Improving Convergence Rates in Multiagent Learning Through Experts and Adaptive Consultation

by

Greg Hines

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Multiagent learning (MAL) is the study of agents learning while in the presence of other agents who are also learning. As a field, MAL is built upon work done in both artificial intelligence and game theory. Game theory has mostly focused on proving that certain theoretical properties hold for a wide class of learning situations while ignoring computational issues, whereas artificial intelligence has mainly focused on designing practical multiagent learning algorithms for small classes of games.

This thesis is concerned with finding a balance between the game-theory and artificial-intelligence approaches. We introduce a new learning algorithm, FRAME, which provably converges to the set of Nash equilibria in self-play, while consulting experts which can greatly improve the convergence rate to the set of equilibria. Even if the experts are not well suited to the learning problem, or are hostile, then FRAME will still provably converge. Our second contribution takes this idea further by allowing agents to consult multiple experts, and dynamically adapting so that the best expert for the given game is consulted. The result is a flexible algorithm capable of dealing with new and unknown games. Experimental results validate our approach.

Acknowledgements

First and foremost, I would like to thank my supervisor Kate Larson. The freedom she gave me as a graduate student along with the support and motivation made her a unique and outstanding supervisor. I would also like to thank my readers, Robin Cohen and Pascal Poupart for their excellent feedback and comments.

This thesis is based on work done in several submitted papers. I would like to thank the reviewers of those papers for their comments. I would also like to thank the faculty and students at the Toyota Technical Institute at Chicago for the feedback they gave me. As well, I would like to thank Gord Hines from the University of Guelph for the numerous discussions on probability theory and help with Lemma 3.

Finally, I would like to thank my fellow graduate students, past and present; Tyrel Russell, Reid Kerr, Laurent Charlin, Martin Talbot, Rob Warren, Fred Kroon, John Whissell, Mattt Enss and Kevin Reagan. Besides the technical help they so often provided, their friendship has helped me through many difficult periods and made my Master's experience a very enjoyable one.

Dedication

To my parents,

I owe the two of you the world.

To Lisa,

I cannot imagine going through all of this without your constant love and support.

Contents

List of Figures

List of Tables

Glossary

A	The set of possible joint actions, 7
A_i	The set of possible actions for agent i, 7
A_{-i}	The set of possible joint actions for all agents except agent i, 7
a_i	A specific but unspecified action by agent i, 7
a_{-i}	A specific but unspecified joint action by all agents but agent i, 7
$BR_i^\epsilon(\sigma_{-i})$	Agent i's ϵ-best response to σ_{-i}, i.e. the set of agent i's strategies which achieve a utility with in ϵ of agent i's maximum possible utility with respect to σ_{-i}, 14
$BR_i(\sigma_{-i})$	Agent i's best response to σ_{-i}, i.e. the set of agent i's strategies which maximize agent i's utility with respect to the joint strategy σ_{-i}, 14
C	A set of possible prediction schemes to be used in an online decision problem, 28
c_j	A prediction scheme used in an online decision problem, 28
$d(G)$	The smallest ϵ value such that every agent in the game G has an ϵ-subdominant strategy, 14
δ_l	A step size for a "loosing agent using WoLF, 43
δ_w	A step size for a "winning agent using WoLF, 43

δ_{a_i}	A probability distribution over A_i with all mass concentrated at a_i, 26
Dominant Strategy	A strategy is a best response regardless of what σ_{-i} is, 14
E	A set of experts, 44
E_i	Agent i's set of experts in adaptive-FRAME, 75
ϵ-subdominant strategy	A strategy that is an ϵ-best response regardless of what σ_{-i} is, 14
η	The probability of FRAME and adaptive-FRAME reseting to a strategy chosen uniformly at random from Σ, 49
$e_i(\cdot)$	Agent i's expert in FRAME, 48
$\mathbf{æ}_i$	An experts algorithm, 76
G	A stage game, 7
Γ^G	A repeated game based on the stage game G, 16
γ_i	The smallest probability of choosing any particular strategy at random from Σ_i^h, 33
h_i	The parameter for Σ_i^h for agent i in Regret Testing, 33
$L^T(H)$	The overall loss from using the set of prediction schemes H over T periods of time in an online decision problem, 28
$L_{c_i}^t$	The loss from using prediction scheme c_i at time t in an online decision problem, 28
Λ	The probability of choosing an action uniformly at random in Regret Testing, 33
κ	The decay factor in an agent's expected utility in a repeated game, 16
λ	The "smoothness parameter for LFP and LERRM, 41
μ	A parameter in the HMC algorithm, 30

m_i	The size of A_i, 7
N	Set of all agents, 7
$\mathcal{N}^G$	The set of all strategies for the game G which are Nash equilibria, 9
$\mathcal{N}_\epsilon^G$	The set of all strategies for the game G which are ϵ-Nash equilibria, 9
$\mathcal{N}_\epsilon^c$	The set of all strategies for the game G which are not ϵ-Nash equilibria, 9
ν	A step size in IGA, 42
n	Number of agents, 7
Φ	A finite set of linear maps between strategies, 25
$p_{e_i}^t$	The probability of agent i consulting expert e_i at time t in adaptive-FRAME, 76
ϕ	A linear map between two strategies, 25
p	The probability of agent i consulting expert e_i in FRAME, 49
$p_{c_j}^t$	The probability of using prediction scheme c_j at time t in an online decision problem, 28
q	An initial probability distriubtion over possible states and signals for a repeated game, 15
$R^T(H)$	The regret due to the set of prediction schemes H over T periods of time, 29
$R_{h_j}^T(H)$	The regret due to prediction scheme h_j over T periods of time, 29
ρ_i	The maximum regret value for which agent i does not change its strategy in Regret Testing, ERT and ALERT, 33, 36, 38
r^∞	The limit of r^t as t approaches infinity if it exists, 51
r^t	The regret of a repeated game at time t, 16
$r_i(\sigma)$	Agent i's regret with respect to the joint strategy σ, 13

S_i	The set of all possible signals for agent i in a repeated game, 15
Σ	The set of all possible uncorrelated strategies, 8
Σ_i^h	The set of all strategies for agent i where probability in each strategy can be expressed as a multiple of $1/h$, 33
Σ_i	The set of all agent i's possible strategies, 8
Σ_{-i}	The set of all possible uncorrelated strategies for all agents except agent i, 8
σ	An uncorrelated probability distribution over all possible joint actions, 8
σ^∞	The limit of σ^t as t approaches infinity if it exists, 51
σ_i^t	The strategy for agent i at time t in a repeated game, 16
σ_A	A probability distribution over all possible joint actions, 10
σ_i	A probability distribution over agent i's possible actions, 8
σ_{-i}	A specific joint strategy for all agents except agent i, 8
s_i	A specific signal sent to agent i in a repeated game, 15
Θ	The set of all possible states of a repeated game, 15
$\mathbb{T}$	A transition function for a repeated game which maps a state and joint action to a probability distribution over possible states and signals, 15
$\mathcal{T}$	The number of iterations in Regret Testing and ERT for which agents must keep the same strategy, 33, 36
θ	A specific state of a repeated game, 15
t	A specific iteration in a repeated game, 15
u_i	Utility function for agent i, 7

v_i	The value of a repeated game for agent i, 16
Ξ^t_{-i}	The cumulative frequency of play for agent i's opponents up until time t, 40
ζ	The probability that even with low regret an agent chooses a strategy in ERT and ALERT, 36, 38

Chapter 1

Introduction

Multiagent learning (MAL) is the study of agents learning while in the presence of other agents who are also learning. To best understand MAL, it helps to consider the study of single agent learning. A simple example of a single agent learning problem is given in Figure 1.1.

Agent 1	$a_{1,1}$	2
	$a_{1,2}$	5

Figure 1.1: A simple single agent learning problem

In this problem an agent is given the choice of two actions, $a_{1,1}$ and $a_{1,2}$. If the agent chooses action $a_{1,1}$ it receives a utility of 2 and if the agent chooses action $a_{1,2}$ it receives a utility of 4. This process repeats infinitely often. Over the course of these repetitions the agent must learn which action gives it the highest utility. There are many well understood techniques to solve this problem and much harder ones. In fact single agent learning has many real world applications; from helping people with dementia through different activities, to helping computers understand human dialogue and even to helping helicopters fly autonomously [6, 42, 47].

However, it can become more difficult when there are multiple agents trying to all learn autonomously. Suppose the problem in Figure 1.1 was generalized to a *game* involving two agents as shown in Figure 1.2. In this game, agent 1 and agent 2 each simultaneously choose an action. The cell at the intersection of these two actions gives the utility for both agents. For example, if agent 1 and agent 2 choose

actions $a_{1,2}$ and $a_{2,2}$ respectively than agent 1 will receive a utility of 5 and agent 2 will receive a utility of 0.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	2,2	0,5
	$a_{1,2}$	5,0	1,1

Figure 1.2: A multiagent version of Figure 1.1

As with the agent in Figure 1.1, the agents in Figure 1.2 must devise a way of maximizing utility through the use of some learning algorithm . However, the game in Figure 1.2 is harder for several reasons. First, the learning algorithm agent 1 uses has an affect on agent 2 and vice-versa. For example, it is possible for agent 1's learning algorithm to prevent agent 2 from ever receiving a "decent utility". Therefore, an additional challenge in multiagent learning can be trying to create learning algorithms which minimize the negative impact they have on other agents. The second issue is that unlike the problem in Figure 1.1, which a clear and unique solution, there are several different possible solutions to the game in Figre 1.2. The maximum utility for both agents can be either 1, 2 or 2.5 depending on how we specify the model for the game. The problem is that models for MAL can be more flexible than those for single agent learning. Therefore, we must find a way to be more specific in how we specify MAL models.

One common solution to both of these issues is the use of *game theory*. Game theory provides a method for understanding how agents interact with each other. (A formal introduction to game theory is given in Chapter 2.) Games can be used to describe many different phenomena. A simple example is the game in Figure 1.2. More complex examples include auctions for advertisement slots on Google's search results pages or routing on the Internet [17, 53]. In other words, game theory can be used to help analyze situations worth billions of dollars or of fundamental importance to modern society. With respect to single agent learning, game theory allows for a generalization to a situation with multiple agents. This could include helping to mediate between multiple autonomous helicopters or optimizing routing over the Internet.

Since single agent learning and game theory have such different backgrounds, it is not surprising that researchers in MAL have varied backgrounds and hence also varied goals. Shoham *et. al.* have identified five main areas or agendas in MAL: [49]

1. Computational
2. Descriptive
3. Normative
4. Prescriptive, cooperative
5. Prescriptive, non-cooperative

The *computational* deals with developing computational approaches to determining properties of games. The *descriptive agenda* deals with building models of how people learn in multiagent systems. The *normative agenda* tries to understand which learning approaches are in equilibrium with each other.[1] Finally, the *prescriptive* agendas deal with how agents should learn in cooperative and non-cooperative settings, respectively. In a cooperative setting, agents are trying to improve the group as a whole while in a non-cooperative setting, agents are only concerned with helping themselves.

Shoham *et. al.* have argued that the last agenda is the most interesting [49]. We agree with this view. However, Shoham *et. al.* ignore the fact that the boundary between these agendas is not always clear. Specifically, this thesis argues that the normative agenda plays an important role in the prescriptive, non-cooperative agenda.

Although many MAL algorithms use both machine learning and game theory ideas, the algorithms can often be characterized by their balance between the two. MAL algorithms that are mostly machine learning tend to focus on achieving results for specific situations. On the other hand, game-theoretic MAL tends to be more concerned with the universality of the properties of the learning algorithms than with their computational properties.

1.1 Approach

This thesis is concerned with trying to get the best of both the machine learning and game theory worlds. An ideal learning algorithm would be universal in its application but also as practical as possible.

[1] An equilibrium is some strategy from which no agent wants to unilaterally deviate. A formal definition is given in Chapter 2.

The approach will focus on recent work in the game theory community which developed a learning algorithm, ALERT, that can be used for basically all games [30]. However, agents using ALERT make very naive decisions, and as a result the algorithm is not practical for even the simplest of games. (We measure an algorithm's practically by how easily a computer could implement it.) By using ideas from machine learning, we are able to help agents make much more efficient decisions. We also rely on experts and experts algorithms, which are ideas from single agent learning.

1.2 Contributions

Specifically, this thesis introduces three new algorithms. The first algorithm is FRAME. FRAME is a multiagent learning algorithm which achieves a strong balance between theoretical concerns and practical ones. On the theoretical side, FRAME can be used in any game to find a solution from which no agent will want to unilaterally deviate from. On the practical side, FRAME, in part through the use of consultation of a single expert, is shown to be a useful algorithm in realistic situations on computers.

Our second algorithm is adaptive-FRAME. Adaptive-FRAME achieves the same theoretical guarantees as FRAME does. However, adaptive-FRAME is able to consult multiple experts and does so adaptively (i.e., better experts are consulted more often). Thus adaptive-FRAME achieves a considerable improvement in performance over FRAME. As a result, adaptive-FRAME can be used in many situations where FRAME cannot. Part of the improvement in adaptive-FRAME comes from the use of experts algorithms.

The last algorithm, LERRM, is an experts algorithm designed specifically for adaptive-FRAME. We show that LERRM is competitive with existing standard experts algorithms, and in some cases LERRM can out perform them.

1.3 Guide to the Thesis

Chapter 2 This chapter provides background on multiagent learning. This also covers different metrics for examining learning algorithms.

Chapter 3 This is the related research chapter. This chapter is broken up into three parts. The first introduces the game-theoretic algorithms which FRAME

generalizes. The second part introduces the machine learning MAL ideas that FRAME and adaptive-FRAME use to help achieve better performance. Finally, the third part talks about the machine learning idea of experts algorithms, which is used in LERRM.

Chapter 4 This chapter introduces FRAME. We prove that FRAME achieves convergence to the set of Nash equilibria and in many cases can converge to a single Nash equilibria.

Chapter 5 This chapter shows the experimental results of FRAME. Our experiments involved several games covering a wide range of types of games.

Chapter 6 This chapter introduces adaptive-FRAME, which is a generalization of FRAME. We prove that adaptive-FRAME is also able to guarantee convergence to the set of Nash equilibria and in many cases can converge to a single Nash equilibria. We also present experimental results that show adaptive-FRAME is able to achieve considerable improvement in the convergence rate in many games over FRAME. Finally, we introduce LERRM, an experts algorithm designed specifically for adaptive-FRAME.

Chapter 7 This is the conclusion. We also suggest areas for future research.

Chapter 2

Background

The setting for our work is repeated games. Repeated games are based on stage games, which are introduced in Section 2.1. Section 2.2 describes the idea of regret which is used throughout this work. In Section 2.3 we introduce repeated games. Section 2.4 introduces learning in repeated games and discusses some properties with which we measure potential solutions.

2.1 Stage Games

A n-player *stage game* is a tuple $G = \langle N, A = A_1 \times \ldots \times A_n, u_1, \ldots, u_n \rangle$ where $N = \{1, \ldots n\}$ is the set of agents in the game, A_i is the set of actions available for agent i to play, A is the set of possible joint actions and $u_i : A \rightarrow \mathbb{R}$ is the utility function for agent i. We denote the size A_i by m_i. We let a_i denote a specific action taken by agent i. As is standard in the literature, we will use A_{-i} to denote the joint actions of all agents but agent i, i.e. $A_{-i} = \{A_1 \times \ldots \times A_{i-1} \times A_{i+1} \times \ldots \times A_n\}$, and $a_{-i} \in A_{-i}$, $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ to be a particular joint action. Agents are all *self-interested*, in that their only concern is maximizing their own utility.

Figure 2.1 shows Battle of the Sexes, a classic example of a stage game. The basic idea behind Battle of the Sexes is that agents would like to coordinate on an action but cannot agree on which action to coordinate on.

When a stage game is given in matrix form, the game is said to be in *normal form*. Agent 1 and agent 2 will each pick an action simultaneously. Agent 1's action can be thought of as picking a row in the matrix. Likewise, agent 2's action can be thought of as picking a column in the matrix. The cell at the intersection of the

row and column gives the utility for both agents. The first value in that cell gives the utility for agent 1 and the second value gives the utility for agent 2. All of the concepts are generalized in the obvious manner for games with more than 2 agents.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	$1, 0.5$	$0, 0$
	$a_{1,2}$	$0, 0$	$0.5, 1$

Figure 2.1: An example of a stage game: Battle of the Sexes

We assume that agents play *strategies*.

Definition 1 *A strategy, σ_i, for agent i, is a probability distribution over its action set A_i, stating with what probability the agent will play each action. A pure strategy is one in which the agent plays one action with probability equal to one. All other strategies are called mixed strategies. The set of all possible strategies for agent i is Σ_i. The profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a joint strategy among all agents and $\Sigma = \times_{i=1}^{n} \Sigma_i$ is the set of all possible joint strategies.*

We define $\Sigma_{-i} = \Sigma_1 \times \ldots \times \Sigma_{i-1} \times \Sigma_{i+1} \times \ldots \times \Sigma_n$ and σ_{-i} to be an element of Σ_{-i}.

By abuse of notation, we define

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{(a_i, a_{-i}) \in A} u_i(a_i, a_{-i}) \sigma_i(a_i) \sigma_{-i}(a_{-i}). \tag{2.1}$$

In words, agent i's expected utility with respect to its strategy σ_i and its opponents' joint strategy σ_{-i} is the sum of the utility over all possible joint actions of the utility for agent i of a joint action multipled by the probability of that joint action happening due to the strategies σ_i and σ_{-i}. We also define an agent's utility with respect to playing a specific again as

$$u_i(a_i, \sigma_{-i}) = \sum_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}) \sigma_{-i}(a_{-i}). \tag{2.2}$$

In words, agent i's expected utility with respect to playing action a_i given its opponents' joint strategy σ_{-i} is the sum of the utility over all possible joint actions that include a_i of the utility for agent i of such a joint action multipled by the probability of the that joint action happening due to the joint strategy σ_{-i}.

Referring back to Figure 2.1, a possible strategy for agent 1 would be $\sigma_1 = (1/2, 1/2)$. If agent 2 also had the same strategy, i.e. $\sigma_2 = (1/2, 1/2)$, then we would calculate the utility of agent 1 as follows;

$$\begin{aligned} u_1(\sigma_1, \sigma_{-1}) &= u_1((a_{1,1}, a_{2,1}))\sigma_1(a_{1,1})\sigma_{-1}(a_{2,1}) \\ &+ u_1((a_{1,1}, a_{2,2}))\sigma_1(a_{1,1})\sigma_{-1}(a_{2,2}) \\ &+ u_1((a_{1,2}, a_{2,1}))\sigma_1(a_{1,2})\sigma_{-1}(a_{2,1}) \\ &+ u_1((a_{1,2}, a_{2,2}))\sigma_1(a_{1,2})\sigma_{-1}(a_{2,2}), \\ &= 1 \cdot \frac{1}{2} \cdot \frac{1}{2} + 0 + 0 + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2}, \\ &= \frac{3}{8}. \end{aligned} \tag{2.3}$$

By the symmetry of the game, agent 2's utility would be the same.

By definition, the agents' strategies are a *Nash equilibrium* if no agent is willing to change its strategy, given that no other agents change theirs [41].

Definition 2 *A strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ is a Nash equilibrium if for every agent i*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i', \sigma_{-i}^*), \ \forall \sigma_i' \neq \sigma_i^*. \tag{2.4}$$

A strategy profile σ^* is an ϵ-Nash equilibrium if for every agent i

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i', \sigma_{-i}^*) - \epsilon, \ \forall \sigma_i' \neq \sigma_i^*. \tag{2.5}$$

In the game in Figure 2.1, an example of an ϵ-Nash equilibrium would be the joint strategy

$$\{\sigma_1, \sigma_2\} = \{(1, 0), (3/4, 1/4)\} \tag{2.6}$$

for $\epsilon = 1/4$.[1] To show that this is an ϵ-Nash equilibrium, we must show that neither agent could increase its utility by more than ϵ by unilaterally deviating from this joint strategy. Given agent 2's strategy, agent 1's best response is $(1, 0)$ which is its current strategy. Therefore, agent 1 cannot increase its utility at all. However, given agent 1's strategy, agent 2's best response is also $(1, 0)$ which is not its current strategy.[2] Agent 2's current strategy provides a utility of $\frac{1}{4}$ while its best response provides a utility of $\frac{1}{2}$. Therefore, agent 2 could increase its utility by at most ϵ by changing its strategy. Therefore, $\{(1, 0), (1 - \epsilon, \epsilon)\}$ is an ϵ-Nash equilibrium.

[1] This actually holds for all ϵ. For simplicity, we examine only one case.

[2] As long as one action provides more expected utility than any other action, playing only that action maximizes utility. Thus, we need only check every pure strategy to find the best response.

We denote the set of all Nash equilibria for some game G by $\mathcal{N}^G$, and the set of all ϵ-Nash equilibria by $\mathcal{N}_\epsilon^G$. When it is clear from the context, we will drop the game index G and use $\mathcal{N}$ and $\mathcal{N}_\epsilon$ respectively. Finally, let $\mathcal{N}_\epsilon^c$ be the set of all joint strategies that are not ϵ-Nash equilibria.

If we assume that agents play actions (i.e. instead of just playing a strategy, agents are forced to pick an action according to their strategy), then we can also define joint strategies in terms of the probabilities that each joint action is played.

Definition 3 *Define σ_a to be the probability that the joint action a is played. We can now define a* correlated strategy*, σ_A, as the probability distribution*

$$\sigma_A = \{\sigma_a | a \in A\}. \tag{2.7}$$

This is a correlated strategy because

$$\sigma_A((a_i, a_{-i})) = \sigma_i(a_i)\sigma_{-i}(a_{-i}) \tag{2.8}$$

does not have to hold. Instead the strategy one agent plays can have an effect on the strategies other agents choose to play.

We define $\sigma_{A_{-i}}$ to be the correlated strategy for all agents but agent i. Since the strategies are correlated, agent i's choice of strategy will have an effect on what $\sigma_{A_{-i}}$ is. Thus, we can consider the probability $P(\sigma_{A_{-i}}|\sigma_i)$. Whereas with uncorrelated strategies, all strategies had to be known before an expected utility could be calculated, with correlated strategies agent i only needs to know its strategy to calculate its expected strategy. This means that we can calculate the expected utility for agent i based solely on its own strategy as

$$u_i(\sigma_i) = \int_{\sigma_{A_{-i}} \in \Sigma_{A_{-i}}} P(\sigma_{A_{-i}}|\sigma_i) u_i(\sigma_i, \sigma_{A_{-i}}). \tag{2.9}$$

Using Equation 2.9, we can calculate agent i's expected utility as

$$u_i = \int_{\sigma_i \in \Sigma_i} P(\sigma_i) u_i(\sigma_i). \tag{2.10}$$

In games with simultaneous moves, the easiest way to have correlated strategies is for each agent to receive a private signal from a third party suggesting which action to play. In order for these signals to be correlated, each agent must know the relative frequency that each joint signal is used. Since there are only a finite number of actions per agent, there are only a finite number of possible joint signals.

If we assume that each agent follows its suggested action, each agent has only a finite number of possible strategies. As a result, we can simplify Equations 2.9 and 2.10 to

$$u_i(\sigma_i) = \sum_{\sigma_{A_{-i}} \in \Sigma_{A_{-i}}} P(\sigma_{A_{-i}}|\sigma_i) u_i(\sigma_i, \sigma_{A_{-i}}), \tag{2.11}$$

and

$$u_i = \sum_{\sigma_i \in \Sigma_i} P(\sigma_i) u_i(\sigma_i), \tag{2.12}$$

respectively.

As an example, consider the game in Figure 2.1. Suppose that a third party can send one of two signals, B and S, to each of the agents. There is a probability of 0.5 that the third party sends B to both agents and a 0.5 probability that the third party sends S to both agents. If the agents receive the signal B, agents 1 and 2 will choose actions $a_{1,1}$ and $a_{2,1}$ respectively and otherwise they will choose actions $a_{1,2}$ and $a_{2,2}$ respectively. This results in a correlated strategy of

$$\begin{aligned} \sigma_A = \{&\sigma_A((a_{1,1}, a_{2,1})) = 0.5, \sigma_A((a_{1,1}, a_{2,2})) = 0, \\ &\sigma_A((a_{1,2}, a_{2,1})) = 0, \sigma_A((a_{1,2}, a_{2,2})) = 0.5\}. \end{aligned} \tag{2.13}$$

A graphical respresentation of this correlated strategy is shown in Figure 2.2.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	0.5	0
	$a_{1,2}$	0	0.5

Figure 2.2: A graphical representation of the correlated strategy in Equation 2.13

Assuming that both agents follow their suggested strategy, we can use Equations

2.11 and 2.12 to calculate agent 1's expected utility as follows[3]

$$\begin{aligned}
u_1 &= \sum_{\sigma_1 \in \Sigma_1} P(\sigma_1) u_1(\sigma_1) \\
&= P(a_{1,1}) u_1(a_{1,1}) + P(a_{1,2}) u_1(a_{1,2}) \\
&= \frac{1}{2} \left[\sum_{\sigma_{A_2} \in \Sigma_{A_2}} P(\sigma_{A_2} | a_{1,1}) u_1(a_{1,1}, \sigma_{A_2}) \right] \\
&\quad + \frac{1}{2} \left[\sum_{\sigma_{A_2} \in \Sigma_{A_2}} P(\sigma_{A_2} | a_{1,2}) u_1(a_{1,2}, \sigma_{A_2}) \right] \\
&= \frac{1}{2} \left[P(a_{2,1} | a_{1,1}) u_1((a_{1,1}, a_{2,1})) + P(a_{2,2} | a_{1,1}) u_1((a_{2,2}, a_{1,1})) \right] \\
&\quad + \frac{1}{2} \left[P(a_{2,1} | a_{1,2}) u_1((a_{2,1}, a_{1,2})) + P(a_{2,2} | a_{1,2}) u_1((a_{2,2}, a_{1,2})) \right] \\
&= \frac{1}{2}(1 \cdot 1 + 0 \cdot 0) + \frac{1}{2}(0 \cdot 0 + 1 \cdot 1) \\
&= \frac{3}{4}
\end{aligned}$$

By the symmetry of the game, agent 2 will also have an expected utility of $\frac{3}{4}$. It is important to note that each agent receives a higher utility than they would from the mixed Nash equilibrium of this game.

In fact, σ_A is actually an equilibrium, specifically a *correlated equilibrium* [2]. A correlated equilibrium is a correlated strategy in which every time agent i plays the strategy σ_{A_i}, there is no strategy that could have achieved a higher utility. Formally, for agent i and all strategies σ_{A_i} such that $P(\sigma_{A_i}) > 0$,

$$\sum_{\sigma_{A_{-i}} \in \Sigma_{A_{-i}}} P(\sigma_{A_{-i}} | \sigma_{A_i}) u_i(\sigma_{A_i}, \sigma_{A_{-i}}) \geq \sum_{\sigma_{A_{-i}} \in \Sigma_{A_{-i}}} P(\sigma_{A_{-i}} | \sigma'_{A_i}) u_i(\sigma'_{A_i}, \sigma_{A_{-i}}) \quad (2.14)$$

for all $\sigma'_{A_i} \in \Sigma_{A_i}$ and for all $i \in N$.

We can now address the assumption that each agent plays its suggested action. An agent will only play its suggested action if that action maximizes its utility. By definition, if a correlated strategy is also a correlated equilibrium, then that strategy maximizes utility. Hence, for the strategy in Equation 2.13, we are able

[3]For simplicity, we use $a_{1,1}$ to also denote the strategy of playing action $a_{1,1}$ with probability 1.

to assume that agents played their suggested strategy because doing so maximized their utility.

