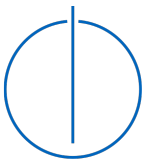
DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Information Systems

No-Regret Learning in Finite Games

Henning Heyen



DEPARTMENT OF INFORMATICS

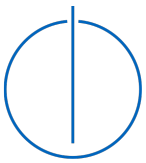
TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Information Systems

No-Regret Learning in Finite Games

No-Regret Learning in endlichen Spielen

Author:	Henning Heyen
Supervisor:	Prof. Dr. Martin Bichler
Advisor:	Matthias Oberlechner
Submission Date:	15.03.2022



I confirm that this bachelor's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 15.03.2022 Henning Heyen

Acknowledgments

First of all, I would like to thank my advisor Matthias Oberlechner for his outstanding support throughout the past five months. Matthias has spent hours of Zoom meetings with me and always provided valuable feedback. Additionally, I wish to extend my special thanks to my girlfriend, Carolin Echterbeck. She has always encouraged me with this project, especially in the final phase. Lastly, I would like to express my gratitude to my parents, Michaela Temme and Thorsten Heyen, and my sister Svenja Heyen for their unconditional support that goes way beyond this project.

Abstract

This thesis addresses the outcome of no-regret dynamics in finite games. Can the outcome be characterized by traditional game-theoretic solution concepts like Nash equilibria? The general answer to this question is no. Nevertheless, there are some games where Nash convergence under no-regret learning has been observed before. The thesis aims to give a neat and compact overview on sufficient conditions under which no-regret learning converges to Nash. These conditions are empirically confirmed by employing two concrete instances of no-regret algorithms on simple two-player games. Plots are provided further to give an intuition of the algorithms' behavior. The bottom line is that only strict Nash equilibria survive under no-regret dynamics. Moreover, the empirical frequency of play converges to interior Nash equilibria in two-player zero-sum games.

Contents

1 Introduction

For decades scientists are interested in the game-theoretic solution concept of Nash equilibria. Even though we know there exists one Nash equilibrium in all finite games that allow mixed strategies, it is computationally intractable to find one in general. This problem has been a topic of intense research in game theory and its applications in economics, optimization, and more recently, machine learning. Since there is not much hope for an algorithm that can compute Nash equilibria efficiently we can instead ask the question: Are Nash equilibria learnable by no-regret dynamics? In contrast to deterministic best response dynamics like fictitious play, no-regret dynamics can potentially learn pure and mixed Nash equilibria as they are algorithms by which players assign probabilities to strategies. The fundamental question is if all players employ a no-regret update policy in a repeated game, do their strategies converge to a Nash equilibrium. Unfortunately, the general answer is no. Even in simple two-player games, we find cycling behavior and strictly dominated strategies assigned with positive probability, which is highly irrational to play. The underlying reason is that no-regret dynamics are known to converge to the game's set of coarse correlated equilibria, a much weaker solution concept. However, the set of Nash equilibria is properly included in the set of coarse correlated equilibria, and for some games, we do observe Nash convergence under no-regret learning. The bottom line is that only strict Nash equilibria survive, and the empirical frequency of play converges to Nash in two-player zero-sum games that yield an interior equilibrium. For a better intuition, this thesis visualizes the behavior of two concrete no-regret algorithms in simple two-player games.

Chapter 2 will introduce the basic concepts of online convex optimization and regret minimization. Later in the chapter, we will derive the family of online mirror descent algorithms. Then, in chapter 3, finite games and their mixed extensions will be formally defined. Also, we study the hierarchy of game-theoretic solution concepts, including Nash and correlated equilibria. After that, in chapter 4, we will review the literature considering the convergence behavior of no-regret dynamics in finite games. Finally, in chapter 5, we will run two no-regret algorithms, namely projected online gradient ascent and entropic gradient ascent, on simple two-player games and empirically verify the results from the literature. In the end, we introduce three games that yield non-strict (or weak) pure Nash equilibria. Interestingly, I found convergent behavior but not to Nash equilibria.

2 No-Regret Learning

Regret dynamics are one the best-studied algorithmic frameworks in online convex optimization. This chapter will introduce the concept of no-regret learning, similar to [7, Chapter 2]. Later we will cover two concrete no-regret algorithms, namely *projected online gradient descent* and *entropic gradient descent*. In the simulations of chapter 5, we will then run those two algorithms on simple two-player games. Note that throughout chapter 2, we will minimize losses, which is the convention in optimization. In chapter 3, on the other hand, we are maximizing utilities, which is the convention in game theory.

2.1 Notation and Basic Definitions

Let us first define some basic concepts that we will make use of. Throughout sets are denoted by upper case letters and vectors by lower case letters. In online learning, a sequence of play is considered. Therefore, we denote x_t the t-th vector in the sequence of $x_1, \dots, x_T$ where T is the number of iterations. In a slight abuse of notation, we will also use index notation for the i-th element of a vector, but it will be clear from the context.

The *inner product* of two vectors x and y with dimension n is defined as

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$

The *norm* l_p of a vector x is defined as

$$\|x\|_p = (\sum_i (|x_i|^p)^{1/p}$$

In particular, the l_2 (or *Euclidean*) norm is then $\|x\|_2 = \sqrt{\langle x, x \rangle}$.

The *gradient* of a differentiable function $f : \mathcal{X} \to \mathbb{R}$ is denoted by ∇f. A function f is called $L-$*Lipschitz* over a set $\mathcal{X}$ with respect to some norm $\|\cdot\|$ if for all $x, y \in \mathcal{X}$ we have that,

$$|f(x) - f(y)| \leq L\|x - y\|$$

A function $f : \mathcal{X} \to \mathbb{R}$ is said to be *convex* if for all x, y and $\lambda \in [0, 1]$ we have that

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Likewise, a function $f : \mathcal{X} \to \mathbb{R}$ is said to be *concave* if for all x, y and $\lambda \in [0, 1]$ we have that

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

A function $f : \mathcal{X} \to \mathbb{R}$ is $\sigma-$*strongly-convex* over $\mathcal{X}$ with respect to some norm $\|\cdot\|$ if there is some $x \in \mathcal{X}$ such that for all $y \in \mathcal{X}$ we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\sigma}{2}\|y - x\|^2$$

A set is called *convex* if for all $x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$ we have that

$$\lambda x + (1 - \lambda)y \in \mathcal{X}$$

Throughout this paper, $\mathcal{V}$ will denote a finite-dimensional real space with some norm $\|\cdot\|$ and $\mathcal{X} \subseteq \mathcal{V}$ a closed convex subset thereof. We will write $\mathrm{ri}(\mathcal{X})$ as the relative interior of $\mathcal{X}$ and $\mathrm{diam}(\mathcal{X}) = \sup\{\|x' - x\| : x, x' \in \mathcal{X}\})$ for its diameter. Also, we will denote $\mathcal{Y} \equiv \mathcal{V}^*$ for the algebraic dual of $\mathcal{V}$, $\langle y, x \rangle$ for the canonical pairing of $y \in \mathcal{Y}$ and $x \in \mathcal{V}$ and $\|y\|_* \equiv \sup\{\langle y, x \rangle : \|x\| \leq 1\}$ for its dual norm. Given an extended real-valued function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ its effective domain is defined as $\mathrm{dom} f = \{x \in \mathcal{V} : f(x) < \infty\}$.

2.2 The Basic Model

In online optimization, the goal is to minimize the aggregate loss incurred against a sequence of unknown loss functions. Formally, at every time step $t = 1, \ldots T$, the optimizer selects an *action* x_t from a closed convex subset $\mathcal{X}$ of an n-dimensional normed space $\mathcal{V}$. After that, the optimizer suffers a convex loss $l_t(x_t)$ based on an a priori unknown loss function $l_t : \mathcal{X} \to \mathbb{R}$. Based on that, the optimizer then updates its action and repeats. See figure 2.1 for a pseudo-code description.

input: convex action set $\mathcal{X}$, sequence of convex loss functions $l_t : \mathcal{X} \to \mathbb{R}$
for $t = 1, 2, \ldots, T$ **do**
  select action $x_t \in \mathcal{X}$
  incur loss $l_t(x_t)$
  update $x_t \leftarrow x_{t+1}$
end

Figure 2.1: pseudo-code for the online convex optimization framework

Based on the properties of the loss function, we distinguish between two disjoint subclasses of online optimization. Throughout, we assume l_t to be differentiable and that it attains a minimum in $\mathcal{X}$.

1. *Online strongly convex optimization*: each l_t is assumed to be α_t-strongly convex, i.e. with respect to $\|\cdot\|$ there is some $x \in \mathcal{X}$ such that for all $x' \in \mathcal{X}$ we have that

$$l_t(x') \geq l_t(x) + \langle \nabla l_t(x), x' - x \rangle + \frac{\alpha_t}{2}\|x' - x\|^2 \tag{2.1}$$

   for some $\alpha_t > 0$

2. *Online linear optimization*: each l_t is assumed to be linear, i.e.

$$l_t(x) = -\langle v_t, x \rangle \tag{2.2}$$

   for some *payoff vector* $v_t \in \mathcal{V}^*$

Depending on the information available to the optimizer, we can specify two feedback assumptions:

1. *Full information*: the entire loss function l_t is revealed to the optimizer at each time step.

2. *First order information*: only the perfect gradient $\nabla l_t(x_t)$ for some input $x_t \in \mathcal{X}$ is revealed.

Obviously, *first order information* is a much lighter assumption than *full information* and therefore applicable to a broader range of problems.

2.3 Regret Minimization

The notion of *regret* is a widely used performance measure of online algorithms. In words, it measures how "sorry" the optimizer is not to have followed a fixed competing action in hindsight. It is the difference between the cumulative loss of the actual sequence of play induced by an algorithm and the cumulative loss of the best fixed action.

Definition 2.1 *The regret incurred by a sequence of actions $x_t \in \mathcal{X}$ and a sequence of loss functions $l_t : \mathcal{X} \to \mathbb{R}$ for an algorithm running for T iterations is defined as*

$$reg(T) = \sum_{t=1}^{T} l_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} l_t(x)$$

The learner's goal is to have the lowest regret possible. In fact, we say an algorithm exhibits *no-regret* if the regret tends to zero as T, the number of iterations, goes to infinity.

Definition 2.2 *An algorithm exhibits no-regret iff $reg(T)$ grows sublinearly with T, i.e.*

$$reg(T) = o(T)$$

The fundamental question in online convex optimization is whether no-regret is achievable. We will address this in the following section.

2.4 Leader Following Policies

The first no-regret candidate is based on the simple update rule: at time $t+1$, select the optimal action in hindsight up to including time step t. This policy is known as *follow the leader* (FTL).

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \sum_{i=1}^{t} l_i(x) \tag{FTL}$$

This approach, however, needs *full information feedback*, i.e. the knowledge of the entire loss function l_t once x_t is chosen, plus the ability to compute the argmin operator in the update step. Both requirements are much lighter in *online linear optimization*, where loss functions are in the form of equation 2.2. But even for *online linear optimization* problems, the no-regret property is not guaranteed under FTL. Consider the following example.

Let $\mathcal{X} = [-1, 1]$ and set the linear loss function as follows

$$l_t(x) = \begin{cases} -x/2 & \text{if } t = 1 \\ x & \text{if } t \text{ is even} \\ -x & \text{if } t > 1 \wedge t \text{ is odd} \end{cases}$$

Apart from $t = 1$, where x_t could actually be set arbitrarily in $[-1, 1]$ according to FTL policy we have that $x_t = (-1)^t$ for all $t > 1$. The incurred loss after T iterations is then in the form of $\sum_{t=1}^{T} l_t(x_t) = T - x_1/2 - 1$. Comparing that with the fixed action $x_t = 0$ for all t, we have that $reg(T) \sim T$, which is not sublinear with T. We can conclude that FTL does not guarantee no-regret.

In some sense, FTL seems to be "unstable". The predictions shift drastically from round to round. One way to stabilize FTL is to add a so-called regularization (or penalty) term. It makes sure that the prediction in the upcoming round is not too "far" off from the current one. That leads to a policy known as *follow the regularized leader* (FTRL). It can be formulated as follows

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \sum_{i=1}^{t} l_i(x) + \frac{1}{\gamma} h(x) \right\} \tag{FTRL}$$

The regularization function is denoted by $h : \mathcal{X} \to \mathbb{R}$, and $\gamma > 0$ is a tunable step size parameter that adjusts the weight on the regularization term. In order to obtain a stabilizing effect, it is common to assume that h is K-strongly convex and continuous. Considering the regret analysis of FTRL we have the following result [7, Theorem 2.1].

Proposition 2.1 *Suppose FTRL is run on a sequence of $l_1, \ldots, l_T$ of convex loss functions. Further, assume each l_t is L_t-Lipschitz with respect to some norm $\|\cdot\|$ and $L \equiv \sup_t L_t < \infty$. Let $H \equiv \max h - \min h$ be the "depth" of h and assume h to be K-strongly convex and set $\gamma = \frac{1}{L}\sqrt{\frac{HK}{T}}$. Then we have that*

$$reg(T) \leq 2L\sqrt{(H/K)T} = o(T)$$

Proposition 2.1 shows that under some assumptions, no-regret is indeed achievable. These include that *(i)* the optimizer has full information on the entire loss function up to the current time step, *(ii)* the minimization problem in the FTRL update rule can be solved efficiently, and *(iii)* the horizon of play is known in advance. While *(iii)* can be resolved easily by the *doubling trick* method [11], the other two are much harder to overcome. The easiest way to minimize a loss function that requires only *first order information* is based on an algorithm known as *(projected) online gradient descent.*

2.5 Projected Online Gradient Descent

The most straightforward method to minimize a loss function in optimization theory is based on gradient descent dynamics. The algorithm simply takes a step in the direction of the objective's gradient. Then, if the problem is constrained, the result is projected back to the feasible region, which is the probability simplex in finite games with mixed extensions. The process repeats. This policy is known as *projected online gradient descent* (POGD) and can be formulated by the following recursive update rule

$$x_{t+1} = \Pi(x_t + \gamma_t v_t) \tag{POGD}$$

where

$$v_t = -\nabla_t = -\nabla l_t(x_t) \tag{2.3}$$

denotes the gradient of the loss function at x_t, $\gamma_t > 0$ it the step size and $\Pi : \mathcal{V} \to \mathcal{X}$ is the *Euclidean projector*

$$\Pi(x) = \underset{x' \in \mathcal{X}}{\operatorname{argmin}} \|x' - x\|_2^2$$

For a pseudo-code description and a schematic representation[1] of POGD see figure 2.2.

input: step size sequence $\gamma_t > 0$
for $t = 1, 2, \ldots, T$ **do**
  incur loss $l_t(x_t)$
  receive feedback $v_t \leftarrow -\nabla l_t(x_t)$
  update $x_{t+1} = \Pi(x_t + \gamma_t v_t)$
end

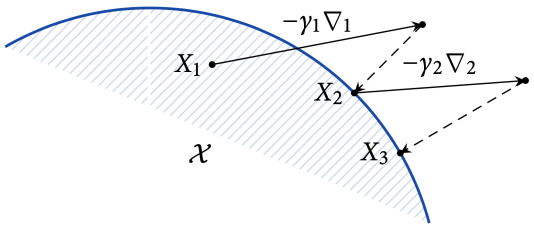


Figure 2.2: pseudo-code and schematic representation of POGD

It is worth mentioning that the addition $x_t + \gamma_t v_t$ in the update step is not well-defined in general, as we add a primal with a dual vector. Assuming $\mathcal{V}$ to be the Euclidean space, however, we have that its dual space $\mathcal{V}^*$ is canonically identified with $\mathcal{V}$. This assumption can only be made when the Euclidean norm is used [7]. Let us move to the regret analysis of POGD [7, Theorem 2.2].

