Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Διπλωματική Εργασία για το

Μ.Δ.Ε. στην Ραδιοηλεκτρολογία/Ηλεκτρονική

Φτάνοντας σε Ισορροπία: Μάθηση και Θεωρία Παιγνίων

Ελευθέριος Λαμπίρης
ΑΜ: 2012109

Επιβλέπων: Άρης Λ. Μουστάκας

Αθήνα 2015

Εξεταστική Επιτροπή

Α. Μουστάκας Α. Πολύδωρος Χ. Ευθυμιόπουλος

Forward

Let us assume two merchants meeting in the middle of a desert, both bearing goods. Since their profession is to trade they try to exchange these goods.

But there appears to be a problem. Both have different valuations of their products' prices relative to the other trader's goods. As a result, they both agree to do business in an alternative way. They start with a number of products, each product with a fixed quantity. Then, they choose one of the products and trade simultaneously.

After the exchange both receive a reward (a measurement of "happiness") depending on their evaluations. If the game is played only once, then they would both try to maximize their one shot payoff. On the other hand, if the game continues for more rounds, then, their intention is to explore their options and end up with as better products as possible.

The above two interactions are completely different. The first one is a one-shot process and is described by Game Theory, i.e., the interactions of rational agents. There, an agent is looking to achieve a high score by implementing in a game only once.

On the other hand, a repeated interaction needs a new framework. In this case, merchants have an abundance of time to try out choices and shoot for maximization of their long-term payoff, thus forming a learning environment. For the description of this environment, but also for the ways

agents learn and adapt, we will focus on two theories, the Multi-armed Bandits and Evolutionary Game Theory.

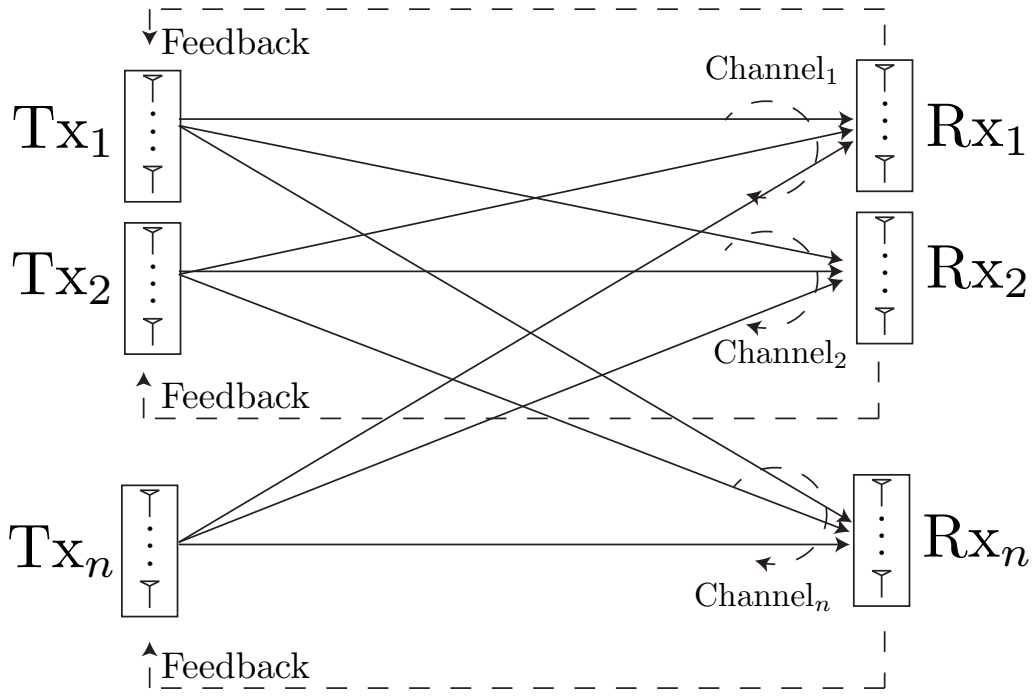
Figure 1: Merchants in the Desert, source: Red Sea Goverment

Merchants in Telecommunications We can treat User Equipment (UE) as merchants in an unknown telecom environment, where they strive for resources. More specifically, we are interested in the k-user Interference Channel (k-IC), figure 2, and how a UE can modify its actions in order to achieve better outcomes, while keeping the overall resources fairly distributed among the UEs.

The main interest in this network is the optimization of the interactions, regarding any of the following quantities:

- Connectivity

Figure 2: The n-User MIMO Interference Channel



- Power
- Capacity
- Bit Error Rate
- Spectrum Sharing

This environment is extremely adversarial due to the co-existence of other UEs and, also, noise. As a result, a user has to overcome interference from other users along with noise in order to deliver a message. In other words, if users want to communicate messages $\mathbf{x}$ the messages received at the receiver would be $\mathbf{y}$ according to the following equation

$$\mathbf{y} = H \cdot \mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y}$ is the resulting vector of messages of dimension $k \times 1$, H is the interference matrix $(k \times k)$, $\mathbf{x}$ is the vector of the initial messages $(k \times 1)$ and $\mathbf{n}$ is the noise at each receiver, which is assumed to be white gaussian $n_i \sim \mathcal{N}(0, 1)$.

One action against interference would be to treat it as noise. But, as the users grow, interference is growing along and, so, this solution is infeasible. Also, there are cases when, even 2 users cannot communicate due to strong interference among them, [1].

Usual approaches to the interference channel optimization problem are:

- Cooperation in order to achieve successful precoding
- or the introduction of a central authority that can manage the resources of the system, such as, power and frequency.

Precoding can be applied if users know the interference matrix H and, then, use this knowledge to transform the transmitted messages as follows

$$\mathbf{x}' = H^{-1}\mathbf{x} \tag{2}$$

Thus, the problem would degenerate to a noise removal problem. The difficulty arises when one comes to calculate these k^2 parameters, which is a task that is not easy to do, since it requires a lot of overhead, even for a small number of users, [2]. On top of that, if one requires the optimization to go further and split users into channels and after perform encoding, the problem would be even more difficult.

The second approach is based on the way cellular systems operate. Although in some cases one could appoint a moderator to the channel, there are multitude of scenarios

where no such moderator could be found. For example machine to machine communications form an IC which cannot be expected to have a moderator, due to the high mobility, e.g. in car to car communications, or the low computing resources of Internet of Things (IoT) devices.

Another example we can consider is in an urban environment with a dense configuration of wireless routers. Each router has to find an appropriate frequency and to adjust the power level so as to accommodate communications inside the coverage area, but, also, in a way that causes as low interference as possible. In this scenario, none of these devices has an overview of the entire network (as happens, for example, in a cell) so as to manage resources.

Thesis Approach

The reason we want to model interactions in telecommunications through game theory is because

- UEs can be designed with a *specific rationality*, which results in
- an *expected behaviour*, which leads to
- *favourable results*.
- Resource management takes the form of a *decentralised* optimization problem, which game theory has the ability to address.

Analyzing the above a bit further, the rationality is attributed to the programmable nature of the UEs.