Correlated strategies are dependent on agents receiving signals that are correlated, even if they are private. For example, in the game in Figure 2.1, a set of uncorrelated signals could be agent 1 receiving B $\frac{2}{3}$'s of the time and S $\frac{1}{3}$ of the time while agent 2 receives B $\frac{1}{3}$ of the time and S $\frac{2}{3}$'s of the time. In this case equation Equation 2.11 would simplify to

$$u_i(\sigma_A) = \sum_{a \in A} P(\sigma_{A_{-i}}) \cdot u_i(\sigma_{A_i}, \sigma_{A_{-i}}), \tag{2.15}$$

and we would have uncorrelated play. More importantly, the specific set of uncorrelated signals would result in agents playing a Nash equilibrium. Therefore, the set of correlated equilibria contains the set of Nash equilibria. By using correlated strategies, a richer set of outcomes is possible. However, using correlated strategies requires a more complex model. Before trying to understand such a model, it makes sense to make sure that we understand the simpler model needed for uncorrelated strategies. Thus, for the rest of this thesis, with the exception of the examination of related work, we will only be concerned with uncorrelated strategies.

2.2 Regret

Another notion that agents may use to evaluate their choice of strategy is that of *regret*.

Definition 4 *Given a joint strategy σ, agent i's regret is*

$$r_i(\sigma) = \max_{\sigma_i' \in \Sigma_i} \left[u_i(\sigma_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \right]. \tag{2.16}$$

Given σ, the regret of a game is the maximum regret among all agents, i.e. $r(\sigma) = \max_{i \in N}(r_i(\sigma))$. When σ is obvious, we shall just use r. This may be seen as a measure of how much agent i is hurt by playing strategy σ_i as opposed to any other strategy.

Two other types of regret are external and internal regret [32, 21]; external regret measures regret against all pure strategies while internal regret measures regret for having played action a instead of action b for all $a \neq b$. These notions are expanded upon in Chapter 3.

For an example of regret, we return to the Battle of the Sexes game in Figure 2.1. Suppose both agents use the joint strategy in Equation 2.6. We have already shown that agent 1 cannot increase its utility by changing its strategy. Therefore, agent 1's regret, $r_1(\sigma)$, is 0. We have also shown that agent 2 can increase its utility by at most 1/4 by changing its strategy. Therefore, agent 2's regret, $r_2(\sigma)$, is 1/4.

We can use regret to give another way of defining a Nash equilibrium. If all agents have no-regret about the strategies they are playing, i.e. $r_i = 0, \forall i \in N$, then the strategy profile is a Nash equilibrium. Similarly, if $r_i \leq \epsilon$ for all i, then we have an ϵ-Nash equilibrium.

Another important notion related to regret and Nash equilibria is *best response*.

Definition 5 *The best response for agent i, if all other agents are playing σ_{-i}, is*

$$BR_i(\sigma_{-i}) = \{\sigma_i \in \Sigma_i | r_i(\sigma_i, \sigma_{-i}) = 0\}. \tag{2.17}$$

.

We can also define the *ϵ-best response* in a similar fashion.

Definition 6 *The ϵ-best response for agent i is*

$$BR_i^\epsilon(\sigma_{-i}) = \{\sigma_i \in \Sigma_i | r_i(\sigma_i, \sigma_{-i}) \leq \epsilon\}. \tag{2.18}$$

For Battle of the Sexes, if agent 1's strategy is $(1, 0)$, then agent 2's best response is the set $\{(1, 0)\}$ since any strategy in this set maximizes agent 2's utility with respect to agent 1's strategy. Assuming the same strategy for agent 1, an ϵ-best response for agent 2 would any strategy $\{(1 - \omega, \omega)\}$ for any $\omega \leq \epsilon$. Any such strategy would give agent 2 a regret of at most ϵ.

A stronger notion than best response is *dominant strategy*. A strategy is dominant if it is strictly the best response regardless of what joint strategy the agents' opponents are playing. A slightly weaker notion is an *ϵ-subdominant strategy*.

Definition 7 *A strategy σ_i is an ϵ-subdominant strategy if*

$$\forall \sigma_{-i} \in \Sigma_{-i}, \sigma_i \in BR_i^\epsilon(\sigma_{-i}). \tag{2.19}$$

For a game G, we let $d(G)$ be the least $\epsilon \geq 0$ such that at least one agent has a ϵ-subdominant strategy.

Battle of the Sexes does not have a dominant strategy for either agent. A game with a dominant strategy for both agents is shown in Figure 2.3. In this game, regardless of what strategy the other agent is going to play, it is always in the best interests of the agent to play the strategy $(0, 1)$.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	$2, 2$	$0, 5$
	$a_{1,2}$	$5, 0$	$1, 1$

Figure 2.3: A game with a dominant strategy for both agents

2.3 Repeated Games

In this section we introduce the idea of repeated games, which form the basis for this thesis. Repeated games allow for far greater flexibility than stage games. As a result, many results are possible in repeated games which were not possible in stage games. This section shows what the differences between stage games and repeated games are and how the rest of the thesis will treat repeated games.

Given a stage game, a *repeated game* is one in which all agents play that stage game over and over again. As is standard, we assume that a repeated game is either of infinite length or its length is unknown to all agents. To examine repeated games in a formal setting, we consider the following model based on the one by Mertens *et al.* [37].

Definition 8 *Given a stage game, $G = \langle N, A = A_1 \times \ldots \times A_n, u_1, \ldots, u_n \rangle$, we define the repeated game Γ^G as follows.*

Let Θ be a finite set of possible states for Γ^G. A state in a repeated game is the same idea as a state in a deterministic finite automata (DFA). We define $\theta^t \in \Theta$ to be the state of Γ^G at time t.[4]

At the beginning of turn t, agent i will receive a signal $s_i^t \in S_i$ where S_i is the set of all signals for agent i. These signals may tell the agent which state the game is currently in, although this is not required. Agents can use this signal to help them choose a strategy; $\sigma_i[s_i]$ is agent i's strategy when it receives the signal s_i and σ_i is the set of strategies for all possible signals. If an agent's strategy is dependent on the time, we denote it by σ_i^t, otherwise the strategy is said to be stationary.

The transition from one state to another is given by $\mathbb{T} : A \times \Theta \to \Delta(\Theta \times S)$, where $\Delta(\Theta \times S)$ is the set of probability distributions over all possible state and signal combinations.

[4]Mertens *et al.* also generalize u_i to be dependent on Θ. The resulting game is better known as a stochastic game or a competitive Markov Chain Process. For simplicity, we will keep u_i independent of Θ.

Finally, we define $q \in \Delta(\Theta \times S)$ to be the initial probability distribution over all possible state and signal combinations.

Thus, we define $\Gamma^G = \langle N, A, u, \Theta, S, \mathbb{T}, q \rangle$. We can also think of the repeated game as a stage game plus the states and transitions or $\Gamma^G = \langle G, \Theta, S, \mathbb{T}, q \rangle$. The utilities of the repeated game Γ^G are exactly the same as those of the stage game G. Specifically, the set of possible utilities is independent of which state Γ^G is in at any particular time or what signals the agents receive. The importance of states and signals is that it allows agents to have correlated strategies. An example of this is given later in this section.

Definition 9 *A game has complete* state information *if every agent always knows which state the game is currently in. That is, for every signal $s_i \in S_i$, there exists a single state $\theta \in \Theta$ such that if agent i receives s_i then θ must be the current state. It is important to note that for each θ there can exist more than one possible s_i.*

We denote a strategy for agent i in a repeated game by $(\sigma_i^t)_{t=0}^{\infty}$, where σ_i^t is the strategy played in the stage game at time t. Similarly, we denote the regret in a repeated game by $(r^t)_{t=0}^{\infty}$, where r^t is the regret at time t.

In a repeated game, agents are no longer necessarily just concerned with their immediate utility. A lower immediate utility might be acceptable if it leads to a greater utility in the future. Likewise, a higher immediate utility might not be acceptable if it comes with a lower utility in the future.

Given a sequence of utilities $u_i^1, u_i^2, \ldots$ for agent i starting at time 1, one possible way of balancing immediate versus future utility is by examining the κ-*discounted* sum of the utilities, given by: [5] [6]

$$v(u_i^1, u_i^2, \ldots) = (1 - \kappa) \sum_{t=1}^{\infty} \kappa^{t-1} u_i^t. \tag{2.20}$$

By "decaying" utilities over time, utilities in the near future are worth more than utilities from futher into the future.

[5] In machine learning, this is known as the *value* of a state (assuming one state), and hence we denote it by v.

[6] The $(1 - \kappa)$ term is needed to normalize the sum, i.e. $(1 - \kappa) \sum_{t=1}^{\infty} \kappa^{t-1} = 1$.

Definition 10 *The* value *of a game, given its starting state θ^0 and the set of stationary strategies σ, is*

$$v_i(\theta^0, \sigma) = (1-\kappa)\sum_{t=1}^{\infty}\sum_{\theta\in\Theta}\sum_{s\in S}\kappa^{t-1}P(\theta^t|\sigma)P(s|\theta^t)u_i(\sigma[s]), \tag{2.21}$$

where $P(s|\theta^t)$ is the probability of the joint signal $s = \{s_1, \ldots, s_N\}$ being sent to the agents given the state θ^t, and $P(\theta^t|\sigma)$ is the probability of the game being in state θ at time t given the stationary strategy σ. As with correlated strategies in stage games, we assume that agents know the probabilities of receiving each possible joint signal in any state.

The value due to the repeated game at time t is equal to the probability of the game being in state θ at time t, times the probability of the signal s being sent to all agents given the state θ^t, times the utility of the resulting joint strategy.

As a result, we must generalize our notion of equilibrium.

Definition 11 *A stationary strategy σ is in equilibrium if and only if for all $i \in N$*

$$v_i(\theta^0, \sigma) \geq v_i(\theta^0, (\sigma_i', \sigma_{-i})) \tag{2.22}$$

for all $\sigma_i' \in \Sigma_i$.

Since we have generalized our notion of equilibrium, we now ask: do G and Γ^G necessarily have the same equilibria? To examine this question, we first introduce some notation.

Definition 12 *Let σ be a stationary strategy. Let $Y_i(\sigma, a_i, \theta, \kappa)$ be the value of Γ^G for agent i, assuming that all agents follow σ except for at $t = 0$, when agent i will play action a_i and all agents will play σ_{-i}.*

This allows us to redefine v_i as follows.

$$v_i(\theta) = \sum_{a_i\in A_i}\sigma_i[\theta](a_i)Y_i(\sigma, a_i, \theta, \kappa), \forall\theta \in \Theta. \tag{2.23}$$

For σ to be an equilibrium, it must be to no agent's advantage to unilaterally change its strategy. Since we are dealing with stationary strategies, an agent cannot change its strategy at a specific time. Instead the agent can only change its strategy

in a specific state (assuming complete knowledge). Under these conditions, we can achieve an equilibrium with the following theorem, which is known as one of the *Folk theorems* [39].[7]

Theorem 1 *Given a repeated game with complete state information and stationary joint strategy σ, if there exists a bounded vector v such that*

$$v_i(\theta) = \max_{a_i \in A_i} Y, \tag{2.24}$$

and Equation 2.23 is met, then σ is an equilibrium of the repeated game with the κ-discounted sum.

The importance of this theorem is that we can show that strategies in a stage game which are not in equilibrium may be in equilibrium in a repeated game. Consider the example in Figure 2.4 based on one by Myerson [39].

		Agent 2	
		cooperate	defect
Agent 1	cooperate	$0.5, 0.5$	$0, 1$
	defect	$1, 0$	$0.25, 0.25$

Figure 2.4: Prisoners' Dilemma

In the Prisoner's Dilemma, the only Nash equilibrium is for both agent to defect. Furthermore, defecting is also the only correlated equilibrium. However, in a repeated game of Prisoners' Dilemma, it is possible to have an equilibrium of both agents cooperating.

Let $\Theta = \{c, d\}$ where the initial state is c and the state d means that at least one agent defected in the previous round. An equilibrium strategy would be $\sigma(\text{cooperate}|c) = 1$ and $\sigma(\text{defect}|d) = 1$. (This is commonly known as the trigger strategy.) To see why, we first find the value of the strategy.

$$v_i(c) = v_i(\text{cooperating}, c) = (1 - \kappa)0.5 + \kappa(v(c)) \tag{2.25}$$
$$v_i(d) = v_i(\text{defecting}, d) = (1 - \kappa)0.25 + \kappa(v(d)) \tag{2.26}$$

[7]The term Folk theorem comes from the fact that these theorems were generally accepted long before they were actually proved. The name is not overly descriptive of the theorems themselves and Myerson instead refers to them as *general feasibility theorems*[39].

If agent i instead chose to defect in state c, its new value would be

$$v_i'(c) = (1-\kappa)1 + \kappa(v(d)). \tag{2.27}$$

Likewise, if agent i instead chose to cooperate in state d, its new value would be

$$v_i'(d) = 0 \tag{2.28}$$

For the strategy to be in equilibrium we need Equation 2.25 greater than or equal to Equation 2.27 and Equation 2.26 greater than or equal to Equation 2.28. The second condition is met unconditionally and the first is met when $\kappa \geq 2/3$. Thus, if agent i cares enough about its future utility, then cooperating can become an equilibrium strategy.

Is this example an exceptional case? If not, we must take extra care when defining exactly what we mean by a solution in repeated games. As Fudenberg and Maskin show, the above example is most likely not exceptional at all [29].

Theorem 2 *For a stage game G, let $\hat{v}_i$ be the minimax utility when considering only pure strategies. Let F denote the set of all possible utilities that could be obtained due to some correlated strategy in G.*

Then for any repeated game, let x be any vector in F such that $F \cap \{z \in \mathbb{R}^N | \hat{v}_i \leq z_i \leq x_i, \forall i \in N\}$ has a nonempty interior relative to $\mathbb{R}^N$. Then there exists some number $0 < \bar{\kappa} < 1$ such that for every κ, such that $\bar{\kappa} \geq \kappa < 1$, there exists a correlated strategy μ such that μ is a subgame-perfect publicly correlated equilibrium of the repeated game, with a discount factor of κ and a value of x_i for each agent.[8]

Although this result deals with correlated strategies, it does suggest that there are many uncorrelated equilibria in repeated games which are not possible in stage games. The advantage, as shown by the above example, is that these equilibria may be much "nicer" (in the sense that both agents receive a higher utility). The disadvantage is the complexity of determining if a given strategy is an equilibrium. Thus, although we consider the more general case to be a valid and interesting area of research, for now we will consider the more simple case of $\kappa = 0$. This will restrict us to equilibria in the repeated game which are also equilibria in the stage game.

[8]Suppose at time t, the repeated game Γ^G is in state θ^t. The repeated game Γ^G in state $\theta^{t'}$ is a subgame if it is possible for Γ^G to go from state θ^t to state $\theta^{t'}$ in some finite number of turns. A strategy is subgame perfect if it is an equilibrium in Γ^G at state θ^t as well as for all possible subgames [39].

2.4 Learning in Repeated Games

As agents play the repeated game they begin to learn which are the best strategies to play in the stage game. In particular, their strategies may change and evolve as they gain more experience. A *learning algorithm* is any algorithm an agent uses to help it play a repeated game.[9]

A solution to a repeated game is some joint strategy where no agent wants to change their strategy. There has been considerable discussion recently over what sorts of solutions agents should try to achieve. The *AI Agenda* argues that a Nash equilibrium may be a poor solution choice since agents do not necessarily have any incentive to try to reach a Nash equilibrium [49]. For example, in the Prisoners' Dilemma, agents would prefer to both cooperate than play the Nash equilibrium.

Thus, the AI agenda instead has agents developing best-response strategies to specific classes of agents. For example, Powers and Shoham have developed an algorithm which can achieve a best-response against agents with limited memory [44]. However, it seems unrealistic that a single agent will have some unique advantage over all other agents. Therefore, we are interested in repeated games between identical agents (i.e. *self-play*). We argue that for self-play, a Nash equilibrium is still a valid solution concept. To show this, we now consider several different possible solution concepts and examine their relative strengths and weaknesses.

1. Social welfare maximization - this requires maximizing the sum of the utilities of all the agents. Since our agents are only self-interested, they might not agree with the social welfare maximizing outcome. For example, suppose that agent i is given a choice between two actions, $a_{i,1}$ and $a_{i,2}$. Action $a_{i,1}$ gives agent i a utility of 1 and all agents a social welfare of 10. Action $a_{i,2}$ gives agent i a utility of 10 and all agents a social welfare of 1. The social welfare maximization solution is for agent i to choose action $a_{i,1}$. However, since agent i is only concerned with its only utility, it will chooose action $a_{i,2}$.

2. Pareto optimal solution - this requires finding a joint strategy for which no agent could change their strategy to improve its utility without hurting the utility of another agent. However, again since our agents are self-interested, they might not agree with this type of solution either.

[9] It should be noted that in the game theory literature, what we refer to as a learning algorithm is called a strategy. To avoid the obvious confusion, we will stay with the term learning algorithm.

3. Nash equilibria - this is the type of solution we will be interested in. Since agents are self-interested and concerned with only their immediate utility, Nash equilibria are the only enforceable outcomes.

   The problem is that finding a Nash equilibrium (or even an ϵ-Nash equilibrium) is PPAD-complete [10, 11]. Hence, we do not expect to be able to create an algorithm which can be used in practice for all repeated games.

4. Maximizing utility - an alternative solution could be for agents to try and maximize their utility. As has already been seen with the Prisoners' Dilemma, converging to a Nash equilibrium does not always provide the highest long term utility for an agent. However, as shown, to have agents both agree to cooperate requires a far more complex model than the one used in this thesis.

   Another example where maximizing utility differs from converging to a Nash equilibrium is when there are multiple Nash equilibria for a game. In this case, it is possible that one Nash equilibrium is Pareto optimal. In this case, it would make sense to try to converge to the first Nash equilibrium. Besides the fact that finding a Pareto optimal Nash equilibrium is NP-complete, for a given Nash equilibrium $\mathcal{N}_1$, there may be multiple Nash equilibria that are Pareto optimal to it [12].

For a repeated game G, if there existed some time t' such that for all $t \geq t'$, the joint strategy σ^t was a Nash equilibrium, i.e. $\sigma^t \in \mathcal{N}_G$, then we could say that the game had obtained a solution at time t'. However, it is rare for a noncooperative multiagent learning algorithm to actually reach a Nash equilibrium at some finite point in time. It is much more common that as time goes on, σ^t gets closer and closer to a Nash equilibrium, even though it never actually reaches that equilibrium, i.e.

$$\forall \omega > 0 \ \exists t, \mathcal{N}_i \in \mathcal{N} \text{ such that } \forall t' \geq t, ||\sigma^{t'} - \mathcal{N}_i|| < \omega. \tag{2.29}$$

In words this means that there exists some Nash equilibrium, $\mathcal{N}_i$, such that for any $\omega > 0$, there exists some finite time t such that all joint strategies starting at time t are a distance of at most ω away from $\mathcal{N}_i$. This leads to the idea *convergence* to a Nash equilibrium, i.e.

$$\lim_{t \to \infty} \sigma^t \in \mathcal{N}. \tag{2.30}$$

This is a standard goal for learning algorithms trying to reach a Nash equilibrium.

However there are a couple of alternative notions of convergence that can be considered. The first is *convergence to the set of Nash equilibria*. This means that

		Agent 2 R	Agent 2 L
Agent 1	T	1,1	0,0
	B	0,0	1,1

Figure 2.5: A simple game

$\lim_{t\to\infty} \sigma^t$ does necessarily exist. For example, for the game in Figure 2.1, we could have the following sequences of joint strategies,

$$\{(.9,0),(.9,0)\},\{(0,.9),(0,.9)\},\{(.99,0),(.99,0)\},\{(0,.99),(0,.99)\},\ldots \quad (2.31)$$

This sequence has no limit. However, it is made up of two separate subsequences, each of which does have a limit. More importantly, each of these limits is a Nash equilibrium. This is an example of convergence to the set of Nash equilibria. Formally, if the sequence of joint strategies $\{\sigma^0,\sigma^1,\ldots\}$ can be partitioned into a (possibly infinite) set of distinct subsequences, $\{\sigma^{\mathbb{S}_1},\sigma^{\mathbb{S}_2},\ldots\}$ such that $\forall i$,

$$\lim_{t\to\infty} \sigma^{\mathbb{S}_i^t} \in \mathcal{N}, \quad (2.32)$$

then the original sequence is said to converge to the set of Nash equilibria. This is also a valid notion for a solution. Note that convergence to a Nash equilibrium is a strictly stronger notion. However, there are other notions of convergence which are not as valid. Both of the above notions relate to *period-by-period behaviours*. An alternative one relates to the *cumulative empirical frequency of play*. The difference is best demonstrated by an example. The game in Figure 2.5 has 3 Nash equilibria, $\{(1,0),(0,1)\}$, $\{(0,1),(0,1)\}$ and $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$.

Suppose Agent 1 and Agent 2 play the following repeated sequence of actions,

$$(T,L),(B,R),\ldots \quad (2.33)$$

Each turn both agents get 0 utility. However, the cumulative empirical frequency of play, or the average number of times each agent played each action, corresponds to the Nash equilibrium $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$. Thus, we can say that the cumulative empirical frequency of play converges to a Nash equilibrium. The problem with this definition is that just because the cumulative empirical frequency of play converges, does not mean that agents get the utility from the corresponding Nash equilibrium.

Which notion of convergence is used is not the only criterion that can be used to judge a learning algorithm. Another is the level of cooperation allowed (or

required) between agents. The whole point of learning an equilibrium (as opposed to finding one through some centralized process) is that agents can remain self-interested and do not have to trust any other agents or third party. This allows us to use our results in a wide variety of situations from interactions on the Internet to economics. These are situations where agents (or people) would often not be willing to give up self-interest or be willing to trust others.

In light of this, cooperation between agents should be kept to a minimum, and if possible, any cooperation should be justified as being in the agents' best interests. The least amount of cooperation that can be obtained is when agents are not even aware of the existence of other agents. This means that agents are not even aware they are playing a game. All agents know is that they are picking an action (or strategy) and each time they do so they receive a utility. Agents know nothing about the connection between the actions they choose and the utility they receive. An algorithm that can work in such a setting is called *radically uncoupled* [22]. Since agents cannot make any assumptions about the other agents they are playing against, a radically uncoupled algorithm can work in any setting. If a learning algorithm is aware of the existence of other agents but agents can still keep their payoff functions private, then the learning algorithm is called *uncoupled*. There are some learning algorithms that are not even uncoupled; these will be talked about later. If we must ever use a non-uncoupled algorithm, we will either justify it as being in the agents' best self-interest or making a substantial difference in the time it takes to converge to a Nash equilibrium.

Finally, we must address what sort of complexity is to be allowed in a learning algorithm. This is a more theoretical concern. Algorithms which are too complex will be either impractical to code or impractical to run. However, in the next section we will discuss a number of possibility and impossibility results from the game theory community. These results often assume that a learning algorithm has *R-recall* and is *stationary* [35]. A learning algorithm has R-recall if the algorithm depends on only the last R turns. In other words, the learning algorithm has only a finite amount of memory. If a learning algorithm has R-recall and does not depend on t (the current time), the algorithm is said to be stationary. If a learning algorithm does not have R-recall then a common alternative is that it must be based only on summary statistics.

Chapter 3

Related Research

In this chapter we examine the related research in the field of multiagent and machine learning. We examine three of the main categories of learning algorithms in repeated games: regret-based learning, fictitious play and gradient ascent learning. Finally, we study the machine learning literature on experts algorithms.

3.1 Regret-Based Learning

A regret-based learning algorithm is any algorithm that is trying to minimize an agent's regret. Sometimes this is done explicitly and sometimes this is an "after the fact" observation. To best understand how regret can be used by a learning algorithm, it helps to consider a more general definition of regret. Consider regret to be the comparison between the utility from a given strategy, σ_i, and the best possible utility that could have been achieved by any strategy in some set of strategies, $\Sigma_{\sigma_i} \in \Sigma_i$. If $\Sigma_{\sigma_i} = \Sigma_i$, then we have our original definition of regret. We can also consider Σ_{σ_i} to be a strict subset of Σ_i. This could be done by using a set of mappings that take σ_i and map it into the desired set. Greenwald and Jafari used Φ, the finite subset of the set of linear maps $\phi : \Sigma_i \rightarrow \Sigma_i$ [31]. In this case, regret is redefined as

$$r_\Phi(\sigma_i, \sigma_{-i}) = \max_{\phi \in \Phi}(u_i(\phi(\sigma_i), \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i})). \tag{3.1}$$

Following Greenwald and Jafari's work, we can now generalize the linear map

$\phi_i : \Sigma_i \rightarrow \Sigma_i$ to $\phi_i : \Sigma_A \rightarrow \Sigma_A$ as[1],

$$\phi_i(\sigma_A)(a_i, a_{-i}) = \sum_{a'_i \in A_i} \sigma_A(a'_i, a_{-i})\phi_i(\delta_{a'_i})(a_i), \tag{3.2}$$

where $\delta_{a'_i}$ is the Dirac function which creates a probability distribution over A_i with all mass concentrated at a'_i .

Definition 13 *The correlated joint strategy, σ_A, is a Φ-equilibrium if and only if $u_i(\sigma_A) \geq u_i(\phi_i(\sigma_A))$ for all agents i and for all $\phi_i \in \Phi_i$.*

Definition 14 *If a learning algorithm is able to minimize Φ_i-regret, i.e.*

$$\lim_{t \rightarrow \infty} r_\Phi(\sigma_i^t, \sigma_{-i}^t) = 0, \tag{3.3}$$

then that algorithm achieves no-Φ_i-regret *and the algorithm is a* no-Φ_i-regret learning algorithm.

Greenwald and Jafari prove the following theorem.

Theorem 3 *Given a game G, if all agents i play via some Φ_i-no-regret learning algorithm, then the joint empirical distribution of play converges to the set of Φ-equilibria, almost surely.*

The importance of this theorem is that, by choosing the right Φ_i, it might be possible to get some sort of convergence. Thus we start by exploring the two most common types of Φ_i-no-regret learning algorithms: *no-external regret* and *no-internal regret* algorithms. It should be noted that both of these types of learning algorithms are often referred to simply as "no-regret learning algorithms". We reserve the term "no-regret learning algorithm" for a specific type of learning algorithm and will, in this case, keep our terminology in line with the game theory literature.

3.1.1 No-External Regret

No-external regret is obtained by letting $\Phi_i = \{\phi_{a_i} | a_i \in A_i\}$ where $\phi_{a_i}(\sigma_i) = \delta_{a_i}$. Hence, no-external regret compares a given strategy against all pure strategies. No-external regret is also known as *Hannan consistency* or *universal consistency* [32].

[1]The value $\sigma_A(a'_i, a_{-i})$ is the probability of the joint action (a'_i, a_{-i}) according to σ_A.

There are a large number of no-external regret algorithms in the literature including work done by Blackwell, Foster and Vohra, Freund and Schapire, Fudenberg and Levine and Hannan [5, 20, 23, 26, 32]. The *Logistic Fictitious Play* algorithm presented later in this chapter, achieves ϵ-no-external regret.

A no-external regret equilibrium can also be thought of as any joint strategy where each agent receives at least as much utility as they would have from a *generalized minimax equilibrium*. A traditional *minimax equilibrium* is defined for a *zero sum game*, which is any two-person game where the utilities for any outcome add to zero. In these games the agents are purely adversarial meaning that one agent's gain is the other agent's loss. An agent's optimal strategy in a zero sum game is then to minimize the other agent's maximum potential utility. These strategies are called *minimax strategies* and the resulting utility is called the *minimax utility*. If both agents are playing minimax strategies then the joint strategy an equilibrium [54]. This sort of equilibrium is called a minimax equilibrium.

Minimax equilibria can also be used to determine the minimum utility which an agent can be guaranteed to receive in any game. For example, to determine the minimum utility agent i can be guaranteed to receive in some game G, we can transform G into a zero sum game G'. In the game G' agent i's utilities remain unchanged however, its opponent's utilities now become the negative of agent i's. An example of this conversion is shown in Figures 3.1 and 3.2. The maximum utility agent i can receive in G' is the minimum guaranteed utility it can receive in G. Furthermore, if agent i plays its minimax strategy from G' in G it will achieve this utility. If all agents play such strategies in G, then the resulting joint strategy is called a generalized minimax equilibrium. Furthermore, any other joint strategy that achieves at least as high a utility as the minimax utility for all agent is also a generalized minimax equilibrium.