Proposition 2.2 *Suppose POGD is run on a sequence of convex and L_t-Lipschitz loss functions l_t with a constant step size $\gamma_t \equiv \gamma > 0$ where* $\operatorname{diam}(\mathcal{X}) \equiv \max\{\|x' - x\|_2 : x, x' \in \mathcal{X}\}$ *denotes the diameter of $\mathcal{X}$. In particular, if $L \equiv \sup_t L_t < \infty$ and $\gamma = \frac{\operatorname{diam}(\mathcal{X})}{L\sqrt{T}}$, then the regret is bounded by*

$$reg(T) \leq \operatorname{diam}(\mathcal{X}) L \sqrt{T}$$

We can conclude that up to a multiplicative constant, POGD enjoys the same regret bound as FTRL (proposition 2.1). Both algorithms have regret in the size of $\mathcal{O}(\sqrt{T})$ and thereby, both are no-regret algorithms. FTRL, however, needs *full information* on the

[1] borrowed from [7, Chapter 2]

loss function l_t at each time step, whereas POGD just needs *first-order information* in the form of the gradient. That makes POGD more lightweight and applicable to a wide range of problems [7].

Before we move on let us address the connection between *follow the regularized leader* and *online gradient descent*. Consider the unconstrained linear optimization problem with $\mathcal{X} = \mathbb{R}^n$, Euclidean regularizer $h = \frac{1}{2}\|x\|_2^2$ and linear losses of the form $l_t(x) = -\langle v_t, x\rangle$ for some sequence $v_t \in \mathbb{R}^n$. According to FTRL this yields [7]

$$\begin{aligned} x_{t+1} &= \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \sum_{i=1}^{t} l_i(x) + \frac{1}{\gamma}h(x) \right\} = \operatorname*{argmin}_{x\in\mathbb{R}^n} \left\{ \|x\|_2^2 - 2\gamma \sum_{i=1}^{t} \langle v_i, x\rangle \right\} \\ &= \operatorname*{argmin}_{x\in\mathbb{R}^n} \left\| x - \gamma \sum_{i=1}^{t} \langle v_i, x\rangle \right\|_2^2 = \gamma \sum_{i=1}^{t} v_i = \gamma \sum_{i=1}^{t-1} v_i + \gamma v_t = x_t + \gamma v_t \end{aligned}$$

That is simply the unprojected update policy of POGD. It is an example of a much more general link between *leader following* and *gradient* dynamics. The main idea is to "linearize" FTRL in the sense that we replace $l_t(x)$ with its *linear surrogate* $\tilde{l}(x)$.

$$\tilde{l}_t(x) = l_t(x_t) + \langle \nabla l_t(x_t), x_t - x\rangle$$

Applying the linear surrogate to FTRL leads to *follow the linearized leader* FTLL [7].

$$x_{t+1} = \operatorname*{argmax}_{x\in\mathcal{X}} \left\{ \gamma \sum_{i=1}^{t} v_s - h(x) \right\} \tag{FTLL}$$

The signal v_s simply denotes the perfect gradient of the loss function l_t at x_t analog to equation 2.3. Note that by this modification FTLL only requires first order information, just like POGD. Writing FTLL recursively leads to the *dual averaging* framework. For more details on that, refer to [7, 9].

2.6 Online Mirror Descent

As discussed in [11, 7], there are cases where the problem's underlying geometry may allow considerably sharper regret bounds as in POGD. The reason is that the Lipschitz constant L is a multiplicative factor in the regret, and it depends on the underlying norm. In POGD, we constrain ourselves on the l_2 norm. Allowing an adaptive Lipschitz constant, however, can be beneficial. For instance, in the *multi-armed bandit* problem, when using

the Euclidean norm, the Lipschitz constant L can be bounded by $\sqrt{n}$. Using l_∞ norm instead of l_2 norm, however, we can bound L by 1. For a more detailed survey, refer to [11, 7]. The natural question arises whether running POGD with non-Euclidean norm can lead to better regret bounds. The *online mirror descent* framework addresses that question.

To understand the idea of online mirror descent, let us revisit projected online gradient descent and formulate it more abstractly. Given an input point $x \leftarrow x_t$ and an impulsive vector $y \leftarrow \gamma_t v_t$, then POGD returns an output point $x^+ \leftarrow x_{t+1}$ defined as

$$\begin{aligned} x^+ = \Pi(x+y) &= \operatorname*{argmin}_{x' \in \mathcal{X}} \left\{ \|x + y - x'\|_2^2 \right\} \\ &= \operatorname*{argmin}_{x' \in \mathcal{X}} \left\{ \|x - x'\|_2^2 + \|y\|_2^2 + 2\langle y, x - x' \rangle \right\} \\ &= \operatorname*{argmin}_{x' \in \mathcal{X}} \left\{ \langle y, x - x' \rangle + D(x', x) \right\} \end{aligned} \tag{2.4}$$

where

$$D(x', x) = \frac{1}{2}\|x' - x\|_2^2 = \frac{1}{2}\|x'\|_2^2 - \frac{1}{2}\|x\|_2^2 - \langle x, x' - x \rangle$$

is the squared Euclidean distance between x and x'. The function $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called *Bregman divergence* and can be generalized by

$$D(x', x) = h(x') - h(x) - \langle \nabla h(x), x' - x \rangle$$

The Bregman divergence is induced by some regularization function $h : \mathcal{X} \to \mathbb{R}$. The basic idea of online mirror descent is to replace the Euclidean distance in POGD with some Bregman divergence induced by a regularization function that is not necessarily based on the Euclidean norm. By that, we can make use of the problem's geometric properties and hope for a better regret bound. The regularization function can be viewed analog to the one introduced in FTRL, so again, we assume h to be continuous and K-strongly convex with respect to some norm $\|\cdot\|$ [7].

Depending on the Bregman divergence D induced by some regularization function h we can define a more general projection $Q : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ named *mirror map* as

$$Q_x(y) = \operatorname*{argmin}_{x' \in \mathcal{X}} \left\{ \langle y, x - x' \rangle + D(x', x) \right\} \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y}$$

Finally, we can formulate the online mirror descent (OMD) update policy.

$$x_{t+1} = Q_{x_t}(\gamma_t v_t) \tag{OMD}$$

where $\gamma_t > 0$ again is the step size sequence and $v_t = -\nabla l_t(x_t)$ is the first-order information of the loss functions l_t in the form of the gradient. Note that the mirror map Q is solely induced by the choice of h, the regularization function. Considering the regret analysis of OMD, we have the following result [7, Theorem 2.4]

Proposition 2.3 *Suppose OMD is run on a sequence of $l_1, \dots, l_T$ of convex loss functions. Further, assume each l_t is L_t-Lipschitz with respect to some norm $\|\cdot\|$ and $L \equiv \sup_t L_t < \infty$. Let $H \equiv \max h - \min h$ be the "depth" of h and assume h to be K-strongly convex and set $\gamma = \frac{1}{L}\sqrt{\frac{2HK}{T}}$. Then we have that*

$$reg(T) \leq L\sqrt{(2H/K)T}$$

In terms of regret bounds, the main difference between POGD (proposition 2.3 and OMD (proposition 2.2) is the factor $2H/K$ and the norm defining the Lipschitz constant L. The factor fully depends on the choice of the regularizer h. So adjusting h to the problem may lead to improved efficiency. Let us introduce two examples of regularization functions and their corresponding mirror maps.

Consider the *Euclidean regularizer*.

$$h(x) = \frac{1}{2}\|x\|_2^2$$

Obviously, the induced mirror map is simply the Euclidean projection in the form of equation 2.4. So, in that case, OMD coincides with POGD.

$$Q_x(y) = \operatorname*{argmin}_{x' \in \mathcal{X}} \left\{ \langle y, x - x' \rangle + \frac{1}{2}\|x' - x\|_2^2 \right\} = \Pi(x + y)$$

Another commonly used regularization function is the *entropic regularizer*. Let $\mathcal{X} = \Delta(\mathcal{A})$ be the standard probability simplex of $\mathbb{R}^n$. The entropic regularization function is defined as

$$h(x) = \sum_{a \in \mathcal{A}} x_a \log(x_a)$$

A straightforward calculation shows that h is 1-strongly convex with respect to the l_1 norm [7]. The induced mirror map can be written as

$$Q_x(y) = \frac{(x_a \exp(y_a))_{a \in \mathcal{A}}}{\sum_{a \in \mathcal{A}} x_a \exp(y_a)}$$

When we apply this mirror map to the OMD framework we obtain an algorithm named *entropic gradient descent* (EGD). For a survey and a more detailed explanation on EGD refer to [11]. For a pseudo-code description and a schematic representation[2] of EGD see figure 2.3.

$$x_{a,t+1} = \frac{x_{a,t}\exp(\gamma_t v_{a,t})}{\sum_{a' \in \mathcal{A}} x_{a',t}\exp(\gamma_t v_{a',t})} \tag{EGD}$$

input: $\gamma_t > 0$, $\mathcal{X} = \Delta(\mathcal{A})$
for $t = 1, 2, \ldots, T$ **do**
  incur loss $l_t(x_t)$
  receive feedback $v_t \leftarrow -\nabla l_t(x_t)$
  update $x_{t+1} = \frac{(x_{a,t}\exp(\gamma_t v_{a,t}))_{a \in \mathcal{A}}}{\sum_{a' \in \mathcal{A}} x_{a',t}\exp(\gamma_t v_{a',t})}$
end

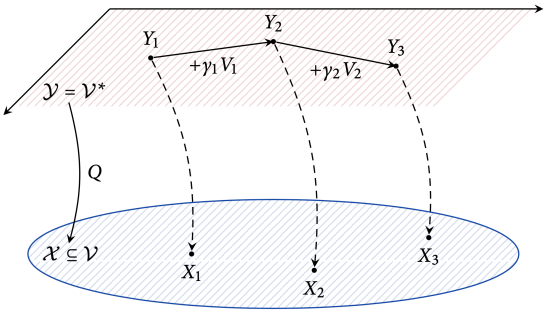


Figure 2.3: pseudo-code and schematic representation of EGD

To be precise, the figure shows the *lazy* variant EGD. It results from the FTLL framework with entropic regularization. It is *lazy* because it aggregates the gradients' steps *lazily*, i.e. the gradient step is not performed on the projection but on the previous gradient step. Figure 2.4 illustrates the difference between *lazy* and *eager* gradient descent[3]. An important observation we need to make is that the entropic regularizer becomes infinitely *steep* at the boundary of $\mathcal{X}$. So, the effective domain of the entropic regularization function is the relative interior of $\mathcal{X}$, i.e. $\mathrm{dom}h = \mathrm{ri}(\mathcal{X})$. Therefore projections of the mirror map will always return interior points of $\mathcal{X}$, as figure 2.3 suggests. Note that for steep regularizers, lazy and eager algorithms coincide [7]. Therefore the shown sequence of play in figure 2.3 is the same as is in the eager version of EGD. For the Euclidean regularizer, on the other hand, projections lead to points on the boundary of $\mathcal{X}$ like shown in figure 2.2. The Euclidean regularizer is said to be *non-steep*.

Comparing the regret bound between POGD and EGD in the *multi-armed bandit* problem, for example, we find that [7]

$$reg_{POGD}(T) \leq 2\sqrt{nT} \qquad \text{and} \qquad reg_{EGD}(T) \leq \sqrt{2T\log(n)}$$

[2] borrowed from [7, Chapter 2]
[3] also borrowed from [7, Chapter 2]

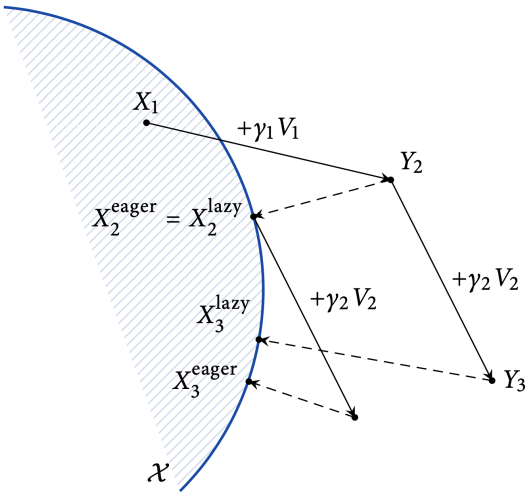


Figure 2.4: lazy vs. eager gradient descent

As a result, even though both POGD and OMD enjoy the same regret bound of $\mathcal{O}(\sqrt{T})$ the multiplicative constant involved can have an enormous impact on the algorithm's efficiency. Therefore, OMD can be of particular interest for problems that operate in high-dimensional spaces.

By now, we have familiarized ourselves with the online mirror descent framework and encountered two concrete no-regret algorithms for constrained problems, namely *projected online gradient descent* (POGD) and *entropic gradient descent* (EGD). We will later employ these two algorithms on simple two-player games and elaborate on their behavior in chapter 5. Before that, let us define finite games more formally.

3 Finite Games

In a mathematical context, games are models of strategic behavior between players that have individual interests. All games have the same basic structure. They consist of players, each player's set of actions, and some payoff that depends on all players' actions. By defining these three elements, we can derive well-defined solution concepts such as Nash equilibria. The fundamental question of this thesis is: If all players choose their actions according to a *no-regret* update policy in a repeated game, will the outcome be some kind of equilibrium state. Before answering that question, let us introduce the concept of games and equilibria more formally.

3.1 Notation and Definitions

Formally, a finite game is 3-tuple $\Gamma \equiv (\mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}})$ where each *player* $i \in \mathcal{N} = \{1, \dots, N\}$ chooses a *pure strategy* (or *action*) a_i from a finite set $\mathcal{A}_i$ of *pure strategies*. The fact that there is only a finite number of players and strategies makes it a *finite* game. Let $\mathcal{A} = \prod_i \mathcal{A}_i$ be the *action space* of pure strategies. The players' *payoff*, also called *utility* or *reward*, depends on all players' *strategies*. There are no assumptions on the *players' payoff* function of pure strategies $u_i : \mathcal{A} \to \mathbb{R}$.

