best-response. The third column gives the average discounted reward obtained against the opponent.

	FP-Q			Q			WoLF-PHC			Random			CE-Q			Nash-Q		
FP-Q	7	7	80.20	10	10	85.66	10	10	73.53	4	10	87.83	10	10	83.83	9	9	80.19
Q	9	9	78.37	10	10	82.02	10	10	73.43	10	10	89.53	10	10	85.65	10	10	83.84
WoLF-PHC	10	10	91.11	10	10	91.07	0	6	76.58	0	9	85.88	8	9	84.93	10	10	81.99
Random	0	0	1.60	0	0	1.45	0	0	3.32	0	0	22.43	0	0	5.61	0	0	4.12
CE-Q	9	9	80.19	4	5	63.82	3	5	64.97	0	3	74.21	7	8	85.28	3	5	68.07
Nash-Q	5	5	60.10	8	8	65.57	7	7	60.91	1	3	72.37	7	8	69.90	5	7	66.13

Table 6.3: Results on Grid Game 3 for 10 runs. The results for each player (row) are presented with respect to each opponent (column). There are three columns per opponent. The first column gives the number of time the player obtained reward within 1% of the best-response. The second column gives the number of times the player obtained reward within 5% of the best-response. The third column gives the average discounted reward obtained against the opponent.

Chapter 7

Experimental Evaluation of Smooth FP-Q

The preceding chapter offered empirical evidence that FP-Q reliably converges to an ϵ-best-response against other learning opponents in a *finite* amount of training. This reflects FP-Q's *Rationality*, which was proven in Chapter 5.

In the empirical investigation presented in this chapter Smooth FP-Q is able to obtain the safety value of a simple zero-sum game with unique, mixed Nash equilibrium. In addition, it demonstrates the ability to learn useful mixed policies for a large stochastic game in a test repeatedly discussed in the literature.

7.1 Biased RPS

Consider the matrix game presented in Figure 7.1, which is another modification of Rock Paper Scissors (Figure 2.1). This game is zero-sum and has a unique, mixed Nash equilibrium. Unlike Rock Paper Scissors, however, both agents are biased towards playing the second action, which is why the game is called Biased RPS. Specifically, the only Nash equilibrium of the game is $\langle(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}), (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})\rangle$.

The only way a player could guarantee itself the safety value, 0, is to play the Nash

$$R_1 = \begin{pmatrix} 0 & 100 & -200 \\ -100 & 0 & 100 \\ 200 & -100 & 0 \end{pmatrix} R_2 = \begin{pmatrix} 0 & -100 & 200 \\ 100 & 0 & -100 \\ -200 & 100 & 0 \end{pmatrix}$$

Figure 7.1: Biased RPS: a modification of Rock Paper Scissors.

	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 2.0$	$\lambda = 3.0$	$\lambda = 5.0$	$\lambda = 10.0$
$M = 50$	-19.32	-9.33	-5.56	-3.58	-1.47	-3.72
$M = 100$	-10.20	-4.95	-2.62	-1.85	-0.82	-3.61
$M = 500$	-1.90	-1.08	-0.39	-0.44	-0.38	-3.47
$M = 1000$	-0.91	-0.43	-0.50	-0.43	-0.12	-3.32
$M = 5000$	-0.29	-0.22	**-0.03**	-0.12	-0.04	-3.23

Table 7.1: Biased RPS: Rewards of Smooth FP-Q versus Exploiter for the last 10,000 of 100,000 games. M is size of memory window, λ is smoothing parameter. Optimal performance yields reward 0, worst case is -200. Averaged over 50 runs.

strategy. Therefore, any learning algorithm that cannot learn mixed policies, such as FP-Q or naïve Q-learning, can be exploited by a superior opponent and will obtain an average reward of at most -100 (for playing the second action every time). Pure randomization will not suffice either, as an optimal opponent would always choose the third action, and the average reward would be -33.33.