		Agent 2	
		L	R
Agent 1	T	$6, 6$	$2, 7$
	B	$7, 2$	$0, 0$

Figure 3.1: The game of Chicken

Continuing the example in Figures 3.1 and 3.2, agent 1's minimax strategy in Figure 3.2 is B. Likewise, agent 2's minimax strategy would be R. In Figure 3.1, the joint strategy $\{B, R\}$ would give both agents a utility of 0. Thus, for Figure 3.1, any joint strategy that gives both agents a utility of at least 0 is a generalized

		Agent 2	
		L	R
Agent 1	T	$6, -6$	$2, -2$
	B	$7, -7$	$0, 0$

Figure 3.2: A zero sum version of the game in Figure 3.1 for agent 1

minimax or a no-external regret equilibrium. In this case, the set of no-external regret equilibria includes all possible strategies. This includes the game's 3 Nash equilibria, $\{(\frac{2}{3}, \frac{1}{3}), (\frac{2}{3}, \frac{1}{3})\}$, $\{(1, 0), (0, 1)\}$ and $\{(0, 1), (1, 0)\}$, as well as all of the correlated equilibria. (See next section for a discussion of this game's correlated equilibria.) This is proof by example that no-external regret equilibria include all of the correlated and Nash equilibria; a formal proof is given by Greenwald and Jafari [31]. Hence, no-external regret is the weakest form of regret.

3.1.2 No-Internal Regret

No-internal regret is obtained by letting $\Phi_i = \{\phi_{a_i, a_i'} | a_i' \neq a_i \in A_i\}$ where

$$(\phi_{a_i, a_i'}(\sigma_i))_{a_i''} = \begin{cases} \sigma_i(a_i'') & \text{if } a_i'' \neq a_i, a_i', \\ 0 & \text{if } a_i'' = a_i, \\ \sigma_i(a_i) + \sigma_i(a_i') & \text{if } a_i'' = a_i'. \end{cases} \tag{3.4}$$

In words, internal regret is the regret agent i feels every time it plays action a_i instead of a_i' for any $a_i' \neq a_i \in A_i$.

Hart and Mas-Colell Algorithm

Foster and Vohra originally defined no-internal-regret with respect to the *on-line decision problem* (ODP) [21]. A classic ODP is to try to minimize the error in predicting a sequence of 0's and 1's [55]. This is done using a prediction scheme c. The sequence is picked by a purely adversarial agent who knows c. In this case, the best case possibility is to have c be a randomized scheme that picks 0 and 1 with equal probability. In this case the average number of incorrect predictions, or the loss, is .5. However, suppose that there are multiple prediction schemes $C = \{c_1, \ldots, c_l\}$, with no restriction placed on the possible methods each scheme can use. Each turn we must decide which scheme to use. Then we can ask how well

our selection of schemes did compared to how well we could have done. If every time we picked scheme c_j we wished we had picked scheme c_i, we would experience regret. Formally, given some sequence of length T, let $L^T(C)$ be the overall loss the prediction schemes achieved. At turn t, there was a probability of $p^t_{c_j}$ of using scheme c_j, which if used would have received a loss of $L^t_{c_j}$. Now if c_j had been used every time instead of c_i, (assuming the sequence remained constant) our loss would have been

$$L^T(C) - (\sum_{t=1}^{T} p^t_{c_j} L^t_{c_j} - \sum_{t=1}^{T} p^t_{c_j} L^t_{c_i}). \tag{3.5}$$

If $\sum_{t=1}^{T} p^t_{c_j} L^t_{c_j} - \sum_{t=1}^{T} p^t_{c_j} L^t_{c_i} > 0$, we would experience regret for having used c_j instead of c_i. Thus the overall regret for scheme c_j is defined as

$$R^T_{c_j}(C) = \sum_{c_i \in C} \max \left[0, \sum_{t=1}^{T} p^t_{c_j} (L^t_{c_j} - L^t_{c_i}) \right], \tag{3.6}$$

and the overall regret for the prediction schemes H is

$$R^T(C) = \sum_{c_j \in C} R^T_{c_j}(C), \tag{3.7}$$

also known as *internal regret*.

Foster and Vohra also presented the first algorithm to achieve no-internal regret [21]. Due to its greater simplicity, we instead present the no-internal regret algorithm by Hart and Mas-Colell (HMC) [33].

HMC

At time t, agent i plays an action, a_i, according to σ^t_i. Agent i then examines all the times action a_i has been played (including at time t). Agent i compares the average utility from these times against the average utility that it would have received if every time it had played a_i, it instead had played some other action $a'_i \in A_i \backslash \{a_i\}$. If agent i would have been better off always playing a'_i, it feels regret. That regret $R^t(a_i, a'_i)$ is the additional utility agent i could have gained. The regret is calculated for all $a'_i \in A_i \backslash \{a_i\}$. The value of $\sigma^{t+1}_i(a'_i)$ is then proportional to that regret.

The version of the HMC algorithm presented here has been adjusted slightly so that agents can play strategies instead of actions.

Given σ^t_{-i}, $\mathbb{D}^t_i(a_{i,j}, a_{i,k})$, defined as

$$\mathbb{D}^t_i(a_i, a'_i) = \sum_{a_{-i} \in A_{-i}} \sigma^t_i(a_i)\sigma^t_{-i}(a_{-i})[u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})], \tag{3.8}$$

is a measure of how much agent i prefers action a'_i to a_i. If we define a^t_i to be an action picked according to σ^t_i, then we can define σ^{t+1}_i as follows:

$$R^t(a_i, a'_i) = \max\{0, \mathbb{D}^t_i(a_i, a'_i)\} \tag{3.9}$$

$$\sigma^{t+1}_i(a_i) = \begin{cases} \frac{1}{\mu} R^t_i(a^t_i, a_i), & \text{if } a_i \neq a^t_i \\ 1 - \sum_{a'_i \neq a^t_i} \sigma^{t+1}_i(a'_i) & \text{otherwise.} \end{cases} \tag{3.10}$$

The parameter μ is a constant which can have any value greater than $|A_i - 1|$. We let $\mu = |A_i|$.

The main result for HMC is

Theorem 4 *If every agent plays according to HMC, then the empirical distribution of play a^t_i converges almost surely as $t \to \infty$ to the set of correlated equilibria of G.*

In fact, Greenwald and Jafari show that achieving no-internal regret always guarantees convergence to the set of correlated equilibria [31].

We return to our example in Figure 3.1 to find its set of correlated equilibria. We start by considering agent 1. Suppose that agent 2 has decided to use strategy σ_2. (We place no restrictions on σ_2. In fact, the following proof must hold for all σ_2 which agent 2 has a positive probability of playing.) Agent 1 then decides to respond with the correlated strategy σ_{A_1}. For σ_{A_1} to result in no-internal regret, every time agent 1 plays action T, it must yield at least as high a utility as playing action B would have. Specifically, the expected utility agent 1 gets from playing action T is,

$$\sigma_{A_1}(T)(6\sigma_2(L) + 2\sigma_2(R)). \tag{3.11}$$

However, if every time it played T, it instead played B, it would have received an expected utility of

$$\sigma_{A_1}(T)(7\sigma_2(L) + 0\sigma_2(R)). \tag{3.12}$$

Therefore, we have

$$\sigma_{A_1}(T)(6\sigma_2(L) + 2\sigma_2(R)) \geq \sigma_{A_1}(T)(7\sigma_2(L) + 0\sigma_2(R)), \tag{3.13}$$

or

$$\sigma_{A_1}(T)((-1\sigma_2(L) + 2\sigma_2(R)) \geq 0, \tag{3.14}$$

or

$$-1\sigma_A(TL) + 2\sigma_A(TR) \geq 0. \tag{3.15}$$

Equation 3.15 gives us our first constraint.

To obtain our second constraint, we consider the expected utility agent 1 gets from playing B,

$$\sigma_{A_1}(B)(7\sigma_2(L) + 0\sigma_2(R)). \tag{3.16}$$

This must be greater than the utility that agent 1 would have expected if it played T every time instead of B, which is

$$\sigma_{A_1}(B)(6\sigma_2(L) + 2\sigma_2(R)). \tag{3.17}$$

This results in our second constraint of,

$$1\sigma_A(BL) - 2\sigma_A(BR) \geq 0. \tag{3.18}$$

By symmetry of the game, we also obtain the following two constraints for agent 2

$$-1\sigma_A(LT) - 2\sigma_A(LB) > 0, \tag{3.19}$$

and

$$1\sigma_A(RT) - 2\sigma_A(RB) > 0. \tag{3.20}$$

Equations 3.15, 3.18, 3.19 and 3.20 are the constraints which define the set of correlated equilibria. It can be checked that all of the Nash equilibria meet these constraints. However, there are also other correlated strategies which meet these constraints; one example is in Figure 3.3. The importance of this example is that both agents achieve a higher utility than from any of the Nash equilibria.

	L	R
T	1/3	1/3
B	1/3	0

Figure 3.3: An example of a correlated equilibrium.

3.1.3 No-Regret

Possibility and Impossibility Results

Greenwald and Jafari show that out of the class of Φ-no regret learning, no-internal regret gives the tightest convergence guarantee. In other words, no Φ-no regret learning algorithm can achieve convergence to any subset of the set of correlated equilibria. Thus, if we wish to achieve convergence to the set of Nash equilibria using no-regret learning, we will have to somehow extend Φ.

A further impossibility result is given by Hart and Mas-Colell who showed that no deterministic radically uncoupled algorithm could guarantee convergence to the set of Nash equilibria [34]. Furthermore, Hart and Mas-Colell also proved the following [35]:

Theorem 5 *For every small enough $\epsilon > 0$, there are no uncoupled, finite recall, stationary learning algorithms that guarantee, in every game, the almost sure convergence of the behaviour probabilities to ϵ-Nash equilibria of the stage game.*[2]

However, we are able to obtain a positive result if we weaken R-recall to R-memory.

Definition 15 *A learning algorithm has R-memory if it requires at most $|A_i|^R$ states.*

Theorem 6 *For every M and $\epsilon > 0$ there exists an integer R and an uncoupled, R-memory stationary learning algorithm that guarantees, in every game with payoffs bounded by M, the almost sure convergence of the behaviour probabilities to the set of ϵ-Nash equilibria [35].*

These results give a sense of what can and cannot be achieved.

Regret Testing

The idea of using randomized sampling and ϵ-Nash equilibria to achieve convergence in two agent games was used in the algorithm *Annealed Regret Testing* (ART) by Foster and Young [22]. ART is based on the learning algorithm *Regret Testing*, also

[2] If an event happens "almost surely" then it happens with probability 1.

by Foster and Young [22]. Both ART and Regret Testing are radically uncoupled algorithms (so agents do not have to even know they are playing a game). Since these are radically uncoupled algorithms, agents are required to play actions each turn instead of strategies.

The goal is that for any $\epsilon > 0$, any two agents in a repeated game can use Regret Testing to eventually guarantee that their joint strategy is an ϵ-Nash equilibrium with a probability of $1 - \epsilon$. To do this, Regret Testing uses a grid to discretize the strategy space of each agent; the smaller ϵ is, the finer the grid is. Agents each start off by picking a strategy somewhere on their grid. Agents will each play with their starting strategy for a period of some length of time. After this period is up, each agent estimates its regret by comparing how it did when playing with its strategy versus when it played randomly. If the regret is too high, a new strategy is chosen. Otherwise, the old strategy is kept. Agents again play using their new strategies for a fixed period of time and the process repeats.

Regret Testing:

Agent i has a set of possible strategies, $\Sigma_i^h \subseteq \Sigma_i$ where Σ_i^h contains every strategy in each probability that can be expressed as a multiple of $1/h$. The larger h is, the more closely any strategy can be approximated by a strategy in Σ_i^h. Agent i's initial strategy σ_i^h is picked uniformly at random from Σ_i^h.

The repeated game is divided up into periods of length $\mathcal{T}$.

1. Each turn, i plays according to Σ_i^h, except when, with a probability of Λ, i plays an action at random.

2. At the end of each period, i calculates $\tilde{u}$, its average utility over the last period when it did not play a random action. Agent i also calculates $\tilde{u}_{a_i}$ for all $a_i \in A_i$, the average utility over the last period from when it played an action at random and chose action a_i.

3. If $\tilde{u}_{a_i} - \tilde{u} > \rho_i$ for any a_i and some ρ_i, agent i picks a new strategy at random from Σ_i^h, with each strategy being picked with a probability of at least γ_i. Otherwise, agent i keeps the same strategy.

Regret Testing basically reduces the repeated game to a *Markov chain*.

Definition 16 *A process $X = X_1, X_2, \ldots$ over the set of states $S = \{S_1, \ldots, S_n\}$ is a Markov Chain if*

$$P(X_n = s_i | X_{n-1}) \tag{3.21}$$

for all $n \geq 1$. In other words, the transition from X_{n-1} to X_n is determined solely by X_{n-1}.

In Regret Testing, each possible strategy is a different state. Some of these states are ϵ-Nash equilibria and some are not. Since the strategy space is discretized, there are only a finite number of states and they can be exhaustively searched, a random search will guarantee that this happens. By using Regret Testing, agents can estimate if their current state is indeed an ϵ-Nash equilibrium. Since agents are only playing a certain state for a finite period of time, the empirical frequency of play during that period is only an approximation of the actual strategies agents were using. Thus, there is some error in the agent's estimations. However, with a fine enough grid and enough testing, agents can be reasonably sure of knowing whether or not they have reached an ϵ-Nash equilibrium. Although agents can leave an ϵ-Nash equilibrium by mistake, they will spend "enough" time there.

Formally, Foster and Young prove the following theorem.

Theorem 7 *Let G be a two-agent game and let $\epsilon > 0$. If both agents use regret with the parameters listed below, then at all sufficiently large times t their joint strategy at t will be an ϵ-Nash equilibrium with probability at least $1 - \epsilon$.*

$$\rho_i \leq \begin{cases} d^2(G)/48 & \text{if } d(G) > 0 \\ \epsilon^2/48 & \text{otherwise,} \end{cases} \tag{3.22}$$

$$\Lambda_i \leq \tau/16, \tag{3.23}$$

$$h_i \geq 8\sqrt{|A_i|}/\tau, \tag{3.24}$$

$$\gamma_i \leq 1/|P_i(h_i)|, \tag{3.25}$$

$$\mathcal{T} \geq (10^3 m^2/\Lambda\tau^2)\ln(10^5 m/\epsilon^2\Lambda^7), \tag{3.26}$$

where $|P_i(h_i)|$ is the number of strategies that can created for agent i given h_i, $\tau = \min\{\tau_1, \tau_2\}$, $\Lambda = \min\{\Lambda_1, \Lambda_2\}$, $\gamma = \min\{\gamma_1, \gamma_2\}$ and $m = \max\{m_1, m_2\}$.

The above conditions can be slightly overwhelming and the justifications for them are left to Foster and Young's paper; however, there are a few simple things that these conditions imply [22]. First, as ϵ decreases, the probability of finding an ϵ-Nash equilibrium increases. However, this is balanced by s strictly monotonically increasing as ϵ decreases. The conditions also show why letting $\epsilon = 0$ will not work.

Annealed Regret Testing

Instead of setting ϵ to 0, it might be possible to slowly decrease ϵ and the associated values so that we get convergence to the set of Nash equilibria. This is what Foster and Young's *Annealed Regret Testing* (ART) does [22].

It turns out that any positive sequence $\epsilon_1 > \epsilon_2 > \epsilon_3 > \ldots$ decreasing towards zero will work. However, since Foster and Young are after a radically uncoupled algorithm, it is impossible to know for sure when to move to the next ϵ. Instead, ART uses a probabilistic rule for moving to the next ϵ. At the beginning of each period, each agent has a probability of moving from ϵ_k to ϵ_{k+1} of

$$p_k \equiv \frac{\epsilon_{k+1}^2}{2k^2 T(\epsilon_{k+1})}, \tag{3.27}$$

where $T(\epsilon_{k+1})$ is defined as follows.

Definition 17 *Let $P_G(\epsilon)$ be the finite-state Markov process determined by G and the parameters in Equations 3.22 and 3.26. Let $E_G(\epsilon)$ be the finite subset of states that induce an ϵ-equilibrium of G.*

Definition 18 *Let P be a finite Markov process. Then P is acyclic if there is no chance of there ever being a cycle in P. This means that P can only run for a finite period of time.*

Definition 19 *Let P be an acyclic, finite Markov process and $\mathcal{A}$ a subset of P's states. For each $\epsilon > 0$, let $T(P, \mathcal{A}, \epsilon)$ be the first time (if any) such that, for all $t \geq T(P, \mathcal{A}, \epsilon)$ and all possible initial states, the probability of the process being in $\mathcal{A}$ at time t is at least $1 - \epsilon$.*

Foster and Young prove the following about $T(P, \mathcal{A}, \epsilon)$:

Lemma 1 *For any $\epsilon > 0$, there exists $T(\epsilon)$ such that $T(\epsilon) \geq T(P_G(\epsilon), E_G(\epsilon), \epsilon)$ for all G such that $d(G) \notin (0, \epsilon)$.*

Theorem 8 *Fix an action space $A = A_1 \times A_2$. ART has the property that, for every game G on A, the joint strategy converges in probability to the set of Nash equilibria of G.*

Experimental Regret Testing

The drawback of Foster and Young's algorithm is that it only works for two agents. Germano and Lugosi created two algorithms, ERT and ALERT, to deal with this drawback [30].

Like Regret Testing, ERT does not guarantee convergence. Instead, after a certain period of time ERT can guarantee an ϵ-Nash equilibrium with a probability $1-\epsilon$. (Of course, once the agents find an ϵ-Nash equilibrium, there is a chance of them leaving it.) ERT corresponds to Regret Testing when $\Lambda = 0$ and where agents still occasionally experiment with new strategies even when they have low regret. Although ERT has agents playing actions, if we let $T = \infty$, then we can convert ERT so that agents play strategies.

Experimental Regret Testing (ERT):

Parameters - $(\mathcal{T} \in \mathbb{N}, \rho \in \mathbb{R}_{++}, \zeta \in (0,1))$

1. $t = 0$, each agent chooses $\sigma_i^0 \in \Sigma_i$ uniformly at random.
2. Loop:
   (a) Each agent plays according to σ_i^t for a period of $\mathcal{T}$ rounds
   (b) Each agent calculates its maximum average regret over the period,

$$r_i^t = \max_{a_i \in A_i} \frac{1}{\mathcal{T}} \sum_{\tau=t}^{t+\mathcal{T}-1} (u_i(a_i, a_{-i}^\tau) - u_i(a^\tau)). \tag{3.28}$$

   (c) If $r_i^t < \rho$ then with a probability of $1-\zeta$, $\sigma_i^{t+\mathcal{T}} = \sigma_i^t$. Otherwise, $\sigma_i^{t+\mathcal{T}}$ is selected uniformly at random from Σ_i. (I.e. if $r_i^t < \rho$ then with a probability of ζ agent i will update its strategy, and if $r_i^t \geq \rho$, then agent i will always update its strategy.)
   (d) Set $t = t + \mathcal{T}$ and repeat the loop.

The basic idea of ERT is that if at some point there are $J < N$ agents who have regret less than ρ, there is a positive probability of there being $J-1$ agents having regret less than ρ at the next turn. Since this process repeats indefinitely, at some point in the future all agents will have regret greater than ρ. At this point they will choose a new joint strategy at random from Σ and there is a positive probability of the new joint strategy being an ϵ-Nash equilibrium [30]. Once the agents find an ϵ-Nash equilibrium, the chances of leaving are low.

ERT works for all *generic games*. The exact definition of generic games is unnecessary for this thesis and complex enough that we refer the interested reader to the relevant related work [30, 52]. Instead, we present only the relevant aspect of generic games for this thesis.

Definition 20 *Let $G = \langle N, A, u\rangle$ be a stage game. Then $G' = \langle N, A', u'\rangle$ is a pure subgame of G if $A' \subset A$, and u' is induced by u, that is for $a \in A'$, $u'(a) = u(a)$.*

As well, $G_{\sigma_J} = \langle N', A'', u''\rangle$ is an induced subgame of G if

1. *$N = N' \cup J$ and*
2. *$u'' = u(\sigma_J)$.*

Then G is generic if it, and every possible pure subgame and induced subgame of it, have only a finite number of Nash equilibria.

Germano and Lugosi show that the vast majority of games are generic [30]. As well non-generic games tend to be degenerate.[3]

The main result for ERT is:

Theorem 9 *Let G be a generic N-agent normal form game. There exists a positive number ϵ_0 such that for all $\epsilon < \epsilon_0$ the following holds: there exists positive constants $c_1, \ldots, c_4$ such that if ERT is used by all agents with parameters*

$$\rho \in (\epsilon, \epsilon + \epsilon^{c_1}), \tag{3.29}$$

$$\zeta \leq c_2 \epsilon^{c_3}, \tag{3.30}$$

$$\mathcal{T} \geq -\frac{1}{2(\rho - \epsilon)^2} \log(c_4 \epsilon_3^c), \tag{3.31}$$

$$\tag{3.32}$$

then for all $M \geq \log(\epsilon/2)/\log(1 - \zeta^N)$,

$$P_M(\mathcal{N}_\epsilon^c) = P(\sigma^{M\mathcal{T}} \notin \mathcal{N}_\epsilon) \leq \epsilon. \tag{3.33}$$

In words this means that at the end of $M \cdot \mathcal{T}$ iterations, the probability of not being at an ϵ-Nash equilibrium is at most ϵ.

[3] An example of a degenerate game would be a 2x2 game with both agents receiving a utility of 1 no matter which action they choose.

Annealed Localized Experimental Regret Testing

Germano and Lugosi were able to take their initial algorithm and convert it into one able to achieve convergence. Their new algorithm is called *Annealed Localized Experimental Regret Testing* (ALERT).

Like ART, the basic idea of ALERT is to slowly anneal the value of ϵ and repeat ERT for each value of ϵ. Any sequence of ϵ_l for $l = 1, 2, \ldots$ such that $\sum_{l=1}^{\infty} \epsilon_l < \infty$ will work; however, Germano and Lugosi choose to use $e_l = 2^{-l}$. The set of all periods of play for a particular ϵ_l is called a *regime*. The set of all regimes is indexed by l. The number of periods in the l^{th} regime is given by

$$M_l \equiv 2 \left\lceil \frac{\log \frac{2}{\epsilon_l}}{\log \frac{1}{1-\zeta_l}} \right\rceil . \tag{3.34}$$

(T, ρ, ζ) must be generalized to depend on l, and so the following values are used

$$\mathcal{T}_l = \left\lceil -\frac{1}{2\epsilon_l^{2l}} \log(\epsilon_l^l) \right\rceil , \tag{3.35}$$

$$\rho_l = \epsilon_l + \epsilon_l^l, \tag{3.36}$$

$$\zeta_l = \epsilon_l^l. \tag{3.37}$$

$\sigma_i^{[l]}$ is σ_i at the beginning of the l^{th} regime. $D_\infty^i(\sigma_i, \epsilon)$ is the L_∞-ball of radius ϵ centered around σ_i.[4]

ALERT:

1. Each agent chooses $\sigma_i^0 \in \Sigma_i$ uniformly at random.
2. Loop by $l = 1, 2, \ldots$ with parameters $(\mathcal{T}_l, \rho_l, \zeta)$ (each iteration of l is called a regime)
   (a) Loop for M_l periods
      i. Each agent plays an action according to σ_i^t for $\mathcal{T}_l$ turns.
      ii. Each agent calculates their regret according to Equation 3.28.
      iii. Agents update their strategies according to
         - if $r_i^t \leq \epsilon_l^{2/3}$ then select $\sigma_i^{t+\mathcal{T}_l} \in \Sigma_i$ uniformly at random

[4] A L_∞-ball can be thought of as a hyper-cube.

- else if $\rho_l \leq r_i^t \leq \epsilon_l^{2/3}$
  - if at some time $t' < t$ during the current regime, $\sigma_i^{t'+\mathcal{T}_l}$ had been selected uniformly at random from Σ_l, then select $\sigma_i^{t+\mathcal{T}_l}$ uniformly at random from Σ_l
  - otherwise, select $\sigma_i^{t+\mathcal{T}_l} \in D_\infty^i(\sigma_i^{[l]}, \sqrt{\epsilon_l})$
- otherwise
  - with probability $1 - \gamma_l$, $\sigma_i^{t+T_l} = \sigma_i^t$
  - otherwise select $\sigma_i^{t+T_l} \in D_\infty^i(\sigma_i^{[l]}, \sqrt{\epsilon_l})$

The main result for ALERT is the following theorm.

Theorem 10 *Let G be a generic N-agent game and $\{\epsilon_l\}_{l=1}^\infty$ be defined $e_l = 2^{-l}$. If each agent plays according to ALERT and using the parameters in Equations 3.34 through 3.37, then*

$$\lim_{r\to\infty} \sigma^t \in \mathcal{N} \tag{3.38}$$

almost surely.[5]

In words this means that the limit of the sequence of the joint strategies is a Nash equilibrium.

Note that as presented, ALERT is an uncoupled algorithm. However, Germano and Lugosi also present a variant that is radically uncoupled.

The one drawback of ALERT is its rate of convergence. Since ALERT is uncoupled, the rate of convergence is independent of the game. This can be seen in the game parameters, where $\mathcal{T}_l$ and M_l are both dependent on only ϵ_l and ζ_l. The downside is that for just about any game of interest ALERT's rate of convergence is impractical.

3.2 Fictitious Play Algorithms

Fictitious play (FP) is probably the oldest form of learning for repeated games [9]. The idea is that an agent assumes it is playing against opponents with unknown but static or unchanging strategies. Each round gives the agent a better idea of

[5]Germano and Lugosi claim that ALERT achieves convergence to a single Nash equilibrium. However, it is not clear if they actually mean convergence to the set of Nash equilibria.

what its opponents' strategies actually are. By playing a best response to the empirical distribution of its opponents' play, the agent will eventually arrive at a best response for that opponent.

We first present the original definition of fictitious play for two agents, which has each agent playing actions (instead of strategies).

Definition 21 *Consider a game with two agents. Each agent assumes the other agent is playing a fixed but unknown strategy and attempts to learn it during the repeated game. To do this, each agent uses a weight function k_i^t given by*

$$k_i^t(a_{-i}) = k_i^{t-1}(a_{-i}) + \begin{cases} 1 & \text{if } a_{-i}^{t-1} = a_{-i} \\ 0 & \text{otherwise} \end{cases} \tag{3.39}$$

with $k_i^0(a_{-i}) \in \mathbb{R}_+$. Note the initial weights do not have to be equal.

Agent i then calculates a probability $\Theta_i^t(a_{-i})$ of its opponent playing the joint action a_{-i} at time t (assuming a static strategy) where

$$\Xi_i^t(a_{-i}) = \frac{k_i^t(a_{-i})}{\sum_{a'_{-i} \in A_{-i}} k_i^t(a'_{-i})}. \tag{3.40}$$

Fictitious play is any rule system that chooses σ_i^t such that $\sigma_i^t \in BR_i(\Xi_i^t)$.

Theorem 11 *Under fictitious play, if the empirical distributions of play converges, then those distributions correspond to a Nash equilibrium [27].*

Although a complete characterization of the types of games in which fictitious play can achieve convergence (in the empirical distribution of play sense) is not known, the set includes at least all 2x2 generic games, zero-sum games and those that can be solved by iterated strict dominance [45, 38, 40]. On the other hand, fictitious play is known to not converge in Shapley's game and 3-player Matching Pennies, among others [48, 36]. In both of these games, fictitious play will most likely result in exponential cycling. However, fictitious play can achieve convergence to the set of correlated equilibria in these games.

We are now ready to expand the definition of fictitious play to allow agents to play strategies (again for two players).

Definition 22 *Given the joint strategy σ_{-i}, fictitious play is any rule system that selects a strategy, σ_i, such that $\sigma_i \in BR_i(\sigma_{-i})$.*

3.2.1 Logistic Fictitious Play

Stochastic fictitious play (SFP) is an attempt to allow fictitious play to achieve convergence in the behavioural sense and not just the empirical distribution of play [25]. To understand SFP, consider fictitious play with agents playing actions. At time t, agent i has a historical record of play to help it decide which action to play, again assuming σ_{-i} is static. However, at any finite point, agent i's historical record may not perfectly describe agent i's opponents' actual joint strategy. Hence, there is always a chance that FP will pick the incorrect action to play. Instead, a SFP algorithm will try to estimate the probability that each action is the best action to play. This distribution, $\overline{BR}_i(\Xi^t_{-i})$, gives the probability by which each action is played. Since agent i can have a mixed strategy, convergence in behaviour is now possible.