Further, the expected behaviour stems from the utility function, which maps outcomes from actions to payoffs. This part will not concern us, since it is the interest of game theory.

Finally, a UE has to know how to learn so as to reach a favourable result, and has to do so in a decentralized manner. The way UEs should be designed to learn, is what this thesis is all about.

Specifically the motivation of this work, along with the key points that are addressed are

- How can a player adapt in an unknown environment?
- How to reach a favourable outcome?
- How learning algorithms perform compared to one another?

What we try to achieve is to present methods from two different areas that are designed to attack the same problem, although from alternative perspectives. These areas are Multi-armed Bandits and Evolutionary Game Theory.

The theory of Multi-armed Bandits (MAB) is presented in Chapter 1. This theory deals with regret instead of equilibrium, meaning that an agent adjusts his behaviour in order to regret as less as possible in the future.

The other is Evolutionary Game Theory (EGT) presented in Chapter 2. Unlike MAB, in EGT the equilibrium is the main concern of the dynamics, that is how an updating rule evolves and what are the guarantees provided regarding the equilibrium.

Applications of Game Theoretic Optimization in Wireless Telecommunications.

The use of game theory in telecommunications is extended due to the rationality inputed in the UEs. Some examples of these applications are the following

Cognitive Radio One of the scenarios that models agents in a telecommunication environment is that of Cognitive Radio (CR), where the problem of Spectrum Allocation comes into question. Examples can be found in [3–7].

Load Balancing In a cellular environment the use of Hetnets (Heterogeneous Networks) is gaining momentum. The idea is to introduce femtocells, e.g. Wifi routers, that could off-load the cellular network and, thus, increase capacity. The main issue, relevant to our approach, is the ability to group users either in the cellular, or the wifi network so as to increase a variable of interest.

Game-theoretic solutions have been proposed for different scenarios of optimization for these networks. For a survey on the various solutions proposed for HetNets see [8]. For proposed solutions involving game theory and HetNets see, for example, [9–11].

ARQ and HARQ The protocols of Automatic Repeat Request (ARQ) and its variant Hybrid ARQ (HARQ) are designed to provide error control through the transport layer by sending an appropriate feedback. In the original protocol (ARQ) the messages available were

- Stop-and-Wait
- Go back N
- Selective Repeat

On the other hand, hybrid ARQ is building on top of ARQ by adding error correction coding.

Examples of game theoretic ARQ mechanisms can be found in [12–15], while for HARQ in [16, 17].

The interested reader can consult the the following sources for game theoretic applications for wireless optization problems, [18–21].

Thoughts

As with many theories the concern is to start from a basic model and move to more complex ones. This basic model in our situation is a two person game. At first, this problem may seem easy to solve, but in reality there is not a solution that is optimal, i.e., reaching an equilibrium with probability close to one, for every type game[1]. That can be evidenced if one considers the difficulty of calculating all equilibria of a bi-matrix game. It is known that the computation of the equilibria is PPAD, even for a 2-person game, [22]. As a result, trying to compute equilibria in a decentralized manner should require at least the time of a direct approach.

This last comment motivates us to choose the number of actions to be small, i.e. $K \leq 10$, so as the algorithms presented would adapt faster.

[1]Game theorists have split games in so many categories (and many times in subcategories) that we will not be able to cover - and not interested to do so - here. Instead, what we want to discuss, is the various learning schemes and how they compare to each other

Contents

Chapter 1

Multi-Armed Bandits

Bandits is yet another area of probabilities that was inspired by gambling, where slot machines are referred to as "one-armed Bandits". The initial problem was that of distributing funds among different slots, arms from now on, where each one returns gains from a, possibly, different and unknown distribution to the player.

This player has two ways to determine the best strategies, the first is exploration and the other is exploitation. As the terms suggest, exploration is a form of sampling the available actions and their payoffs. Since payoffs are stochastic, the higher the exploration term the better. Furthermore, by engaging in exploitation one specifically selects the actions that are the most promising.

An algorithm that keeps the two above balanced is the most successful, since it is easily seen that neither exploration nor exploitation should act alone.

This chapter will focus on the opaque multi-arm bandit problem, where only one reward is received at each round. As we have seen in Forward, a single reward can provide valuable feedback to the transmitter. On top of that, we are interested in communications that need as little feedback as possible.

First and foremost, this chapter introduces the notation, part of which will be kept the same for the Evolutionary Game Theory chapter. Moreover, we will describe the different kinds of bandits. Finally, the last two sections are concerned with two things that are required in a bandit problem. The first is the regret, which is a measure of either success, or failure, of a strategy and the second is the proposed learning algorithms, which are called policies.

Notation

In this section we will present the needed notation. A bandit will be assumed to have K available arms, or actions, and the time horizon is T, which is assumed to be finite. Although, most policies presented here have an "any-time" counterpart we will not present them here, but the interested reader can consult the original publications.

Bandits may receive gains from an action, presented with $g_{i,t} \in [0, 1]$, or losses, $l_{i,t} \in [-1, 0]$. Subscript i, t denotes that the i-th arm has yielded a gain g or a loss l at time t. In this thesis we will concern ourselves with using only the gains. Also, the symbol $I_t \in \{1, ..., K\}$, denotes the choice made from the user at time t. More, a symbol frequently used is $\mathbb{1}_{I_t=i}$, where,

$$\mathbb{1}_{I_t=i} = \begin{cases} 1, & I_t = i \\ 0, & \text{else} \end{cases} \tag{1.1}$$

In addition, when computing the probability vector one makes use of estimators which are conditioned to the gains received. These estimators are denoted as $\tilde{g}_{i,t}$. A capital $G_{i,t}$ (or $\tilde{G}_{i,t}$) denotes a summation of all g's ($\tilde{g}_{i,t}$) up to time t. Also, a recurring variable is the learning rate which is denoted by η_t.

Finally, each arm has a probability to be drawn, which is time-variable and is denoted by $p_{i,t}$. The bold analog of this, $\mathbf{p}_t$, is the vector containing all probabilities at time t, i.e. $\mathbf{p}_t = \{p_{1,t}, p_{2,t}, ..., p_{K,t}\}$.

1.1 Bandits Preliminaries

1.1.1 Types of Bandits

Adversarial Bandits

The first appearance of this type of problems was in [26] as an extension to the original bandits problem. In this setting the gains are chosen from the player and its adversary simultaneously. A way to imagine this is through the rigged casino problem, where the player chooses which arm to play but an opponent keeps track of those choices and acts for his own self interest. It is important to note that zeroing all gains is against that interest since no one would be interested in playing in this casino. This setting is more suitable to describe individual agents under a game theoretic environment where each one plays, then receives a payoff and, accordingly, adjusts his strategy.

In the literature, one can find adversarial bandits to be called non-stochastic due to their inherent difference. This thesis will stick to the adversarial term, since, I believe, describes better the underlying problem.

Since we are in a game-theoretic environment the interest of this thesis is on the adversarial setting, i.e. how bandits interact using various Policies.