One test of the ability of Smooth FP-Q to meet the *Safety* criterion would be to train it against a superior opponent on Biased RPS. If successful, Smooth FP-Q should be able to obtain an average reward close to 0.

Smooth FP-Q was trained for 100,000 iterations of Biased RPS. At every iteration an Exploiter observed its *full intended strategy*. Thus, at *every* game iteration, Exploiter computed the expected value of each action and played the action that maximized its expected reward and minimized the expected reward of Smooth FP-Q. Equation 6.3 describes the learning rate used for this and all other experiments on Biased RPS. No explicit exploration schedule was implemented. All Q-values were normalized in the interval $[0, 10]$ prior to computing Equation 5.3, but these normalized values were not stored in the Q-matrices.

Experiments were conducted with several different values for the smoothing pa-

rameter λ (Section 5.3) and the memory window size M (Section 6.2.2). Table 7.1 demonstrates the average reward obtained by Smooth FP-Q during the last 10,000 game iterations, averaged over 50 runs. As demonstrated by the numbers, Smooth FP-Q is able to obtain reward close to 0 for several different sets of parameters. For all λs, the best performance is obtained using the largest memory setting, $M = 5000$. The best performance overall is with $\lambda = 0.2$, which gives an average reward of -0.03. Considering the size of the rewards in the Biased RPS, this is a very good performance.

For comparison, FP-Q was also subjected to the same test using the memory window that proved optimal in Smooth FP-Q ($M = 5000$). Its average reward for the last 10,000 games against the optimal Exploiter was -122. This is lower than the reward FP-Q could have obtained for playing its second action every time (row or column 2 of the matrix game). The reason is that, under the empirical frequencies of the opponent's play, choosing other actions often had higher *expected* utility.

A possible explanation for Smooth FP-Q's good performance is that it was facing an optimal *teacher*. The Exploiter it faced would take an optimal action against it every iteration during learning, which provides a strong corrective and may be the reason behind the success of the experiment. What would happen if we pit Smooth FP-Q against a less powerful opponent?

In another set of 50 runs per parameter set, Smooth FP-Q faced WoLF-PHC. The latter used the learning rate and policy adjustment step size that were suggested by Bowling and Veloso [2002] for the Rock Paper Scissors game:

$$\alpha(t) = \frac{1}{10 + \frac{t}{10000}} \qquad \delta_w(t) = \frac{1}{20000 + t} \qquad \delta_l(t) = 4\delta_w(t). \tag{7.1}$$

The same parameters allowed WoLF-PHC to exhibit strong performance against Exploiter, securing an average reward of -0.24 (over 50 runs).

Table 7.2 summarizes the results of Smooth FP-Q vs. WoLF-PHC for the different sets of parameters. Once again, Smooth FP-Q was able to secure rewards close to 0, the safety value of the game. This time, however, the best memory setting was $M = 50$. Best performance overall was with $\lambda = 0.5$ or $\lambda = 1.0$, yielding average

	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 2.0$	$\lambda = 3.0$	$\lambda = 5.0$	$\lambda = 10.0$
$M = 50$	**0.22**	**0.22**	0.00	0.20	-0.53	-2.39
$M = 100$	0.18	0.09	-0.18	-0.30	-0.39	-2.96
$M = 500$	0.13	-0.10	-0.18	-0.14	-0.38	-2.98
$M = 1000$	-0.30	-0.32	-0.21	-0.11	-0.73	-2.96
$M = 5000$	0.06	-0.08	-0.27	-0.34	-0.58	-3.12

Table 7.2: Biased RPS: Rewards of Smooth FP-Q versus WoLF-PHC for last 10,000 of 100,000 game iterations. M is size of memory window, λ is smoothing parameter. WoLF-PHC uses RPS parameters. Game is zero-sum. Averaged over 50 runs.

reward of 0.22.