Logistic fictitious play (LFP) is one specific example of SFP where $\overline{BR}_i(a_i)$ is defined as

$$\overline{BR}_i(\Xi^t_{-i})[a_i] \equiv \frac{e^{(1/\lambda)u_i(a_i,\Xi^t_{-i})}}{\sum_{a'_i \in A_i} e^{(1/\lambda)u_i(a'_i,\Xi^t_{-i})}}, \tag{3.41}$$

where λ is a smoothness parameter [25]. (As λ approaches 0, LFP behaves more and more like basic FP.)

Theorem 12 *For every game G and $\epsilon > 0$, there exists a λ such that LFP is ϵ-universally consistent [28].*

Thus, the best that LFP can be guaranteed is ϵ-no-external regret.

If agents play strategies, then given σ_{-i}, LFP is defined as

$$\sigma_i(a_i) = \frac{e^{(1/\lambda)u_i(a_i,\sigma_{-i})}}{\sum_{a'_i \in A_i} e^{(1/\lambda)u_i(a'_i,\sigma_{-i})}}, \tag{3.42}$$

for all $a_i \in A_i$.

To generalize all of these types of fictitious play, we assume that agents play uncorrelated strategies.

3.3 Infinitesimal Gradient Ascent Algorithms

Suppose that agent i is interested in maximizing its utility assuming that its opponents' strategies are fixed. Agent i could begin by calculating its utility as

$$u_{\sigma_{-i}}(\sigma_i) = \sum_{(a_i,a_{-i}) \in A} u_i(a_i, a_{-i})\sigma_i(a_i)\sigma_{-i}(a_{-i}). \tag{3.43}$$

Equation 3.43 could be written as a function of $m-1$ variables as

$$\begin{aligned}
&u_{\sigma_{-i}}(\sigma_i(a_1),\ldots,\sigma_i(a_{m-1})) \\
&\quad= \sum_{j=1}^{m-1}\sigma_i(a_j)\sum_{a_{-i}\in A_{-i}}\sigma_{-i}(a_{-i})u_i(a_j,a_{-i}) \\
&\quad\quad+\left(1-\sum_{j=1}^{m-1}\sigma_i(a_j)\right)\sum_{a_{-i}\in A_{-i}}\sigma_{-i}(a_{-i})u_i(a_m,a_{-i}), \\
&\quad= \sum_{j=1}^{m-1}\sigma_i(a_j)\left[\sum_{a_{-i}\in A_{-i}}(\sigma_{-i}(a_{-i})u_i(a_j,a_{-i})-\sigma_{-i}(a_{-i})u_i(a_m,a_{-i}))\right]+1. \quad (3.44)
\end{aligned}$$

Taking the partial derivative of Equation 3.44 with respect to $\sigma_i(a_j)$ for $j < m$ gives

$$\frac{\partial u_{\sigma_{-i}}}{\partial\sigma_i(a_j)} = \sum_{a_{-i}\in A_{-i}}(\sigma_{-i}(a_{-i})u_i(a_j,a_{-i})-\sigma_{-i}(a_{-i})u_i(a_m,a_{-i})) \quad (3.45)$$

Therefore, u_i is differentiable with respect to all $\sigma_i(a_j)$. The vector

$$<\frac{\partial u_{\sigma_{-i}}}{\partial\sigma_i(a_1)},\ldots,\frac{\partial u_{\sigma_{-i}}}{\partial\sigma_i(a_{m-1})}> \quad (3.46)$$

is called the *gradient* of u_i [51]. One of the important properties of the gradient is that it gives the direction for the maximum change in u_i. Thus by moving its strategy along the gradient, agent i would be able to maximize u_i.

The problem with having multiple agents all simultaneously using such an approach is that one agent's change in strategy would affect the gradients for all other agents. Hence, an approach would have to allow agents to still reach an equilibrium. *Infinitesimal Gradient Ascent* (IGA) is such an approach that works for two agents each having an action space of size 2 [50]. Both agents update their strategies according to the rule,

$$\sigma_i^{t+1}(a_{i,0}) = \sigma_i^t(a_{i,0}) + \eta\frac{\partial U_i(\sigma_i^t,\sigma_{-i}^t)}{\partial\sigma_i(a_{i,0})}, \quad (3.47)$$

where ν is some step size.

Theorem 13 *If both agents follow IGA, where $\nu \to 0$, 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.*

3.3.1 WoLF

A well known IGA algorithm is the *Win or Learn Fast* (WoLF) algorithm by Bowling and Veloso [8]. WoLF uses a variable learning rate to achieve convergence in cases where IGA can not. The specific version of WoLF we use is *WoLF-Policy Hill-Climbing* (WoLF-PHC). It has been slightly modified for playing stage games and so that agents can play strategies. Note that WoLF assumes the opponents' strategies from pervious iterations are observable.

1. Let $t = 0$, $\delta_w, \delta_l \in (0, 1]$ and $\sigma_i^0 = \frac{1}{|A_i|}$.
2. Repeat
   (a) Play σ_i^t and observe u_i^t. Let $t = t + 1$.
   (b) Update the estimate of the average strategy

$$\forall a_i \in A_i \; \bar{\sigma}_i^t(a_i) = \bar{\sigma}_i^{t-1}(a_i) + \frac{1}{t}(\sigma_i^{t-1}(a_i) - \bar{\sigma}_i^{t-1}(a_i)) \tag{3.48}$$

   (c) Update the strategy

$$\sigma_i^t(a_i) = \sigma_i^{t-1}(a_i) + \Delta_{a_i} \tag{3.49}$$

   where

$$\Delta_{a_i} = \begin{cases} -\delta_{a_i} & \text{if } a_i \neq \arg\max_{a_i' \in A_i} u_i(a_i, \sigma_{-i}^t) \\ \sum_{a_i' \neq a_i} \delta_{a_i'} & \text{otherwise,} \end{cases} \tag{3.50}$$

$$\delta_{a_i} = \min\left(\sigma_i^t(a_i), \frac{\delta}{|A_i| - 1}\right) \tag{3.51}$$

$$\delta = \begin{cases} \delta_w & \text{if } \sum_{a_i' \in A_i} \sigma_i^t(a_i') u_i(a_i', \sigma_{-i}^t) > \sum_{a_i' \in A_i} \bar{\sigma}_i^t(a_i') u_i(a_i', \sigma_{-i}^t) \\ \delta_l & \text{otherwise.} \end{cases} \tag{3.52}$$

Theorem 14 *In a two-person, two-action, repeated general-sum game, if both agents follow the WoLF-PHC (with $\delta_l > \delta_w$), then their strategies will converge to a Nash equilibrium [8].*

WoLF has also been shown to converge in other games such as 3-Player Matching Pennies. However, WoLF does not converge for Shapley's Game, among others. More importantly, convergence in some games may depend on the specific δ_w and δ_l values used.

3.4 Experts Algorithms

A common problem for learning is that there is no one algorithm that is strictly better than all the others. A learning algorithm that does very well for one type of game can often fail to achieve convergence in another type of game. For example, LFP and WoLF achieve convergence in different sorts of games. There could be considerable benefit from combining these two algorithms into a new learning algorithm which can achieve convergence for any game which either LFP or WoLF by itself could achieve convergence for. The difficulty is finding some way of combining algorithms so that each algorithm is used optimally. For example, a new algorithm based on LFP and WoLF that used LFP when WoLF should have been used is not a useful algorithm.

This is a common problem in machine learning. In machine learning terms, LFP and WoLF would be considered *experts*. An algorithm that makes use of experts and optimizes when to use each expert is called an *experts algorithm* [1]. Experts algorithms are also known as *ensemble algorithms* [16].

Let $E = \{e_0, \ldots, e_r\}$ be a set of $r + 1$ experts. At time t an agent will want to know which expert to consult. The agent will first consult the experts algorithm which will provide a policy p^t, which is a probability vector for consulting each agent. It is the job of the experts algorithm to try to optimize p^t according to some metric.

3.4.1 Hedge

The first experts algorithm, Hedge, was created by Auer et al [1]. Here we present the version of Hedge given by Freund and Schapire [23]. The basic idea of Hedge is at time t to consult expert e_i with a probability proportional to some "weight", $w^t_{e_i}$. Initially, these weights are chosen at random.

Hedge starts by assigning a "weight", $w^1_{e_i}$, to each expert e_i and then consults an expert with a probability equal to that expert's weight proportional to all of the weights. At time t, every expert is asked for a suggested strategy even though only one of those strategies is used in the end. Each expert must then calculate the regret its suggested strategy would have obtained had that strategy been used. This regret is denoted by $r^t_{e_i}$. At time $t + 1$, each expert's weight is decayed by a factor, $\psi < 1$, raised to $r^t_{e_i}$. Thus, as time proceeds, experts who suggest strategies that would have incurred a high regret are consulted less and less often.

Hedge:

Provided $\psi \in (0,1)$ and an initial weight vector $w_{e_i}^1 \in [0,1]^{|E|}$ such that $\sum_{e_i \in E_i} w_{e_i}^1 = 1$.

At time $t > 1$:

1. Calculate p_i^t as:
$$p_i^t(e_i) = \frac{w_i^t(e_i)}{\sum_{e_i' \in E_i} w_i^t(e_i')}.$$

2. Use suggested strategy from expert selected according to p_i^t. However, all experts must still calculate a suggested strategy. Each expert then calculated $r_{e_i}^t$ which is the regret that would have been obtained if expert e_i's suggested strategy had been used.

3. Update the weights according to
$$w_i^{t+1}(e_i) = w_i^t(e_i)\psi^{r_{e_i}^t}.$$

3.4.2 Strategic Experts Algorithm

The second experts algorithm we examine is by Pucci de Farias and Megiddo [15]. Their algorithm, *Strategic Experts Algorithm* (SEA), differs in two respects. First, once an expert is picked, it is used for a number of consecutive rounds, instead of just one. Secondly, an expert is judged only by how it actually does, as opposed to how it could have done when it was not being consulted.

The main difference of SEA compared to Hedge is that a new expert to consult is not chosen every turn. Instead, when expert e_i is chosen, it is then consulted for a period of N_{e_i} turns. Initially $N_{e_i} = 1$ but every time expert e_i is consulted N_{e_i} is increased by 1. This means that the more often expert e_i is chosen, the longer it will be consulted for. The other difference between Hedge and SEA is that SEA measures the performance of experts is based on measuring utility, not regret. Specifically, M_{e_i} is used to denote the "average" utility that expert e_i's strategies have obtained. When choosing a new expert to consult at time t, with a probability of $1/t$, SEA chooses the expert with the highest M_{e_i} value. Otherwise an expert is chosen at random. Thus, as time goes on, the expert with the highest average utility is consulted more and more often.

SEA:

1. Set $M_{e_i} = 0$ and $N_{e_i} = 1$. Set $t = 1$.

2. With probability $1/t$ perform an *exploration phase*, namely, choose an expert e_i uniformly at random; otherwise, perform an *exploitation phase*, namely, choose an expert e_i uniformly at random from those experts with maximum M_{e_i}.

3. Set $N_{e_i} = N_{e_i} + 1$. Follow e_i's instructions for the next $N_{e_i} - 1$ stages. Denote by $\tilde{R}$ the average payoff accumulated during those N_{e_i} stages, and set

$$M_{e_i} = M_{e_i} + \frac{2}{N_{e_i} + 1}(\tilde{R} - M_{e_i}). \tag{3.53}$$

4. Set $t = t + 1$ and repeat.

To examine SEA, we consider $\tilde{u}^t_{e_i}$, the average utility obtained by the expert e_i up until time t and $\tilde{u}^t_{SEA}$, the average utility obtained by SEA overall up until time t. This gives us the following result.

Theorem 15

$$P(\lim_{t\to\infty} \inf \tilde{u}^t_{SEA} \geq \max_{e_i} \lim_{t\to\infty} \inf \tilde{u}^t_{e_i}) = 1. \tag{3.54}$$

In words this means that, in the limit, the average utility achieved by SEA is at least as much as the maximum average utility achieved by any of the individual experts. This is useful if, for example, one of the experts is playing a minimax strategy. In this case, SEA is guaranteed to achieve, on average, at least the maximin value of the game.

Another use of Theorem 15 is that, under certain conditions, SEA can obtain a higher utility than would have been achieved by converging to a Nash equilibrium. For example in the Prisoner's Dilemma, an agent using SEA can wind up always cooperating instead of converging to the Nash equilibrium. To achieve this, SEA requires that an agent's opponents all be *flexible*.

Definition 23 *An agent is flexible if for all t there exists some time $t' < t$ such that the agent's strategy at time t does not depend on anything that has happened before time t'.*

While the idea of flexibility is a more theoretical one, a realistic situation where agents could be flexible is one where they would have extremely limited amounts of memory. For example, agents working in embedded systems might have very little memory to work with.

If an agent i, using SEA, happens to be playing against flexible opponents, then that agent achieves almost surely an average utility that is asymptotically as large as what agent i's best expert could achieve against the same opponents.

Chapter 4

FRAME

In this chapter we introduce our algorithm, FRAME. In Section 4.1, we present FRAME and how it builds upon ALERT. In Section 4.2, we discuss and prove FRAME's properties.

4.1 Introduction

Although ERT and ALERT, as introduced in Sections 3.1.3 and 3.1.3 respectively, are theoretically important, their practical use is limited by two issues:

1. Since ERT and ALERT were designed as uncoupled algorithms, agents using ERT cannot know with certainty when they have reached an ϵ-Nash equilibrium. Instead agents are only able to bound the probability of not being at an ϵ-Nash equilibrium. Obtaining the necessary bound can require an impractical amount of time. This is exasperated by ALERT calling ERT repeatedly and needing a non-trivial decrease in the size of ϵ with each call [30].

2. ERT and ALERT pick new strategies uniformly at random. Using this brute force method to find an ϵ-Nash equilibrium is a major reason why ALERT takes so long to converge.

Our algorithm, a *Framework for Regret Annealing Methods using Experts* or *FRAME*, is inspired by ALERT but explicitly addresses these two issues while still providing the theoretical guarantees of ALERT.

To address the first issue, we start by making a number of assumptions.

We first assume that at any given point in time, agents' strategies are fully observable for all past time periods. This is a common approach taken by many algorithms [3, 8]. This assumption has no effect on the correctness of our algorithm, instead it removes the need for experimentally determining regret which can be very costly timewise. ALERT could get around this assumption by having agents fix their strategy for a certain period of time. At the end of this period, through simple observation of the actions played by each agent, agents will know all of their opponents' strategies. Thus no privacy is lost by this assumption.

We next assume that the maximum regret of all agents is publicly known. Again, this has no effect on the correctness of our algorithm but removes one of the major performance constraints in ALERT. Although this is not as common an assumption, there are other algorithms that make the stronger assumption that agents can determine a potential equilibrium in advance [4, 13]. Determining a potential equilibrium in advance requires agents to share their utility functions, which are more private than strategies since utility functions cannot necessarily be determined experimentally. As well, determining an equilibrium in advance is a computationally complex problem [10, 14].

Finally we assume that while agents are self-interested, they are willing to cooperate to a certain degree. Specifically, we assume that agents will agree to move to a new joint strategy only if it decreases the maximum regret over all the agents. Since we are only interested in self-play, an equilibrium is often the best outcome for all agents. Hence, although this is our strongest assumption, some cooperation among the agents is in their best interests as it allows for faster convergence rates.

Since the maximum regret is publicly known, agents can now know for certain when a better ϵ-Nash equilibrium has been found. This can potentially be much faster than the ERT and ALERT approach (which requires obtaining a probabilistic bound), and also allows us to use a greedy approach when picking a new ϵ-Nash equilibrium. This approach is different enough that our proof does not follow directly from Germano and Lugosi's work [30].

The second problem with ERT and ALERT is that they choose new strategies naively, i.e. uniformly at random. In contrast, FRAME allows an agent, with some probability, to consult an *expert*, which returns a possible new strategy. Any expert will work, even one who makes only useless suggestions. If the expert is able to find new strategies that lead to better ϵ-Nash equilibria, then the agent can take advantage of this to greatly speed up convergence. However, part of the goal of FRAME is that even with useless experts, convergence is still guaranteed.

The FRAME algorithm for agent i is shown in Algorithm 1. We use the following notation in our algorithm: $\mathcal{U}(X)$ denotes a value picked uniformly at random from

the set X, $e_i(\cdot)$ is the expert and $B(x, d)$ is a bounded search region centered at x with minimum radius $d > 0$.

The FRAME algorithm, with respect to agent i, works as follows. At time $t = 0$, agent i chooses a strategy σ_i^0 uniformly at random from Σ_i. At any subsequent time $t > 0$, FRAME can consult the provided expert, $(e_i(\cdot))$, to obtain a new strategy. Each agent independently consults $e_i(\cdot)$ with a provided probability of p_i. If consulted, the expert returns a possible strategy β_i^{t+1}. To provide protection against poor experts, FRAME checks to see if β_i^{t+1} is inside the region $B(\sigma_i^t, d(r^t))$.[1] If β_i^{t+1} is not, or if the expert was not consulted, β_i^{t+1} is chosen uniformly at random from the bounded search region, $B(\sigma_i^t, d(r^t))$. (This may be thought of as consulting the *Naive Expert*, which is an expert that picks strategies uniformly at random.) Agent i then calculates $r(\beta^{t+1})_i$. If $r(\beta^{t+1}) < r(\sigma^t)$, then $\sigma^{t+1} = \beta^{t+1}$, otherwise, $\sigma^{t+1} = \sigma^t$. To avoid the off chance of getting stuck at a locally optimal joint strategy, each agent chooses an alternative strategy τ_i^{t+1} uniformly at random from Σ_i. If the regret at τ^{t+1} is less than half the current regret, then with a given probability η, the game *resets* to τ^{t+1}. (Any constant fraction less than one will work; one half was chosen for simplicity.) Resetting the joint strategy to τ just means that τ becomes the new joint strategy.[2]

This process repeats until the regret is zero.

4.2 Theoretical Properties

In this section, we discuss the theoretical properties of FRAME. In particular, we prove that FRAME is guaranteed to converge to the set of Nash equilibria. To show convergence, we show that the limit of the sequence of regret of the agents, all using FRAME, is 0, since 0 regret is the same thing as a Nash equilibrium. Formally, if agents start off with a joint strategy σ^0 then for the infinite sequence of regret, $(r^t(\sigma^0))_{t=0}^{\infty}$, we must show that

$$\lim_{t\to\infty} r^t(\sigma^0) = 0. \tag{4.1}$$

We start by examining the case where $\eta = 0$, i.e. the game never resets, for which case we will derive the more relaxed condition,

$$\lim_{t\to\infty} r^t(\sigma^0) = r^\infty \leq r(\sigma^0). \tag{4.2}$$

[1] Any function $d()$ may be used so long as $d(x) > 0$, for $x > 0$.

[2] As will be discussed later, the problem with resetting strategies is that it causes random changes in agents' utilities. Therefore, if possible, resetting should be avoided.

Algorithm 1 $FRAME_i(p_i,\, \eta,\, e_i(\cdot),\, d())$

Require: $0 \leq p_i < 1$, $0 < \eta \leq 1$, $d(\epsilon) > 0, \forall \epsilon > 0$

$\sigma_i^0 = \mathcal{U}(\Sigma_i)$

//Let β be a temporary strategy.

$\beta_i^0 = \sigma_i^0$

for $t = 0, 1, \ldots$ **do**

$x_i = \mathcal{U}([0, 1])$

// With probability p, consult the expert

if $x_i < p_i$ **then**

β_i^{t+1} is the strategy returned by $e_i(\cdot)$

// If β_i^{t+1} is outside of bounded region then must

// choose a new strategy at random

if $\beta_i^{t+1} \notin B(\sigma_i^t, d(r(\sigma^t)))$ **then**

$x = p_i$

end if

end if

//Otherwise, choose a random strategy

if $x_i \geq p_i$ **then**

$\beta_i^{t+1} = \mathcal{U}(B(\sigma_i^t, d(r(\sigma^t))))$

end if

$\tau_i = \mathcal{U}(\Sigma_i)$

// If new regret is less than current regret, then

// update current regret and use new joint strategy

if $r(\beta^{t+1}) < r(\sigma^t)$ **then**

$\sigma^{t+1} = \beta^{t+1}$

else

$\sigma^{t+1} = \sigma^t$

end if

$x = \mathcal{U}([0, 1])$

//If the regret of τ is less than half the current regret,

//with probability η, the joint strategy will reset to τ

if $x < \eta$ and $r(\tau) < r(\sigma^t)/2$ **then**

$\sigma^{t+1} = \tau$

end if

end for

This condition lays the foundation for one of the main propositions regarding the correctness of FRAME.

Proposition 1 *Let σ^∞ be the limit of the joint strategies of agents all using FRAME when $\eta = 0$, i.e.*

$$\lim_{t\to\infty} \sigma^t = \sigma^\infty. \tag{4.3}$$

Then one of the following two conditions must hold:

1. *$r^\infty = 0$, i.e. σ^∞ is a Nash equilibrium*
2. *$r^\infty > 0$ and the agent with the highest regret at σ^∞ is not unique. In this situation σ^∞ is called a critical strategy.*[3]

Proof: This is proved in Section 4.2.1. □

To avoid the second condition in Proposition 1, it is necessary to be able to jump to a completely new joint strategy. This can be done by having $\eta > 0$. In this case, we can achieve the following, stronger result:

Proposition 2 *If $\eta > 0$, then*

$$\lim_{t\to\infty} r^t(\sigma^0) = 0. \tag{4.4}$$

Proof: This is proved in Section 4.2.2. □

As will be shown, the problem with having $\eta > 0$ is that the joint strategy may repeatedly jump to a completely new joint strategy. This can cause chaotic gameplay and is why we can only guarantee convergence to the set of Nash equilibria, as opposed to convergence to a specific Nash equilibrium. This can result in continually random changes in the utilities for the agents. Obviously there is no way to tell in advance if σ^∞ is a critical strategy, but our experimental results chapter shows that this case is rare. Hence, we were able to let $\eta = 0$ for all our experiments and rely solely on Proposition 1 for our correctness.[4]

Example

To understand the two conditions in Proposition 1, consider the game in Figure 4.1. If we use FRAME with $\eta = 0$, this game has two possible outcomes. The first

[3] Formally, we define a joint strategy σ to be a *critical strategy* if $r(\sigma) > 0$ and $\sigma \notin \mathcal{N}$. Although this is not a standard term it is related to the idea of a critical point in multivariate calculus.

[4] Although having $\eta > 0$ does not make a difference in runtime asymptotically, in practice, having to randomly select a joint strategy and compare it every turn is costly.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	$-\epsilon$, $-\epsilon$	0, 0
	$a_{1,2}$	0, 0	1, 1

Figure 4.1: A game with a locally optimal and critical joint strategy.

is that $\sigma^\infty = \{(0,1),(0,1)\}$. In words, this means that the game has converged to the joint strategy $(a_{1,2}, a_{2,2})$ which is the game's only Nash equilibrium. This outcome falls under the first condition of Proposition 1.

The second possible outcome is that $\sigma^\infty = \{(1,0),(1,0)\}$ or the joint strategy $(a_{1,1}, a_{2,1})$. Why is this outcome possible? Consider the initial starting strategy $\sigma_0 = \{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$. For σ_0, agent 1's regret as a function of agent 2's strategy is[5]

$$\begin{aligned} r_1((\sigma_1,\sigma_2)) &= (1-\sigma_2(a_{2,1})) - (\frac{1-\sigma_2(a_{2,1})}{2} - \frac{\epsilon\sigma_2(a_{2,1})}{2}), \\ &= \frac{1}{2} - \frac{\sigma_2(a_{2,1})}{2} + \frac{\epsilon\sigma_2(a_{2,1})}{2}. \end{aligned} \tag{4.5}$$

By symmetry of the game, agent 2 has an equivalent regret function. The importance of this function is that as $\sigma_2(a_{2,1})$ decreases, agent 1's regret will increase. Hence, as σ moves from $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$ to $\{(1,0),(1,0)\}$, the agent's regret will actually increase (at least for a while). Since FRAME only allows new joint strategies to be adopted if they decrease the overall regret, then if $\sigma_0 = \{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$, FRAME will actually not be able to achieve convergence to the Nash equilibrium. Instead FRAME will converge to the joint strategy $\{(1,0),(1,0)\}$, since that will decrease the overall regret.

Once in the region of $\{(1,0),(1,0)\}$, FRAME will not be able to escape. Hence, $\{(1,0),(1,0)\}$ is a locally optimal joint strategy. It also happens to be a critical strategy since $r_1(\{(1,0),(1,0)\}) = \epsilon = r_2(\{(1,0),(1,0)\})$. Therefore this outcome is covered by the second condition in Proposition 1. This is obviously not a proof that FRAME will always result in one of the two conditions in Proposition 1. However, it does give an idea of how those conditions can arise. To avoid the second condition, it would be necessary to somehow be able to jump from a joint strategy in the region around $\sigma = \{(1,0),(1,0)\}$ to a joint strategy in the region around $\sigma = \{(0,1),(0,1)\}$. This is why Proposition 2 is required.

[5]Note that agent 1's regret is equal to the maximum utility it could have obtained: in this case $1-\sigma_2(a_{2,1})$) minus the utility it did obtain $\frac{1-\sigma_2(a_{2,1})}{2} - \frac{\epsilon\sigma_2(a_{2,1})}{2}$.

4.2.1 Proof of Proposition 1

In order to prove Proposition 1, we start by proving the following two lemmas.

Lemma 2 *Consider the joint strategy σ. Let $\sigma_i^* \in BR_i(\sigma_i)$. Assuming that $\sigma_i \notin BR_i(\sigma_i)$, consider the line segment l from σ_i to σ_i^* such that*

$$l(\Delta) = \sigma_i + \Delta\rho,\ 0 \leq \Delta \leq 1, \tag{4.6}$$

where $\rho = \sigma_i^* - \sigma_i$.

Then for $0 < x \leq 1$, $u_i(l(x), \sigma_{-i}) > u_i(\sigma_i, \sigma_{-i})$ and $r_i(l(x), \sigma_{-i}) < r_i(\sigma_i, \sigma_{-i})$.

Proof: We first write out u_i as

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{a_i} \sum_{a_{-i}} P(a_i|\sigma_i) P(a_{-i}|\sigma_{-i}) u_i(a_i, a_{-i}). \tag{4.7}$$

The total differential, du_i, of equation 4.7 is

$$du_i = \frac{\partial u_i}{\partial \sigma_i(a_{i,1})} d\sigma_i(a_{i,1}) + \ldots + \frac{\partial u_i}{\partial \sigma_i(a_{i,|A_i|})} d\sigma_i(a_{i,|A_i|}), \tag{4.8}$$

$$= \bigtriangledown u_i \cdot < d\sigma_i(a_{i,1}), \ldots, d\sigma_i(a_{i,|A_i|}) > . \tag{4.9}$$

If we are only interested in the total differential along l then we can simplify equation 4.9 to

$$du_i = \bigtriangledown u_i \cdot \rho d\Delta. \tag{4.10}$$

Since Equation 4.7 is just a summation of linear terms, each of the partial derivatives is constant, and therefore $\bigtriangledown u_i$ is also a constant. Therefore, the rate of change is constant along l and must be increasing. Since the utility is increasing the regret must be decreasing. □

Lemma 3 *For a given ϵ-Nash equilibrium σ, let $f_\sigma(\sigma^\infty) : \mathbb{R}^{N|A|} \rightarrow \mathbb{R}$ be the change in regret from moving from the strategy σ to the new strategy σ^∞, i.e.,*

$$f_\sigma(\sigma^\infty) = r(\sigma) - r(\sigma^\infty). \tag{4.11}$$

If there is some strategy σ' such that $f_\sigma(\sigma') > 0$ and $|\sigma' - \sigma| < d(\epsilon)$, then there exists some region $Y \subseteq \Sigma$ such that

$$P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0, \tag{4.12}$$

and furthermore, for all $\sigma'' \in Y$, $f_\sigma(\sigma'') > 0$. In words, if there is at least one strategy, σ', within a bounded region around σ which has less regret than σ, then there is a positive probability of picking a strategy uniformly at random from that bounded region that has regret less than σ. Furthermore, this region includes σ'.