1.1.2 Other Bandits

Stochastic Bandits

The theory of bandits was initially developed for dealing with the uncertainty of payoffs as a result of drawing from a set of arms. The main premise was that each arm would output a payoff which would be i.i.d from an unknown distribution. This type of bandits would later be called stochastic.

Applications of Stochastic Bandits One of the interesting problems that bandit terminology was designed to address is that of clinical trials[1]. When presented with various patients the problem is to provide them with the most appropriate medication, which, of course is not known in advance.

An area where bandits have seen tremendous application is internet content placement. In short, the problem is to use the limited feedback received by users to display more relevant content, such as advertisements in a website or news in a newspaper's website. This variant of stochastic bandits is called Contextual and some approaches can be found in [27–29]

As with the adversarial bandits, the stochastic case has its own algorithmic family which is the Upper Confidence Bound (UCB). There are many variants in this family which provide bounds that are, in fact better than those in the adversarial setting, i.e., the exp3 algorithms. Another popular algorithm is the Thomson sampling.

[1] Although the literature is lacking extended exploration of this subject, as well as real world applications of it, as noted in [23]

Markovian Bandits

As the name implies the gains in this setting are a result of Markov processes. Each arm is associated with a Markov chain and at each round the bandit selects an arm which yields a gain drawn from its own Markov process.

By selecting an arm one gets a payoff and, only, that arm changes state according to a state transition probability.

Applications of Markovian Bandits The applicability of this bandit variant is found in problems that have many possible alternatives, and each alternative is driven by its own Markov chain. An example from telecommunications is the area of Cognitive radios where one wants to perform distributed time-division, see, e.g. [30–32].

As it is natural, the different nature of this variant from stochastic and adversarial bandits requires algorithms tailored to that specific nature. The most popular policy used in this setting is called index policy, or Gittin's Index, [33], which provide better results compared to UCB variants, see for example [34, 35].

A Note on the difference between Stochastic and Adversarial Bandits

At first sight, one could consider the two settings to be equal, or, at least to produce the same results. The main argument against is:

> Acting in the adversarial case changes the way the environment selects gains, while in the stochastic case, gains are always tied to a distribution.

As a result, we will see that algorithms designed for the adversarial setting have an exploration term, which increases

the regret, in order to make sure that any changes in the gains can be exploited.

For the stochastic case, the main algorithm that is used is the UCB (Upper Confidence Bound) policy[2], while for the adversarial case the exp3 (Exponential weights for Exploration and Exploitation) policy. The best known bounds for UCB applied to the stochastic problem is $\mathcal{O}(\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i})$, [36], with $\Delta_i = \max_{j=1,\dots,K} \mu_j - \mu_i$, while, by applying the exp3 at the adversarial setting, one cannot do better than $\mathcal{O}(\sqrt{KT \ln K})$, as proven in [37]. Taking all these in account we can see that algorithms designed for the stochastic case can only yield suboptimal results as opposed to the case of algorithms designed for these environments.

Finally, in section (1.2.3), we will present an algorithm that requires no assumption regarding the environment, but ensures that if the environment is stochastic then the regret will perform close to UCB and, if not, then the algorithm will "switch" to an exp3 algorithm.

1.1.3 Regrets

The quantity that is used in the bandits' literature to measure the effectiveness of a learning algorithm is the regret. The regret quantifies how worse a policy performs compared to the optimal gain either in each round or in expectation.

A quantity appearing in the following regrets is that of the total reward given by

$$G_A = \sum_{t=1}^{T} g_{I_t}(t), \tag{1.2}$$

or, else, the sum of the rewards received at each round.

[2] To be exact, from the initial UCB have stemmed many variants, so that UCB forms an algorithmic family

Worst-case Regret

For any set $J \subset \{j_1, j_2, ..., j_K\}^T$ the worst-case regret is noted as follows

$$R_w = G_J - G_A, \tag{1.3}$$

where

$$G_J = \sum_{t=1}^{T} g_{J(t)}, \tag{1.4}$$

or, else, the sum of the rewards if the chosen action set would be J.

Cumulative Regret

Using the quantity G_J from the previous regret, we could come up with Cumulative Regret, by setting

$$G_{J_{max}} \equiv \max_j \sum_{t=1}^{T} g_j(t), \tag{1.5}$$

The main objective of a bandit is to maximize the sum of gains received at each round, or, else to minimize

$$R_n = \max_{i=1,...,K} \sum_{t=1}^{n} g_{i,t} - \sum_{i,t}^{n} g_{I_t,t}. \tag{1.6}$$

Expected Regret By taking the expectation in 1.6 we end up with

$$\mathbf{E}R_n = \mathbf{E}\left\{ \max_{i=1,...,K} \sum_{t=1}^{n} g_{i,t} \right\} - \mathbf{E}\sum_{t=1}^{n} g_{I_t,t}. \tag{1.7}$$

The goal, when dealing with a cumulative reward is to minimize R_n and, as a result, try to achieve in each round the maximum available gain.

Pseudo-Regret

By taking the expectation before the maximum, we end up with pseudo-regret, which is

$$\overline{R}_n = \max_{i=1,\dots,K} \mathbf{E} \sum_{t=1}^{n} g_{i,t} - \mathbf{E} \sum_{t=1}^{n} g_{I_t,t}. \tag{1.8}$$

Pseudo-regret can be seen as a more relaxed term, compared to expected regret, since one does not compare against the best strategies, but against the best in expectation. Having said that, some of the results will be based on this regret since its easier to manipulate, because one observes only one strategy in each round.

1.2 How Adversarial Bandits Adapt - Policies

In this section we will study different ways that a player can increase the regret while providing guarantees over the expected loss. Most part of this section is devoted to exp3 and its variants. That is due to the fact that the results of other algorithmic families, e.g. UCB (Upper Confidence Bound), are tailored to deal with stochastic settings.

A bandits' formula

Before delving further into the adversarial algorithms, I feel the need to break down the way a policy works. The sequence of actions a player performs are:

1. starts from a uniform distribution for the arms,
2. draws an arm randomly according to the current distribution $\boldsymbol{p}_t$,
3. receives a gain $g_{I_t,t}$ and updates $\tilde{g}_{i,t}$ $\forall i \in \{1, 2, ..., K\}$,
4. calculates $\boldsymbol{p}_{t+1}$ according to $\tilde{g}_{i,t}$,
5. reiterates from step 2.

Examining the above, we can see that a policy is comprised of two things, an update rule and an estimator. By changing those two one can have a unique algorithm. As we will see later on, these two are the key points in the design of algorithms for bandits. Moreover, in this thesis we are concerned with a gain received from just one arm, and specifically that which was played the previous round. Although there are scenarios where one could receive different kinds of feedback, e.g., the highest payoff strategy, of the payoff of each action, e.t.c., this thesis will not be concerned with those.

.

1.2.1 Exp3 and Variants

The exp3 algorithm (EXPloration and EXPloitation with EXPonential weights) was first introduced in [26]. The motivation of the authors was to solve a new bandit problem[3], which they named non-stochastic, and later became known as adversarial (see section 1.1.1). In this paper, the authors, along with the original exp3 algorithm, propose different variants in order to bound the regret and/or take advantage

[3]Compared to the already known and studied stochastic MAB problem

of more complex environments, e.g. updating rules that use knowledge from experts.