Why is it that best performance against an optimal Exploiter was exhibited with a relatively large memory, while it was better to maintain a small memory size against WoLF-PHC? As time goes by, WoLF-PHC converges more and more accurately on the Nash policy. Its empirical model would most accurately reflect the Nash strategy if it considers only the most recent several iterations. A model that takes into account less recent iterations will reflect WoLF-PHC's actions when it had not yet converged on the Nash policy. This information would be outdated, and would introduce noise in the model.

Just like Smooth FP-Q, WoLF-PHC is also sensitive to parameter settings. Experiments with several different cooling schedules for α and δ were conducted, and WoLF-PHC's performance against Smooth FP-Q was not always as strong. One experiment, however, yielded quite interesting results. When using the cooling schedules suggested by Bowling and Veloso [2002] for the grid-world domain (Equations 6.3 and 6.4), Smooth FP-Q was ordinarily able to obtain positive rewards. Surprisingly, for a relatively large M and small λ, its reward was significantly higher than for all other settings. This held even if the rewards were averaged over 100 runs. The results of this experiment are summarized in Table 7.3.

The only reasonable explanation for the observed rewards appears to be that the large memory size allows Smooth FP-Q to determine and exploit patterns in WoLF-PHC's learning. The latter's susceptibility to exploitation has been demonstrated before in the literature [Chang and Kaelbling, 2002]. Note that the grid-world α and

	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 2.0$	$\lambda = 3.0$	$\lambda = 5.0$	$\lambda = 10.0$
$M = 50$	1.36	1.11	0.87	1.19	2.71	-2.62
$M = 100$	1.78	1.38	0.93	1.03	3.00	-2.98
$M = 500$	3.75	2.49	1.63	1.76	3.32	-3.10
$M = 1000$	6.59	4.18	2.85	3.53	3.13	-3.24
$M = 5000$	**24.69**	19.19	11.06	6.79	2.29	-3.30

Table 7.3: Biased RPS: Rewards of Smooth FP-Q versus WoLF-PHC. M is size of memory window, λ is smoothing parameter. WoLF-PHC uses grid-world parameters. Game is zero-sum. Averaged over 50 runs.

δ cooling schedules are not altogether unreasonable for WoLF-PHC, as they allowed for an average reward of -0.28 against the Exploiter.

7.2 Grid Soccer

The previous experiment demonstrated that Smooth FP-Q can get close to the safety value of a zero-sum matrix game against different opponents by learning appropriate mixed policies. Because of its ability to learn a meaningful mixed policy for this game, Smooth FP-Q can satisfy the *Safety* criterion put forth in Chapter 4.

However, these tests do not provide sufficient evidence that Smooth FP-Q can learn mixed policies for *multi-state* environments. A number of the design decisions that had to be made in incorporating smooth fictitious play into the JAL framework may not allow for the learning of useful mixed policies in a multi-state setting. For example, the use of identical values for the λ parameter at each state may not be viable.

This section illustrates the quality of Smooth FP-Q mixed policies learned in self-play on the large grid-world game of Soccer (Figure 7.2[1]). The self-play tests conducted here do not directly address the evaluation criteria. However, they demonstrate the ability to learn meaningful mixed policies in a multi-state environment. This was the very purpose for the introduction of Soccer by Littman [1994].

[1]Reproduced from Figure 2 in [Littman, 1994].