Proof: This proof is left for Appendix A. □

Using these two lemmas, we can prove the following proposition.

Proposition 3 *For any non-critical ϵ-Nash equilibrium σ, the region $B(\sigma, d(\epsilon))$ contains a region S, such that $P(\mathcal{U}(B(\sigma, d(\epsilon))) \in S) > 0$ and for all $\sigma' \in S$, $r(\sigma') < \epsilon$.*

Proof: Since σ is a non-critical strategy, there exists an unique agent i such that $r_i(\sigma) = r(\sigma) = \epsilon$. Consider agent i's strategy σ_i versus its opponents' joint strategy σ_{-i}. For σ_{-i}, agent i has a best response strategy $\sigma_i^* \in BR_i(\sigma_{-i})$ such that $u_i(\sigma_i^*, \sigma_{-i}) > u_i(\sigma_i, \sigma_{-i})$. Let l be a line segment from σ_i to σ_i^*. Note $|l| > 0$ since $\sigma_i^* \neq \sigma_i$. When we adjust agent i's policy, by some amount $\Delta(\sigma_i)$, along l towards σ_i^*, we decrease i's regret using Lemma 2.

However, at the same time we may increase another agent's regret. In the worst case, suppose that agent j has the second largest regret, r_j, and its increase, γ, with respect to $\Delta(\sigma_i)$, is the largest among all agents. Since we want to decrease the overall amount of regret, we want to choose some $\Delta(\sigma_i)$ such that $r_j + \gamma\Delta(\sigma_i) < r$. Set

$$\Delta(\sigma_i) = \min[d(\epsilon), \frac{r - r_j}{2\gamma}], \tag{4.13}$$

since this guarantees that $\Delta(\sigma_i) < d(\epsilon)$. Now $r(\sigma_i + \Delta(\sigma_i), \sigma_{-i}) < \epsilon$, that is we have found a better ϵ-Nash equilibrium. By Lemma 3, $P(r(\mathcal{U}(B(\sigma, d(\epsilon)))) < r(\sigma)) > 0$. In words, this means that a joint strategy picked uniformly at random from the region $B(\sigma, d(\epsilon))$ has a positive probability of having less regret than σ. □

We are now ready to prove Proposition 1.

Proof: We start by showing that for all non-critical joint strategies, there is always a new joint strategy close by which is closer to being an equilibrium. By close by, we mean within some bounded region centered on the current joint strategy.[6] Proposition 3 shows that such joint strategies do exist and that FRAME has a positive probability of finding them.

Now suppose that agents play a repeated game for an infinite number of turns using FRAME. Agents will move to a new joint strategy if it decreases the overall regret. Therefore, if for some subsequence, $Q = \{q_1, \ldots\}$, of all turns, the sequence r_t^q converges to a specific value, say r^∞, then the sequence of regret for all turns must be at most r^∞.

[6]In our code, this bounded region is denoted by $B(\sigma_i^t, d(r(\sigma^t)))$. In our implementation we used the bounded region of a $L\infty$-ball $D_\infty(\sigma_i^t, d(r(\sigma^t)))$, which can be thought of as a hyper-cube centered around σ_i^t with width $2d(r(\sigma^t))$.

Every turn there is a $(1-p_e)^n > 0$ chance of all agents picking a new strategy at random. Therefore, let Q be the infinite subsequence of turns where all agents update their strategies at random.

We now prove Proposition 1 by contradiction. Suppose that $r^\infty > 0$ and σ^∞ is not a critical strategy. Now consider some finite point in time, $t-1$, where agent i has the largest regret with respect to σ^{t-1}. Let us assume the worst case, where agent j has both the second largest regret and r_j's rate of increase with respect to σ_i^{t-1}, γ, is the largest among all agents. (If the agent is not unique then j may be any of them.) Define

$$D_p(\sigma^{t-1}) = r_i(\sigma^{t-1}) - r_i(\sigma_i^{t-1} + \Delta^t(\sigma_i), \sigma_{-i}^{t-1}) - \xi, \tag{4.14}$$

for some small $\xi > 0$, where

$$\Delta(\sigma_i) = \min\left[d(r(\sigma^{t-1})), \frac{r(\sigma^{t-1}) - r_j(\sigma^{t-1})}{2\gamma}\right]. \tag{4.15}$$

By Proposition 3, at time t, there is a positive probability of FRAME being able to reduce the overall regret by at least $D_p(\sigma^{t-1})$ versus the regret at time $t-1$. We would like to be able to say something about the behaviour of $D_p(\sigma^t)$ as t approaches infinity. While in general, we would expect $D_p(\sigma^t)$ to be decreasing, unfortunately it is not necessarily a monotonically decreasing function. Furthermore, even if σ^∞ is not a critical strategy, it is possible that at some finite time t, σ^t might be one. (This is possible since critical strategies may include non-locally optimal strategies or locally optimal strategies from FRAME can still escape from.) Hence, $D_p(\sigma^t)$ may at times even be 0. However, there must exist some time T^c after which no critical strategy is encountered (since the game is approaching a non-critical strategy). We thus define

$$D_{\inf}(\sigma) = \inf\{D_p(\sigma^t) | t \in Q, t \geq T^c\}, \tag{4.16}$$

where inf or infimum is the greatest lower bound. Note that $\delta_{inf}(r) > 0$.

Now consider the actual decreases in regret given by

$$D_a(\sigma^{t-1}, \sigma^t) = r(\sigma^t) - r(\sigma^{t-1}). \tag{4.17}$$

We know that $\lim_{t\to\infty} D_a(\sigma^{t-1}, \sigma^t) = 0$, and therefore there exists a point in time $T \in Q$ greater than or equal to T^c such that

$$\forall t' \geq T,\ D_a(\sigma^{t'}, \sigma^{t'+1}) < \delta_{inf}(\sigma). \tag{4.18}$$

By Proposition 3, for all $t' \geq T$ there exists a positive probability of finding a new joint strategy that reduces the overall regret by at least $\delta_{\inf}(\sigma)$. Therefore this must happen once which is a contradiction of Equation 4.18. Therefore σ^∞ cannot be a critical strategy and Proposition 1 is proven. □

4.2.2 Proof of Proposition 2

If agents get stuck in a locally optimal region, FRAME will have to jump to a completely different region of the strategy space. A key part of Proposition 2 is Lemma 4, which says that if FRAME does pick a joint strategy uniformly at random from all possible joint strategies, there is a positive probability of finding a strategy closer to equilibrium.

Lemma 4 *Given σ such that $r(\sigma) > 0$, there is a positive probability of picking a joint strategy $\sigma' \in \Sigma$ uniformly at random such that $r(\sigma') \leq r(\sigma)/2$.*

Proof: This is proved in Appendix A. □

Thus the proof for Proposition 2 will require showing that by picking a joint strategy uniformly at random from all possible strategies enough times, FRAME will never get stuck in a locally optimal region.

Proof: We now consider the case where $\eta > 0$. It should be noted that with $\eta > 0$, new joint strategies can now come from Σ. However, it is still the case that these joint strategies will be picked only if they decrease the overall regret. Hence, cases where convergence was achieved when $\eta = 0$ will still achieve convergence when $\eta > 0$. The difference is that we can now deal with cases where the limiting strategy is a critical strategy.

Suppose that when $\eta = 0$, the limiting strategy is indeed a critical one. (In this case FRAME would be unable to achieve convergence.) Let this strategy be σ^∞ and the corresponding regret $r^\infty > 0$. Now set η to any value such that $\eta > 0$. In this case, FRAME now choose a new joint strategy from all possible joint strategies. The trick is picking a new strategy such that FRAME is no longer stuck (i.e. the limiting strategy is still σ^∞). As previously mentioned, there is no way to know in advance if σ^∞ is a critical strategy or what σ^∞ will be; hence we must assume the worst case that, σ^∞ is indeed a critical strategy. However, since we do not have know what σ^∞ will be, there is no way of picking a single new joint strategy such that we guarantee FRAME will not be stuck at σ^∞.

Instead we will use an infinite sequence $\sigma^{T'} = \{\sigma^{t'_1}, \sigma^{t'_2}, \ldots\}$ such that, no matter what σ^∞ actually is, we can guarantee that FRAME will not get stuck. One possibility is a series of such that $r(\sigma^{t'_{i+1}}) \leq r(\sigma^{t'_i})/2$. That way, no matter what σ^∞ is, there exists some time j such that for all $i \geq j$, $r(\sigma^{t'_i}) < r(\sigma^\infty)$. Hence $\sigma^{T'}$ will not get stuck at σ^∞.

To prove that $\sigma^{T'}$ can exist, we must show that given any σ there is a positive probability of finding $\sigma' \in \Sigma$ such that $r(\sigma') \leq r(\sigma)/2$. This is done using Lemma

4. By setting $\eta > 0$, we guarantee that FRAME is able to make $\sigma^{T'}$ a subsequence of σ^t.

Therefore, if $\eta > 0$, σ^∞, if it exists, cannot be a critical strategy and we have convergence to the set of Nash equilibria. □

It is important to note that throughout all of these proofs, any iteration of a game where agents consult an expert were explicitly ignored. Thus, suggestions made by experts have no impact on the correctness of FRAME.

4.3 Conclusion

In this chapter we introduced the algorithm FRAME. FRAME builds upon ERT and ALERT and is able to keep the theoretical results they achieve. However, whereas ALERT cannot be used in practice even for the most simple games, FRAME designed with the goal of being a practical algorithm. One of the reasons for this is that FRAME allows agents the chance to consult an expert for possible new strategies; when the expert makes good suggestions, the agents are able to benefit and the convergence rate improves. However, when the expert makes useless or even hostile suggestions, convergence is still guaranteed.

The second half of this chapter introduced the main theoretical properties of FRAME, and proved them. Specifically, it was proven that FRAME is able to achieve convergence to the set of Nash equilibria for all games. Furthermore, it was proven that under common conditions, FRAME is able to achieve convergence to a single Nash equilibrium.

Chapter 5

FRAME Experimental Results

In this chapter we discuss our findings from a series of experiments using FRAME. We first describe our experimental setup, including which experts were chosen and why, as well as which games were used in the experiments. We then report our findings, and illustrate that FRAME is a practical learning algorithm.

5.1 Experimental Setup

While any expert will work in theory, ones that make gradual adjustments to the strategies of the agents are considered to be better, since it is easier to observe their effect. In our experiments we used three such experts; the Hart and Mas-Colell algorithm (HMC), logistic fictitious play (LFP) and Win or Learn Fast (WoLF). These experts were chosen because all of them work by making gradual adjustments in strategies. Furthermore, they represent the three basic approaches to multiagent learning. Given the fundamental difference between these experts, it is not surprising that each of them has its own area of expertise, or types of games it is best suited for. By experimenting using these different areas of expertise we are able to clearly contrast these experts.

5.1.1 Experts

We briefly review each of the experts.

Hart and Mas-Colell algorithm (Section 3.1.2)

At time t, agent i plays an action a_i^t according to σ_i^t (with σ_i^0 being the uniform distribution). Based on σ_{-i}^t, agent i measures its regret for not having played any other action. At time $t+1$, each action is then played with a probability proportional to that regret.

Formally we define $D_i^t(a_{i,j}, a_{i,k})$ as

$$\mathbb{D}_i^t(a_i, a_i') = \sum_{a_{-i} \in A_{-i}} \sigma_i^t(a_i)\sigma_{-i}^t(a_{-i})[u_i(a_i', a_{-i}) - u_i(a_i, a_{-i})]. \tag{5.1}$$

This is a measure of how much agent i prefers action a_i' to a_i. Thus we define agent i's regret as,

$$R^t(a_i, a_i') = \max\{0, \mathbb{D}_i^t(a_i, a_i')\}. \tag{5.2}$$

Finally, agent i picks σ_i^{t+1} as follows,

$$\sigma_i^{t+1}(a_i) = \begin{cases} \frac{1}{\mu} R_i^t(a_i^t, a_i), & \text{if } a_i \neq a_i^t \\ 1 - \sum_{a_i' \neq a_i^t} \sigma_i^{t+1}(a_i') & \text{otherwise,} \end{cases} \tag{5.3}$$

where μ is a constant which can have any value greater than $|A_i - 1|$. We let $\mu = |A_i|$.

Logistic Fictitious Play (Section 3.2)

Given σ_{-i}, Logistic Fictitious Play (LFP) gives a strategy σ_i in which the probability of playing action a_i is calculated by

$$\sigma_i(a_i) = \frac{e^{(1/\lambda)u_i(a_i, \sigma_{-i})}}{\sum_{a_i' \in A_i} e^{(1/\lambda)u_i(a_i', \sigma_{-i})}}, \tag{5.4}$$

where λ is a smoothness parameter. As λ approaches 0, LFP approaches playing a strict best response to σ_{-i}.

WoLF (Section 3.3)

Given σ_{-i}^{t-1} and σ_i^{t-1}, WoLF calculates σ_i^t by adjusting σ_i^{t-1} towards $BR_i(\sigma_{-i}^{t-1})$. This is done in incremental steps. The key to WoLF is that the step sizes vary depending on whether the agent is "winning" or "losing". To determine if an agent is winning or losing, WoLF compares the performance of the agent's current strategy against the performance of an "average" strategy (Equation 5.5 gives the definition of the average strategy). If the current strategy is doing worse than the average strategy, the agent is losing. In this case WoLF tries to change quickly by having large step sizes. If the current strategy is doing better than the average one, the agent is winning; in this case WoLF is more cautious and takes smaller step sizes.

1. Let $t = 0$, $\delta_w, \delta_l \in (0, 1]$ and $\sigma_i^0 = \frac{1}{|A_i|}$.
2. Repeat
   (a) Play σ_i^t and observe u_i^t. Let $t = t + 1$.
   (b) Update the estimate of the average strategy

$$\forall a_i \in A_i \; \bar{\sigma}_i^t(a_i) = \bar{\sigma}_i^{t-1}(a_i) + \frac{1}{t}(\sigma_i^{t-1}(a_i) - \bar{\sigma}_i^{t-1}(a_i)) \tag{5.5}$$

   (c) Update the strategy

$$\sigma_i^t(a_i) = \sigma_i^{t-1}(a_i) + \Delta_{a_i} \tag{5.6}$$

   where

$$\Delta_{a_i} = \begin{cases} -\delta_{a_i} & \text{if } a_i \neq \arg\max_{a_i' \in A_i} u_i(a_i, \sigma_{-i}^t) \\ \sum_{a_i' \neq a_i} \delta_{a_i'} & \text{otherwise,} \end{cases} \tag{5.7}$$

$$\delta_{a_i} = \min\left(\sigma_i^t(a_i), \frac{\delta}{|A_i| - 1}\right) \tag{5.8}$$

$$\delta = \begin{cases} \delta_w & \text{if } \sum_{a_i' \in A_i} \sigma_i^t(a_i') u_i(a_i', \sigma_{-i}^t) > \sum_{a_i' \in A_i} \bar{\sigma}_i^t(a_i') u_i(a_i', \sigma_{-i}^t) \\ \delta_l & \text{otherwise.} \end{cases} \tag{5.9}$$

5.1.2 Implementation Issues

FRAME is implemented using C++ in Linux. The simulations were run on two systems; Pilatus and Vidal at the University of Waterloo. Pilatus is a 64-bit system

composed of 64 Itanium2 processors with 192 gigabytes of memory [19]. Vidal is a cluster consisting of 20 nodes, each with 2 Opteron processors and 4 gigbytes of memory [19]. The runtime for the batch of 1000 trials ranged from a few seconds to a couple of hours.

5.1.3 Games

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	1, 0.5	0, 0
	$a_{1,2}$	0, 0	0.5, 1

Figure 5.1: Battle of the Sexes

		Agent 2		
		$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
Agent 1	$a_{1,1}$	0, 0	1, 0	0, 1
	$a_{1,2}$	0, 1	0, 0	1, 0
	$a_{1,3}$	1, 0	0, 1	0, 0

Figure 5.2: Shapley's Game

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	1, 1, 0	0, 0, 0
	$a_{1,2}$	0, 1, 1	1, 0, 1

Agent 3 - $a_{3,1}$

	Agent 2	
	$a_{2,1}$	$a_{2,2}$
$a_{1,1}$	1, 0, 1	0, 1, 1
$a_{1,2}$	0, 0, 0	1, 1, 0

Agent 3 - $a_{3,2}$

Figure 5.3: 3-Player Matching Pennies: agent 1 chooses the row, agent 2 chooses the column, and agent 3 chooses the matrix

We ran experiments on the games shown in Figures 5.1 through 5.3. (Additional results for different games are included in Appendix B. The results presented in this chapter are the most informative.) For each of these games, we ran 1000 trials. While the starting strategies have a definite impact on the convergence rates and possibly on the relative performance of each of the experts, to avoid an overload in information, we examined only one starting strategy for each game. Since only a small value for $1 - p$ was needed to obtain a high degree of randomization, results are only shown for $p = 0.75, 0.95$ and 0.98. Where ever possible, the parameters for each expert were based on the existing literature. Convergence was measured to 2 decimal places.

All results are shown in histogram format. For each run, our data was divided up into 20 intervals. For example, if for some run, the fastest convergence time was 10 iterations and the slowest was 110, then the interval size for that run's histogram

would be 5. Thus, the x-axis in each of our graphs is the convergence time divided up into intervals and the y-axis the percentage of trials that fell into each interval.

For each of these games, given the starting strategies and the experts used, there is no risk of running into a locally optimal joint strategy or a plateau. Thus, for these games, FRAME's correctness can rest solely on Proposition 1 and we are able to set $\eta = 0$. This allows for a faster convergence time.

We examine each of the games in turn.

5.2 Experimental Results

Battle of the Sexes

Battle of the Sexes (BoS) has 3 Nash equilibria;

$$\{((1,0),(0,1)),((0,1),(0,1)),((\frac{2}{3},\frac{1}{3}),(\frac{1}{3},\frac{2}{3}))\}.$$

Different learning algorithms can have a bias towards one or two of the Nash equilibria (usually either the pure or mixed equilibria). Thus, we chose this game because it balances a simple joint action set with a complex set of Nash equilibria. We used a starting strategy of $\sigma^0 = \{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$.

As a reference point, we first had both agents only consulting the Naive Expert. The results are shown in Figure 5.4. The convergence rates for ALERT on BoS would be off the charts compared to these results (ALERT has a fixed runtime independent of any parameters of the game). Thus, we have already shown that FRAME can be an effective method for learning. However, with the proper use of experts, we can do even better.

The first expert we examined was LFP with a parameter of $\lambda = 0.5$. The results are shown in Figure 5.5. If $p_e = 1$, it would take around $160,000$ iterations for convergence. However, when $p_e = 0.98$ the convergence rate improves significantly. This shows that occasionally consulting the Naive Expert not only provides theoretical guarantees, it can also be practical.

The next expert we used was WoLF, with parameters $\delta_w = \frac{1}{20000+t}$ and $\delta_l = 2\delta_w$. The results are shown in Figure 5.6. As shown, WoLF converges very quickly for BoS. In general, however, for such a small game, there is not much difference between a randomized approach and WoLF. The exception is when FRAME forces WoLF to converge to an equilibrium it would not normally converge to. In the case

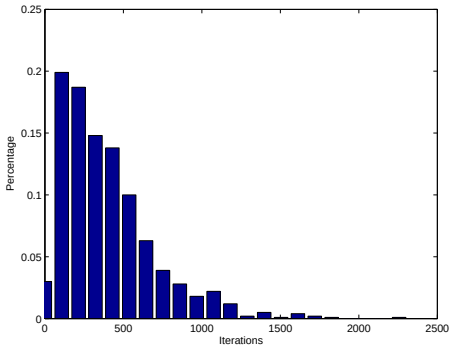

Figure 5.4: Convergence rates for BoS using a purely random learning algorithm.

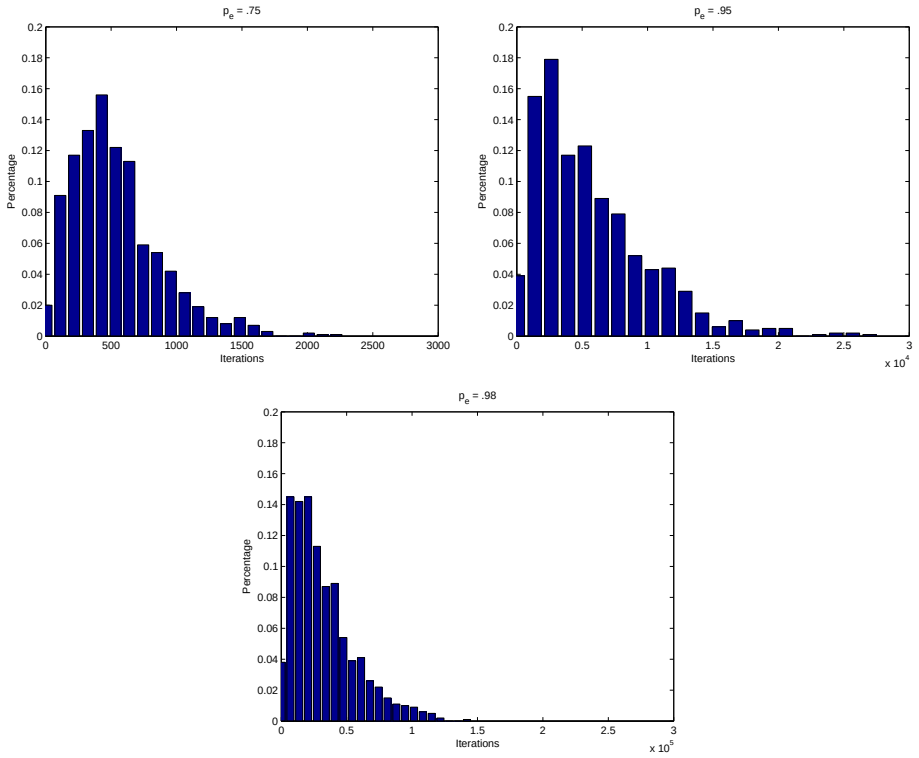

Figure 5.5: Convergence rates for BoS using FRAME with LFP. Note the difference in scale.

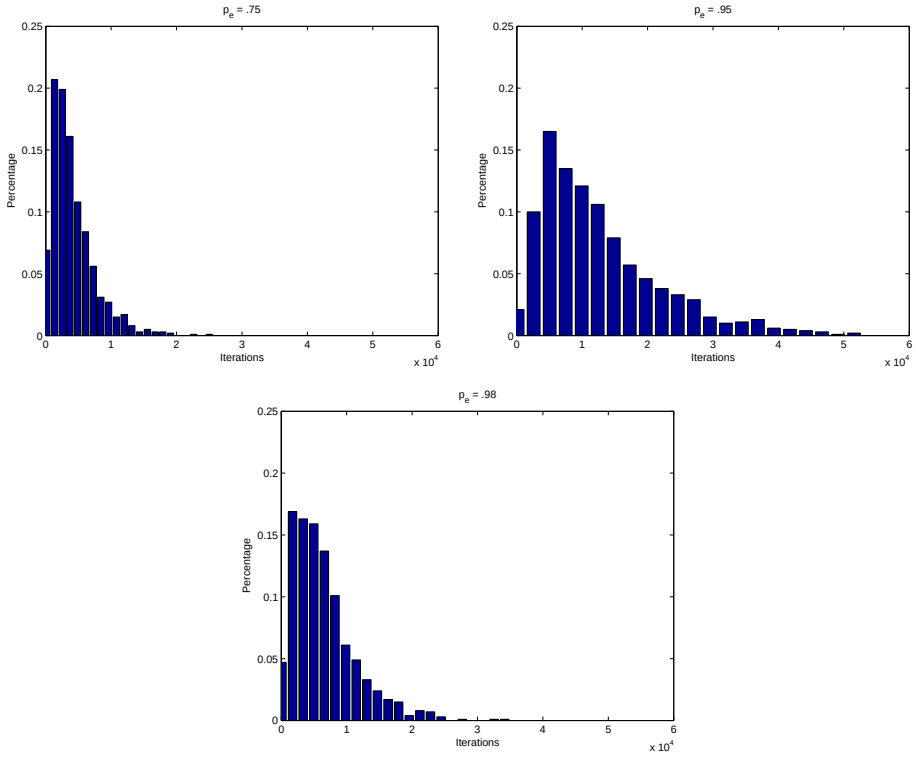

Figure 5.6: Convergence rates for BoS using FRAME with WoLF.

of BoS, by itself, WoLF would never converge to the mixed equilibrium. Hence, when FRAME forces WoLF to do so, convergence takes much longer. This explains why the convergence rate decreases so much for $p_e = 0.95$. When $p_e = 0.98$, there are too few random jumps for FRAME to force WoLF to converge to the mixed equilibrium.

Finally, we used HMC with a parameter of $\mu = 2$. The results are shown in Figure 5.7. Like WoLF, HMC is able to achieve convergence quickly. Thus, as expected, as $p_e \to 1$, the convergence rate improves.

Shapley's Game

Shapley's Game, shown in Figure 5.2, is a classic game because it was the first game in which fictitious play was shown to not converge in any sense. It is still regarded as a hard game for learning algorithms, with more recent algorithms such as WoLF still unable to achieve convergence. However, LFP, given the right value

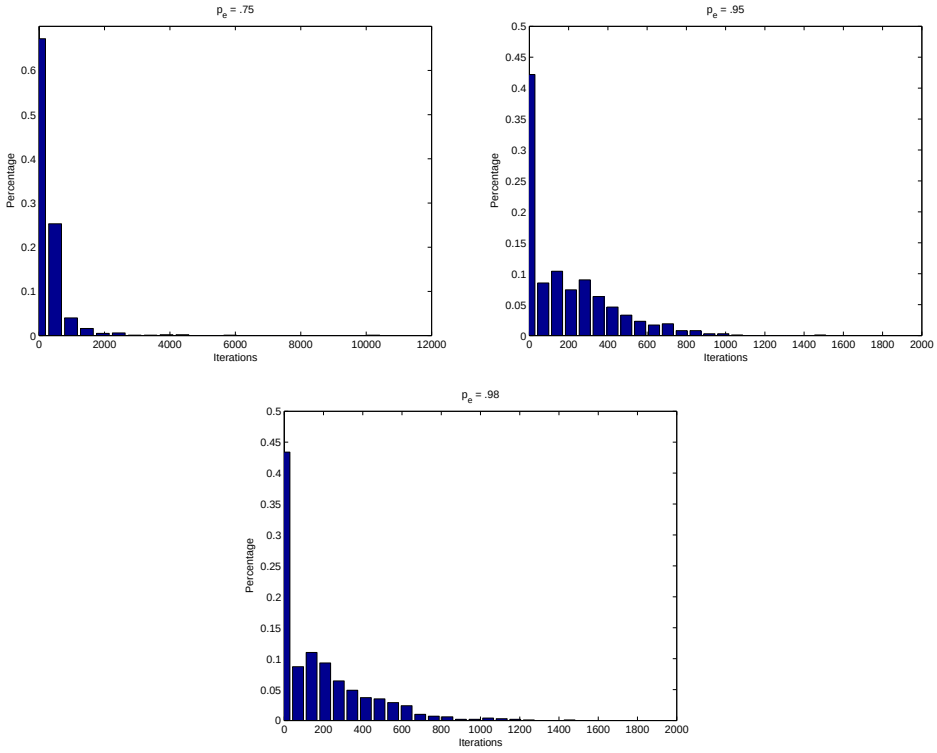

Figure 5.7: Convergence rates for BoS using FRAME with HMC. Note the difference in scale.

for λ, converges very quickly. Shapley's game has an unique Nash equilibrium of $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})\}$.

Again, for reference, we first show, in Figure 5.8, the results for learning using a purely random learning algorithm.

The results for LFP using a value of $\lambda = 0.5$ are shown in Figure 5.9. Since LFP converges very quickly for Shapley's game, as p_e increases we could expect to see faster convergence times, which is exactly what happens.