As we will see in the Evolutionary Dynamics (Section 2.5.4), the exp3 algorithm degenerates to logit and population dynamics in a full information environment.

Vanilla Exp3

The original exp3 algorithm was introduced in [26] and it was the first attempt to tackle the adversarial problem. It's inspiration come by the Hedge algorithm introduced in [38].

Choose: a non-increasing sequence of η_t, $t \in \mathbb{N}$.
Set: $p_{i,1} = \frac{1}{K}$ and $\tilde{G}_{i,1} = 0$, $\forall i \in \{1, ..., K\}$
for $t = 2$ **to** n **do**
  Draw arm $I_t \sim \boldsymbol{p}_t$;
  for $i = 1$ **to** K **do**

$$\tilde{g}_{i,t} = \frac{g_{i,t}}{p_{i,t}} \mathbb{1}_{I_t = i}, \text{ Compute estimated gain} \tag{1.9}$$

$$\tilde{G}_{i,t} = \tilde{G}_{i,t-1} + \tilde{g}_{i,t}, \text{ update estimated cumulative gain.} \tag{1.10}$$

  end
  Compute $\boldsymbol{p}_t = \{p_{1,t}, ..., p_{K,t}\}$

$$p_{i,t} = \frac{\exp\left(\eta_t \tilde{G}_{i,t}\right)}{\sum_{j=1}^{K} \exp\left(\eta_t \tilde{G}_{j,t}\right)} \tag{1.11}$$

end

Algorithm 1: Original Exp3

The estimator that is used in this algorithm is an unbiased estimator, meaning that the expectation on the gains on time t is equal to the actual gain received in step t,

$$\mathbf{E}_{I_t \sim \mathbf{p}(t)}\{\tilde{g}(t)\} = g_{I_t}(t). \tag{1.12}$$

Although this property is extremely helpful so as to bound the sum of regrets, it allows the variance of cumulative regret to be arbitrarily large, i.e.,

$$\mathbf{E}_{I_t \sim \mathbf{p}(t)} \left\{ \tilde{g}^2(t) \right\} = \frac{g_{I_t}^2(t)}{p_{I_t}} \tag{1.13}$$

and, as a result, this algorithm may yield unpredictable regrets.

Exp3.P

The exp3.P variant is an alternative to the vanilla exp3 which was proposed in [26], in order to bound the regret with high probability and solve the problem of variance.

Choose: $\gamma, \beta \in (0, 1]$ and $\eta_t : \eta_n \leq \eta_{n-1}, \forall n \in \{0, ..., T\}$
Set: $w_i(1) = \exp\left(\frac{\alpha\gamma}{3}\sqrt{\frac{T}{K}}\right)$, $\forall i \in \{1, ..., K\}$
Set: $p_{i,t} = \frac{1}{K}$, $\forall i \in \{1, 2, ..., K\}$
for $t = 2$ **to** n **do**
  Draw arm $I_t \sim \boldsymbol{p}_t$;
  for $i = 1$ **to** K **do**

$$\tilde{g}_{i,t} = \frac{g_{I_t,t}\mathbb{1}_{I_t=i} + \beta}{p_{i,t}} \tag{1.14}$$

$$w_i(t) = w_i(t-1)\exp\left(\eta_t \tilde{g}_{i,t}\right) \tag{1.15}$$

  end
  Compute $\boldsymbol{p}_t = \{p_{1,t}, ..., p_{K,t}\}$

$$p_{i,t} = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K} \tag{1.16}$$

end

Algorithm 2: Exp3.P

The differences between the original algorithm and exp3.P is the introduction of parameters γ and β. Parameter γ helps

the exploration, while β creates a skewed estimation of the gains, i.e.,

$$\tilde{g}_i(t) = \frac{g_{I_t}(t)\mathbb{1}_{i=I_t} + \beta}{p_i(t)} \tag{1.17}$$

which can be shown (see, [39]), that with probability at least $1-\delta$ leads to an upper bound to the real gains

$$\sum_{t=1}^{T} g_i(t) \leq \sum_{t=1}^{T} \tilde{g}_i(t) + \frac{\ln\left(\delta^{-1}\right)}{\beta} \tag{1.18}$$

On the other hand, parameter γ is used to create a mixture of the exponential weight distribution with that of a uniform distribution, which further promotes exploration.

Using the following values for the exp3.P algorithm

$$\beta = \sqrt{\frac{\ln\left(K\delta^{-1}\right)}{TK}}, \quad \eta = 0.95\sqrt{\frac{\ln K}{TK}}, \quad \gamma = 1.05\sqrt{\frac{K\ln K}{T}} \tag{1.19}$$

leads to, with probability at least $1-\delta$, the bound

$$R_T \leq 5.15\sqrt{TK\ln\left(K\delta^{-1}\right)} + \sqrt{\frac{TK}{\ln K}}. \tag{1.20}$$

Lower Bound on Regrets A lower bound helps to ensure that the upper bounds that have been calculated above cannot be improved further. A proof for this bound can be found in [39].

$$\inf\sup\left(\max_{1,2,\ldots,K} \mathbf{E}\sum_{t=1}^{T} y_{i,t} - \mathbf{E}\sum_{t=1}^{T} y_{I_t,t}\right) \geq \frac{1}{20}\sqrt{TK} \tag{1.21}$$

Variables $y_{i,1}, y_{i,2}, ...$ are assumed to be i.i.d. gains and the supremum is taken over all possible distributions of rewards. Also, the infimum is taken over all forecasters, which means that the upper bounds are at least of order $\mathcal{O}\left(\sqrt{TK}\right)$.

The role of parameter γ In the adversarial setting, where exp3.P was introduced, the role of γ was to retain exploration as an option at all times. That was understandable, since, when one deals with an adversary whose main goal is to return the least possible gain, the player has to reevaluate all arms in case the adversary's strategy has changed.

On the other hand, in a game theoretic environment, apart from the zero-sum game, players don't have an incentive to "trick" each other, but to maximize individual payoffs. That means that an exploration parameter could be harmful since it can potentially force a game away from an equilibrium.

Exp3.IX

In [40], the author proposes a new variant to the exp3 algorithm that achieves to bound the real regret, instead of the pseudo-regret as was the case in previous policies. The IX stands for Implicit eXploration, due to the removal of the exploration term which results in an exploration of $\Omega(\sqrt{KT})$.

Following the logic of previous works, the exp3.IX algorithm is created in order to introduce a variance-reducing effect, by calculating the estimator with

$$\tilde{g}_{t,i} = \frac{g_{t,i}}{p_{t,i} + \gamma} \mathbb{1}_{I_t=i}, \tag{1.22}$$

where $\gamma > 0$ is an appropriately chosen variable.