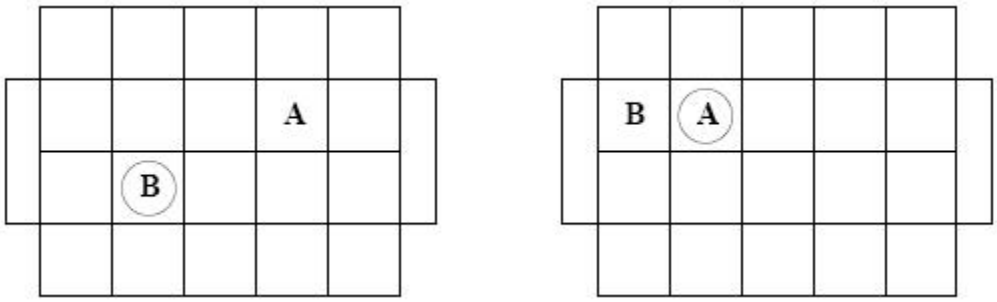

Figure 7.2: Grid Soccer. An initial grid state (left) and a situation requiring a non-deterministic choice for A (right).

As depicted in Figure 7.2, the game is played on a 4x5 grid. The starting position of the players is always the same, but initial possession of the ball is determined at random. The goal of each player is to carry the ball, represented by a circle, to the goal on the opposing side of the field. The game is terminated as soon as a goal is scored, which yields a reward of 1 for the winner and -1 for the loser.

Players occupy distinct squares and can choose from five possible actions – up, down, left, right, and stand. They choose an action simultaneously, but actions are executed in random order, which adds non-determinism to the game. When a player takes an action that would take it to the square currently occupied by the other player, possession of the ball goes to the stationary player, and the move fails. Therefore, the defending player can get the ball by standing where the other player wants to go.

Any deterministic policy by the player on the offensive can be blocked indefinitely by a clever defender. Therefore, the former *must* use a mixed policy. This is illustrated by the right side of Figure 7.2 – player A must choose randomly between standing and moving down in order to hope for an opening that would allow it to score in subsequent moves.

Note that this game is quite big compared to the other games discussed in this thesis – for any position of player A, there are 19 possible positions for player B, and any one can have the ball. Thus, there are a total of 760 possible states. For comparison, the games used in Chapter 6 have 72 states.

Littman [1994] implemented this game in order to illustrate the performance of Minimax-Q. Since the game is zero-sum, Minimax-Q is guaranteed to converge on the Nash equilibrium policy in the limit of time [Littman and Szepesvari, 1996]. Playing the Nash policy would also guarantee the safety value of the game, which is 0. Therefore, the policy learned by Minimax-Q should not be susceptible to exploitation.

Littman trained Minimax-Q in self-play for one million steps, fixed its policy, and then trained a challenger using Q-learning against it for another million steps. He then tested Minimax-Q's fixed policy against the fixed policy of the challenger in a subsequent evaluation stage, keeping track of the number of games won against the challenger. Convergence on the Nash policy in training would guarantee winning *at least* 50% of the games completed.

Smooth FP-Q was tested using the same experimental setup: one million steps of training in self-play and the same number for training of a Q-learning Challenger against its fixed policy. To make the results comparable, during the final evaluation stage, there was a probability of 0.1 that the game would end after each step. Littman used this design to simulate a common discount factor.

Smooth FP-Q and its Challenger used parameters identical to the ones in Littman's test. Exploration was ϵ-greedy, with ϵ fixed at 20%. The learning rate $\alpha(t)$ at time t was cooled as follows:

$$\alpha(t) = \alpha(t-1) \times 10^{\log 0.01/10^6}, \tag{7.2}$$

and the starting α was set to 1. As in Biased RPS, the Smooth FP-Q Q-values were normalized in $[0, 10]$ prior to computing Equation 5.3. In addition, the same λ was used in every state.

The results for different sets of λ and M are represented by Table 7.4. Using $\lambda = 0.8$ and $M = 50$ yielded success in 55.7% of all games completed during the final evaluation stage. This clearly illustrates that the learned mixed policies in self-play are close to the Nash equilibrium ones. The Q-learning Challenger was not always able to learn an optimal exploitation policy despite the fact that it used the same learning and exploration rates.

Exploiting a deterministic learner, however, was an easy task for the Challenger.