The results for WoLF, using $\delta_w = 1/(100 + t)$ and $\delta_l = 3\delta_w$, are shown in Figure 5.10. Note the difference in scale and that data is presented up to the 98^{th} percentile. Since WoLF does not converge for Shapley's Game, as p_e increases, we would expect to see slower convergence rates. This is indeed what happens. More importantly, though, is that as p_e approaches 1, while the convergence rates may increase, we are still achieving convergence. This is an example of FRAME being able to deal with an expert poorly suited for a particular game.

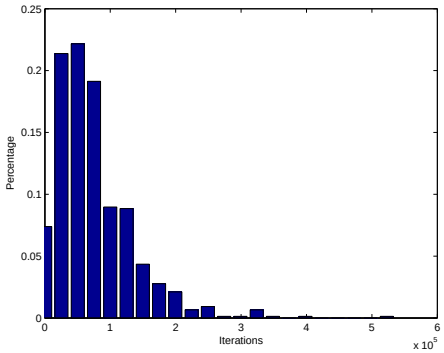

Figure 5.8: Convergence rates for Shapley's Game using a purely random learning algorithm.

Finally, in Figure 5.11, we present the results for Shapley's Game using HMC as the expert. Although HMC does not achieve convergence by itself, it is able to achieve convergence to the set of correlated equilibria. As p_e increases, there is no major change in the rate of convergence. This suggests that HMC is no better but also no worse than a purely randomized approach to Shapley's Game.

3-Player Matching Pennies

3-player Matching Pennies, as shown in Figure 5.3, is another game which can be very difficult to achieve convergence in. However, unlike Shapley's game, WoLF is able to achieve convergence while LFP is not.

The first expert we used was LFP. Unlike with Shapley's Game, by itself LFP cannot achieve convergence in 3-Player Matching Pennies. As a result, the more an agent consults LFP, the slower convergence should be. However, we should still be seeing convergence. The results shown in Figure 5.12 confirm these expectations.

The convergence rates for WoLF are shown in Figure 5.14. Since WoLF converges quickly in 3-Player Matching Pennies, we would expect to see faster convergence rates. This is what happens, which shows that a poor expert for one game may actually be an excellent expert in another. This is a strong argument in favour of exploring many different experts.

The results for 3-Player Matching Pennies using HMC are shown in Figure 5.15. We see that, although HMC is not as well suited for 3-player Matching Pennies as

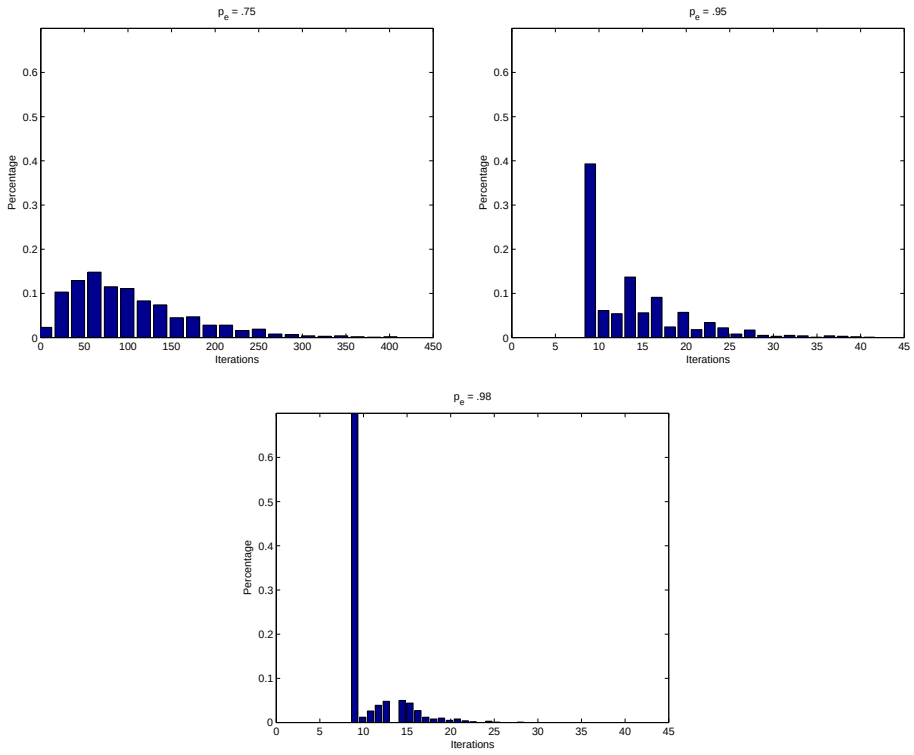

Figure 5.9: Convergence rates for Shapley's Game using FRAME with LFP. Note the difference in scale.

it was for Shapley's game, it is still able to perform decently well. This is reflected in the moderate decrease in convergence rates as p_e increases.

5.3 Conclusion

In this chapter we reported on the experimental results for FRAME; these experiments involved several different experts and games. The results confirmed the two key benefits of FRAME. The first is that the practice of consulting experts can have a significant impact on the convergence rate: when experts provided good strategies, agents were able to improve the convergence rate. The second benefit is that convergence is always guaranteed. Even when experts do not provide useful strategies and the convergence rate decreases, convergence is to the set of Nash equilibria still achieved.

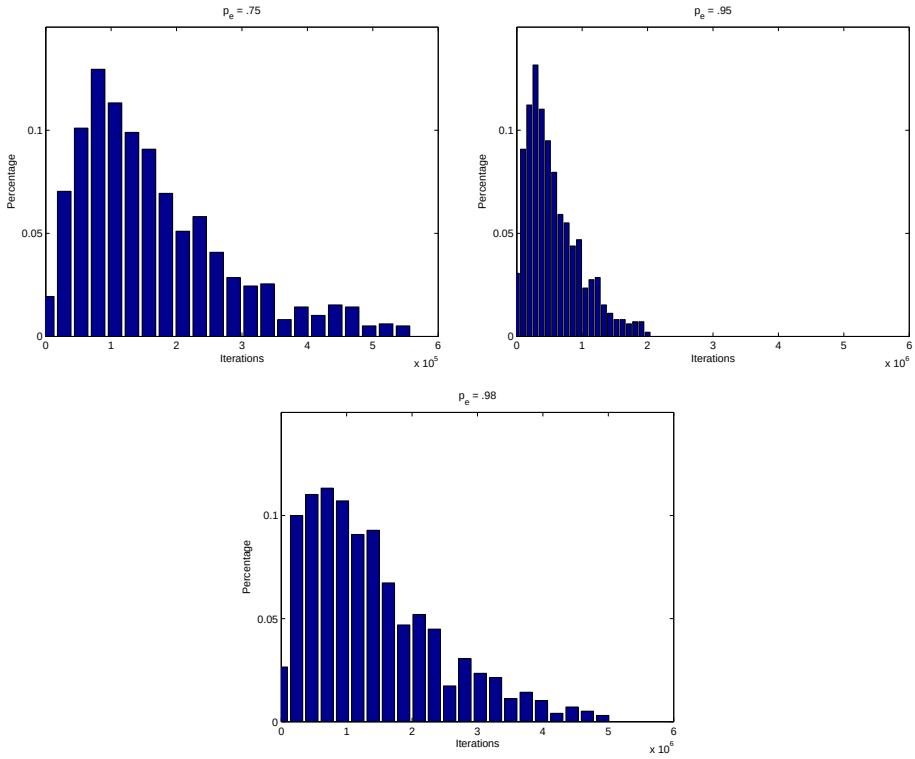

Figure 5.10: Convergence rates for Shapley's Game using FRAME with WoLF. Note the difference in scale. Each graph is shown up to the 98^{th} percentile.

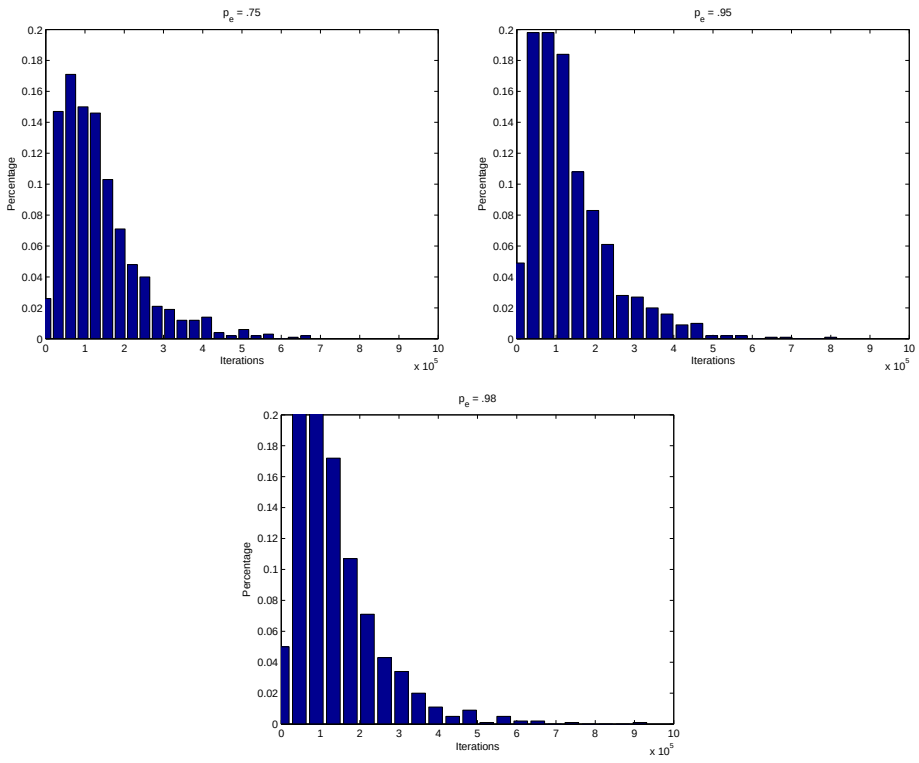

Figure 5.11: Convergence rates for Shapley's Game using FRAME with HMC.

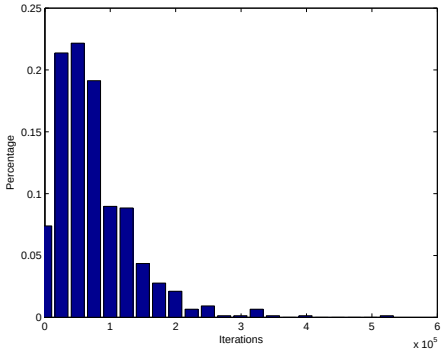

Figure 5.12: Convergence rates for 3-Player Matching Pennies using a purely random learning algorithm.

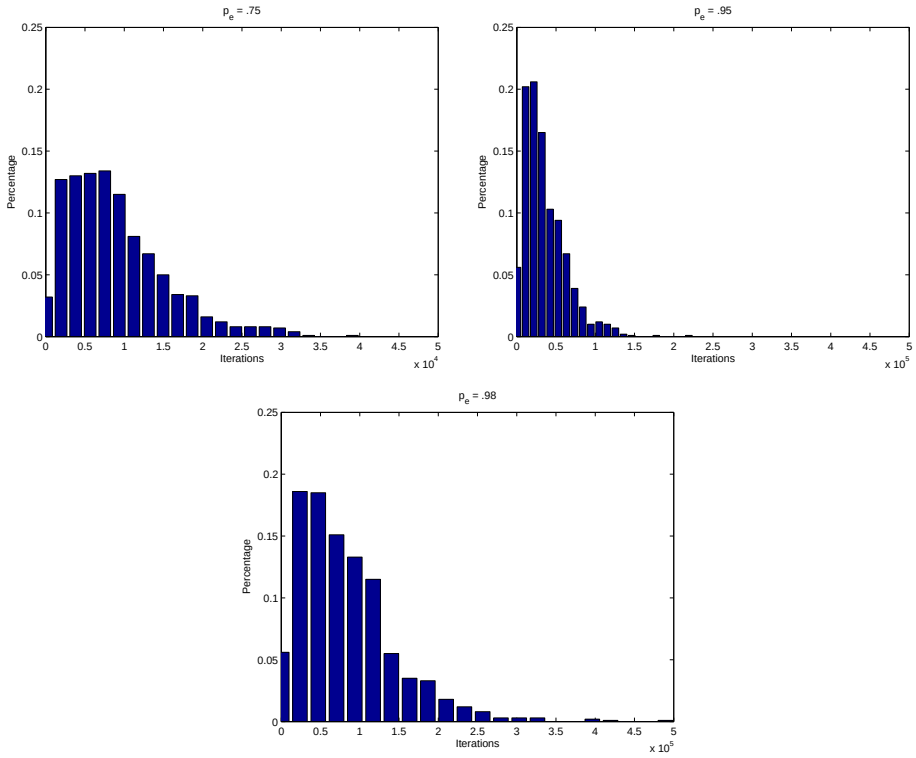

Figure 5.13: Convergence rates for 3-Player Matching Pennies using FRAME with LFP. Note the difference in scale.

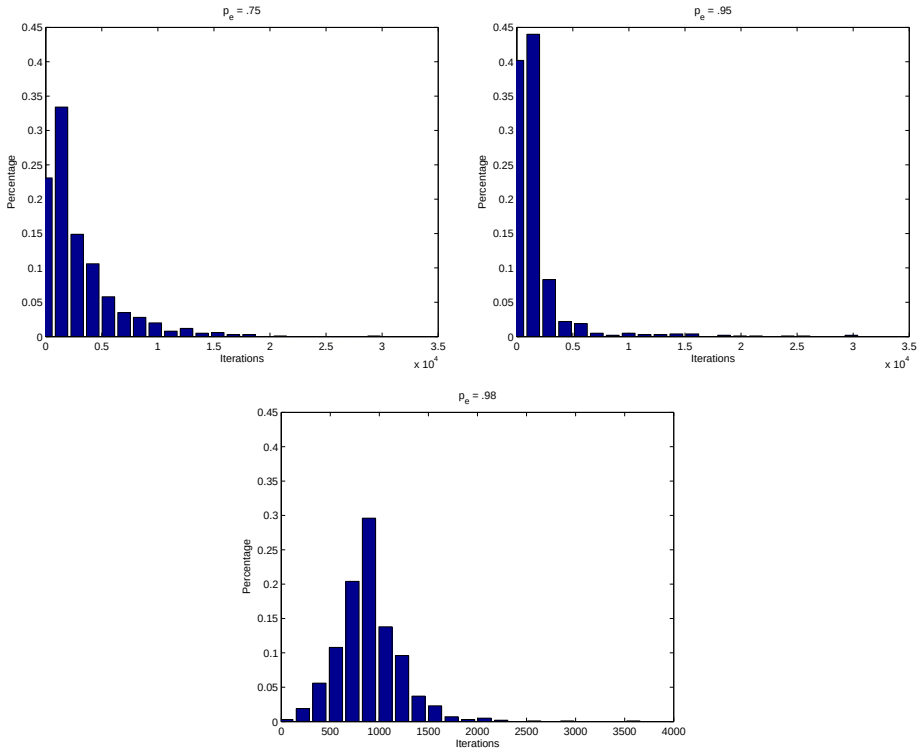

Figure 5.14: Convergence rates for 3-Player Matching Pennies using FRAME using WoLF. Note the difference in scale.

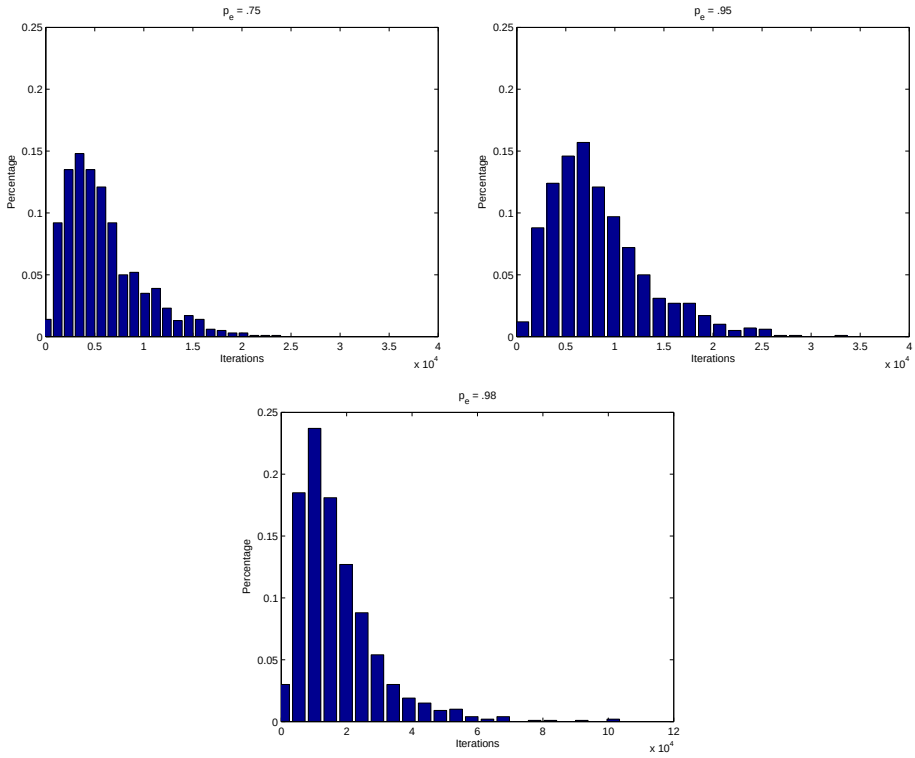

Figure 5.15: Convergence rates for 3-Player Matching Pennies using HMC. Note the difference in scale.

Chapter 6

Adaptive-FRAME

In the previous chapter, we showed the experimental results for FRAME. These results confirmed that FRAME is able to balance theoretical guarantees and practical concerns. However, FRAME is limited by allowing agents to only consult one expert. Since different experts are better suited for different games, this limits the flexibility of FRAME.

The solution is to allow agents to consult multiple experts. The naive way of doing this is to have agents consult each expert with an equal probability; we call this approach the *Naive Experts Algorithm* (NEA). However, for a given game, if one expert is providing better strategies than another expert, consulting the first expert more would improve convergence rates. Thus, we would like some sort of an adaptive approach to consulting agents.

In order to do so, we have to define some metric for comparing the performance of experts. Based on this metric, we will have to develop some adaptive approach to consulting the experts. It turns out that there are several possible metrics and adaptive approaches. This chapter is concerned with examining different metrics and approaches; these include existing methods as well as ones designed specifically for FRAME.

6.1 An Adaptive Approach

We generalize FRAME in two ways: first, instead of one expert, each agent has a set of experts $E_i = \{e_{i,0}, \ldots, e_{i,|e_i|}\}$ to consult. (For simplicity, $e_{i,0}$ is always the *Naive Expert* which suggests a strategy picked uniformly at random from a bounded

region.) Since different experts are better suited for different games, this will allow an agent more flexibility. With slight abuse of notation, we define e_i to be some specific but undefined expert for agent i. Expert e_i is consulted with probability p_{e_i} and returns a suggested strategy β_{e_i}.

Secondly, we allow the probabilities of consulting each expert to vary over time, that is we generalize p_{e_i} to $p^t_{e_i}$. As a result we are no longer required to decide on and fix the probabilities in advance. We are now able to tune the probabilities to best suit the current game. The most practical way of doing this is to adjust the probabilities while playing the game, since this allows us to deal with new and unknown games. Algorithms that allow us to adjust the probabilities of consulting different experts during game play are called *experts algorithms* [1]. Agent i's experts algorithm is denoted by æ$_i$ and p_i is called æ's policy.

The resulting algorithm, *adaptive-FRAME*, is shown in Algorithm 2. For correctness, we only require that

$$\sum_{t=1}^{\infty} p^t_{e_{i,0}} = \infty. \tag{6.1}$$

In words this means that the Naive Expert is consulted infinitely often. As long as Equation 6.1 holds, the correctness for adaptive-FRAME follows directly from Propositions 1 and 2. If Equation 6.1 does not hold, then by the Borel-Cantelli Lemma, there will only be a finite number of turns where all agents consult the Naive Expert. This would violate the conditions in Proposition 1.

In practice, this condition does not have to hold. In particular, our experiments were conducted using experts algorithms which did not necessarily satisfy Equation 6.1. If convergence rates are fast enough then Equation 6.1 can be relaxed. Hence, it is more of a theoretical condition than a practical one. However, to maintain theoretical correctness of adaptive-FRAME we could set a maximum rate of decay of consulting the Naive Expert, i.e.,

$$\tilde{p}^t_{e_{i,0}} = \max\{g(t), p^t_{e_{i,0}}\} \tag{6.2}$$

for any $g(t)$ such that $\sum_{t=0}^{\infty} g(t) = \infty$ (for example $g(t) = 1/(t^{0.9})$), and renormalize the other probabilities.

6.2 Experts Algorithms

In this section we briefly review some the standard experts algorithms in the literature. We also introduce our experts algorithm, designed specifically for use with

Algorithm 2 adaptive-FRAME$_i$ ($æ_i(\cdot)$, E_i, $d()$, η)

$\sigma_i^0 = \mathcal{U}(\Sigma_i)$
for $t = 0, 1, \ldots$ **do**
  • $p_i^t = æ_i(\cdot)$ is the probability distribution over the experts E_i for agent i at time t
  • β_i^{t+1} is the strategy returned by consulting $e_{i,j}(\cdot)$, where $e_{i,j}$ was determined according to p_i^t
  if β_i^{t+1} is not in the bounded region $B(\sigma_i^t, d(r(\sigma^t)))$ **then**
    • β_i^{t+1} is the strategy picked uniformly from $B(\sigma_i^t, d(r(\sigma^t))))$
  end if
  if the regret of β is less than the regret of σ^t **then**
    $\sigma^{t+1} = \beta^{t+1}$
  else
    $\sigma^{t+1} = \sigma^t$
  end if
  • τ_i is strategy picked uniformly at random from $\mathcal{U}(\Sigma_i)$
  if the regret of τ is less than half the regret of σ^{t+1} **then**
    • with probability η, set $\sigma^{t+1} = \tau$
  end if
end for

adaptive-FRAME. Experts algorithms vary in both how they measure the performance of the experts and how they adapt to the relative performance of the experts. The review of the existing experts algorithms provides a context for our algorithm.

Hedge (Section 3.4.1)

Hedge measures the performance of an expert by using regret [23]. The regret is based upon comparing the actual utility an expert's decision would have achieved versus the best utility achieved amongst all other experts. The probability of consulting each expert is proportional to some weight value for that agent, where the initial weight vector is $w_i^1 \in [0,1]^{|e_i|}$. This weight decays according to some decay factor, $\psi \in [0,1]$, raised to the expert's regret.

1. Calculate p_i^t as

$$p_i^t(e_i) = \frac{w_i^t(e_i)}{\sum_{e_i' \in E_i} w_i^t(e_i')}.$$

2. Use suggested strategy from expert selected according to p_i^t. However, all experts must still calculate a suggested strategy. For all experts, we calculate the regret, $r(\beta_{e_i}^t, \beta_{-i}^t)$, that using their suggested strategy would have given agent i.

3. Update the weights according to

$$w_i^{t+1}(e_i) = w_i^t(e_i)\psi^{r(\beta_{e_i}^t, \beta_{-i}^t)}.$$

4. Repeat.

Strategic Experts Algorithm (Section 3.4.2)

On the other hand, Strategic Experts Algorithms (SEA) measures the performance of an expert by the utility it did achieve and not by comparing the expert to any others.

1. Set $M_{e_i} = N_{e_i} = 0$. Set $t = 1$.

2. With probability $1/t$ perform an *exploration phase*, namely, choose an expert e_i uniformly at random; otherwise, perform an *exploitation phase*, namely, choose an expert e_i uniformly at random from those experts with maximum M_{e_i}.

3. Set $N_{e_i} = N_{e_i} + 1$. Follow e_i's instructions for the next N_{e_i} stages. Denote by $\tilde{R}$ the average payoff accumulated during those N_{e_i} stages, and set

$$M_{e_i} = M_{e_i} + \frac{2}{N_{e_i} + 1}(\tilde{R} - M_{e_i}). \tag{6.3}$$

4. Set $t = t + 1$ and repeat.

6.2.1 Logistic Expected Regret Reduction Maximization

Logistic Expected Regret Reduction Maximization (LERRM) is an experts algorithm created specifically for adaptive-FRAME, inspired by LFP [25]. The metric LERRM uses to measure the performance of an expert is the Expected Regret Reduction (ERR). At time T, for agent i, expert e_i's ERR is defined as,

$$ERR(e_i)_i^T = \frac{\sum_{t=0}^{T-1}(r_i(\beta^t) - r_i(\beta_{e_i}^{t+1}, \beta_{-i}^{t+1}))}{T}. \tag{6.4}$$

ERR is a measurement of how much an expert's suggested strategies could have, or did, reduce an agent's regret. Specifically, at time T, assuming that all agents other than agent i played the same strategies for $t = 0$ to $t = T$, ERR measures the average reduction in agent i's regret if agent i had always consulted expert e_i.

ERR is a better measure of what we are trying to achieve in adaptive-FRAME than an expert's regret or actual utility. Since we are interested in trying to reduce regret, it makes sense to actually measure the ability of each expert to do that.

Example: To demonstrate how ERR is calculated, consider the following example. Suppose that two agents, both using adaptive-FRAME, are playing the repeated game of Battle of the Sexes as shown in Figure 6.1.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	1, 0.5	0, 0
	$a_{1,2}$	0, 0	0.5, 1

Figure 6.1: An example for calculating ERR

Suppose that agent 1 has two possible experts to consult each turn, $\{e_{1,1}, e_{1,2}\}$. We wish to calculate the ERR value for both of these experts at time $t = 3$. Table 6.1 shows the suggested strategies by both experts over the time $t = 0$ to $t = 3$ as well as which expert agent 1 actually consulted each turn. Following adaptive-FRAME, both agents choose their inital strategies uniformly at random instead of consulting an expert. Therefore, $\beta^0_{e_{1,j}}$ is not defined.

t	0	1	2	3
$\beta^t_{e_{1,1}}$	-	(1,0)	(1,0)	(1,0)
$\beta^t_{e_{1,2}}$	-	(0.75,0.25)	(0.8,0.2)	(0.7,0.3)
Expert consulted	-	$e_{1,1}$	$e_{1,1}$	$e_{1,2}$
β^t_1	(1,0)	(1,0)	(1,0)	(0.7,0.3)

Table 6.1: An example of calculating ERR continued

Suppose that agent 2's strategy over the same period is given by Table 6.2.

We first calculate expert $e_{1,1}$'s ERR. It does not matter that expert $e_{1,1}$ was not always consulted. What we are interested in is what would have happened if expert $e_{1,1}$ was always consulted, assuming that this would not have changed any of

t	1	2	3	4
β_2^t	(0,1)	(0.1,0.9)	(0.5,0.5)	(0.6,0.4)

Table 6.2: An example of calculating ERR continued

agent 2's strategies. To calculate expert $e_{1,1}$'s ERR, we start by noting that at time $t = 0$, agent 1's regret was 0.5. At time $t = 1$, if agent 1 had gone with the strategy suggested by expert $e_{1,1}$ (which it actually did), agent 1's regret would have become 0.35. Thus by consulting expert $e_{1,1}$ for σ_1^1, agent 1 would have reduced its regret by 0.15. Similarly, by consulting expert $e_{1,1}$ for σ_1^2, agent 1 would have reduced its regret by 0.35 since $\beta_{e_{1,1}}^2$ was an optimal strategy. Finally, the strategy $\beta_{e_{1,1}}^3$ would have reduced agent 1's regret by 0 since both β_1^2 and $\beta_{e_{1,1}}^3$ were optimal strategies. Thus we can calculate expert $e_{1,1}$'s ERR as

$$\begin{aligned} ERR(e_{1,1})_1^3 &= \frac{0.15 + 0.35 + 0}{3}, \\ &= \frac{1}{6}. \end{aligned}$$

Through similar reasoning, we can show that $ERR(e_{1,2})_1^3 = 0.1225$.