Choose: $\gamma \in (0,1),\ \eta > 0$
Initialize: $\tilde{G}_{i,0} = 0, \quad w_i(1) = 1 \quad \forall i \in \{1,...,K\}$
Set: $p_{i,1} = \frac{1}{K}$, $\forall i \in \{1,...,K\}$
for $t = 1$ **to** n **do**
  Draw arm $I_t \sim \boldsymbol{p}_t$;
  for $i = 1$ **to** K **do**

$$\tilde{g}_{i,t} = \frac{g_{I_t,t}\mathbb{1}_{I_t=i}}{p_{i,t} + \gamma} \tag{1.23}$$

$$\tilde{G}_{i,t} = \tilde{G}_{i,t-1} + \tilde{g}_{i,t} \tag{1.24}$$

  end
  Compute $\boldsymbol{p}_{t+1} = \{p_{1,t+1}, ..., p_{K,t+1}\}$

$$p_{i,t+1} = \frac{\exp\left(\eta_t \tilde{G}_{i,t}\right)}{\sum_{j=1}^{K} \exp\left(\eta_t \tilde{G}_{j,t}\right)} \tag{1.25}$$

end

Algorithm 3: Exp3.IX

The proposed values for the parameters $\eta,\ \gamma$ are

$$\eta = 2\gamma = \sqrt{\frac{2\log K}{KT}} \tag{1.26}$$

which result in a bound of

$$R_T = 2\sqrt{2KT\log K} + \sqrt{\frac{2KT}{\log K}}\log\left(\frac{2}{\delta}\right) + \log\left(\frac{2}{\delta}\right) \tag{1.27}$$

A remark in the exponential weighted algorithmic family. As we have already seen, when updating probabilities, there appears the auxiliary variable w called weight. In the literature the proposed ways to weight updating, are, either

sequentially or by recalculating them in each round, i.e.,

$$w_i(t+1) = w_i(t)\exp\left(\eta\tilde{g}_i\right) \quad \text{or} \quad w_i(t+1) = \exp\left(\eta\tilde{G}_i(t)\right). \tag{1.28}$$

These two are equal if we assume that η is constant, but in the case where η is time-varying, then the two update schemes are not the same, as it can easily be seen

$$w_i(t+1) \propto \exp\left(\sum_{j=1}^{t}\eta_j\tilde{g}_i(j)\right) \tag{1.29}$$

$$w_i(t+1) \propto \exp\left(\eta_t\sum_{j=1}^{t}\tilde{g}_j\right) \tag{1.30}$$

Although this difference is not explicitly stated in the literature, both these update schemes are used interchangeably, even for the same algorithms.

1.2.2 Implicitly Normalized Forecaster

A new family of policies for the Adversarial Bandit problem is the Implicitly Normalized Forecaster (INF), presented in [41]. INF can be seen as a generalization of the exp3 forecaster by using a generic update rule.

The main idea is to exchange the exponential function, $\exp(\eta g)+\frac{\gamma}{K}$, with a continuously differentiable function $\psi(g)$, having the following properties

$$\begin{gathered}\psi : \mathbb{R}_-^\star \to \mathbb{R}_+^\star \\ \psi' > 0, \quad \lim_{x\to\infty}\psi(x) < \frac{1}{K}, \quad \lim_{x\to 0}\psi(x) \geq 1\end{gathered} \tag{1.31}$$

Another change that the INF approach brings is to the form of the estimator, which is

$$\tilde{g}_i(t) = -\frac{\mathbb{1}}{\beta}\log\left(1-\frac{\beta g_i(t)}{p_i(t)}\right). \tag{1.32}$$

The incentive behind the estimator is that in a first order approximation is equal to the unbiased estimator, i.e.,

$$\sum_{i=1}^{K} \tilde{g}_{i,t} = \frac{p_{I_t,t}}{\beta} \log\left(1 - \frac{\beta \cdot g_{I_t,t}}{p_{I_t,t}}\right) \tag{1.33}$$

$$= g_{I_t,t} + \mathcal{O}\left(\frac{\beta \cdot g_{I_t,t}}{p_{I_t,t}}\right) \tag{1.34}$$

In this paper, the authors used in the place of ψ function

$$\psi(x) = \left(\frac{\eta}{-x}\right)^q + \frac{\gamma}{K} \tag{1.35}$$

and the resulting policy is the poly INF algorithm. Finally, a normalization function, C, needs to be computed in each step, which cannot be derived analytically, thus the name of the policy, such as $C : \mathbb{R}^K \to \mathbb{R}$

$$\sum_{i=1}^{K} \psi\left(x_i - C(\boldsymbol{x})\right) = 1, \tag{1.36}$$

where $\boldsymbol{x} = \{x_1, x_2, ..., x_K\}$.

By using the poly INF algorithm, the two regret bounds that can be achieved are

$$\overline{R}_T \leq 8\sqrt{KT} \tag{1.40}$$

$$\mathbf{E}R_T \leq 9\sqrt{KT \ln(3K)}. \tag{1.41}$$

A very important remark that this paper makes is that the bound of the expected regret cannot be further improved in the case of a non-oblivious adversary.

A remark on INF's estimator

As we have seen in eq. 1.34, the expected value of the estimator is the received gain plus smaller terms. But a

Choose: Function $\psi(x)$, with the properties of equations 1.31
Initialize: $\tilde{G}_{i,0} = 0, \quad \forall i \in \{1, ..., K\}$
for $t = 1$ **to** n **do**
  Draw arm $I_t \sim \boldsymbol{p}_t$;
  for $i = 1$ **to** K **do**

$$\tilde{g}_{i,t} = -\frac{\mathbb{1}}{\beta} \log\left(1 - \frac{\beta \cdot g_{I_t,t}}{p_{I_t,t}}\right) \tag{1.37}$$

$$\tilde{G}_{i,t} = \tilde{G}_{i,t-1} + \tilde{g}_{i,t} \tag{1.38}$$

  end
  Compute: $C_t = C(\tilde{\boldsymbol{G}}_t)$
  Compute: $\boldsymbol{p}_t = \{p_{1,t}, ..., p_{K,t}\}$

$$p_{i,t} = \psi\left(\tilde{G}_{i,t} - C_t\right) \tag{1.39}$$

end

Algorithm 4: Poly Implicit Normalized Forecaster

calculation of its variance would reveal the same problem as the unbiased estimator of vanilla exp3 algorithm.

Another problem that arises from the use of this estimator is that the term inside the logarithm should always be positive. That can be achieved by selecting $\beta < \gamma/K$, which is the normalizing term of the update rule and, subsequently, the lowest value of any probability. As a result, this estimator's use is limited specifically to update rules which use the explicit exploration term, i.e., using the uniform distribution term.

1.2.3 Other Policies for Adversarial Bandits

Stochastic and Adversarial Optimal (SAO)

An algorithm that is designed to deal with either one of the Stochastic or the Adversarial setting is "Stochastic and Adversarial Optimal", presented in [42]. The goal of this algorithm is to match a regret of $\mathcal{O}(\sqrt{n})$ in the adversarial case, while, if the gains appear to be stochastic the regret will grow as polylog(n).