Additionally, we allow the players to mix strategies. So the strategies that players choose are probability vectors. Formally, in a *mixed extension* of Γ, players can also play *mixed strategies*, i.e. probability distributions x_i drawn from the probability simplex $\mathcal{X}_i = \Delta(\mathcal{A}_i) = \{x_i : \sum_{a_i\in\mathcal{A}_i} x_{i,a_i} = 1 \wedge x_i \geq 0\}$. We can interpret x_{i,a_i} as the probability that player $i \in \mathcal{N}$ assigns to action $a_i \in \mathcal{A}_i$. The expected payoff for some player i depends on a *mixed profile* $x = (x_1, \dots, x_N) \in \mathcal{X} = \prod_i \mathcal{X}_i$ and can be written as

$$u_i(x) = \sum_{a_1\in\mathcal{A}_1} \cdots \sum_{a_N\in\mathcal{A}_N} x_{1,a_1} \dots x_{N,a_N} u_i(a_1, \dots, a_N)$$

When we want to highlight that player i selects strategy x_i, we write $x = (x_i; x_{-i})$ where $x_{-i} = (x_j)_{j\neq i}$ is the ensemble of strategies selected by the other players. Assuming u_i to be continuously differentiable, we can simply write the players' *individual gradients* as their *payoff* vectors.

$$v_i(x) = \nabla_{x_i} u_i(x_i; x_{-i}) = (u_i(a_i; x_{-i}))_{a_i \in \mathcal{A}_i}$$

In other words, the gradient $v_i(x)$ is the payoff $(u_i(a_i; x_{-i}))_{a_i \in \mathcal{A}_i}$ to player i when i selects a_i against the mixed strategy profile of the other players x_{-i}.

A finite game is said to be *zero-sum* if for all pure action profiles $a \in \mathcal{A}$ the cumulative utilities of all players sum up to zero.

$$\sum_{i=1}^{N} u_i(a) = 0 \qquad \forall a \in \mathcal{A} \tag{3.1}$$

In the literature, we sometimes find *constant sum* games, i.e. the equation above holds for some constant $c \in \mathbb{R}$. However, constant sum games are equivalent to zero-sum games as they can be normalized. By simply subtracting $\frac{c}{N}$ from all utilities, we obtain a zero-sum game without any loss of information. If a game is neither zero nor constant sum, we say it is a *general-sum* game.

Let us define the notion of *dominated strategies* [7]. Formally, given a finite game Γ, we say that $a_i \in \mathcal{A}_i$ is *strictly dominated* by $a_i' \in \mathcal{A}_i$ if the following holds.

$$u_i(a_i; x_{-i}) < u_i(a_i'; x_{-i}) \quad \forall x_{-i} \in \mathcal{X}_{-i} \equiv \prod_{j \neq i} \mathcal{X}_j \tag{3.2}$$

If the inequality holds strictly only for some, but not for all $x_{-i} \in \mathcal{X}_{-i}$, then a_i is said to be *weakly dominated* by a_i'. Of course, when a dominated strategy is removed from the game, other strategies may become dominated in the resulting game. Strategies that are removed at some point in this repetitive process are called *iteratively dominated strategies*. Note that choosing a strictly dominated strategy is obviously irrational to play.

Another way to reason about finite games is to consider *pareto dominance*. A strategy profile $a' \in \mathcal{A}$ is said to *pareto dominate* another profile $a \in \mathcal{A}$ if no player gets less utility with a' than with a and at least one player gets better utility with a' than with a. Formally, a' *pareto dominates* a if both statements hold.

1. $\forall i \in \mathcal{N}: \quad u_i(a') \geq u_i(a)$
2. $\exists i \in \mathcal{N}: \quad u_i(a') > u_i(a)$

A strategy profile $a \in \mathcal{A}$ is *pareto optimal* if there is no other profile $a' \in \mathcal{A}$ that pareto dominates a.

Assume the mixed extension of Γ is played over and over again. Let us call this a *repeated game* Γ^∞. Consider $x^t = (x_1^t, \dots, x_N^t)$ as the strategy profile played in round t where each $x_i^t \in \Delta(\mathcal{A}_i)$ denotes player i's mixed strategy in round t. We can apply the concepts from chapter 2 where each player $i \in \mathcal{N}$ predicts a mixed strategy x_i^t in each time step t simultaneously. Then the players receive feedback in the form of the players' individual gradients $v_i(x^t) = \nabla_{x_i^t} u_i(x^t)$. We say a strategy $a_i \in \mathcal{A}_i$ becomes *extinct* if x_{i,a_i}^t goes to 0 as t goes to ∞.

Unfortunately, there is a misleading difference between *game theory* and *optimization* in terms of how the objective is formulated. In optimization, agents aim to *minimize* losses. However, in game theory, the general convention is that players *maximize* utilities. Therefore, we must assume the utility function to be concave instead of convex. The notion of regret as in definition 2.1 can be reformulated into a game-theoretic context. The *regret* of player i is then defined as

$$reg_i(T) = \max_{x_i \in \Delta} \sum_{t=1}^{T} u_i(x_i; x_{-i}^t) - \sum_{t=1}^{T} u_i(x^t)$$

Also, note that the derived algorithms in chapter 2 were *descent* algorithms. As we are maximizing utilities now, they become *ascent* algorithms. In particular, we move in the positive direction of the gradient. For that reason, *projected online gradient descent* (POGD) will be named *online projected gradient ascent* (OPGA) hereafter. Similarly, *entropic gradient descent* (EGD) is now called *entropic gradient ascent* (EGA).

Since the probability simplex $\Delta(\mathcal{A}_i)$ is convex and u_i is linear, which is concave, we can apply these algorithms to *repeated games* and hope that the sequence of play will converge to some notion of equilibrium. Equilibria concepts are discussed in more detail in section 3.3. But before that, let us introduce a well-known example of a finite game to have better intuition.

3.2 An Example for a Two Player Game

The easiest way to define a game in a two-player setting is to draw its *payoff matrix*. For a better understanding, let us consider an example. The zero-sum game Rock Paper Scissors, for instance, has the payoff matrix shown in table 3.1.

Obviously $\mathcal{N} = \{1, 2\}$. Following the convention, we will call payer 1 the *row player* and player 2 the *column player*, respectively. Both have the same action space $\mathcal{A}_1 = \mathcal{A}_2 = \{Rock, Paper, Scissor\} \equiv \{R, P, S\}$. Let us abbreviate the actions due to readability.

	Rock	*Paper*	*Scissors*
Rock	$0, 0$	$-1, 1$	$1, -1$
Paper	$1, -1$	$0, 0$	$-1, 1$
Scissors	$-1, 1$	$1, -1$	$0, 0$

Table 3.1: payoff matrix Rock Paper Scissors

The utilities for pure strategies are given in the payoff matrix, where the first entries refer to the row player and the second entries to the column player. The game is zero-sum because the players' utility sums up to zero for all pure strategy profiles, just like in equation 3.1. So for example, when the row player selects *Paper* while the column player chooses *Rock*, then the row player wins, and the column player loses. In formulas, for $a = (P,R) \in \mathcal{A}_1 \times \mathcal{A}_2$ we have

$$u_1(a) = 1 \qquad \text{and} \qquad u_2(a) = -1$$

We can use matrix notation and write the utility for mixed strategies $x = (x_1, x_2)$ as

$$u_1(x) = {x_1}^T A x_2 \qquad \text{and} \qquad u_2(x) = {x_1}^T B x_2$$

where A,B are the pure strategy payoffs matrices for each player derived from table 3.1.

$$A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

We can think of *mixed strategies* as probability vectors where players assign weights on actions. So for example, the row player can choose to play *Rock* and *Paper* equally likely and not to play *Scissors*. In formulas, $x_1 = (1/2, 1/2, 0)$. When the row player chooses to play solely rock, i.e. $x_2 = (1, 0, 0)$ then the expected payoff for both players would be

$$u_1(x) = 1/2 \qquad \text{and} \qquad u_2(x) = -1/2$$

Note that strategies are called *pure strategies*, when a single action is assigned with probability 1 and the others with probability 0. For instance, x_2 is a pure strategy. Finally, the individual gradients for both players are simply their payoff vectors.

$$v_1(x) = \nabla_{x_1} u_1(x) = A x_2 \qquad \text{and} \qquad v_2(x) = \nabla_{x_2} u_2(x) = x_1^T B$$

3.3 Equilibria Concepts

This chapter will study several game-theoretic solution concepts, starting with Nash equilibria. After that, the more general notion of correlated equilibria will be introduced along with an example. Generally speaking, while Nash equilibria are rational and hard to compute, correlated equilibria are less rational but easy to compute.

3.3.1 Pure and Mixed Nash Equilibria

The most widely used solution concept in game theory is that of a *Nash equilibrium* (NE). It is a state where no player can increase their expected payoff by changing their strategy while the other players keep their strategy unchanged. In other words, Nash equilibria are those strategy profiles that give no player the incentive to deviate unilaterally. In finite games with mixed extensions, we can distinguish between mixed and pure Nash equilibria.

Definition 3.1 *A mixed strategy profile $x^* \in \mathcal{X}$ is called mixed Nash equilibrium (MNE) if*

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \qquad \forall x_i \in \mathcal{X}_i, i \in \mathcal{N}$$

For instance, the unique MNE in Rock Paper Scissors from section 3.2 would be that each player chooses all actions equally likely, i.e. $x_i^* = (1/3, 1/3, 1/3)$ for all $i \in \mathcal{N}$. A special case of MNE is when all players choose pure strategies. Remember, a pure strategy simply means that all player $i \in \mathcal{N}$ select one action $a_i \in \mathcal{A}_i$ with probability 1 and the other actions $a_j \in \mathcal{A}_i \setminus \{a_i\}$ with probability 0. We can then define a *pure Nash equilibrium* as follows.

Definition 3.2 *An action profile $a^* \in \mathcal{A}$ is called pure Nash equilibria (PNE) if*

$$u_i(a_i^*; a_{-i}^*) \geq u_i(a_i; a_{-i}^*) \qquad \forall a_i \in \mathcal{A}_i, i \in \mathcal{N}$$

Cleary, all pure Nash equilibria are also mixed Nash equilibria. All x_i^* are simply unit vectors, then. Let us denote $PNE(\Gamma)$ the game's set of pure Nash equilibria and $MNE(\Gamma)$ as the game's set of mixed Nash equilibria. Then we have that in general

$$PNE(\Gamma) \subset MNE(\Gamma)$$

If a Nash equilibrium is not pure in all actions, i.e. $x_i^* \in \text{ri}(\mathcal{X}_i)$ for all player $i \in \mathcal{N}$, we call it an *interior* or *fully mixed* Nash equilibrium. We can further distinguish between *strict* and *weak Nash equilibria*. A Nash equilibrium x^* is called strict when definition 3.1 holds with strict inequality for all $x_i \neq x_i^*$. In other words, x^* is strict if no player can

deviate unilaterally from x^* without *reducing* their payoff or, equivalently, when every player has a unique best response to x^*. That implies that strict Nash equilibria are pure strategy profiles $x^* = (a_1^*, \dots, a_N^*)$ [7]. So, interior equilibria cannot be strict.

Definition 3.3 *A pure strategy profile $x^* = (a_1^*, \dots, a_N^*)$ is called strict Nash equilibrium if*

$$u_i(a_i^*; a_{-i}^*) > u_i(a_i; a_{-i}^*) \qquad \forall a_i \in \mathcal{A}_i \setminus \{a_i^*\}, i \in \mathcal{N}$$

If a NE is not strict, we will call it *weak*. Whether or not a NE is strict plays an essential role in the convergence behavior of no-regret algorithms. We will also discuss the tractability of computing Nash equilibria. It turns out that it is hard. But more on that later in chapter 4. Before that, less common but computationally tractable solution concepts are introduced, namely *correlated* and *coarse correlated equilibria*.

3.3.2 Correlated and Coarse Correlated Equilibria

Consider the following example[1]

	Stop	*Go*
Stop	$0, 0$	$0, 1$
Go	$1, 0$	$-100, -100$

Table 3.2: payoff matrix Intersection game

Imagine an intersection where the players are drivers that can either *Stop* or *Go*, in short, $\mathcal{A}_i = \{S, G\}$ for both players. The players' goal is to *Go* without a crash. When both play *Go* they crash and when both play *Stop* no one gains any utility. The payoff is set accordingly, as in table 3.2.

There are two pure Nash equilibria, (S, G) and (G, S). Note that at least one player has payoff 0 in that case. There is also a mixed Nash equilibrium. Suppose the row player selects the strategy $x_1 = (p, 1-p)$. Then the column player must be indifferent between both actions, i.e.

$$0 = p - 100(1-p) \iff 101p = 100 \iff p = 100/101$$

So both players choosing *Stop* with probability $p = 100/101$ and *Go* with probability $1 - p = 1/101$ leads to a MNE. In terms of utility, this is even worse as the expected

[1] University of Pennsylvania, Prof. Aaron Roth, NETS 412 Algorithmic Game Theory, Spring 2017, Lecture 8

payoff for both players is 0. Under the MNE, the four possible action profiles have roughly the following probability distribution.

	Stop	*Go*
Stop	98%	$< 1\%$
Go	$< 1\%$	$\approx 0.01\%$

Table 3.3: action profile probability distribution under MNE

A much better outcome would be the subsequent distribution.

	Stop	*Go*
Stop	0%	50%
Go	50%	0%

Table 3.4: action profile probability distribution under CE

Both players have expected utility 1/2 and do not risk a crash. The problem is that there is no MNE that yields this probability distribution. The reason is that Nash equilibria are defined as profiles of mixed strategies that require players to randomize independently, without any communication. The above distribution, however, requires both to correlate their actions. On actual streets, that correlation device is a traffic light. It suggests both drivers whether to *Stop* or *Go*. Following its advice is a best response for everyone. That concept can be formalized.

Definition 3.4 *A correlated equilibrium (CE) is a distribution $\mathcal{D}$ over action profiles $\mathcal{A}$ such that for all player $i \in \mathcal{N}$ and every action $a_i^* \in \mathcal{A}_i$*

$$\mathbb{E}_{a\sim\mathcal{D}}[u_i(a)] \geq \mathbb{E}_{a\sim\mathcal{D}}[u_i(a_i^*; a_{-i})|a_i]$$

In words, a CE is a probability distribution over action profiles such that after a profile a is drawn from this distribution, playing a_i is a best response for player i conditioned on seeing a_i, given that all the other players play according to a. In the intersection game, for instance, conditioned on seeing *Stop*, playing *Stop* is indeed a best response given that the other player sees *Go*. Likewise, conditioned on seeing *Go*, playing *Go* is indeed a best response given that the other player sees *Stop*.

Nash equilibira are also correlated equilibria. The difference is just that the players' actions are drawn from an independent distribution, so being conditioned on a_i provides no additional information to a_{-i}. Therefore the set of correlated equilibria $CE(\Gamma)$ strictly

contains the set of mixed Nash equilibria $MNE(\Gamma)$ in general. Note that in contrast to Nash equilibria, correlated equilibria can be computed efficiently by solving a linear program [7]. An even larger solution concept is the set of *coarse correlated equilibria*.

Definition 3.5 *A coarse correlated equilibrium (CCE) is a distribution $\mathcal{D}$ over action profiles $\mathcal{A}$ such that for all player $i \in \mathcal{N}$ and action $a_i^* \in \mathcal{A}_i$*

$$\mathbb{E}_{a\sim\mathcal{D}}[u_i(a)] \geq \mathbb{E}_{a\sim\mathcal{D}}[u_i(a_i^*; a_{-i})]$$

The difference to CE is that a CCE only requires that following a suggested action a_i when a is drawn from $\mathcal{D}$ is only a best response *before* a_i is seen. One can show that there are instances where $CCE(\Gamma)$ is strictly larger than $CE(\Gamma)$ by constructing a distribution[2] that is a CCE but not a CE. That yields the following general equilibrium hierarchy.

$$PNE(\Gamma) \subset MNE(\Gamma) \subset CE(\Gamma) \subset CCE(\Gamma)$$

Even though a CCE leads to a higher expected payoff, one can construct examples where a CCE assigns positive probability only to strictly dominated strategies [12]. Hence, they fail the most basic assumption that players are rational. Therefore, CCE is a relatively weak solution concept. Even though hard to compute, Nash equilibrium remains the most robust and stable solution concept. To what extent do *no-regret dynamics* converge to these solution concepts? That will be discussed in the next chapter.