If ERR was a perfect measure of an expert's ability to reduce regret, it would make sense to simply consult the agent with the highest ERR. However, at any given time, ERR is only an estimation. Hence, it might be that the agent with the highest ERR is not actually the optimal expert to consult. Furthermore, we want to ensure that there is always a positive probability of consulting the naive expert. Thus, we would like to use ERR to determine some probability of consulting each expert. This is exactly what LERRM does.

$$LERRM(e_i)_i^t = \frac{e^{\frac{1}{\lambda}ERR(e_i)_i^t}}{\sum_{e_i' \in E_i} e^{\frac{1}{\lambda}ERR(e_i')_i^t}}. \tag{6.5}$$

As in LFP, λ is a measure of smoothness. Thus LERRM can serve as a balance between using the expert with the highest ERR and considering other experts.

Example: Continuing the example for calculating ERR, we can use these values for LERRM. Supposing that $\lambda = 1$, at $t = 4$, LERRM will consult expert $e_{1,1}$ with a probability of .51104 and expert $e_{1,2}$ with a probability of .48896.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	0.54, 0.54, 0.54	0.54, 1, 0.54
	$a_{1,2}$	1, 0.54, 0.54	0.46, 0.46,0

Agent 3 - $a_{3,1}$

	Agent 2	
	$a_{2,1}$	$a_{2,2}$
$a_{1,1}$	0.54,0.54,1	0, 0.46, 0.46
$a_{1,2}$	0.46,0,0.46	0.46, 0.46, 0

Agent 3 - $a_{3,2}$

Figure 6.2: 3-player Chicken: agent 1 chooses the row, agent 2 chooses the columm, and agent 3 chooses the matrix

6.3 Experimental Setup

The games used in the experiments were Battle of the Sexes (Figure 5.1), Shapley's Game (Figure 5.2) and 3-player Chicken (Figure 6.2). These games were chosen to best illustrate the adaptive aspect of adaptive-FRAME. For each of these games, there are obvious optimal experts. As shown in Table 6.3, WoLF is the best expert for BoS, LFP is the best expert for Shapley's game and HMC is the best expert for 3-player Chicken (shown in Figure 6.2). Thus the performance of the different experts algorithms in being able to determine the optimal expert should be easy to measure. As well, Shapley's Game and 3-player Chicken are hard games to learn, so these results will help to reinforce the practicality of adaptive-FRAME.

The same experts were used in testing adaptive-FRAME that were used for testing FRAME as well as HMC from Section 3.1.2. All experts were run with the same parameters used in testing FRAME. For experts algorithms, we used NEA as a basis for comparison. We used all three of the experts algorithms mentioned in the previous section; Hedge, SEA and LERRM. LERRM was run with $\lambda = 0.00005$ and Hedge was run with $\beta = 0.00005$. These values were chosen experimentally.

Since adaptive-FRAME is a random process, there will always be a few exceptionally long runs. These runs are not overly representative of the adaptive-FRAME process. Furthermore, showing these results in graphs often forces a loss of detail in the important regions. Hence, when necessary, results are shown for the 98^{th} percentile.

For comparison purposes we first tested each expert on its own without the use of FRAME. These convergence rates are presented in Table 6.3.[1]

[1]DNC = does not converge. NT = not tested.

Game	Number of Iterations to Convergence		
	LFP	WoLF	HMC (average)
BoS	DNC	3509	NT
Shapley's Game	14	DNC	NT
3-player Chicken	DNC	64	< 10

Table 6.3: Convergence rates for each expert without the use of FRAME.

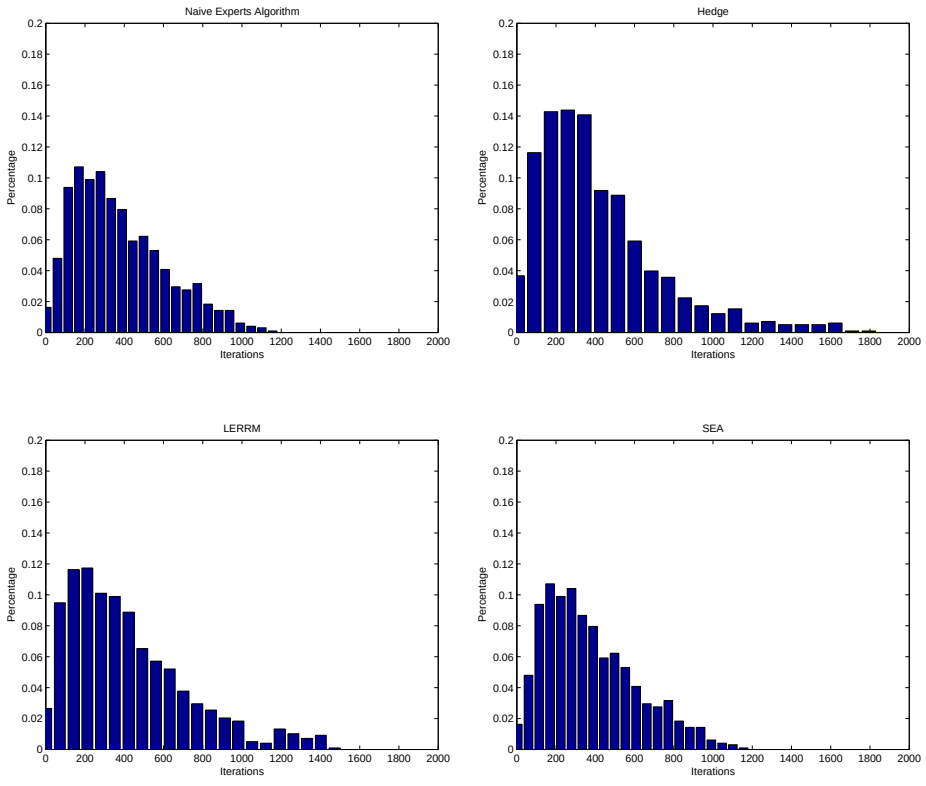

Figure 6.3: Convergence rates for BoS using adaptive-FRAME. Results are given for the 98th percentile.

6.3.1 Battle of the Sexes

BoS was tested using WoLF and LFP as experts. Both WoLF and the Naive Expert do reasonably well by themselves, as shown in Table 6.3 and Figure 5.6, respectively.

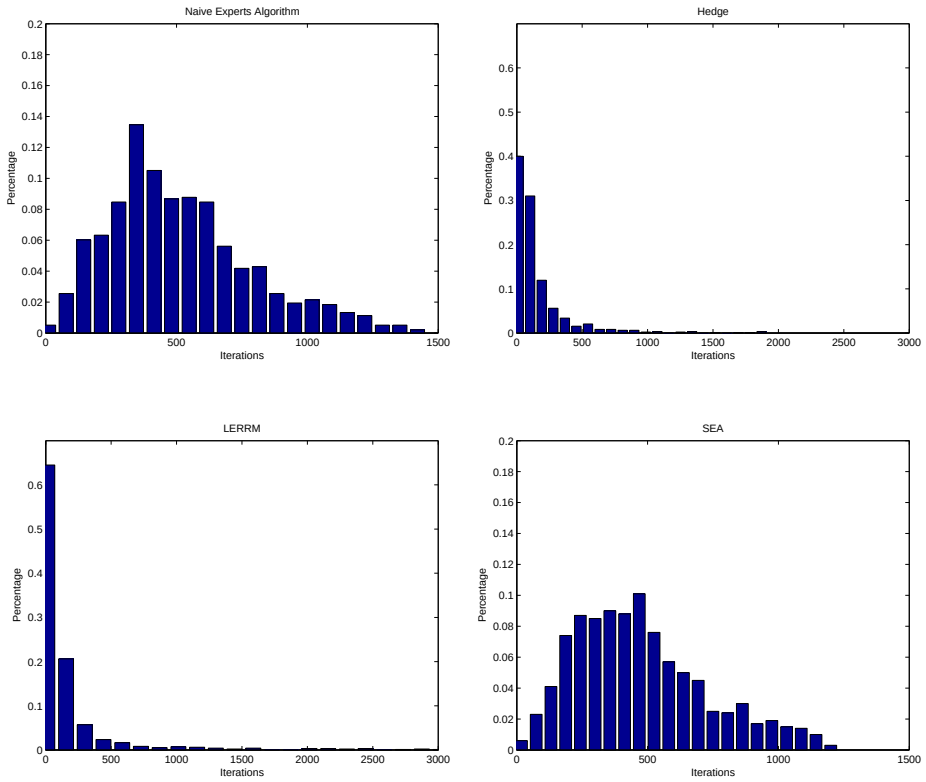

Figure 6.4: Convergence Rates for Shapley's Game using adaptive-FRAME. Note the difference in scale. Results are given for the 98th percentile.

However, as shown in Table 6.3, LFP does not achieve convergence at a practical rate. The results in Figure 6.3 show that all of the experts algorithms are able to outperform the worst expert. These confirm the idea that having multiple experts makes agents more flexible and provides protection against poor experts.

However, Figure 6.3 also shows that NEA does basically as well as the other experts algorithms. NEA is able to perform that well simply because BoS is such a simple game. Since Hedge, LERRM and SEA still outperform the worst expert, this suggests that for simple games there may not be much benefit to using a more sophisticated approach than NEA but there is also no harm in doing so.

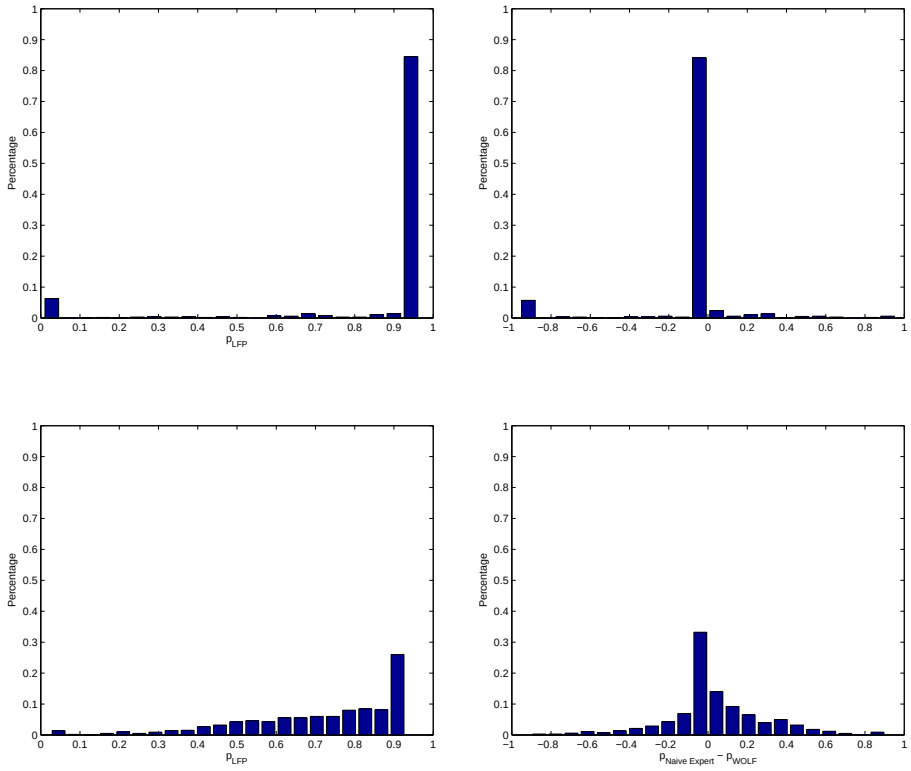

Figure 6.5: Expert usage statistics for Shapley's Game

6.3.2 Shapley's Game

Shapley's Game was tested using LFP and WoLF as experts. Figure 6.4 shows that all of the experts algorithms do much better than WoLF, which is the worst expert for Shapley's Game as shown in Table 6.3. Hedge and LERRM give, on average, much faster convergence rates compared to the NEA. In particular LERRM performs very well. SEA does only about as well as NEA. However, the range of results is much larger for Hedge and LERRM. One possible explanation is that Hedge and LERRM are both very sensitive to initial conditions; having the first few rounds be exceptional cases could throw both of these algorithms off. On the other hand, SEA's performance suggests that it is either poorly suited for use in adaptive-FRAME or convergence happens so quickly that SEA does not have enough time to adapt to consulting the optimal agent.

How are Hedge and LERRM able to achieve this performance? Since LFP

gives the fastest convergence rate, Hedge and LERRM should consult LFP with a very high probability. The left column in Figure 6.5 shows the probability Hedge and LERRM have, respectively, of consulting LFP at the time of convergence. These results show that both experts algorithms, on average, do consult LFP with a very high probability. For Shapley's Game, the other experts are, practically speaking, equally inefficient. Therefore, we would expect both experts algorithms to consult the Naive Expert and WoLF with roughly equal probability. The right column of Figure 6.5 shows the probability of consulting the Naive Expert minus the probability of consulting WoLF at the point of convergence for Hedge and LERRM, respectively. These results show that in fact, on average, there is no major difference in the probability of consulting the two experts. Thus, we are able to see that Hedge and LERRM are able to adapt so they consult the most appropriate expert for the game. LERRM's superior performance can be attributed to it adapting so that it places most of its weight on LFP.

However, we see that both Hedge and LERRM occasionally preform very poorly. Specifically, while the slowest convergence for SEA was 2021 iterations, Hedge and LERRM's slowest convergence was 12012 and 16848 iterations, respectively. Roughly 3% and 2% of LERRM and Hedge's trials took longer than 2021 iterations respectively. While this is a noticeable number, Hedge and LERRM preform well enough on average that these exceptional cases do not have a noticeable impact. To understand these cases, note that as shown in Figure 6.5, both Hedge and LERRM will very occasionally wind up consulting LFP with a very low probability. The problem is that both Hedge and LERRM adapt quickly enough so that they are very sensitive to the results from the first few iterations. Given the random nature of adaptive-FRAME, it is not surprising that these iterations are not always representative of the true state of the game. When this is the case, Hedge and LERRM can wind up with an incorrect idea of which experts are optimal to consult. However, even in these exceptional cases adaptive-FRAME is still achieving convergence. On the other hand, since SEA is so slow to adapt, it does not suffer from this problem.

6.3.3 3-Player Chicken Game

We present two sets of results for 3-player Chicken. The first set of results, shown in Figure 6.6, is with just WoLF and LFP as experts. These results show all of the experts algorithms easily outperforming the worst expert for 3-player Chicken, LFP, as shown in Table 6.3. One of the major differences between 3-player Chicken and Shapley's Game is that SEA can do much better than NEA. This indicates

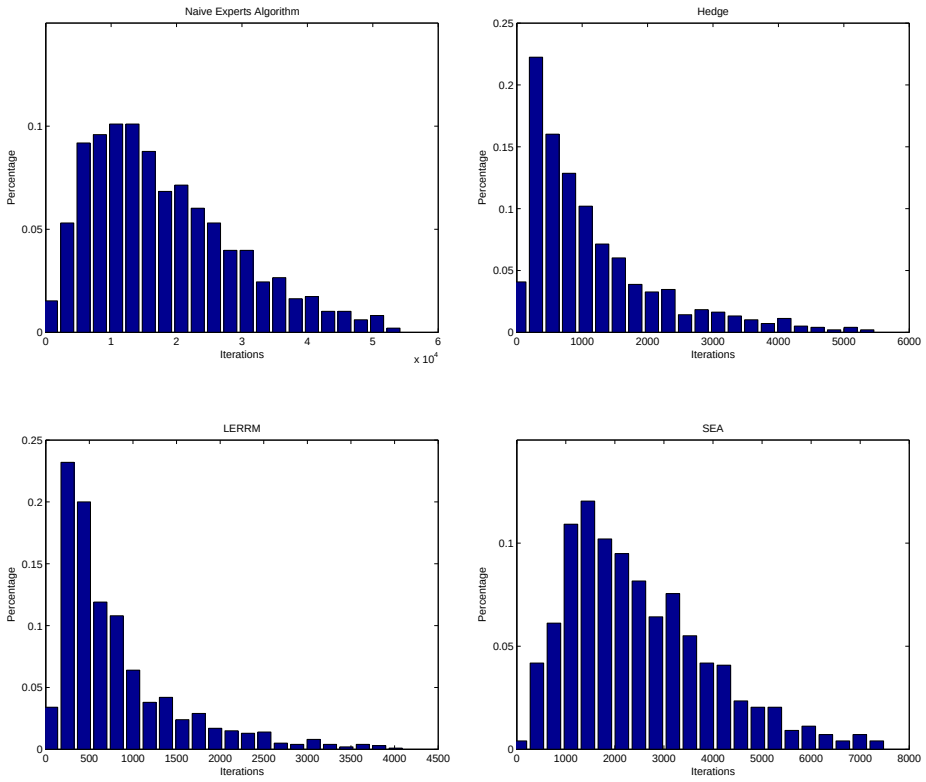

Figure 6.6: Results for 3-Player Chicken with $E_i = \{NaiveExpert, LFP, WoLF\}$. Note the difference in scale.

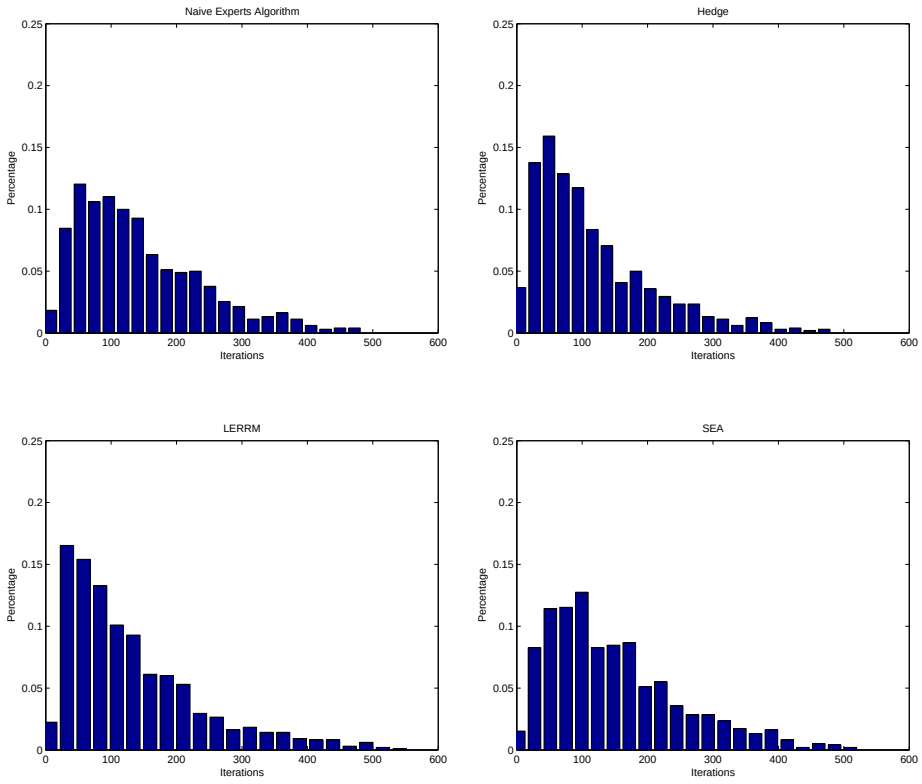

Figure 6.7: Results for 3-Player Chicken with $E_i = \{NaiveExpert, LFP, WoLF, HMC\}$.

that SEA is able to learn which is the optimal expert to consult; it just takes longer to do so than Hedge or LERRM. The second set of results, shown in Figure 6.7, is with WoLF, LFP and HMC as experts. HMC is by far the best expert for 3-Player Chicken, hence we would hope to see a noticeable improvement in the convergence rate. However, it might be possible that with an additional expert, it would take longer for the experts algorithms to find the optimal expert.

How are Hedge and LERRM able to outperform NEA? The left column of Figure 6.8 shows the probability of consulting WoLF in the 3-player Chicken Game. This column shows that both Hedge and LERRM consult WoLF with a very high probability. The right column of Figure 6.8 shows the difference in probability between consulting the Naive Expert and LFP. Since neither expert is very useful, we do not expect to see much difference in how much they are consulted. This is indeed what we see, therefore, we can conclude that Hedge and LERRM perform

well since they both adapt so that they consult the best expert for the game. This time, both Hedge and LERRM consult the best expert with roughly equal probability, which explains their similar results.

As with Shapley's game, the results for 3-player Chicken show that both Hedge and LERRM occasionally have very slow convergence rates. The same analysis applies here.

6.4 Conclusion

In this chapter we introduced the idea of adaptive-FRAME. The goal of adaptive-FRAME is to keep the theoretical guarantees of FRAME while allowing agents to further improve convergence rates and be more flexible. This goal was achieved by allowing agents to consult multiple experts and to do so in an adaptive manner. The use of experts algorithms allows agents to dynamically adapt to the expert best suited for the current game.

We presented results using a number of experts and experts algorithms. This included an experts algorithm, LERRM, we specifically designed for adaptive-FRAME. Our results showed that the use of experts algorithms can give a definite improvement in the convergence rates. As well, LERRM was shown to be competitive with existing standard experts algorithms, and in some cases can even outperform them.

However, we see that both Hedge and LERRM occasionally preform very poorly. Specifically, while the slowest convergence for SEA was 2021 iterations, Hedge and LERRM's slowest convergence was 12012 and 16848 iterations, respectively. Roughly 3% and 2% of LERRM and Hedge's trials took longer than 2021 iterations respectively. While this is a noticeable number, Hedge and LERRM preform well enough on average that these exceptional cases do not have a noticeable impact. To understand these cases, note that as shown in Figure 6.5, both Hedge and LERRM will very occasionally wind up consulting LFP with a very low probability. The problem is that both Hedge and LERRM adaptive quickly enough that they are very sensitive to the results from the first few iterations. Given the random nature of adaptive-FRAME, it is not surprising that these iterations are not always representative of the true state of the game. When this is the case, Hedge and LERRM can wind up with an incorrect idea of which experts are optimal to consult.

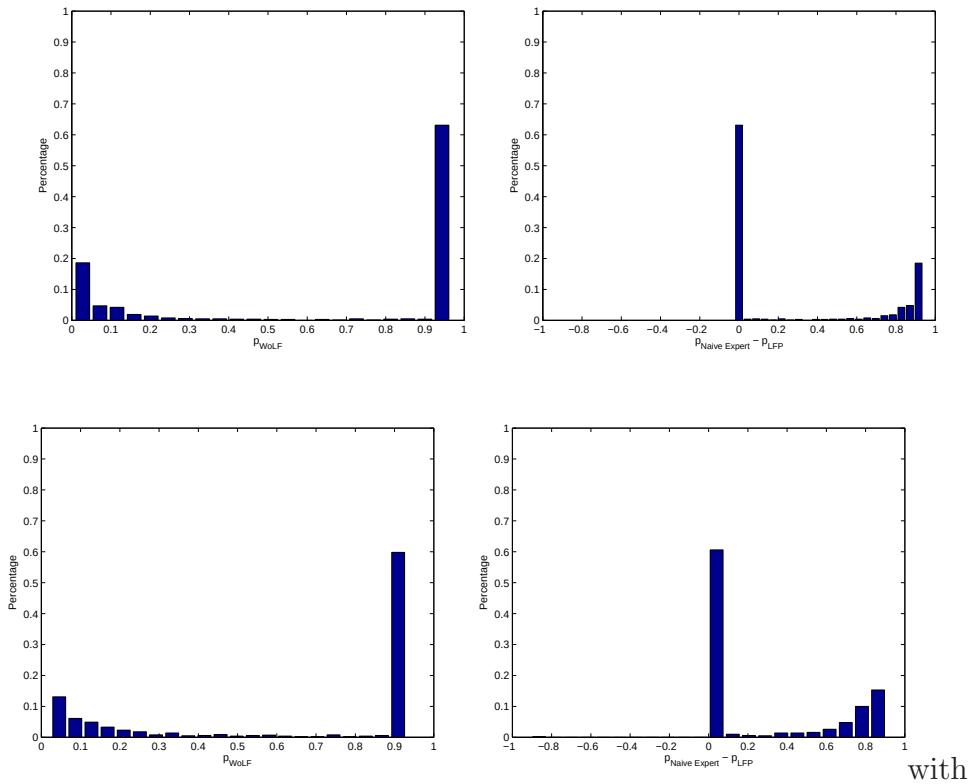

Figure 6.8: Expert usage statistics for 3-Player Chicken with $E_i = \{NaiveExpert, LFP, WoLF\}$.

Chapter 7

Conclusion

This thesis set out to study multiagent learning in repeated games. Our interest was in what Shoham *et. al.* call the prescriptive, non-cooperative agenda [49]. In particular, we were interested in studying how agents could maximize their utility when facing similar agents (i.e. self-play). Specifically, this thesis was interested in developing a bridge between theoretical guarantees of convergence and real-world performance.

Recent algorithms had been able to achieve convergence to Nash equilibria in nearly all games. However, these algorithms could not be used in practice. The goal of this thesis was to create new algorithms which could be used in practice by computers.

7.1 Contributions

The main contributions of this thesis are:

- FRAME: FRAME is a multiagent learning algorithm which uses a greedy method for selecting new strategies that are closer to equilibrium than the current strategy. FRAME also introduces the idea of allowing agents to consult an expert for possible new strategies. When these experts provide good suggestions, the convergence rate improves. Agents are protected, however, against experts who make poor suggestions.
- In Chapter 4, we introduced FRAME and proved the main theoretical results about it. Specifically, FRAME is able to achieve convergence to the set of

Nash equilibria for all games. Furthermore, for a large number of games FRAME is able to achieve convergence to a single Nash equilibrium.

- In Chapter 5, we presented our experimental results. These results confirmed that FRAME is a practical algorithm. FRAME was even able to achieve reasonable convergence rates on games generally considered to be difficult. The results showed how experts could be beneficial for agents and how the agents were still able to achieve convergence with poor experts.

- Adaptive-FRAME: adaptive-FRAME built upon FRAME by allowing agents to consult multiple experts and to do so in a dynamic manner. Hence, agents were able to adapt so that they consulted the expert best suited for a given game. This provided agents with a greater degree of flexibility in dealing with new games. Adaptive-FRAME also showed a noticeable improvement over FRAME in the convergence rates in general.

- Adaptive-FRAME introduced the idea of using experts algorithms in a multiagent setting. Experts algorithms are a common idea in single agent learning but have only been applied to multiagent learning in a very limited setting [24]. We studied the use of two common experts algorithms.

- LERRM: In Chapter 6 we also introduced our own experts algorithm, LERRM. We showed that LERRM was competitive with the existing experts algorithms and at times could even outperform them.

7.2 Directions for Future Work

This work opens up new interesting directions for future work. In this section we outline these directions.

7.2.1 Examining Different Experts and Experts Algorithms

The experiments with adaptive-FRAME showed that agents were able to take advantage of additional experts. Thus having more experts would help agents improve their performance on the games used in the thesis, and deal better with new games. There are several experts that we did not study in this thesis which could be useful, such as Fictitious Play and GIGA-WoLF [7]. The performance of two of the experts we used, LFP and WoLF, are dependent on the parameters they are given;

adaptive-FRAME could be expanded to work with multiple sets of parameters for different experts. It might also be possible to learn online different values that work well for these experts.

Although our experts algorithms were able to adapt to the optimal expert for a given game there is always room for improvement. Our experts algorithm LERRM presents hope that an experts algorithm specifically designed for adaptive-FRAME could have a noticeable advantage over traditional experts algorithms. Analyzing the experimental results also highlighted some roadblocks to better experts algorithm performance. For example, experts algorithms were very sensitive to results during the initial part of the game. By random chance these initial results were not always reflective of the actual situation, and the experts algorithms might benefit from being a little more conservative initially. This could potentially be very advantageous in more challenging games.

7.2.2 Using Different Notions of Regret

Although FRAME and adaptive-FRAME were able to achieve practical convergence rates on many games, there are many more games for which we do not expect to obtain such rates. In particular, the joint strategy space for games with more than 3 players is large enough that FRAME and adaptive-FRAME may never work for them. Recent results by Chen and Deng suggest that finding Nash equilibria in general, let alone learning, is not likely to be done efficiently (e.g. in polynomial time) time [11, 10].

However, there are alternative notions of convergence that might be used instead. In particular, the notion of correlated equilibria is promising. Recent results have shown that in many cases finding a particular correlated equilibrium is relatively easy [43]. Furthermore, correlated equilibria open up the possibility of more cooperation between agents ,and should allow for more mutually beneficial outcomes. One way of achieving convergence to the set of correlated equilibria is through the use of no-internal regret algorithms. Hence, a different notion of regret could be useful.