The algorithm uses the unbiased estimator, $\tilde{g}_{i,t} = \frac{g_{I_t,t}\mathbb{1}}{p_{i,t}}$, and is divided in three steps, which will, for sake of simplicity, be analyzed for the case of $K = 2$ arms. Variables that will be used are $C_{\text{crn}} = 12 \ln T$ and H, which is the time average of G, e.g. $\hat{H} = \frac{1}{t}\hat{G}$, e.t.c.

Parameter C_{crn} comes from bounding a probability using a Chernoff Bound. The interested reader is referred to [42] for a detailed explanation.

In order for the algorithm to decide under which setting to optimize for, the following steps are being used:

- **Exploration** In each time instance, the two arms are sampled with equal probability, i.e. $(\frac{1}{2}, \frac{1}{2})$ and then, the two ending conditions are checked:

$$t > \Omega(12 \ln T) \tag{1.42}$$

$$|\tilde{H}_{1,t} - \tilde{H}_{2,t}| < 24\frac{C_{\text{crn}}}{\sqrt{t}}. \tag{1.43}$$

  If any of the two is met, then the algorithm moves to the next phase. Else, a new sampling followed by conditions checking round is performed.

- **Exploitation** The algorithm enters in this phase at time $t^\star$ and let us assume that, without loss of generality, arm 1 has greater, average payoff, meaning $\tilde{H}_{1,t^\star} > \tilde{H}_{2,t^\star}$.

Then, samples arms with $P = (1 - t^\star/2t,\ t^\star/2t)$. After each step the following conditions are checked:

$$8\frac{C_{\text{crn}}}{\sqrt{t^\star}} \leq \tilde{H}_{1,t} - \tilde{H}_{2,t} \leq 40\frac{C_{\text{crn}}}{\sqrt{t^\star}} \tag{1.44}$$

$$\begin{cases} |\tilde{H}_{1,t} - \hat{H}_{1,t}| \leq 6C_{\text{crn}}/\sqrt{t} \\ |\tilde{H}_{2,t} - \hat{H}_{2,t}| \leq 6C_{\text{crn}}/\sqrt{t^\star}, \end{cases} \tag{1.45}$$

and the exploitation step is iterated as long as the above hold.

- **Adversarial** If the algorithm reaches this point, then the algorithm has decided that it is the adversarial case, so the computing of arms' probabilities is continued through the exp3.P algorithm (see Algorithm 2).

Chapter 2

Evolutionary Dynamics

Introduction

This Chapter is dedicated to a branch of Game Theory called Evolutionary Game Theory (EGT) which studies the ways players can change their strategies through receiving payoffs for their chosen actions and by observation of other players' moves.

The development of EGT was primarily to describe the evolution of populations through interactions with other species, [43]. A famous example was the game of hawk and dove. In this game, there are two infinite populations, one consisting of doves and one of hawks and each member is assigned one of the actions available. Then, each member of the population would interact with a member of the other population, picked at random. After that interaction, all members receive a payoff and according to that they, either change their belief (action) or not. By constantly interacting, species change the distribution of their population for all actions, in such a way that would result in higher returns in the future.

EGT and Telecommunications As we have seen, the contribution of game theory in telecommunication problems is in providing the appropriate tools to model users as rational agents constantly maximizing a specific utility function. On the other hand, EGT is the appropriate tool to analyze the interactions of agents, or, to put it in another way, how users can efficiently update their beliefs so as to receive higher payoffs as time grows.

So, according to our previous analysis, each user is one of the different and interacting species. Each user assigns a probability distribution to the set of available actions, which looks as if an infinite population is split in the available actions. After a round of receiving payoffs the population is redistributed according to the input (payoff or observation of other players' moves).

A requirement of our problem is the need for decentralized and non-cooperative algorithms. These two requirements are 1) the result of an absence of a central authority that has knowledge of what users want and can regulate the interactions appropriately, and 2) due to the assumption that there is no way for users to exchange information with one another, i.e., they cannot "agree" on a solution. The only tool available to our users is the limited feedback received so as to update their beliefs.

Finally, we will focus on evolutionary algorithms that are discrete in time, so as to model the interactions in a realistic way. Although, most dynamics have been investigated, mostly, under a continuous framework, there is significant research on their discrete time analogs.

Structure of this Chapter The first part of this chapter is dedicated to the necessary notation. After that, we will explore some simple update rules and then move to the best

response dynamics.

Finally, the last two sections contain analysis of Smoothed best response and the replicator dynamics.

2.1 Notation

This section will provide the necessary notation for the remaining of this chapter. Although there are differences in notation between EGT and MAB in the literature, we will try to keep things fairly relevant between these two chapters. As a result we will use most of the notation from the previous chapter and hope it won't be confusing or annoying to readers who, already, have a game theoretic background.

We assume that there are two players who interact. Each one has a set of actions of cardinality K, which is kept the same for both, for simplicity reasons. At each time instance a player selects an action, I_t, randomly from his current distribution, $\mathbf{p}_t$, over the set of actions, i.e. $I_t \sim \mathbf{p}_t$. The two symbols p, q denote, respectively, the probabilities of the player we are talking about and his adversary.

A player has a matrix U_i of dimensions $K \times K$, which is called utility matrix and maps all possible pairs of actions to a payoff. Specifically, if I_t and J_t are the two actions played at time t, then the players would receive $U_i(I_t, J_t)$ and $U_j(J_t, I_t)$. A single payoff received will be denoted by $g_{I_t,t}$. It is important to note that in each utility matrix the column player is the one that the matrix refers to.

Finally, due to the matrix formulation of the problem, we could write the expected payoff at time t as follows

$$\mathbf{E}\{U_i(t)\} = \mathbf{p}_t^T U_i \mathbf{q}_t \tag{2.1}$$

In what follows, we will be using two terms, which refer to the information provided to the users. The full informa-

tion environment and the imperfect information. The first one means that a player is up to date with all other players' strategies, while under imperfect information users either keep an empirical distribution or adapt depending on the feedback received.

Infinite Games v. Bandits with finite horizon

As we have seen, the algorithms that were proposed when we considered the bandits' setting include a finite time horizon. That choice was made in order to simplify the presentation of the algorithms, since there are modifications for the case of an infinite time environment.

On the other hand, when dealing with the EGT formulation, we will stick to the model of infinite time. In an finite time setting players can do a backwards induction and come up with the perfect strategy to follow, which not only defeats the purpose of this chapter, but, also, requires complete knowledge of the payoff matrix. Instead, our attention will be focused on agents who are time-unaware and adapt their strategies towards a maximization of their expected payoff.

2.2 Simple Update Rules

In this section we will present some algorithms with simple structure and study their efficiency through simulations.

2.2.1 Cross' Learning Process

A simple update rule is Cross' Learning Process, presented in [44]. It is shown to converge to the continuous time replicator equation in [45], under the assumption of a finite

time horizon. An important requirement of the algorithm is for all payoffs to be in the interval $[0, 1]$.