[2] University of Pennsylvania, Prof. Aaron Roth, NETS 412 Algorithmic Game Theory, Spring 2017, Lecture 8

4 Literature Review

This chapter aims to overview significant results in no-regret dynamics applied to finite games with mixed extensions. First, we will discuss some general game-theoretic findings that motivate the usage of no-regret dynamics in game theory and then summarize results considering the convergence behavior of no-regret algorithms, particularly *follow the regularized leader* (FTRL).

4.1 General Results

John Nash famously proved in 1950 that for every finite game with mixed extensions, there exists at least one mixed Nash equilibrium (MNE), i.e. $MNE(\Gamma)$ is not empty [10]. This theorem became known as *Nash's Theorem*. Another important result is that no polynomial-time algorithm exists for computing Nash equilibria (NE). Finding a Nash equilibrium is *PPAD*-complete. That was first shown by Daskalakis, Goldberg, and Papadimitriou in games with at least three players [3] and later extended by Chen and Deng to two players [2]. That raises the question of whether Nash equilibria are learnable at all.

No-regret algorithms are recipes by which players update probabilities assigned to actions. So if all players employ a no-regret update policy, they could potentially learn, which means converge to pure and mixed Nash equilibria. Convergence can be formulated in various ways. We will differentiate between the players' empirical frequency of play, i.e. time-averaged convergence, and the actual sequence trajectories, last-iterate convergence in short. It was shown that under no-regret dynamics, the empirical frequency of play converges to the game's set of coarse correlated equilibria [4], a relatively weak game-theoretic solution concept. No-regret learning may cycle and weight only strictly dominated strategies [9]. Moreover, the *impossibility result* by Hart and Mas-Colell states that there exist no uncoupled dynamics which guarantee Nash convergence [5]. No-regret dynamics are, by construction, uncoupled in the sense that a player's update rule does not explicitly depend on the payoffs of other players. Therefore the *impossibility result* precludes Nash convergence of no-regret learning in general. That is consistent with the numerous negative complexity results in finding a Nash equilibrium [2, 3].

4.2 Sufficient Conditions for Nash Convergence

Despite these negative results, we empirically observe Nash convergence under no-regret algorithms for some games. In the following, I aim to provide an overview of sufficient conditions under which no-regret learning converges to a NE. In chapter 5 I try to give empirical evidence and visualize the results.

Similar to this thesis, in 2001, Jafari, Greenwald, Gondek, and Ercal studied the behavior of a concrete no-regret algorithm, namely Hedge, on simple finite games. They found that the algorithm's induced sequence of play converges to NE in dominance-solvable games such as the Prisoner's Dilemma, a game where a unique dominant strategy pure Nash equilibrium (PNE) exists [6]. They additionally suggest that in two-player zero-sum games, the empirical frequency of play converges to fully mixed NE under no-regret dynamics by employing the algorithm on games like Rock Paper Scissors and Matching Pennies. However, in the Shapley Games, a general-sum 3x3 game, the Hedge algorithm exhibits non-convergent exponential cycling behavior [6], which suggests that in general-sum games, no-regret algorithms fail to converge to NE in general.Later in 2016, Sandholm and Mertikopoulos formally proved these findings [8, Theorem 4.1 and Theorem 6.1].

Proposition 4.1 *Under FTRL, iteratively dominated strategies become extinct*

Proposition 4.2 *Under FTRL, the empirical frequency of play converges to Nash in two-player zero-sum games with interior equilibrium*

Considering the actual trajectories that players take, we know that in zero-sum games that admit an interior Nash equilibrium, trajectories are non-convergent for all initial strategy profiles other than the equilibrium [1]. More specifically, for 2x2 zero-sum games, even though the time-averaged strategies converge, the actual sequence of strategies are repelled away from the interior equilibrium and will eventually move towards the boundary of the probability simplex [1, Theorem 1].

More recent studies have shown that, in fact, only strict Nash equilibria survive under FTRL dynamics [4]. It turned out that strict NE are so-called *stable* states. Let us introduce the notion of stable states [7, Def. 4.2].

Definition 4.1 *A state $x^* \in \mathcal{X}$ is called variational stable (or simply stable) if there exists a neighborhood U of x^* such that*

$$\langle v(x), x - x^* \rangle \leq 0 \qquad \forall x \in U$$

with equality if and only if $x = x^*$.

In particular, if U can be taken to be all of $\mathcal{X}$, we say that x^* is *globally variationally stable* (or *globally stable* for short). In other words, if x^* is stable, then for all x in the neighborhood of x^*, the players' individual payoff gradients $v(x)$ "point towards" x^* in the sense that the angle between $x^* - x$ and $v(x)$ is acute. Furthermore, we can define *attracting* states [7, Def. 3.2.3].

Definition 4.2 *A state x^* is attracting if there is a neighbourhood U of x^* such that under FTRL the sequence of play $x^t \to x^*$ as $t \to \infty$ whenever $x^0 \in U$.*

We say that a state x^* is *asymptotically stable* if it is *stable* and *attracting* [7, Def. 3.2.4]. Based on this notion of stability, we can draw multiple implications.

It was shown that no interior point, and therefore no *fully mixed Nash equilibrium*, can be asymptotically stable under FTRL dynamics [4, Theorem 1].

Proposition 4.3 *A fully mixed NE cannot be asymptotically stable under FTRL*

Actually, only strict Nash equilibria can be stable under FTRL [4, Theorem 2]. In fact, a stable state is equivalent to a strict Nash equilibrium [9, Prop. 5.2]

Proposition 4.4 *The following are equivalent*

1. *x^* is stable*
2. *x^* is a strict Nash equilibrium*

In particular if x^* is *globally stable* then x^* is a unique Nash equilibrium [9, Prop.2.5].

Proposition 4.5 *If x^* is globally stable, it is the game's unique Nash equilibrium.*

As shown in [9, Theorem 4.7] and [9, Theorem 4.11] respectively, we furthermore have that the following two propositions hold when no-regret dynamics are employed.

Proposition 4.6 *The induced sequence of play converges globally to a globally stable equilibrium with probability* 1

Proposition 4.7 *If x^* is stable, then it is locally attracting with high probability*

In the next chapter, I would like to give empirical evidence for all the discussed findings. I will run two concrete no-regret algorithms on simple two-player games and visualize their convergence behavior.

5 Simulations

In this chapter, I want to give an intuition for the results on no-regret convergence in finite games discussed in chapter 4. I have implemented both, *projected online gradients ascent* (POGA) and *entropic gradient ascent* (EGA). As expected, both algorithms show very similar behavior. The *steep entropic regularizer* used in EGA slightly reshaped the players' trajectories in comparison to the common *Euclidean regularizer* in POGA. In terms of convergence, however, both behave the same. For that reason, I might show plots of only one algorithm. In the vector field plots, the black star (★) will denote an interior Nash equilibrium while the black dot (●) stands for a pure Nash equilibrium. The orange dot (●) denotes an initial strategy profile.

As the results suggest, I found Nash convergence in frequencies in two-player zero-sum games with an interior equilibrium and last-iterate convergence in games where strict Nash equilibria exist. I have limited myself to 2x2 and 3x3 bimatrix games, as higher-dimensional games with more than two players are hard to illustrate. Therefore we have $\mathcal{N} = \{1, 2\}$ throughout this chapter. The outcome of no-regret learning depends on the type of game and the type of equilibria. The chapter is structured accordingly.

5.1 Unique Fully Mixed Nash Equilibrium

Let us first consider games that yield an interior, or equivalently fully mixed, Nash equilibrium. In general, we would expect the algorithms' trajectories not to converge because there exists no pure and therefore no strict Nash equilibrium. In particular as stated in proposition 4.3, no fully mixed Nash equilibrium can be asymptotically stable and therefore no interior equilibrium can be attracting. However, at least in two-player zero-sum games, we expect the empirical frequency of play, i.e. time-averaged strategies, to converge to the interior Nash equilibrium (proposition 4.2).

5.1.1 Matching Pennies

Matching Pennies is a simple two-player zero-sum game. Both players choose between *Heads* and *Tails*, and if they match, then the row player wins, and if they mismatch, the column player wins. The payoff is set accordingly as in table 5.1.

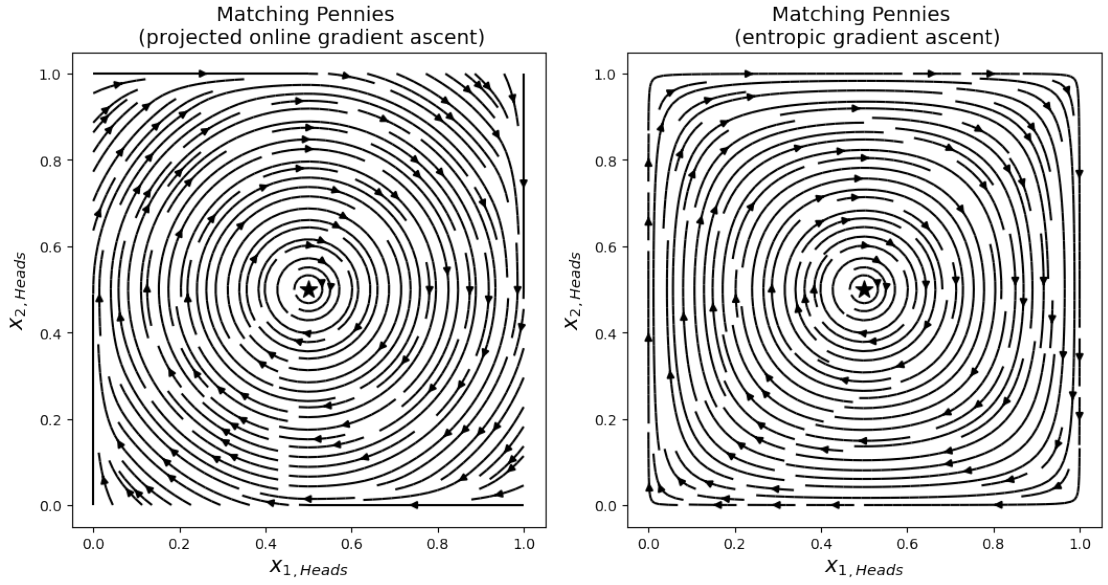
	$Heads$	$Tails$
$Heads$	$1, -1$	$-1, 1$
$Tails$	$-1, 1$	$1, -1$

Table 5.1: payoff matrix Matching Pennies

There is only a single fully mixed Nash equilibrium, i.e. when both players choose *Heads* and *Tails* equally likely.

$$x_i^* = (1/2, 1/2) \qquad \forall i \in \mathcal{N}$$

As stated in proposition 4.3 no fully mixed strategy, and therefore no interior Nash equilibrium, can be asymptotically stable under FTRL algorithms. Since FTRL can only converge to stable states or strict Nash equilibria equivalently (see proposition 4.4), for both EGA and POGA, the induced sequence of play fails to converge. Both exhibit cyclic behavior around the unique MNE as illustrated in the figure 5.1. As we can see, using entropic regularization, the projections always lead to interior strategies, whereas for Euclidean regularization, strategies are potentially projected to the boundary. Note that $x_{i,Tails}$ is implicitly given by $1 - x_{i,Heads}$.

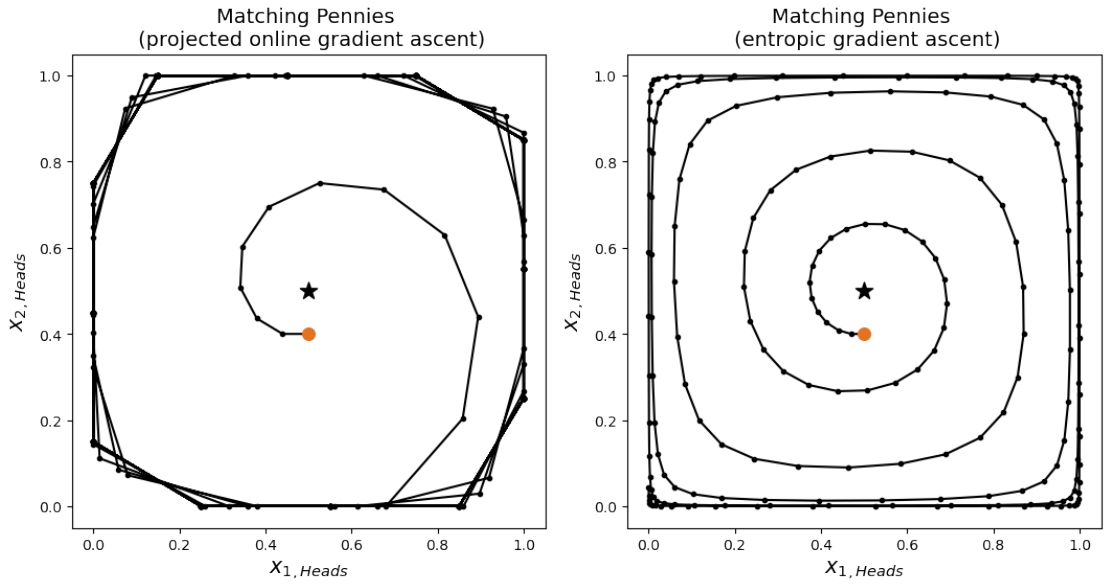


(a) POGA **(b)** EGA

Figure 5.1: vector field in Matching Pennies

The vector field might be misleading, as it looks like the algorithms cycle perfectly around the MNE but that would only hold for infinitesimal step size.

A closer examination of the specific trajectories, however, shows that actually for positive step size, they are repelled from the interior equilibrium as depicted in figure 5.2. Moreover, as mentioned earlier, in every 2x2 zero-sum game with interior equilibrium, trajectories diverge to the boundary under FTRL for all initial strategy profiles other than the equilibrium [1].



(a) POGA **(b)** EGA

Figure 5.2: trajectories with initial strategies $x_1^0 = (\frac{1}{2}, \frac{1}{2})$ and $x_2^0 = (\frac{2}{5}, \frac{3}{5})$, $T = 200$, $\gamma = 0.3$

As expected, the time-averaged trajectories, on the other hand, ultimately converge to the unique MNE. That happened independently from the initial strategies of the players. The amplitude of cycles dampens over time, as illustrated in figure 5.3. That finding perfectly fits to proposition 4.2.