	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
$M = 25$	3.5	29.0	34.9	43.6	29.7	31.1
$M = 50$	5.2	35.0	39.8	42.3	36.7	40.2
$M = 100$	9.7	39.1	40.4	43.1	38.9	42.1
$M = 500$	3.9	41.8	52.5	**55.7**	48.2	48.3
$M = 1000$	8.4	47.6	53.4	54.2	51.0	45.3
	$\lambda = 1.1$	$\lambda = 1.3$	$\lambda = 2.0$	$\lambda = 3.0$	$\lambda = 5.0$	$\lambda = 10.0$
$M = 25$	34.5	30.5	21.5	15.0	10.4	6.1
$M = 50$	34.9	31.6	22.7	15.6	10.8	6.3
$M = 100$	36.1	35.5	24.3	16.3	10.6	6.2
$M = 500$	41.7	39.8	23.9	15.9	10.1	6.4
$M = 1000$	47.9	39.1	23.5	16.6	10.4	6.4

Table 7.4: Grid Soccer: Percentage of games won (out of all games completed) by Smooth FP-Q policy, trained in self-play, tested against a Q-learning Challenger. M is size of memory window, λ is smoothing parameter. Optimal policy would win at least 50%. Averaged over 50 runs.

In order to demonstrate Smooth FP-Q's improvement over FP-Q, the latter was also trained in self-play using identical settings for memory size, learning rate, and exploration schedules. As expected, its learned policy was completely dominated by the Challenger policy: FP-Q was able to win in 0.0% percent of the cases.

For comparison, Minimax-Q obtained 37.5% in the same test [Littman, 1994]. While provably optimal, it may have needed more time to converge. Recall from Section 3.3 that JALs have been shown to exhibit faster learning than Minimax-Q [Uther and Veloso, 2003]. In addition, the Minimax-Q result is averaged over only 3 runs. Smooth FP-Q exhibited significant variability in its performance across test runs. It is possible that further testing of Minimax-Q would have revealed a better average performance.

Other algorithms have also exhibited strong performance on this test. Bowling and Veloso [2002] used the experiment in their analysis of WoLF-PHC. Their algorithm was able to win over 40% of its games against the Challenger, and improved with more time for training.

As a final note, observe that while Smooth FP-Q was able to exhibit strong performance with the right set of parameters, it also exhibited high sensitivity to small tweaks in their value. This means that it may be hard to predict Smooth FP-Q's performance with arbitrary parameters.

7.3 Summary

In this chapter, a series of experiments demonstrated that Smooth FP-Q overcomes the major limitation of previous JAL implementations and can successfully learn mixed policies. The algorithm meets the *Safety* requirement on a zero-sum matrix game against an optimal opponent. In addition, mixed policies learned in self-play perform well against a subsequently trained Q-learning Challenger on a large zero-sum stochastic game.

Chapter 8

Concluding Remarks

This thesis presented an extensive evaluation of Joint Action Learners in competitive stochastic games. The evaluation was conducted with respect to a new set of criteria informed by previous work in the multi-agent learning literature. The adopted criteria were: *Rationality* – the ability to learn a best-response against stationary (or convergent) opponents, *Safety* – the ability to obtain the safety value of a game, and *Constant Adaptability* – the ability to remain equally adaptive to changes in the environment throughout the learning process.

The prospects of obtaining *Stability* in the learning process by reaching a Nash equilibrium were also considered. Previously published results reveal that it is impossible to guarantee that two adaptive agents will converge on a Nash equilibrium unless all rewards are fully observable [Hart and Mas-Colell, 2003]. A new result demonstrated that the impossibility holds even if one of the agents has already converged on a stationary Nash policy.