In the discrete case of interest, the update rule takes the following form

$$p_{k,t+1} = g_{I_t,t}\mathbb{1}_{I_t=k} + (1 - g_{I_t,t}) \cdot p_{k,t}. \tag{2.2}$$

The main premise in order to achieve the continuous limit of replicator dynamics, is that a player receives payoffs during a time interval dt, which are stochastic and, due to the law of large numbers, the result is close to the expected value.

The power of this result lies in the easiness of the updating rule and that the underlying continuous equation is the replicator equation, see Section 2.5.

2.2.2 Linear Reward - Inaction

Another algorithm which resembles in simplicity the one presented in section 2.2.1, and its appearance was in [46]. It is called Linear Reward -Inaction ($L_{R\text{-}I}$) and it draws its inspiration from automata. The updating rule is

$$p_{i,t+1} = p_{i,t} + b \cdot g_{I_t,t} \left(\mathbb{1}_{I_t=t} - p_{i,t}\right), \tag{2.3}$$

where b is an appropriately chosen parameter, $0 < b < 1$, and the gains are bounded, i.e., $g \in [0, 1]$.

For the case of a sufficiently small b, the ODE of the system can be calculated, which is

$$\frac{\mathrm{d}p_{i,t}}{\mathrm{d}t} = f(p_{i,t}), \tag{2.4}$$

where $f(p_{i,t}) = \mathbf{E}\{g_{I_t,t}(\mathbb{1} - p_{i,t})\}$. By analyzing the above ODE the following can be proven

- All pure strategies are stationary points,

- All Nash equilibria are stationary points,
- All stationary points that are not NE are unstable,
- All pure NE are asymptotically stable,
- if the algorithm converges then this point is a NE

Finally, the authors of [46] show that a sufficient condition for convergence to an NE under the $L_{R\text{-}I}$ is

$$\frac{\partial F}{\partial p_{iq}}(P) = c \cdot h_{iq}(P), \tag{2.5}$$

where $c > 0$ is a constant, F is a bounded differential function and h_{iq} is the expected payoff of player i under the strategy profile q.

2.2.3 Simple Models Simulations

In order to see the efficiency of the update rules presented in this section we will simulate interactions of two players for the two players' problem given by the following utility matrices

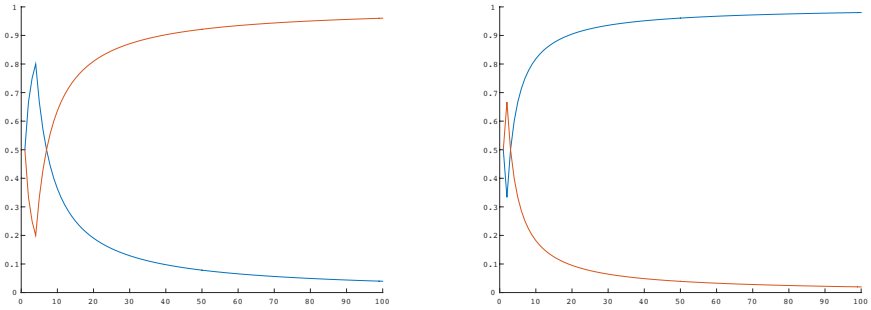
$$U_1 = \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \qquad U_2^T = \begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}. \tag{2.6}$$

which has mixed NE

$$\boldsymbol{p} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad \boldsymbol{q} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} \tag{2.7}$$

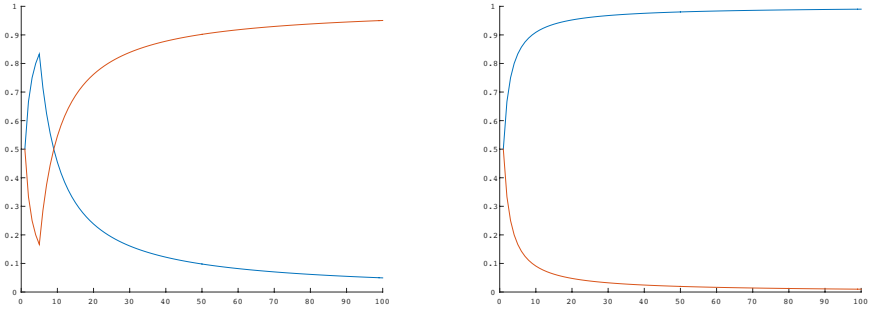
The reason this game is chosen is due to the easiness of analyzing the equilibrium, while the small number of actions should make it simpler to reach an equilibrium. By simulating this game we have

Figure 2.1: Cross' Learning



while for the case of LRI we get

Figure 2.2: Cross' Learning



By taking a look at the results, we can deduce that users under those update rules, at first do respond well to the other player's moves, but they move too quickly to a "safe choice", or, else they make the transition from exploring to exploiting fast. This results in convergence to pure action profiles and consequently "missing the mark".

Another remark about the above figures is that, since players don't have a complete picture of the game, but only act under the feedback they receive, learning fast makes them susceptible to exploiting the wrong action, even in the case of existing pure NEs.

2.3 Best Response Dynamics

Best Response (BR) dynamics is an update rule that is based in the premise of maximizing every strategy given the past feedback, i.e. other player's strategy or empirical distribution. As such, BR dynamics can be considered as a system of "local rationality" and the main interest is whether it can reach an equilibrium, or, even, whether it can maximize expected payoffs.

Since there could be many strategies resulting in the same received payoff (in expectation), conditioned on the other's strategy, usually, BR is a set of maximizing strategies, instead of a specific strategy. That means that one has to carefully select the next strategy profile. The BR set is formed as follows

$$\mathrm{BR}(\boldsymbol{p}_t) \in \arg\max_{\tilde{\boldsymbol{p}}} \tilde{\boldsymbol{p}} U \boldsymbol{q}_t, \tag{2.8}$$

In a full information environment, there are two issues with BR dynamics. First of all, when considering the case of having a few historical data, the NE is unstable, since a new action results in a the system to leave the equilibrium.

Secondly, if the game has evolved to an equilibrium, after many plays, and we consider a change in payoffs (since we are interested in a dynamic environment), that change will not be easily detected, since all players best respond to the other players' strategies. So, this case has two problems, first there is the matter that users are agnostic to the payoffs, and only care how other players strategize. The other problem is that since all players keep historical data, then best responding is in reality best responding mostly to old actions, rather than new.

Concerning the case of pure NE, the BR dynamic is resilient to small perturbations and, thus, a pure NE is stable

under BR dynamics.

On the other hand, when considering an incomplete information setup, the update rule degenerates in

$$p_{i,t+1} = \begin{cases} 1, & \text{if } \arg\max_{i'} U(i', J_t) q_{J_t,t} \\ 0 & \text{else} \end{cases} \tag{2.9}$$

which results in missing mixed NE, but sometimes to miss even pure NE as we will see in the following examples.

Assume the following game,

$$U_1 = \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}. \tag{2.10}$$

which has mixed NE

$$\boldsymbol{p} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad \boldsymbol{q} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} \tag{2.11}$$

It is easy to see that if players best respond to each other then, on average will spend a quarter of their time receiving each of these payoffs, which is not a NE. Even in games where a pure NE exists, BR dynamics may not converge, if there isn't a special underlying structure to the game, as is analyzed in [47].