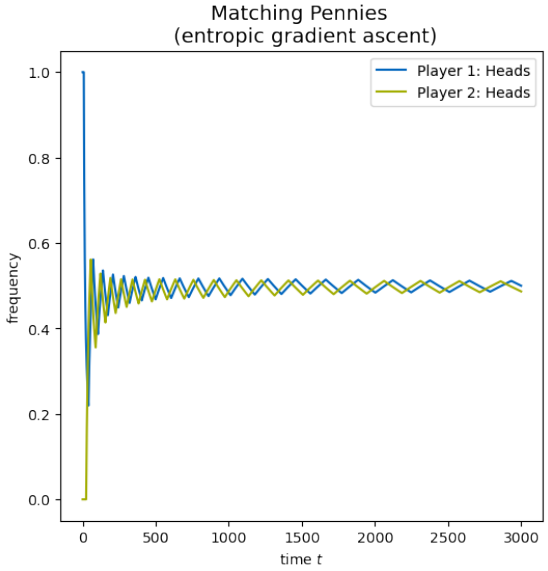


Figure 5.3: EGA empirical frequency of play in Matching Pennies, $\gamma = 0.1$

5.1.2 Rock Paper Scissors

One might think the convergence of empirical frequencies to the game's MNE is an artifact of its simple 2x2 structure, but I found similar behavior in Rock Paper Scissors, a 3x3 zero-sum game.

	Rock	*Paper*	*Scissors*
Rock	$0, 0$	$-1, 1$	$1, -1$
Paper	$1, -1$	$0, 0$	$-1, 1$
Scissors	$-1, 1$	$1, -1$	$0, 0$

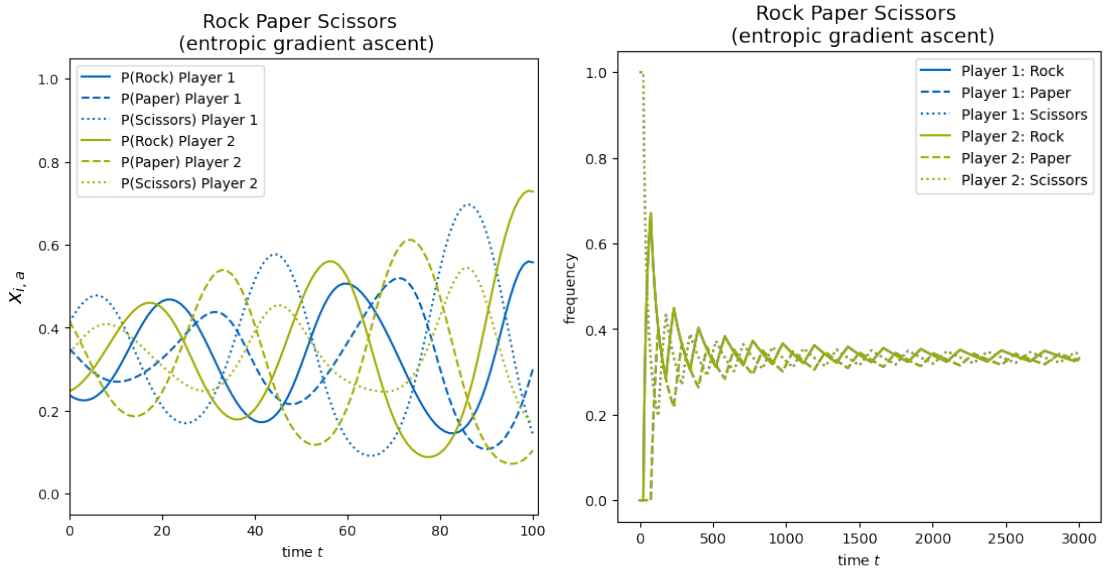
Table 5.2: payoff matrix Rock Paper Scissors

The only Nash equilibrium that exists is the following fully mixed NE.

$$x_i^* = (1/3, 1/3, 1/3) \qquad \forall i \in \mathcal{N}$$

Both algorithms exhibit out-of-sync oscillating behavior. Also, the cyclic period and amplitude increase over time, corresponding to the repelling behavior we observed in Matching Pennies. The players essentially chase one another. An illustration of that behavior is shown in figure 5.4a. Again, the empirical frequencies, on the other hand,

converge to the game's interior Nash equilibrium as in figure 5.4b. Note that the initial strategies are assigned randomly.



(a) last-iterate

(b) time-average

Figure 5.4: EGA behavior in Rock Paper Scissors, $\gamma = 0.1$

5.1.3 Shapley Game

The Shapley Game [6] resembles that of Rock Paper Scissors. The difference is that it is not zero-sum but general-sum. In the payoff matrix (table 5.3), we can see that the utility of an action profile sometimes sums up to 1 and sometimes to 0.

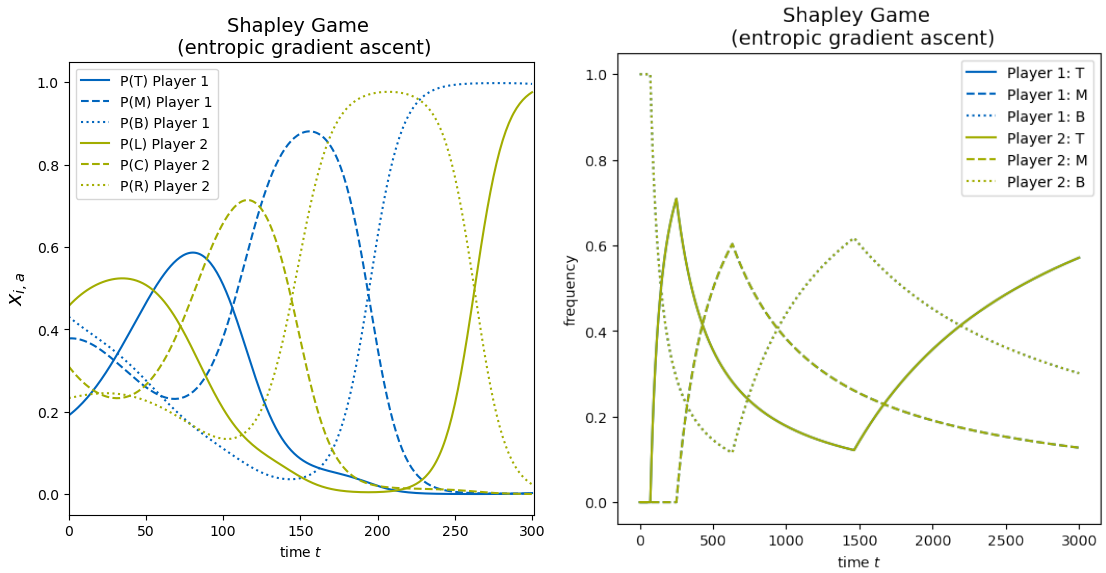
	L	C	R
T	$1, 0$	$0, 1$	$0, 0$
M	$0, 0$	$1, 0$	$0, 1$
B	$0, 1$	$0, 0$	$1, 0$

Table 5.3: payoff matrix Shapley Game

The game's unique NE is the same fully mixed Nash equilibrium as in Rock Paper Scissors.

$$x_i^* = (1/3, 1/3, 1/3) \qquad \forall i \in \mathcal{N}$$

In this case, both EGA and POGA are non-convergent neither in weights nor in frequencies (figure 5.5). Again the weights cycle exponentially through the space of possible strategies. As far as frequencies are concerned, the amplitudes of the cycles do not dampen over time as they did in Matching Pennies or Rock Paper Scissors. They rather grow exponentially. In general sum games, like the Shapley Game, no regret dynamics seem to fail to converge generally. The same behaviour was found for *fictitious play*, a simple *best response* dynamic that is not a no-regret algorithm [6].



(a) last-iterate

(b) time-average

Figure 5.5: divergent behavior of EGA in Shapley Game, $\gamma = 0.1$

5.2 Unique Pure Nash Equilibrium

Next, we consider games with a unique pure Nash equilibrium. When the PNE is globally stable (definition 4.1), then we expect the induced sequence of play to globally converge to the unique PNE (proposition 4.6).

5.2.1 Prisoner's Dilemma

The example I would like to address is the famous Prisoner's Dilemma. The game works as follows. Two bank robbers have been arrested. They are separated from each other, and both can choose to stay silent or betray the other one by admitting the crime. When both stay silent, both are sent to prison for only one year. When both betray, then they

get two years each. However, if one stays silent while the other betrays the one that stayed silent, goes to prison for three years while the other one is set free, see table 5.4.

	Silent	*Betray*
Silent	$-1, -1$	$-3, 0$
Betray	$0, -3$	$-2, -2$

Table 5.4: payoff matrix Prisoner's Dilemma

The only Nash equilibrium is when both players choose *Betray*. So there is no fully mixed Nash equilibrium but only one pure Nash equilibrium.

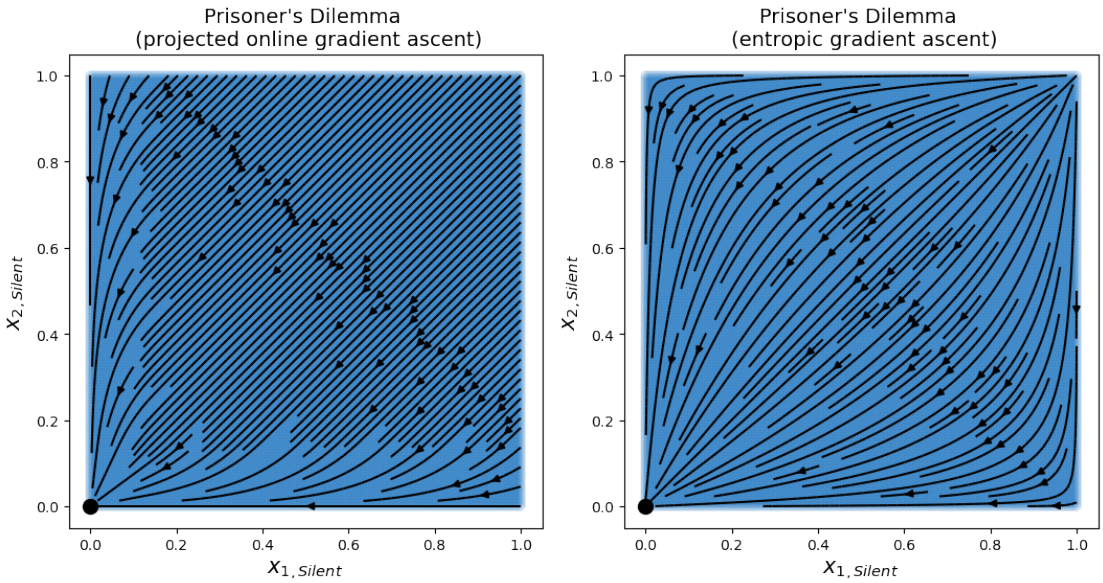
$$x^* = (Betray, Betray) \qquad \textit{strict} \text{ PNE}$$

Note that the PNE is also strict. Any unilateral deviation from the PNE would lead to a reduction in payoff. For instance, if the row player knows that the column player chooses *Betray*, then deviating from *Betray* would decrease the row player's payoff, from -2 to -3. As the game is symmetric, the same holds for the column player.

Another observation is that for both players *Silent* is strictly dominated by *Betray* in the sense of equation 3.2. More precisely, no matter what the column players chooses to play, the row player is always better of playing *Betray*, because $0 > -1$ and $-2 > -3$. Again, since the game is symmetric, the same holds for the column player. So by iteratively eliminating dominated strategies, we can solve this game. Such games are also called *dominance solvable*. As stated in proposition 4.1 we expect that for both algorithms the dominated strategy *Silent* becomes extinct, that means $x_{i,Silent}$ tends to 0 as the number of iterations grows.

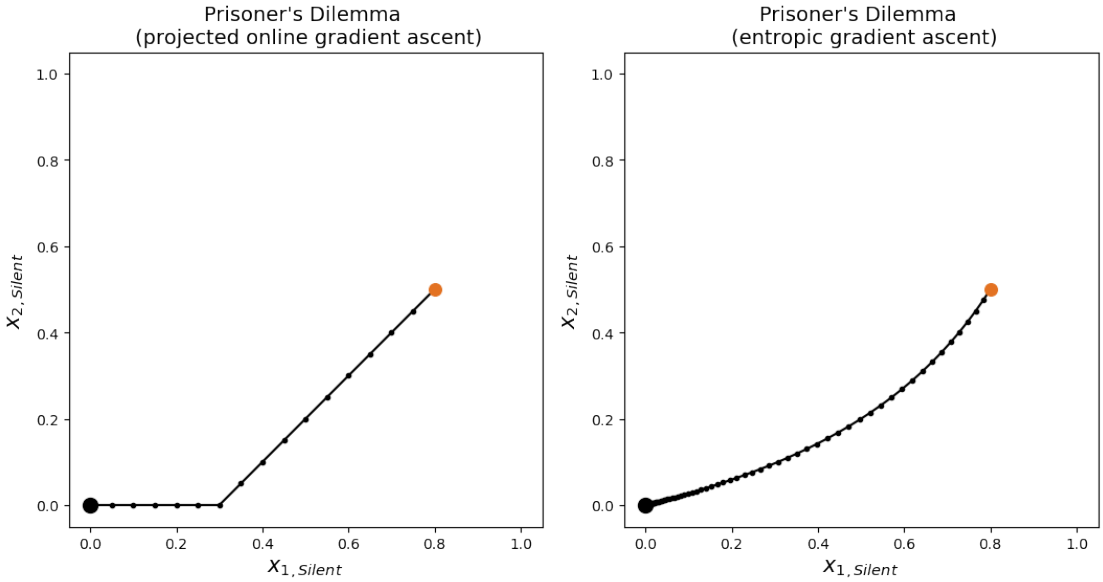
Consider figure 5.6. The blue region indicates strategies in the neighbourhood to the PNE that fulfill the inequality from the definition of stable states (4.1). We can see that the PNE is globally stable. As stated in proposition 4.6, FTRL converge globally to the globally stable equilibrium. As the vector field suggests, both POGA and EGA indeed converge globally to the game's unique PNE. Also, the dominated strategy *Silent* becomes extinct.

Figure 5.7 illustrates some specific trajectories that the algorithms take. We find that EGA does not reach the boundary as the projections always lead to interior strategies under entropic regularization. For OPGA, on the other hand, the Euclidean projections lead to strategies on the boundary at some point. Interestingly, using the same step size of $\gamma = 0.1$, POGA converges much faster than EGA.



(a) POGA **(b)** EGA

Figure 5.6: vector field in Prisoner's Dilemma with stable neighbourhood w.r.t strict PNE



(a) POGA **(b)** EGA

Figure 5.7: trajectories with initial strategies $x_1^0 = (\frac{4}{5}, \frac{1}{5})$ and $x_2^0 = (\frac{1}{2}, \frac{1}{2})$, $T = 200$, $\gamma = 0.3$

5.3 Mixed and Pure Nash Equilibria

In games where both fully mixed and pure Nash equilibria exist, we expect no-regret algorithms to converge to strict pure Nash equilibria solely. We will also see that as mentioned in proposition 4.3 interior equilibria are not asymptotically stable.

5.3.1 Battle Of Sexes

Image a couple of two persons with different interests. One would prefer to watch a boxing fight, say the row player, and the other one prefers to go to a ballet, the column player. Nevertheless, they would rather spend time together than choose different events. There is no communication between both. The payoff is set accordingly as in table 5.5.

	$Fight$	$Ballet$
$Fight$	$3, 2$	$0, 0$
$Ballet$	$0, 0$	$2, 3$

Table 5.5: payoff matrix Battle of Sexes

There are two quite obvious pure Nash equilibria. One where both choose *Fight*, and one where both choose *Ballet*. Moreover, there is also a fully mixed Nash equilibrium, i.e. when both players randomize over the actions. In particular, the row player should choose *Fight* with probability 3/5 and *Ballet* with 2/5 and the column player should choose *Fight* with 2/5 and *Ballet* with 3/5. In formulas, we have the following Nash equilibria.