It was demonstrated that FP-Q, the JAL previously discussed in the literature, can provably meet the *Rationality* criterion. This property translated into a very strong performance in an extensive series of tests against other convergent learning algorithms. In an all-versus-all tournament setting, FP-Q was compared to single-agent Q-learning [Watkins and Dayan, 1992], WoLF-PHC [Bowling and Veloso, 2002], Nash-Q [Hu and Wellman, 2003], CE-Q [Greenwald and Hall, 2003], and a stationary

random player. Each match was played several times for each of three different stochastic games. Observations were recorded of the number of times an algorithm was able to obtain the best-response reward against the policy of its opponent, as well as its average reward across all matches versus the same opponent on a given game. FP-Q outperformed all multi-agent extensions of Q-learning. Single-agent Q-learning also exhibited strong performance. However, FP-Q consistently obtained higher rewards in direct matches against it.

While its performance against stationary and convergent opponents is demonstrably convincing, FP-Q cannot always fare well against adaptive opponents. In particular, it cannot always meet the *Safety* criterion due to its inability to play mixed policies. This was demonstrated in a test against an optimal Exploiter on the Biased RPS game, in which FP-Q's average reward after extensive training was considerably lower than the safety value of the game.

To address this limitation, the thesis presented Smooth FP-Q – a variant of FP-Q that is capable of playing mixed policies. Smooth FP-Q's design was informed by work done in the game theory community on extending the fictitious play algorithm. A randomized action selection mechanism replaced FP-Q's best-response dynamics. With the appropriate parametrization, Smooth FP-Q was able to secure the safety value in Biased RPS against the optimal Exploiter, and perform well against other learners such as WoLF-PHC. Its ability to learn beneficial mixed policies in large stochastic games was demonstrated on the grid-world game of Soccer [Littman, 1994]. The policies it could learn in self-play proved impervious to exploitation by a Q-learning Challenger.

Both FP-Q and Smooth FP-Q remain unable to satisfy the *Constant Adaptability* requirement. No experiments were conducted with respect to this criterion, but none were needed. It is clear that Joint Action Learners or any other Q-learning variants cannot satisfy *Constant Adaptability* due to the decreasing learning rate α. As discussed in Section 3.2.1, decreasing α is required for guaranteeing convergence of Q-learning in single-agent environments. In addition, it is necessary for exhibiting *Rationality* in multi-agent settings (see proof of Theorem 2 in Section 5.1). Investigating how Joint Action Learners can preserve their *Rationality* in stationary settings

but remain constantly adaptive against non-stationary opponents is a primary goal for future research.

The importance of adaptability was demonstrated by Chang and Kaelbling [2002]. They designed a cunning adaptive agent that was able to exploit the rational and convergent WoLF-PHC. The strength of their PHC-Exploiter was in maintaining beliefs about the strategy adopted by the opponent. Joint Action Learners also maintain such beliefs through explicit opponent modeling. Thus, the key to achieving *Constant Adaptability* may be in designing JALs that make better use of their opponent modeling capabilities. For example, it may be possible to design a JAL that is capable of distinguishing between stationary and adaptive opponents and adjusting its learning rate α accordingly.

In addition to designing constantly adaptive JALs, future research could focus on articulating new criteria for the rewards against opponents that are not stationary or convergent. At minimum, the learner should be able to obtain the safety value of the game, as required by the *Safety* criterion. However, it may be possible to guarantee higher rewards, depending on the capabilities of the opponent. Advances in the no-regret literature [Foster and Vohra, 1999] might offer helpful insights. Smooth FP-Q is a promising performer with respect to new criteria from the no-regret literature because it adopts the smooth fictitious play action selection mechanism for each state, and smooth fictitious play satisfies a certain no-regret property in matrix games.

Another venue for future research would be to address the parametrization challenge for Smooth FP-Q. The new JAL algorithm proved to be very sensitive to parametrization and it may be hard to predict the right parameters for a given game. It would be important to understand if parameters that emerge as optimal in self-play are also conducive to good performance against other opponents. If this is the case, the parameters can be tweaked in self-play before the algorithm is deployed against unknown opponents. Alternatively, parameters may be *learnable*. As the agent accumulates information about its performance in history, it can reason about its theoretical performance under different parametrization. It can then adjust its current parameters accordingly.