7.2.3 Adapting FRAME to Stochastic Games

A stochastic game is made up of multiple stage games (also known as states). For all possible joint action, each state provides not only the utilities for each agent but also a transition to the next state. Stochastic games are also known

as competitive-MDPs and can be thought as of multiagent MDPs [18]. Stochastic games can be used to model many different situations in real life, from economics to robotics. However, there is currently no known method for achieving convergence to Nash equilibrium in stochastic games. Creating such an algorithm would not only have benefits in the theoretical domain but could be very useful in different sorts of modeling. We are currently in the process of adapting FRAME to work in stochastic games.

There are several challenges for stochastic games. First of all, the idea of regret has not been as well established in stochastic games. Secondly, decreasing regret in one state may cause an increase in regret in another state.

7.3 Summary

It was the goal of this thesis to help build a bridge between theoretical and practical agendas in multiagent learning. Our algorithm, FRAME, does this by creating a balance between the two. However, as with all balancing acts, compromises had to be made. It is unlikely that there will ever exist some "silver-bullet" multiagent learning algorithm, thus it is hoped that FRAME can serve as a means rather than an end. Ideally, FRAME will serve as a means to understanding the conflicting goals in multiagent learning and examining different means of addressing them.

Appendix A

Measure Theory

This Appendix provides a background on the measure theory used in this thesis. Specifically, measure theory is required for the main Propositions regarding FRAME.

An essential step in proving FRAME's correctness is to examine what happens when a strategy is selected uniformly at random from Σ or some subset of it. We are unable to use basic probability theory since it only deals with with probabilities involving discrete sample spaces or very basic situations involving continuous sample spaces. Instead we must use a generalization of probability theory called measure theory.

Specifically, this Appendix proves Lemmas 3 and 4 from Chapter 4. Lemma 4 is proved first since its proof provides an introduction to measure theory. A more thorough introduction is given by Rosenthal[46].

Lemma 4 *Given σ such that $r(\sigma) > 0$, there is a positive probability of picking a joint strategy $\sigma' \in \Sigma$ uniformly at random such that $r(\sigma') \leq r(\sigma)/2$.*

Proof:

We start by defining a *probability measure space* as the triple $(\Sigma, \mathcal{F}, P)$:[46]

- The joint strategy space Σ is also our sample space.
- The σ-algebra $\mathcal{F}$ is a collection of subsets of Σ such that:
  - The sets Σ and $\emptyset$, the empty set, are both contained in $\mathcal{F}$.
  - $\mathcal{F}$ is closed under complements and countable unions and intersections.

- The measure probability P which is a mapping from $\mathcal{F}$ to $\mathbb{R}$ such that:
  - $0 \leq P(A) \leq 1$ for all $A \in \mathcal{F}$.
  - If $A_1, A_2, \ldots$ are a countably infinite number of subsets of Σ then

$$P(A_1 \cup A_2 \cup \ldots) = P(A_1) + P(A_2 \backslash A_1) + P(A_3 \backslash A_2 \backslash A_1) + \ldots \quad \text{(A.1)}$$

  - $P(\emptyset) = 0$ and $P(\Sigma) = 1$.

  Note that $P(X)$ is defined if and only if $X \in \mathcal{F}$.

Although we are free to choose any σ-algebra, for simplicity we chose one based on L_∞-balls. A L_∞-ball, $D_\infty(\gamma, \epsilon)$, is a hypercube with a center at $\gamma \in \Sigma$ and a width of $2\epsilon \geq 0$. Let $\mathcal{J}$ denote the set of all possible L_∞-balls inside of Σ. Our σ-algebra will be $\mathcal{B} = \sigma(\mathcal{J})$, also known as the Borel set, which is the smallest σ-algebra containing all elements of $\mathcal{J}$.

Finally we must define $P(X)$ for all $X \subseteq \mathcal{B}$.To do this, we will rely on another measure, the *Lebesgue measure* or μ. The Lebesgue measure may be thought of as an extension of volume to a higher dimension. Like P, μ is a mapping from $\mathcal{F}$ to $\mathbb{R}$. However, we do not require that $\mu(\Sigma) = 1$.

We are now define $P(X)$ as

$$P(X) = \frac{\mu(X)}{\mu(\Sigma)}, \quad \text{(A.2)}$$

assuming that $X \in \mathcal{B}$. This definition meets all the requirements for a measure probability and leads to the intuitive idea of what a probability should be.

In the case of this lemma, we are interested in finding $P(\mathcal{N}_\epsilon)$ for some $\epsilon > 0$. In order to find $P(\mathcal{N}_\epsilon)$, we must find $\mu(\mathcal{N}_\epsilon)$ and $\mu(\Sigma)$. Since Σ is a solid region, $\mu(\Sigma)$ has a positive value. (Finding the exact value is unnecessary, the important part is that it is positive.) However we can not find a minimum value for $\mu(\mathcal{N}_\epsilon)$ since it may not exist [46]. In other words, there are subsets of Σ that do not have a measure defined for them. Although these cases are rare in stage games, for completeness we now consider how to deal with them [30].

Since the measure of a set X is defined for all elements in $\mathcal{F}$, if some X does not have a measure defined for it, this means that $X \notin \mathcal{F}$. How can this happen? Returning to the definition of $\mathcal{F}$, we see that if A_1 and A_2 are both in $\mathcal{F}$ then so is $A_1 \cup A_2$, and furthermore $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2 \backslash A_1)$. In the context of our definition of $\mathcal{F}$ for the joint strategy space, if A_1 and A_2 are both L_∞-balls

(i.e. both A_1 and A_2 are in $\mathcal{F}$)) then $A_1 \cup A_2 \in \mathcal{F}$ and furthermore, $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2 \backslash A_1)$. We can expand on this inductively by adding in as many A_i's as we want. In fact, as long as we have a countably infinite number of A_i's all of this will still hold.

The problem arises when we have an uncountably infinite number of A_i's. The difference between countably and uncountably infinite is vital. A set of infinite numbers is countably infinite if they can be enumerated. For example, the set of positive integers is countably infinite because you could start listing off all of them and every positive integer would eventually be included in your list. The same goes for rational numbers. More thought has to be put into your list but there is a way of listing off all the rational numbers such that every one is eventually included. The cardinality of these sorts of sets is $\aleph_0$. For sets that are uncountably infinite there is no way of enumerating them. For example the real numbers are uncountably infinite. No matter what method you try to enumerate them with, there will always be numbers that are never included in your list. The cardinality of reals and similar sets is $\aleph_1$.

The importance of all of this is that $\mathcal{F}$ is not closed under an uncountable number of unions. This means that if X is the union of an uncountably infinite number of L_∞-balls, then its measure may not be defined. To examine this in the context of this thesis, we make the following definition. For a Nash equilibrium $\sigma^{\mathcal{N}_i}$, let $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$ denote the region in Σ that is an ϵ-Nash equilibrium with respect to $\sigma^{\mathcal{N}_i}$. Therefore

$$\mathcal{N}_\epsilon = \cup_{\sigma^{\mathcal{N}_i} \in \mathcal{N}} \mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}). \tag{A.3}$$

Since each $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$ is a subregion of Σ, $\mu(\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}))$ is positive for all i. Therefore

$$\begin{aligned} \mu(\mathcal{N}_\epsilon) &= \sum_{\sigma^{\mathcal{N}_i} \in \mathcal{N}} \mu(\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})), \\ &> 0, \end{aligned} \tag{A.4}$$

as long as $\mathcal{N}$ is at most countably infinite. Under these circumstances, $P(\mathcal{N}_\epsilon)$ is always defined and positive. In fact, Germano and Lugosi approach this problem by basically assuming that there are only a finite number of Nash equilibria [30].

Thus, the problem only arises when there are an uncountably infinite number of Nash equilibria. An example of this is shown in Figure A.1. This game is not actually a problem since every joint strategy is a Nash equilibrium, however it is possible to create more complex games which are. To deal with these problem games we simply consider a single Nash equilibrium, $\sigma^{\mathcal{N}_i}$, out of all possible ones. Since $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}) \subseteq \mathcal{N}_\epsilon$, the probability of randomly picking a strategy that is in $\mathcal{N}_\epsilon$ is

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	1,1	1,1
	$a_{1,2}$	1,1	1,1

Figure A.1: A simple game with an uncountably infinite number of Nash equilibria.

at least as high as the probability of picking a strategy in $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$, which is positive. Thus we simply define $P(\mathcal{N}_\epsilon)$ to be positive in this case as well.

To complete this proof we simply pick some $\epsilon < r(\sigma)/2$. □

Next, we prove Lemma 3 from Chapter 4.

Lemma 3 *For a given ϵ-Nash equilibrium σ, let $f_\sigma(\tilde{\sigma}) : \mathbb{R}^{N|A|} \rightarrow \mathbb{R}$ be the change in regret from moving from the strategy σ to the new strategy $\tilde{\sigma}$, i.e.,*

$$f_\sigma(\tilde{\sigma}) = r(\sigma) - r(\tilde{\sigma}). \tag{A.5}$$

If there is some strategy σ', such that $f_\sigma(\sigma') > 0$, and $||\sigma' - \sigma|| < d(\epsilon)$ then there exists some region $Y \subseteq \Sigma$ such that

$$P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0, \tag{A.6}$$

and furthermore, for all $\sigma'' \in Y$, $f_\sigma(\sigma'') > 0$. In words, if there is at least one strategy σ' within a bounded region around σ which has less regret, then there is a positive probability of picking a strategy uniformly at random from that bounded region that has regret less than σ. Furthermore, this region includes σ'.

Proof: Note that f is continuous (since r is also continuous). Thus by definition of continuity, for every $\tilde{\sigma}$ and every $\delta > 0$ there exists an $\epsilon > 0$ such that if the distance from $\tilde{\sigma}$ to $\tilde{\sigma}'$ is less then ϵ than the distance between $f(\tilde{\sigma})$ and $f(\tilde{\sigma}')$ is less than δ.

Let $\delta = f(\sigma')/3$. By continuity, there exists some ϵ such that if σ'' is within an ϵ-ball of π',

$$f(\sigma') - \delta < f(\sigma'') < f(\sigma') + \delta. \tag{A.7}$$

Considering the first half of the inequality A.7, and substituting in $f(\sigma')/3$ for δ, we get

$$f(\sigma') - \frac{1}{3}f(\sigma') = \frac{2}{3}f(\sigma') < f(\sigma''). \tag{A.8}$$

Now since $f(\sigma') > 0$, $f(\sigma'') > 0$. Since the ϵ-ball has a positive measure, we have found a region of positive measure around σ' where equation 4.11 is positive.

Therefore, by definition of positive measure, $P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0$. $\square$

Appendix B

Additional Results

The results included in this appendix are included for completeness.

We consider two additional games, shown in Figures B.1 and B.2.

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	0.5,0.5	0,1
	$a_{1,2}$	1,0	0.1,0.1

Figure B.1: Prisoners' Dilemma

		Agent 2	
		$a_{2,1}$	$a_{2,2}$
Agent 1	$a_{1,1}$	1,-1	-1,1
	$a_{1,2}$	-1,1	1,-1

Figure B.2: Matching Pennies

Prisoner's Dilemma is a simple game with only one Nash equilibrium, $\{(0,1),(0,1)\}$. As a result, the Naive Expert (e.g. purely random updates to the strategy) is a reasonably effective method, as shown in Figure B.3. Both WoLF and LFP are also reasonable efficient experts for this game, as in Figures B.4 and B.5, respectively.

In fact, LFP and the Naive Expert performance equally as well as shown by the equal performance for all the results in Figure B.5. On the other hand, WoLF by

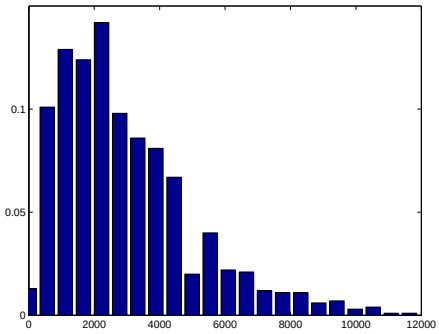

Figure B.3: Convergence rates for Prisoners' Dilemma using FRAME with a purely random learning algorithm.

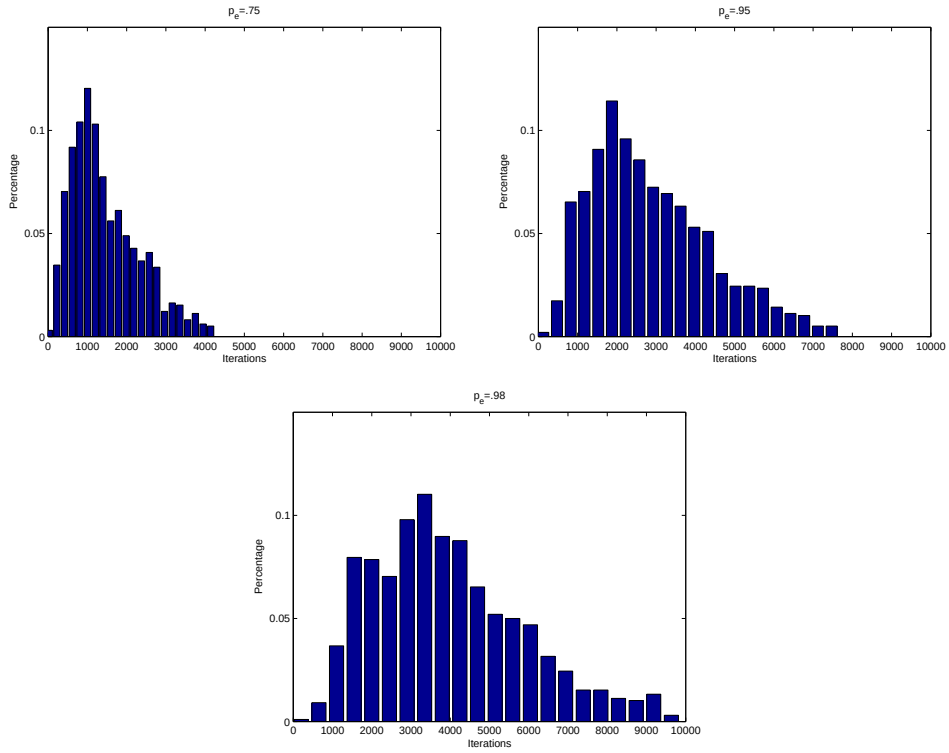

Figure B.4: Convergence rates for Prisoners Dilemma using FRAME with WoLF. Note data is presented to the 98^{th} percentile. Also note that the difference in scale compared with Figre B.3.

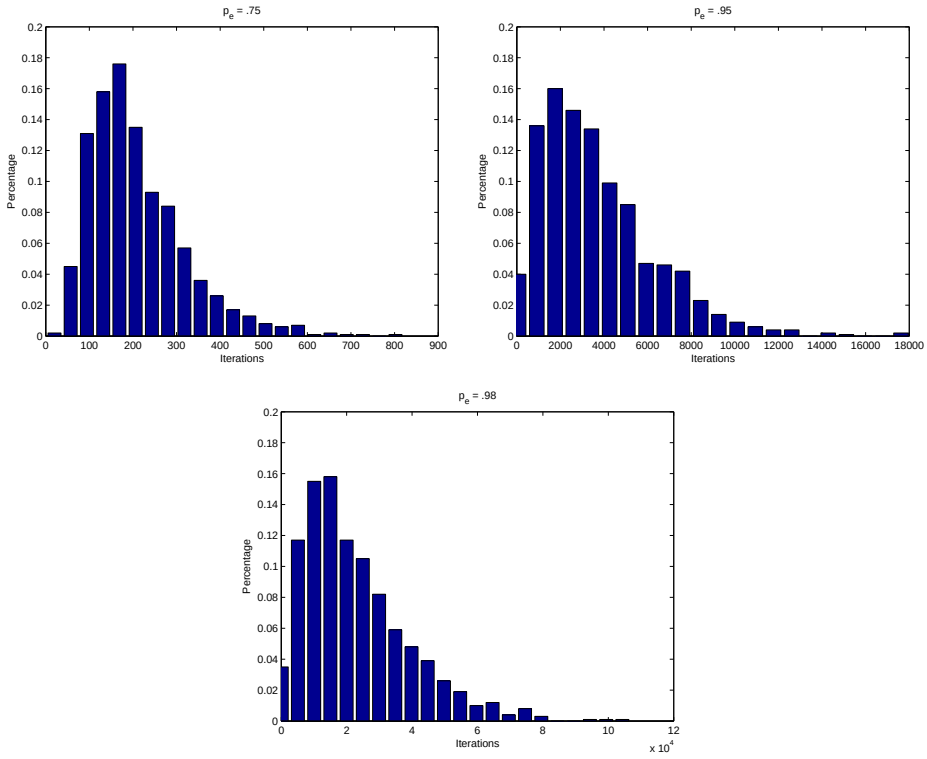

Figure B.5: Convergence rates for Prisoners' Dilemma using FRAME with LFP.

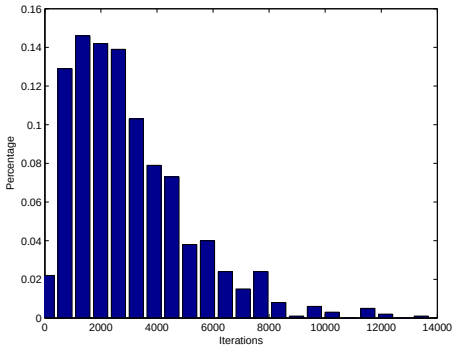

Figure B.6: Convergence rates for Matching Pennies using a purely random learning algorithm.

itself does not do as well as the Naive Expert. However, it is interesting to note that when WoLF is consulted 75 percent of the time, the convergence rate is better than for either of the experts by themselves. This lends evidence to the ideas that the Naive Expert has some practical use and that there is value from simply combining experts.

Matching Pennies is a slightly more complex game with a single mixed Nash-equilibrium of $\{(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2})\}$. Again the Naive Expert is able to achieve a reasonable convergence rate. WoLF's performance suffers because WoLF has a harder time converging to a mixed Nash equilibrium than a pure one. However, the convergence rate is relatively unaffected by the value of p_e. Although the convergence rate is decreasing as p_e increases, the rate of change is small. This supports the idea that p_e can be very close to 1 and the Naive Expert can still have a definite impact. On the other hand, p_e has very little effect on LFP.

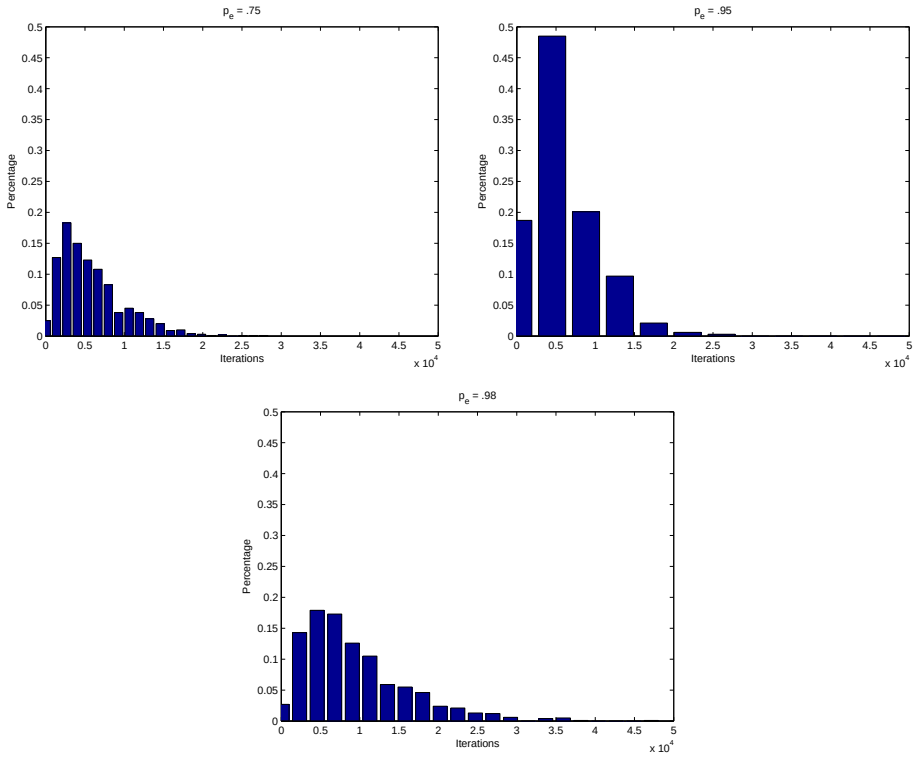

Figure B.7: Convergence rates for Matching Pennies using FRAME with WoLF.

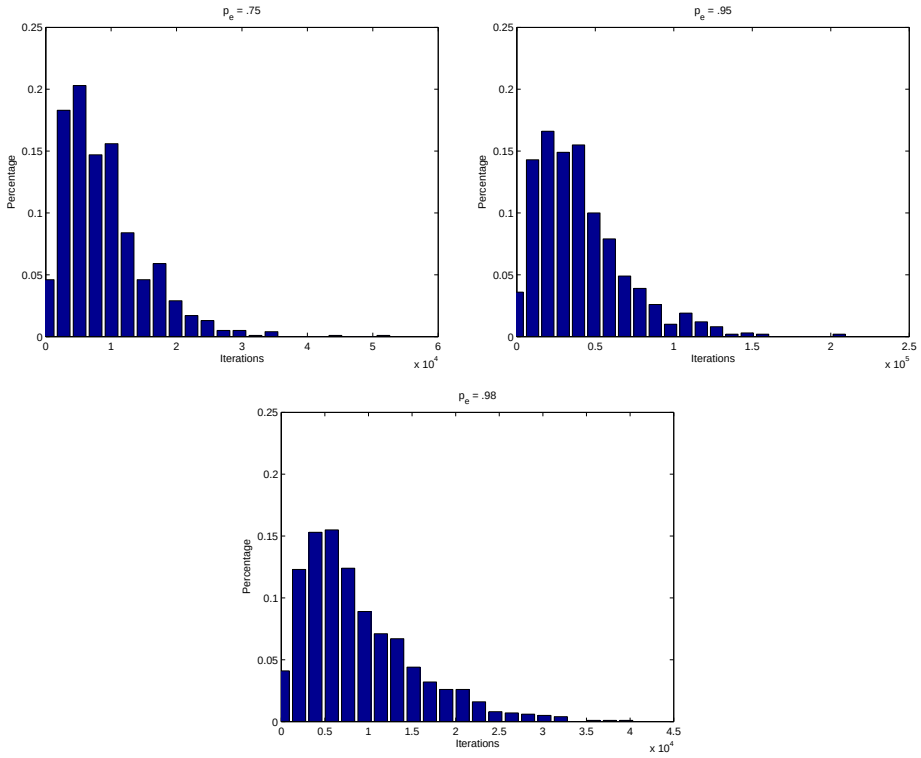

Figure B.8: Convergence rates for Matching Pennies using FRAME with LFP. Note the difference in scale.

Bibliography

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 322–331, 1995.

[2] Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

[3] Bikramjit Banerjee and Jing Peng. Performance bounded reinforcement learning in strategic interactions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 2–7, San Jose, CA, USA, 2004.

[4] Bikramjit Banerjee and Jing Peng. $RV_{\sigma(t)}$: A unifying approach to performance and convergence in online multiagent learning. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems. (AAMAS)*, pages 2–7, Hakodate, Japan, 2006.

[5] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.

[6] Jennifer Boger, Jesse Hoey, Pascal Poupart, Craig Boutilier, Geoff Fernie, and Alex Mihailidis. A planning system based on Markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology and Biomedicine*, 10(2):323–333, 2006.

[7] Michael Bowling. Convergence and no-regret in multiagent learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 209–216, Vancouver, Canada, 2005.

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

[9] George W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, 1951.

[10] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. In *FOCS*, 2006.

[11] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing Nash equilibria: approximation and smoothed complexity. In *FOCS*, pages 603–612, 2006.

[12] Vincent Conitzer and Tuomas Sandholm. Complexity results about Nash equilibria. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 765–771, Acapulco, Mexico, 2003.

[13] Vincent Conitzer and Tuomas Sandholm. Awesome: a general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2006.

[14] Costas Daskalakis, Paul Goldberg, and Christos H. Papadimtriou. The complexity of computing a Nash equilibrium. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC 2006)*, pages 71–78, Seattle, May 2006.

[15] Daniela Pucci de Farias and Nimrod Megiddo. How to combine expert (or novice) advice when actions impact the environment? In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2003.

[16] Thomas G. Dietterich. Ensemble learning. In *The Handbook of Brain Theory and Neural Networks, Second Edition*. The MIT Press, 2002.

[17] Joan Feigenbaum, Vijay Ramachandran, and Michael Schapira. Incentive-compatible interdomain routing. In *ACM Conference on Electronic Commerce*, 2006.

[18] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.

[19] Lawrence Folland. High peformance computing resources. Website. www.cs.uwaterloo.ca/twiki/view/CF/HighPerformanceComputingResources.

[20] Dean Foster and Rakesh Vohra. A randomization rule for selecting forecasts. *Operations Research*, 41:704–709, 1993.

[21] Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40–55, 1997.

[22] Dean P. Foster and H. Peyton Young. Regret testing: a simple pay-off based procedure for learning Nash equilibrium. *Theoretical Economics*, 1(3):341–367, 2006.

[23] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[24] Yoav Freund and Robert Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[25] Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320–367, 1993.

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

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

[28] Drew Fudenberg and David Levine. Conditional universal consistency. *Games and Economic Behavior*, 29:104–130, 1999.

[29] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting and incomplete information. *Econometrica*, 54:533–554, 1986.

[30] Fabrizio Germano and Gabor Lugosi. Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, 2007. To appear.

[31] Amy Greenwald and Amir Jafari. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Conference on Learning Theory (COLT)*, Washington, D.C., 2003.

[32] James Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, pages 97–139. Princeton University Press, 1957.

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

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

[35] Sergiu Hart and Andreu Mas-Colell. Stochastic uncoupled dynamics and Nash equilibrium. *Games and Economic Behavior*, 2005.

[36] James Jordan. Three problems in learning mixed-strategy equilibria. *Games and Economic Behavior*, 1993.

[37] Jean-Francois Mertens, Sylvain Sorin, and Shmuel Zamir. *Repeated Games*. 1989.

[38] Koichi Miyasawa. On the convergence of the learning process in a 2x2 non-zero-sum two-person game. Technical report, Princeton University, 1961.

[39] Roger Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, 1991.

[40] John H. Nachbar. "Evolutionary" selection dynamics in games: convergence and limit properties. *International Journal of Game Theory*, 19(1):59–89, 1990.

[41] John Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[42] Andrew Y. Ng, Jin H. Kim, Michael I. Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. In *Proceedings of the Neural Information Processing Systems conference*, 2004.

[43] Christos H. Papadimitriou. Computing correlated equilibria in multi-player games. In *Proceedings of the 27th Annual ACM symposium on Theory of Computing*, pages 49–56, 2005.

[44] Rob Powers and Yoav Shoham. Learning against opponents with bounded memory. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, UK, 2005.

[45] Julia Robinson. An iterative method of solving games. *Annals of Mathematics*, 54:296–301, 1951.

[46] Jeffrey S. Rosenthal. *A First Look at Rigorous Probability Theory*. World Scientific, 2000.

[47] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000.

[48] Lloyd S. Shapley. Some topics in two-person games. In *Advances in Game Theory*. Princeton University Press, 1964.

[49] Yoav Shoham, Rob Powers, and Trond Grenager. If multiagent learning is the answer, what is the question? *Artificial Intelligence (Special Issue on the Foundations of Research in Multiagent Learning)*, 2007. To appear.

[50] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 541–548, Stanford, CA, 2000.

[51] James Stewart. *Calculus*. Brooks/Cole Publishing Company, 1999.

[52] Eric van Damme. *Stability and Perfection of Nash Equilibria*. Springer-Verlag, 1991.

[53] Hal R. Varian. Online ad auctions. In *ACM Conference on Electronic Commerce*, 2006.

[54] John von Neumann and Oskar Morgenstein. *Theory of games and economic behavior*. Princeton University Press, 1947.

[55] Vladimir Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383, 1990.