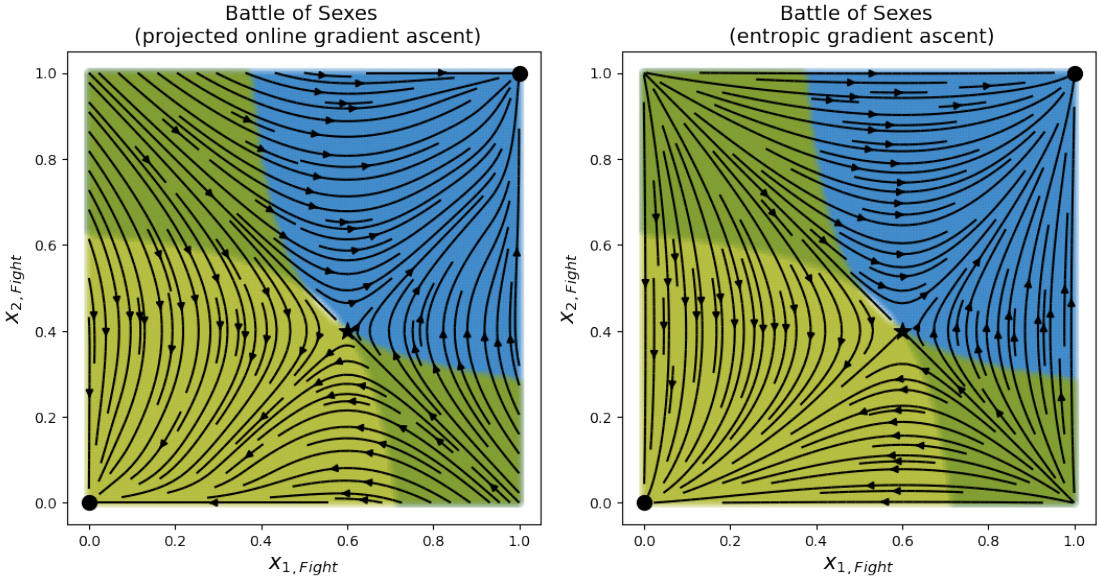
$$x^* = (Fight, Fight) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (Ballet, Ballet) \qquad \textit{strict} \text{ PNE}$$

$$x_1^* = (3/5, 2/5) \qquad x_2^* = (2/5, 3/5) \qquad \text{MNE}$$

Note that again both PNE are strict in the sense of definition 3.3. Neither the row player nor the column player can unilaterally deviate from a PNE without reducing its payoff. According to proposition 4.4 that means both PNE are also stable states as defined in 4.1. Then both PNE must also be locally attracting (proposition 4.7).

Indeed, as shown in figure 5.8, both algorithms converge locally to the corresponding PNE. Notice how the stable regions of the PNE intersect. The attracting regions for both PNE can be implicitly derived from the vector field.

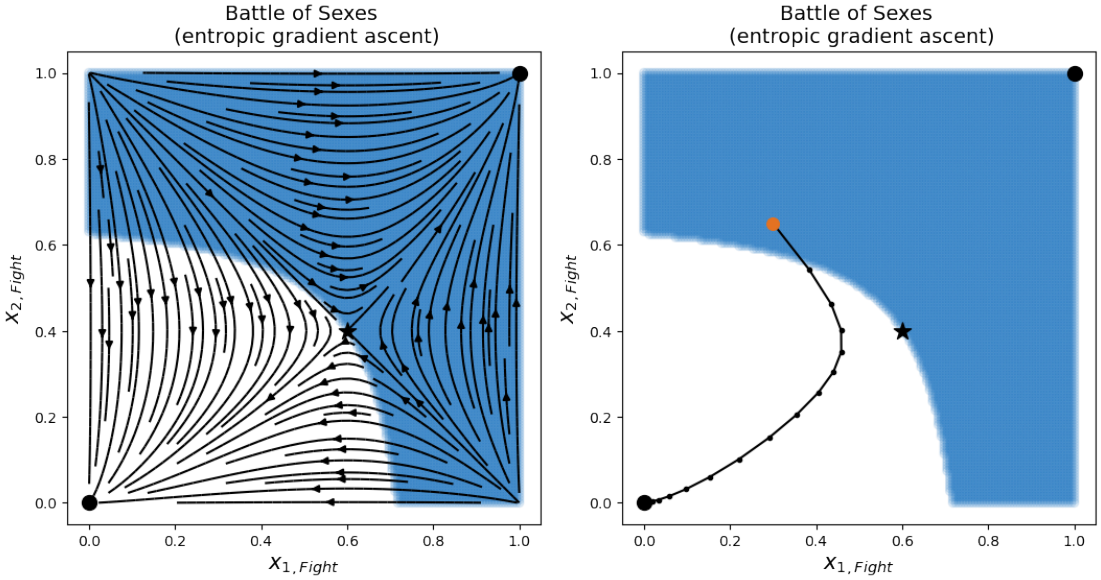


(a) POGA **(b)** EGA

Figure 5.8: vector field in Battle of Sexes, the blue region indicates the stable neighbourhood w.r.t *(Fight,Fight)* and the light green region w.r.t to *(Ballet,Ballet)*, the dark green region is the intersection of both

For clarification I have highlighted only the stable neighbourhood with respect to the PNE *(Fight,Fight)* in figure 5.9a. Also, notice that not all points in the stable region converge to the corresponding PNE. For an example refer to example 5.9b. While the PNE *(Fight,Fight)* is attracting for all initial strategies that are above the diagonal between *(Fight,Ballet)* and *(Ballet,Fight)*, the PNE *(Ballet,Ballet)* is attracting for all initial strategies below that diagonal. We can conclude that the stable neighbourhood is not necessarily equal to the attracting neighbourhood in general.

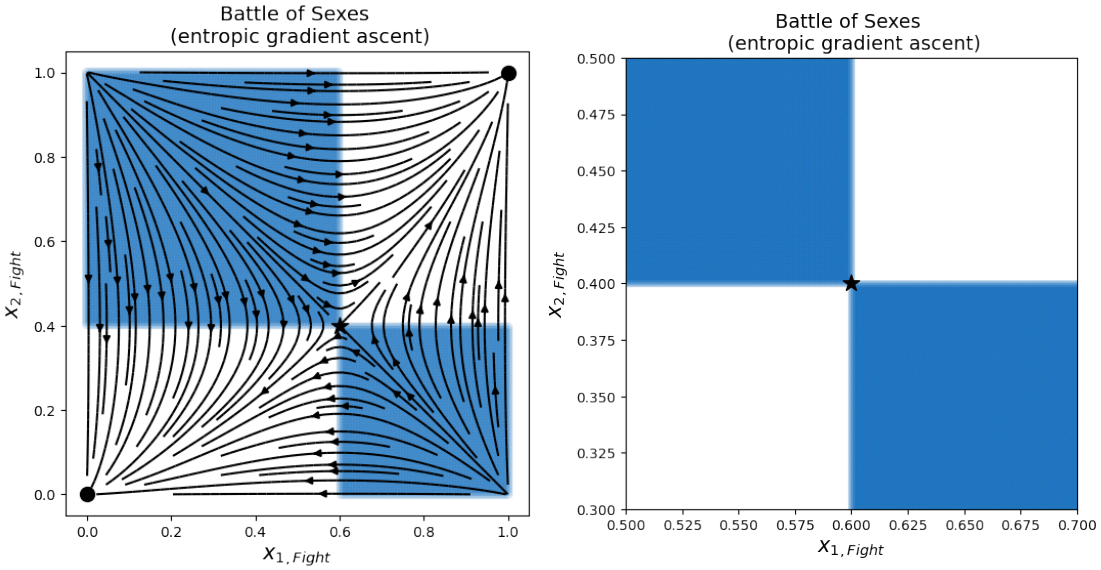
The question might arise whether the fully mixed Nash equilibrium can be stable as well. Even though there is a stable region for the MNE as depicted in 5.10a, we cannot find a neighbourhood for the MNE such that the inequality of the stability definition (4.1) holds for all strategies within the neighbourhood. Intuitively, no matter how far we zoom in to the MNE, we cannot draw a circle around it such that all points within the circle are blue as illustrated in figure 5.10b. Therefore except from the perfect diagonal between *(Fight,Ballet)* and *(Ballet,Fight)* the algorithms never converges towards the interior Nash equilibrium. The same behavior was observed for POGA. That is align with proposition 4.3, stating that no interior point can be asymptotically stable under FTRL.



(a) stable neighbourhood w.r.t *(Fight,Fight)*

(b) $x_1^0 = (0.3, 0.7)$, $x_2^0 = (0.65, 0.35)$, $T = 200$

Figure 5.9: EGA in Battle of Sexes, $\gamma = 0.1$



(a) stable neighbourhood w.r.t interior equilibrium

(b) zoomed in to the interior equilibrium

Figure 5.10: The interior equilibrium is not asymptotically stable under EGA in Battle of Sexes

5.3.2 Intersection Game

Let us revisit the Intersection game from subsection 3.3.2. The game involves two car drivers that need to cross an intersection without a crash. They can either *Stop* or *Go*. The driver's payoff is straightforward as in table 5.6.

	Stop	*Go*
Stop	$0, 0$	$0, 1$
Go	$1, 0$	$-100, -100$

Table 5.6: payoff matrix Intersection game

Similar to the Battle of Sexes, the game has two pure Nash equilibria, namely *(Stop, Stop)* and *(Go,Go)*, and a single fully mixed Nash equilibrium where both drivers choose to *Stop* with probability 100/101 and *Go* with probability 1/101. To sum up, we have the following Nash equilibria.

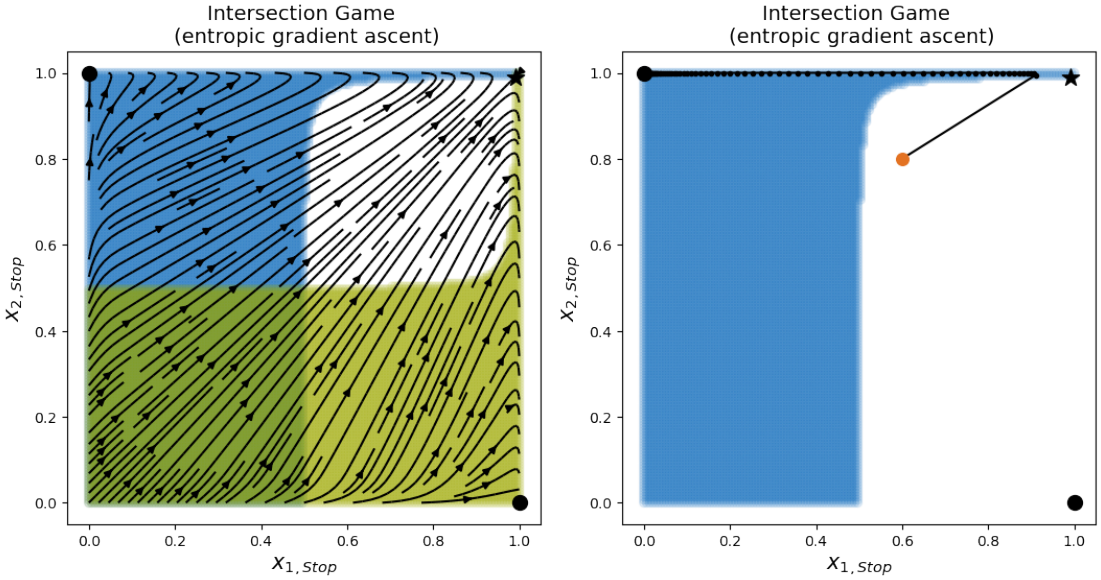
$$x^* = (Go, Go) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (Stop, Stop) \qquad \textit{strict} \text{ PNE}$$

$$x_1^* = x_2^* = (100/101, 1/101) \qquad \text{MNE}$$

Again it is easy to check that both PNE are strict and therefore stable states (proposition 4.4). In figure 5.11a, the stable neighbourhoods for both PNE are colored. We can see that both PNE are indeed locally stable states. Like expected, both no-regret algorithms converge locally to the corresponding PNE (proposition 4.7). In fact, both algorithms converge to the PNE that is the "closest" one from the initial strategy, just like in Battle of Sexes, see figure 5.11a. So the attracting regions are again divided by the diagonal between *(Stop,Stop)* and *(Go,Go)*.

Note that the MNE denoted by the star in the top right corner is indeed a fully mixed NE even though it looks like a PNE. The figure might be misleading as it seems the algorithm converges to the MNE. Even though the MNE seems to be attracting, both algorithms eventually converge to one of the strict PNE for all initial strategies. I have plotted a single trajectory using EGA in figure 5.11b to clarify that. Interestingly, the attracting neighbourhood contains initial strategies that are not included in the stable neighbourhood. So, in contrast to Battle of Sexes, the attracting neighbourhood is not a subset of the stable neighbourhood. An example for such an initial strategy is depicted in the same figure 5.11b.



(a) stable neighbourhood w.r.t *(Go,Stop)* in blue and w.r.t to *(Stop,Go)* in light green

(b) $x_1^0 = (0.6, 0.4)$, $x_2^0 = (0.8, 0.2)$, $T = 200$

Figure 5.11: EGA behavior in the Intersection game, $\gamma = 0.1$

Just like in Battle of Sexes, we cannot find a neighbourhood for the MNE such that the equation of the stability definition 4.1 is fulfilled. As we zoom in to the MNE again, we cannot draw a circle around the MNE such that all points within the circle are colored. The zoomed-in plot is analogue to 5.10b. The same observations were made for POGA.

5.3.3 Coordination Game

The behavior of the Coordination game under no-regret dynamics has already been studied in [6] but other no-regret algorithms than EGA and OPGA were used. As the name suggests, both players aim to cooperate, see table 5.7.

	L	C	R
T	$3, 3$	$0, 0$	$0, 0$
M	$0, 0$	$2, 2$	$0, 0$
B	$0, 0$	$0, 0$	$1, 1$

Table 5.7: payoff matrix Coordination game

There are three pure Nash equilibria and multiple fully mixed Nash equilibria. As we have seen in the previous examples, fully mixed Nash equilibria are not asymptotically stable and therefore not learnable in general-sum games under no-regret dynamics. For that reason, we will neglect interior equilibria from now on. Instead, let us consider the following three PNE.

$$x^* = (T, L) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (M, C) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (B, R) \qquad \textit{strict} \text{ PNE}$$

Obviously all of them are strict PNE as any unilateral deviation from an PNE results in a decrease in payoff. Note that *(B,R)* is *pareto dominated* by *(M,C)* which is once again *pareto dominated* by *(T,L)*, so *(T,L)* is *pareto optimal*. Refer to section 3.1 to recall pareto optimality.