Establishing specific evaluation criteria for learning in competitive games proved to be unexpectedly challenging. Learning literature in AI and game theory puts a great emphasis on investigating the convergence properties of algorithms. Convergence to optimal policy is the natural yardstick in single-agent domains. Convergence to Nash or other equilibria may be of interest in multi-agent settings if the goal being pursued is to design cooperative artificial agents or to describe and predict the behavior of natural agents. In designing competitive artificial agents, however, requiring convergence rarely makes sense. It is only desirable if opponents are stationary, when it is rational to converge to a best-response.

If artificial agents are to supplement humans as decision-makers in economic transactions and other domains, they should be capable of adequate reaction to changes in governments or companies' management, changes in preferences reflecting new trends, or updates in the software of other artificial agents. Thus, agents must remain adaptive with time.

Research in multi-agent learning has not yet converged on the right evaluation criteria, or the algorithms that could reliably meet them in real-world settings. Our hope is that more progress could be made through continuous exploration.

Bibliography

[Bellman, 1957] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Blackwell, 1956] D. Blackwell. Controlled random walks. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 336–338, 1956.

[Bowling and Veloso, 2000] Michael Bowling and Manuela Veloso. An analysis of stochastic game theory for multiagent reinforcement learning. Technical Report CMU-CS-00-165, Carnegie Mellon University, Pittsburgh, PA, 2000.

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

[Bowling, 2005] Michael Bowling. Convergence and no-regret in multiagent learning. Technical Report TR04-11, University of Alberta, 2005.

[Brafman and Tennenholtz, 2004] Ronen I. Brafman and Moshe Tennenholtz. Efficient learning algorithm. *Artificial Intelligence*, 159:27 – 47, 2004.

[Brown, 1951] G. W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*. John Wiley and Sons, New York, 1951.

[Chang and Kaelbling, 2002] Yu-Han Chang and Leslie Kaelbling. Playing is believing: The role of beliefs in multi-agent learning. In *Advances in Neural Information Processing Systems 14*, pages 1483 – 1490. MIT Press, 2002.

[Chen and Deng, 2005a] X. Chen and X. Deng. 3-nash is ppad-complete. Technical Report TR05-134, ECCC, 2005.

[Chen and Deng, 2005b] X. Chen and X. Deng. Settling the complexity of 2-player nash equilibrium. Technical Report TR05-140, ECCC, 2005.

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

[Cottle *et al.*, 1992] Richard W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, New York, 1992.

[Daskalakis *et al.*, 2005] C. Daskalakis, P. W. Goldber, and C. H. Papadimitriou. The complexity of computing nash equilibrium. Technical Report TR05-115, ECCC, 2005.

[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.

[Foster and Vohra, 1997] Dean Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40 – 55, 1997.

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

[Fudenberg and Kreps, 1993] Drew Fudenberg and David Kreps. Learning mixed equlibria. *Games and Economic Behavior*, 5:320 – 367, 1993.

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

[Fudenberg and Levine, 1998] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.

[Fudenberg and Levine, 2006] Drew Fudenberg and David K. Levine. An economists perspective on multi-agent learning. Forthcoming, 2006.

[Greenwald and Hall, 2003] Amy Greenwald and Keith Hall. Correlated-q learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 242–249, 2003.

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

[Hart and Mas-Colell, 2003] Sergiu Hart and Andreu Mas-Colell. Uncoupled dynamics do not lead to nash equilibrium. *American Economic Review*, 93:1830–1836, 2003.

[Hu and Wellman, 2003] Junling Hu and Michael P. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[Kaelbling *et al.*, 1996] Lesley P. Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Joournal of Artificial Intelligence Research*, 4:237 – 285, 1996.

[Kakade, 2003] S. M. Kakade. *On the sample complexity of reinforcement learning.* PhD thesis, University College London, 2003.