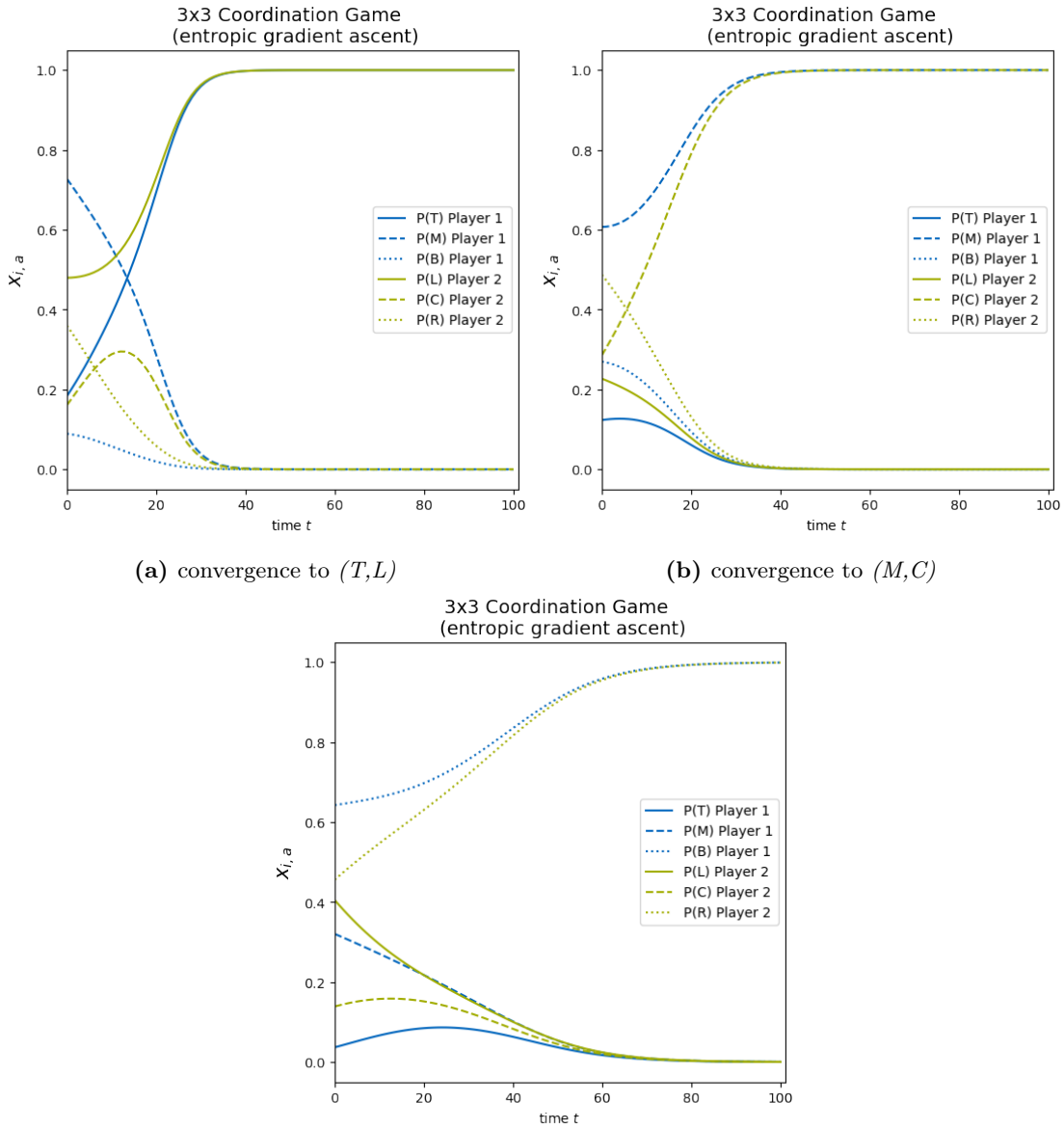
One might assume that for any general-sum game that is not zero-sum no-regret algorithms do not converge as we observed in the Shapley Game in subsection 5.1.3. However, in the Coordination game, I found convergence to one of the above PNE for all initial strategies.

I have randomized the players' initial strategies and actually most of the time both algorithms converged to the *pareto optimal* PNE *(T,L)* (figure 5.12a). Less often I found convergence to the PNE *(M,C)* (figure 5.12b) and even less often to the PNE *(B,R)* (figure 5.12c).

Unfortunately, I could not find a pattern for which initial strategy the algorithms converge to a specific PNE. However, the fact that most of the time, it converges to *(T,L)* might probably have something to do with its *pareto optimality*. For a relative frequency distribution, I have empirically observed, see table 5.8.

(T,L)	*(M,C)*	*(B,R)*
0.64	0.28	0.08

Table 5.8: relative frequencies of convergence in the Coordination game, sampled 10000 random initial strategies



(a) convergence to *(T,L)*

(b) convergence to *(M,C)*

(c) convergence to *(B,R)*

Figure 5.12: EGA behavior in the Coordination game, $\gamma = 0.1$

The reason why both algorithms converge to Nash equilibria in the Coordination game and diverge in the Shapley Game is that there exists no strict NE in the Shapley Game but only an interior equilibrium. As we have concluded in chapter 4 only strict NE survive under no-regret dynamics, which means that in games where no strict NE exists, we need

to expect that the induced sequence of play diverges in general, as it did in the Shapley Game. Note that the empirical frequency of play might still converge even though no strict NE exists as we observed in the two-player zero-sum games Matching Pennies and Rock Paper Scissors.

5.4 Weak Pure Nash Equilibria

Lastly, I would like to address games with a pure weak Nash equilibrium. As we have seen before, the trajectories of both algorithms never converge to fully mixed Nash equilibria which are weak equilibria per definition. That goes align with proposition 4.3 stating that no interior equilibrium can be asymptotically stable under FTRL. Nevertheless, games can be constructed to have pure weak Nash equilibria. When there is another strict PNE besides that, the strict one is locally attracting. Interestingly, I found convergent behavior for initial strategies "close" to the weak PNE. However, they do not converge to the weak PNE but rather somewhere to the boundary of the simplex where one player chooses a pure strategy while the other plays a mixed strategy.

5.4.1 Strict and Weak 2x2

Consider the following 2x2 payoff matrix in table 5.9.

	H	T
H	$2,3$	$1,2$
T	$1,2$	$2,2$

Table 5.9: payoff matrix strict and weak pure Nash equilibria 2x2

The game yields one mixed and two pure Nash equilibria. As the MNE is again not asymptotically stable we will focus on the two PNE.

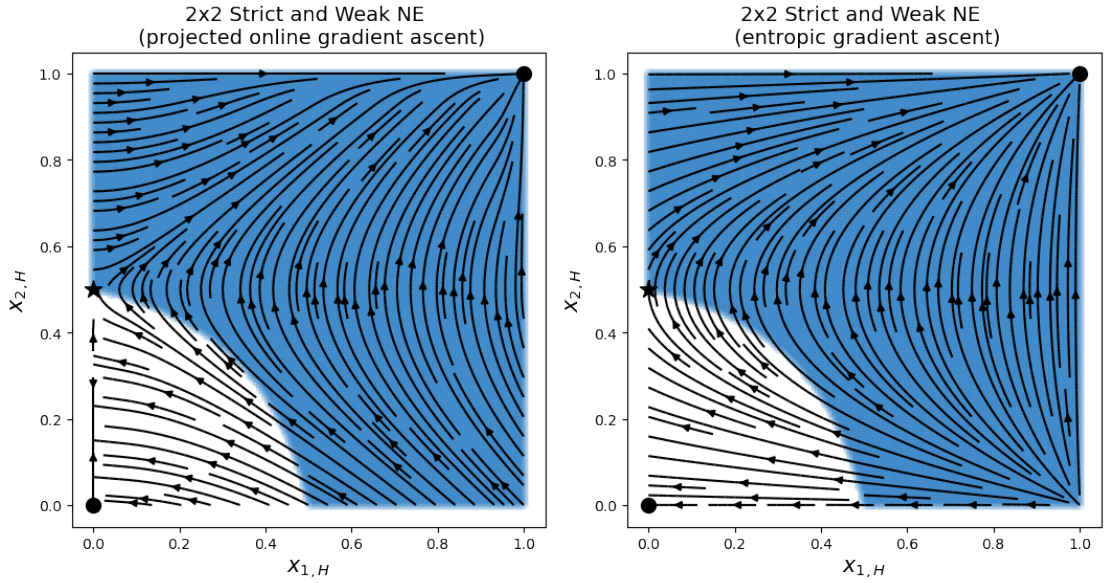
$$x^* = (H, H) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (T, T) \qquad \textit{weak} \text{ PNE}$$

Let us first look at *(H,H)*. It is strict in the sense of definition 3.3 as any unilateral deviation leads to a strict decrease in payoff. For instance, if the row player expects the column player to play H then deviating from H to T would lead to a reduced payoff from 2 to 1. Similarly, expecting the row player to play H, the column player's payoff decreases from 3 to 2 if the column player deviates from H to T. Therefore *(H,H)* is a strict PNE.

The strategy profile *(T,T)* on the other hand, is weak. A PNE is called weak when no player has the incentive to deviate unilaterally. However, if they deviate, they do not necessarily reduce their payoff, or equivalently, there is more than one unique best response to the PNE. In the specific game above *(T,T)* is weak because expecting the row player to play T the column player can deviate from T to H without reducing its payoff as $2 = 2$. Note that *(T,T)* is still a PNE as the payoff also does not strictly increase when deviating.

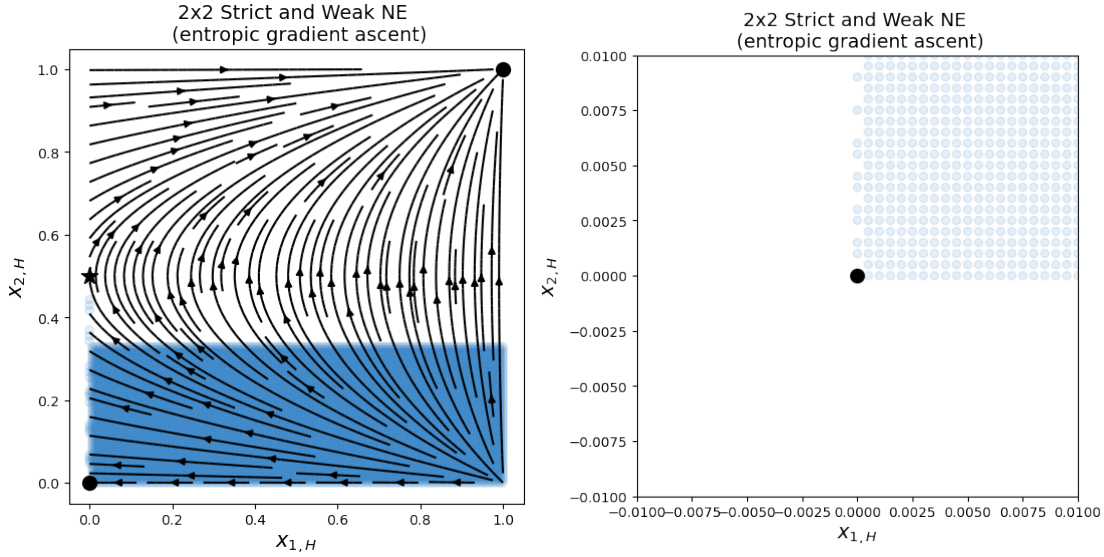
Now I would like to validate the results from chapter 4 that only strict PNE survive under no-regret dynamics. For the strict PNE *(H,H)* I found local stability as depicted by the blue region in figure 5.13. As expected, both algorithms converge to the strict PNE locally.



(a) POGA **(b)** EGA

Figure 5.13: vector field with stable neighbourhood w.r.t the strict PNE *(H,H)* colored in blue

A closer look at the weak PNE *(T,T)* shows that it is not stable in the sense of definition 4.1. There are always strategies in the neighbourhood of the weak PNE that did not fulfill the inequality of the definition. Especially strategies profile where the row player chooses the pure strategy T, there are "gaps" that were not stable for the weak PNE, no matter how much we zoom in, see figure 5.14b. Blue points indicate strategy profiles for which the inequality from the stability definition holds. This observation fits proposition 4.4 saying that only strict PNE are stable. The same results were found for POGA.



(a) stable neighbourhood w.r.t *(T,T)*

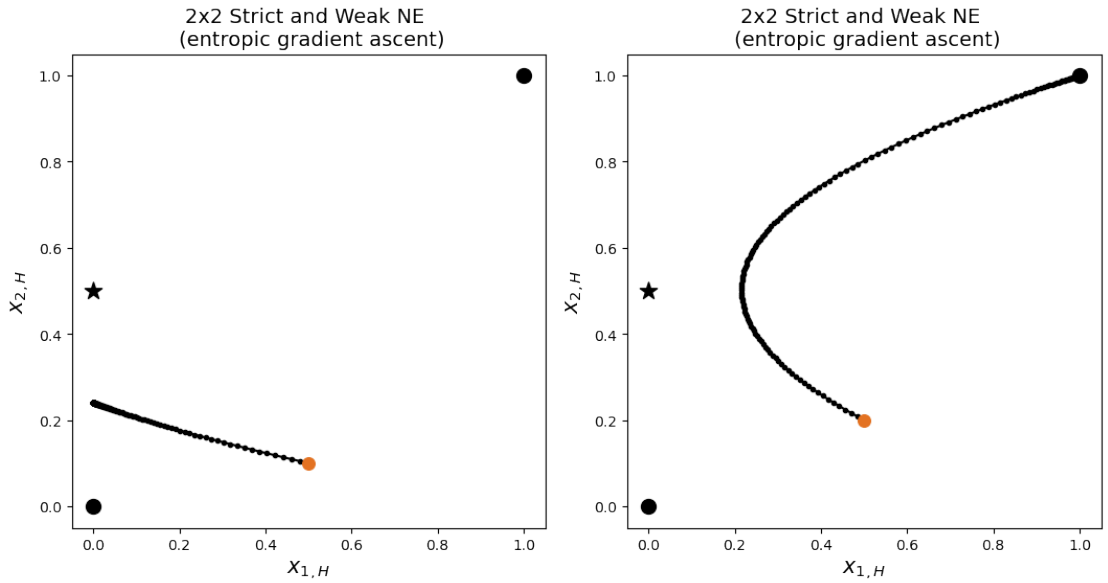
(b) zoomed in to *(T,T)*

Figure 5.14: The weak PNE *(T,T)* is not stable under EGA

The vector fields might be misleading, though, as it looks like the EGA algorithm steps outside the feasible probability simplex. However, a closer examination showed that for some initial strategies that are locally close to the weak PNE, the EGA algorithm converges towards the "left wall". An example trajectory for that phenomenon is shown in figure 5.15a. No matter how many iterations were used, the EGA algorithm did not change its direction. Moving the initial strategy slightly closer towards the strict PNE, however, yields convergence towards the strict equilibrium as illustrated in 5.15b. Again, the same holds for POGA.

So indeed, the strict and therefore stable PNE is also locally attracting (proposition 4.7). Nevertheless, even though there exists a unique strict PNE, as in the Prisoner's Dilemma, this game shows that no-regret dynamics do not necessarily converge to it globally. It seems like the existence of the weak PNE is disrupting the convergence to the strict PNE. Also, it is worth mentioning that for no initial strategy, I have found convergence to the weak PNE but rather convergence towards the "left wall". I came to the same conclusions for both algorithms.

In future research, it might be interesting to look at these strategies on the "left wall" more closely. A reasonable explanation for this behavior could be that the underlying probability distributions are correlated or even coarse correlated equilibria.