[Kalai and Lehrer, 1993] E. Kalai and E. Lehrer. Rational learning leads to nash equilibria. *Econometrica*, 61:1019 – 1045, 1993.

[Krishna and Sjostrom, 1995] V. Krishna and T. Sjostrom. On the convergence of fictitious play. Mimeo, Harvard University, 1995.

[Littman and Szepesvari, 1996] Michael L. Littman and C. Szepesvari. A generalized reinforcement-learning model: Convergence and applications. In *Proceedings of the 13th International Conference on Machine Learning*, pages 310 – 318, Bari, Italy, 1996. Morgan Kaufmann.

[Littman, 1994] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kaufmann.

[Littman, 2001] Michael L. Littman. Firend-or-foe q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322 – 328. Morgan Kaufman, 2001.

[Mannor and Shimkin, 2003] Shie Mannor and Nahum Shimkin. The empirical bayes envelope and regret minimization in competitive markov decision processes. *Mathematics of Operations Research*, 28(2):327 – 345, 2003.

[Massaro and Friedman, 1990] D. Massaro and D. Friedman. Models of integration given multiple sources of information. *Psychological Review*, 97:22 – 252, 1990.

[McKelvey and McLennan, 1996] R. McKelvey and A. McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, J. Rust, Michael D. Instriligator, and Kenneth J. Arrow, editors, *Handbook of Computational Economics*, volume 1. Elsevier, 1996.

[Mitchell, 1997] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Miyasawa, 1961] K. Miyasawa. On the convergence of learning processes in a 2 x 2 non-zero-person game. Research Memo 33, Princeton University, 1961.

[Myerson, 1991] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, MA, 1991.

[Nachbar and Zame, 1996] J. H. Nachbar and W. R. Zame. Non-computable strategies and discountable repeated games. *Economic Theory*, 8:103 – 122, 1996.

[Nachbar, 1990] J. Nachbar. "evolutionary" selection dynamics in games: Convergence and limit properties. 19:59 – 89, 1990.

[Nash, 1951] John Nash. Non-cooperative games. *Annals of Mathematics*, 54:286 – 295, 1951.

[Osborne and Rubinstein, 1994] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[Powers and Shoham, 2005] Rob Powers and Yoav Shoham. New criteria and a new algoithm for learning in multi-agent systems. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2005.

[Robinson, 1951] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296 – 301, 1951.

[Russel and Wefald, 1991] Stuart Russel and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361 – 395, 1991.

[Sandholm *et al.*, 2005] Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. Mixed-integer programming methods for finding nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 491 – 501, Pittsburgh, PA, 2005.

[Sen *et al.*, 1994] Sandip Sen, Mahendra Sekaran, and John Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intellience*, pages 426 – 431, Seattle, WA, 1994.

[Shapley, 1964] L. Shapley. Some topics in two-person games. In M. Drecher, L. S. Shapley, and A. W. Tucker, editors, *Advances in Game Theory*. Princeton University Press, 1964.

[Shoham *et al.*, 2003] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University, 2003.

[Shoham *et al.*, 2006] Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? Forthcoming, 2006.

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

[Singh *et al.*, 2000] Satinder P. Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvari. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Uther and Veloso, 2003] Wiliam Uther and Manuela Veloso. Adversarial reinforcement learning. Technical Report CMU-CS-03-107, Carnegie Mellon University, Pittsburgh, PA, 2003.

[Watkins and Dayan, 1992] Christopher J.C.H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279 – 292, 1992.

[Young, 1993] H. Peyton Young. The evolution of conventions. *Econometrica*, 61:57–84, 1993.

[Zinkevich *et al.*, 2005] Martin Zinkevich, Amy Greenwald, and Michael L. Littman. Cyclic equilibria in markov games. In *Proceedings of the Neural Information processing Systems Conference*, 2005.