(a) $x_1^0 = (0.5, 0.5)$, $x_2^0 = (0.1, 0.9)$, $T = 200$ **(b)** $x_1^0 = (0.5, 0.5)$, $x_2^0 = (0.2, 0.8)$, $T = 200$

Figure 5.15: convergent behavior using EGA, $\gamma = 0.1$

5.4.2 Strict and Weak 3x3

The next game is a 3x3 general-sum game with two strict and one weak pure Nash equilibrium. There are also multiple fully mixed Nash equilibria. However, they are neglected again as the sequence of play never converges to one of them for the very same reason that interior states cannot be asymptotically stable (proposition 4.3). Consider the 3x3 payoff matrix described in table 5.10.

	A	B	C
X	$2, 3$	$1, 2$	$1, 1$
Y	$1, 1$	$2, 1$	$3, 2$
Z	$1, 2$	$2, 2$	$2, 1$

Table 5.10: payoff matrix strict and weak pure Nash equilibria 3x3

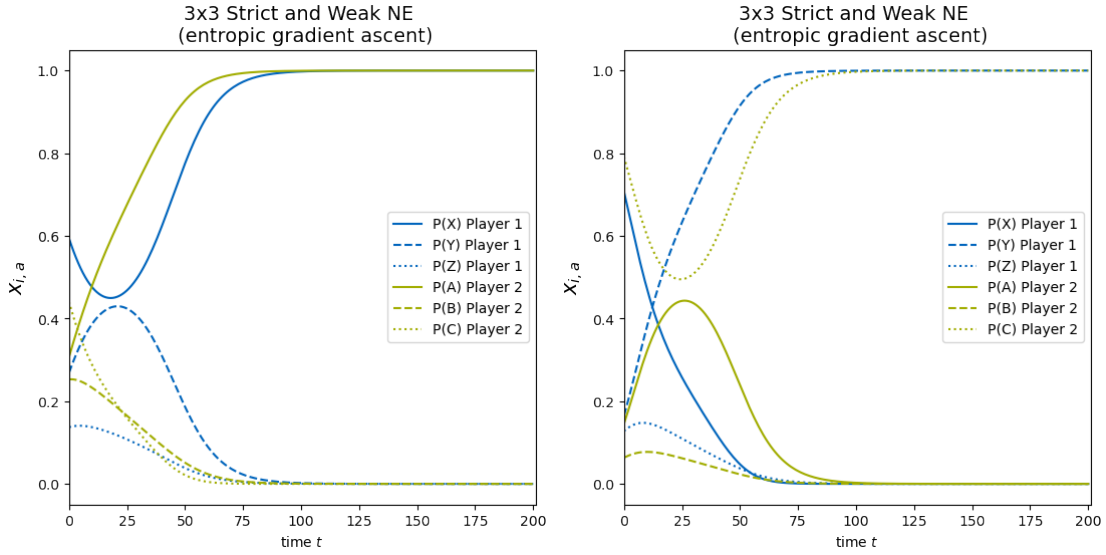
$$x^* = (X, A) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (Y, C) \qquad \textit{strict} \text{ PNE}$$

$$x^* = (Z, B) \qquad \textit{weak} \text{ PNE}$$

As mentioned above the game yields three pure Nash equilibira. One can easily check that *(X,A)* and *(Y,C)* are strict as they are unique best responses. However, *(Z,B)* is not a unique best response for the column player. Assuming the row player chooses Z, the column player can deviate from B to A without losing any payoff. The payoff for the column player stays at 2 for that deviation. Therefore *(Z,B)* is weak.

As far as convergence is concerned, I found similar results as in the 2x2 game discussed previously. The initial strategies were randomized. In most cases, I found convergence of both EGA and OPGA to one of the strict PNE. Figure 5.16 shows an example for convergence to *(X,A)* and *(Y, C)*, respectively.



(a) convergence to *(X,A)* **(b)** convergence to *(Y,C)*

Figure 5.16: Convergent behavior for strict PNE using EGA, $\gamma = 0.1$

For non of the tested initial strategy profiles, I found convergence to neither the weak PNE *(Z,B)* nor any of the fully mixed NE. However, both algorithms do converge sometimes to states that are no Nash equilibria. Figure 5.17 illustrates that behavior.

Note that the initial strategies here are not too "far" from the weak PNE *(Z,B)*. The behavior corresponds to the convergence towards the "left wall" from figure 5.15a. Again, these empirical findings match the general result that only strict Nash equilibria survive under no-regret dynamics.

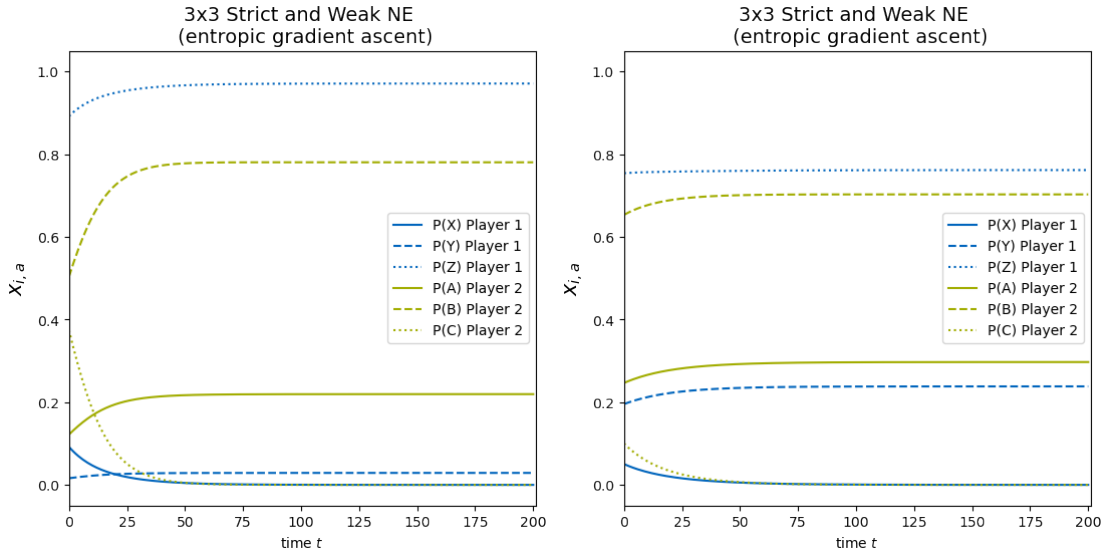


Figure 5.17: Convergent behavior to states that are no Nash equilibria using EGA, $\gamma = 0.1$

5.4.3 Weak 2x2

Lastly, we consider a game with no strict but a weak pure Nash equilibrium. Consider the following payoff matrix in table 5.11.

	H	T
H	$1, 1$	$1, 2$
T	$0, 2$	$2, 2$

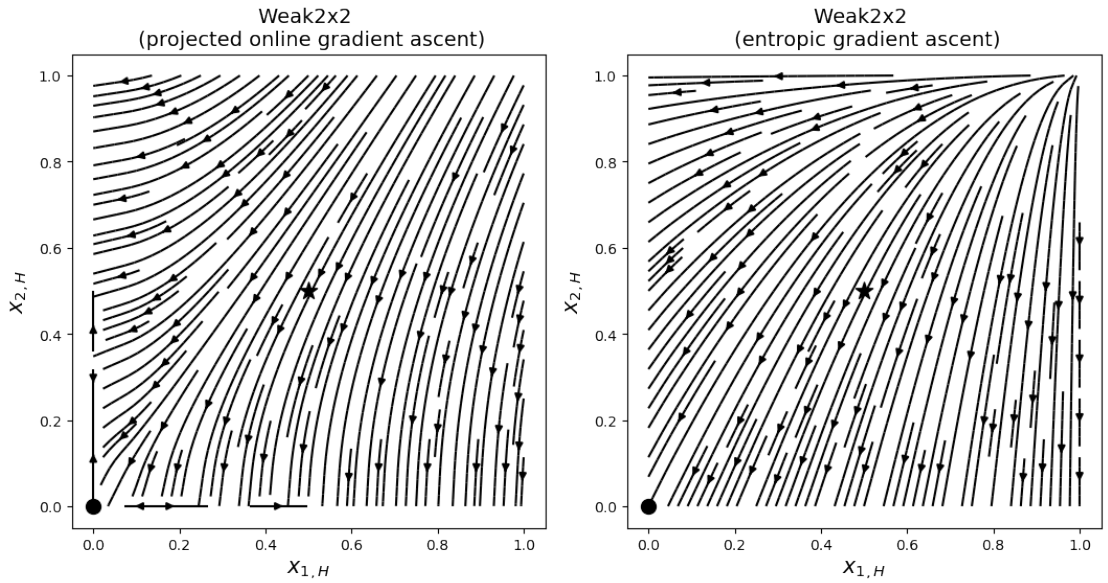
Table 5.11: payoff matrix weak pure Nash equilibrium 2x2

Note that this time there is no strict PNE. However, (T, T) is weak because it is not the unique best response for both players. For example, assuming the row player chooses T, then the column player can deviate from T to H without reducing its utility. Also, there is a mixed Nash equilibrium where both players select each action equally likely.

$$x^* = (T, T) \qquad \textit{weak} \text{ PNE}$$

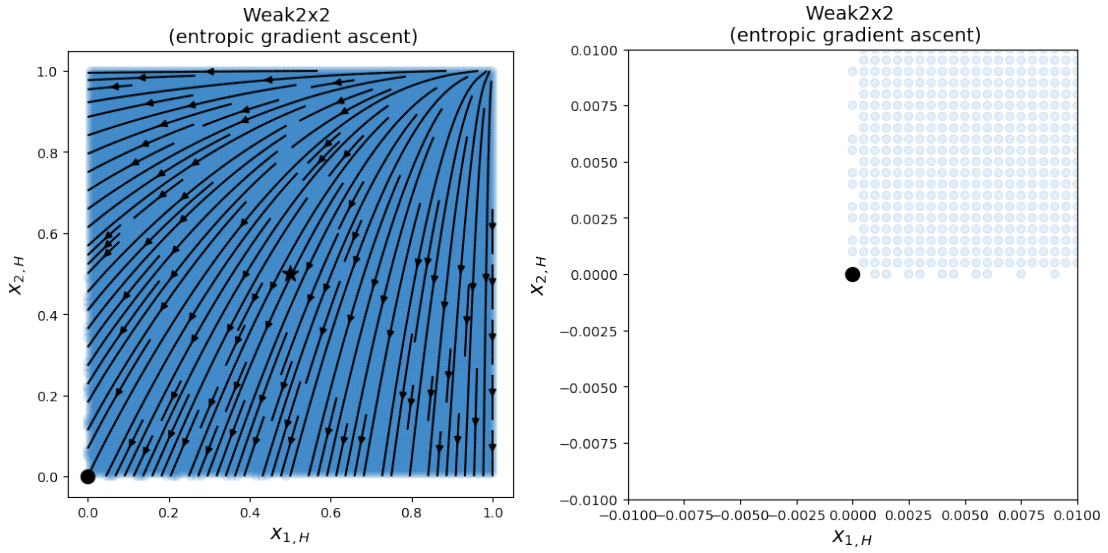
$$x_1^* = (1/2, 1/2) \qquad x_2^* = (1/2, 1/2) \qquad \text{MNE}$$

Consider the following two figures 5.18 and 5.19.



(a) POGA **(b)** EGA

Figure 5.18: vector field in a game with a weak PNE



(a) global perspective **(b)** zoomed in to (T, T)

Figure 5.19: stable neighbourhood with respect to (T, T)

Interestingly, the plots suggest that there is no convergent behavior to Nash for any initial strategy. Neither the interior nor the weak PNE is attracting, see figure 5.18. Instead, both algorithms converge to the boundary where only one player selects the pure strategy T.

When we look at trajectories, we have similar behavior as described previously in figure 5.15a. Considering figure 5.19a it appears like the weak PNE is globally stable. However, as we zoom in, we find these "gaps" on the boundary indicating that the weak equilibrium is not stable, see figure 5.19b.

6 Conclusion

To sum up, we find that indeed only strict Nash equilibria survive under no-regret dynamics. If we average trajectories, we also have Nash convergence for two-player zero-sum games with an interior equilibrium.

For future research, it would be interesting to study weak pure Nash equilibria in more detail. We observe convergent behavior of no-regret algorithms in games with weak pure Nash equilibria, but they do not converge to Nash but rather to the boundary. As we know, no-regret learning generally converges to the game's set of coarse correlated equilibria. Therefore a possible explanation for that strange convergence behavior could be that the outcome is actually a correlated or coarse correlated equilibrium.

Moreover, it might be interesting to make an in-depth comparison between the notions of *attracting* and *stable*. The Intersection game showed that the attracting neighbourhood does not necessarily need to be a subset of the stable neighborhood for some equilibrium.

Additionally, one could compare the convergence rates for different no-regret algorithms and tune the step size accordingly. Maybe it might also be worth extending the visualizations to three-player settings, even though it could be a bit tricky.

More generally speaking, Nash equilibria as the archetype of solution concepts can be questioned overall as it is so hard to compute and only learnable under strong assumptions.

Bibliography

[1] J. Bailey and G. Piliouras. "Fast and furious learning in zero-sum games: Vanishing regret with non-vanishing step sizes." In: *Advances in Neural Information Processing Systems* 32 (2019).

[2] X. Chen and X. Deng. "Settling the complexity of two-player Nash equilibrium." In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE. 2006, pp. 261–272.

[3] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. "The complexity of computing a Nash equilibrium." In: *SIAM Journal on Computing* 39.1 (2009), pp. 195–259.

[4] L. Flokas, E.-V. Vlatakis-Gkaragkounis, T. Lianeas, P. Mertikopoulos, and G. Piliouras. "No-regret learning and mixed Nash equilibria: They do not mix." In: *arXiv preprint arXiv:2010.09514* (2020).

[5] S. Hart and A. Mas-Colell. "Uncoupled dynamics do not lead to Nash equilibrium." In: *American Economic Review* 93.5 (2003), pp. 1830–1836.

[6] A. Jafari, A. Greenwald, D. Gondek, and G. Ercal. "On no-regret learning, fictitious play, and nash equilibrium." In: *ICML*. Vol. 1. 2001, pp. 226–233.

[7] P. Mertikopoulos. "Online optimization and learning in games: Theory and applications." PhD thesis. Grenoble 1 UGA-Université Grenoble Alpes, 2019.

[8] P. Mertikopoulos and W. H. Sandholm. "Learning in games via reinforcement and regularization." In: *Mathematics of Operations Research* 41.4 (2016), pp. 1297–1324.

[9] P. Mertikopoulos and Z. Zhou. "Learning in games with continuous action sets and unknown payoff functions." In: *Mathematical Programming* 173.1 (2019), pp. 465–507.

[10] J. Nash. "Non-cooperative games." In: *Annals of mathematics* (1951), pp. 286–295.

[11] S. Shalev-Shwartz et al. "Online learning and online convex optimization." In: *Foundations and trends in Machine Learning* 4.2 (2011), pp. 107–194.

[12] Y. Viossat and A. Zapechelnyuk. "No-regret dynamics and fictitious play." In: *Journal of Economic Theory* 148.2 (2013), pp. 825–842.