In the following example, there is a pure NE which BR dynamics cannot reach if the initial conditions are any combination of the second and third actions of both players.

$$U_1 = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 0 & 1 & 3 & 0 \\ 1 & 4 & 2 & 1 \\ 1 & 0 & 1 & 10 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 0 & 4 & 1 & 0 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 0 & 10 \end{pmatrix}. \tag{2.12}$$

Remarks on Best Response

The problems that arise from the use of BR dynamics can be summarized in the following

- Considering a full information environment and both players being in an equilibrium. Then, the BR set could contain more than one strategy and as such there is no guarantee that a player would choose the NE strategy indefinitely.
- If only one of the players uses a BR dynamic, his adversary could easily pick his actions to trick and outperform the first. That is the result of the deterministic nature of the update rule.
- Limited information settings, which have more practical applications, are prone to be suboptimal contrary to the expectation of the NE.
- In a limited information environment, one cannot hope to make an accurate prediction of the actual distribution of their opponent. In fact, in [48], the authors prove that there is no predictor, which uses past observations and is asymptotically consistent. Thus, BR cannot reach an equilibrium in all settings.

2.4 Smoothed Best Response

Although BR is proven to be a suboptimal way to interact with other agents, many dynamics have used it in a different way thus giving birth to Smoothed Best Response dynamics. The main premise is to calculate the best response but act only partially according to it.

We can view that as using the best response as a target and not the actual next move. As such, smoothed BR move towards that target while a BR strategy would be to adopt the target at all times.

2.4.1 Logit Dynamics

Logit dynamics was introduced in [50] and belongs to the category of smoothed best reply update rules. It is extending BR dynamics with the introduction of the parameter β, which is called the rationality level of the player. The logit rule is

$$p_{i,t} = \frac{1}{\mathcal{Z}(t)} e^{\beta g(i,t|\mathbf{s}^{-j})}, \tag{2.13}$$

where $\mathcal{Z}(t)$ is a normalizing factor and $g(i,t|s^{-j})$ is the payoff for each action i given the strategy profiles of all other players to be $\mathbf{s}^{-j}$. That means that, instead of applying the BR rule, a player assigns probabilities in each action given the best response but, also, how each action is performing contrary to the whole action set.

Regarding the parameter β the two extremes are $\beta = 0$, where a player would select uniformly all options regardless of the feedback and $\beta \to \infty$, when the updating rule degenerates to the BR rule.

Properties of Logit Dynamics

Some useful properties of Logit Dynamics, which are proven in [47], are

- They form an ergodic Markov Chain,
- Invariance under utility translation, i.e., adding or subtracting does not affect the result, but

- rescaling of the utilities changes the dynamics, since a rescaling by α changes β to $\beta \cdot \alpha$

That last property of Logit dynamics along with the need to select the proper parameter β is what makes its behaviour different in games that are seemingly the same, or, else, we could say that the result of the learning rule is sensitive to the parameter β.

Association of FP and exp3

If we consider a full information setting and a population dynamic environment, then each player updates each action proportionally to the received gain and the up to date strategy profiles of other users. Then the update rule takes the following form

$$\begin{aligned} \mathbf{p}_{t+1} &\propto \mathbf{p}_t \cdot \exp\left\{\beta \cdot \text{diag}[U\mathbf{q}_t]\right\} \\ &\propto \mathbf{p}_{t-1} \cdot \exp\left\{\beta \cdot \text{diag}[U(\mathbf{q}_t + \mathbf{q}_{t-1})]\right\} \\ &\vdots \\ &\propto \mathbf{p}_0 \cdot \exp\left\{\beta \cdot \text{diag}[U \cdot \sum_{k=0}^{t} \mathbf{q}_k]\right\}, \end{aligned} \tag{2.14}$$

which, if compared with the the discrete-time replicator dynamics are the same, see 2.5.

2.4.2 Fictitious Play

Fictitious play (FP) first appeared on [52] as a way to compute the NE of a game. This learning rule is among the first proposed in tackling the learning problem. The update rule

for the discrete case is

$$\boldsymbol{p}_{t+1} = \frac{t\boldsymbol{p}_t + BR_t}{t+1} \tag{2.15}$$

As we have seen in the case of BR one makes jumps from one BR point to the next. Contrary to that, fictitious play can be seen as a rule that updates the strategy profile by selecting a point that lies between the current strategy and the best response. In the first steps, the rule almost best responses and, as time goes by, the strategy barely changes.

The problem of learning NE through Fictitious Play

In [53], the author analyzes the reasons why learning through FP is not a sufficient condition to reach an equilibrium. The author characterizes the result of FP, as weak convergence, meaning that each player reaches the empirical distribution of an equilibrium, but ends up not playing according to that distribution. Contrary to that, a learning rule converges strongly if, not only the player learns the empirical of an equilibrium but, also, implements it.

As an alternative, the author of [53] proposes the following learning rule which is similar to FP but converges strongly to a NE

$$\mathbf{p}_{t+1} \in BR_{t+1}(\hat{\mathbf{q}}_\mathbf{t})\rho_t + \hat{\mathbf{p}}_t(1-\rho_t) \tag{2.16}$$

where ρ_t is a variable which satisfies $\lim_{t\to\infty} \rho_{i,t} = 0$, and $\hat{\mathbf{s}}$ is the empirical strategy profile of $s \in \{p, q\}$.

The incentive of this algorithm is for players to start playing best responses to explore the state space and, as time goes by, to converge to their empirical distribution.

Missed Nash Equilibria with FP

The first evidence of a missed equilibrium came by Shapley in [54]. Shapley introduced a 3×3 game which did not converge to the NE, but, instead, to a limit cycle, for some initial conditions.

As a continuation of that, one may assume that being close enough to a mixed equilibrium should be a sufficient condition of convergence. This is shown to not be true in some cases, see [53], even if the mixed equilibrium is unique.

2.5 Replicator Dynamics

Replicator dynamics is among the most popular learning algorithms in Evolutionary Game Theory and was first presented in [60]. The attention received is probably due to links between human & animal learning and the replicator equation, [61, 62].

This algorithm is associated with the exp3 algorithms of Chapter 1 as we will see in Section 2.5.2 and is designed as a population dynamics equation, while we have already proven the connection of RD and FP in the full information setting.

The continuous-time version of RD is

$$\frac{dp_{i,t}}{dt} = p_{i,t}\left[(U\mathbf{q}_t)_i - \mathbf{p}_t^T U \mathbf{q}_t\right]. \tag{2.17}$$

In section 2.5.2 we will prove that one can derive a discrete time dynamic[1], which is

$$p_{i,t+1} = p_{i,t}\frac{\exp\left[(U\mathbf{q}_t)_i\right]}{\sum_k \exp\left[(U\mathbf{q}_t)_k\right]} \tag{2.18}$$

[1]More precisely we will transition from the discrete time RD to continuous-time RD

A first examination of the replicator equation shows how probabilities of strategies change over time. If a strategy yields a greater payoff than the average payoff, then this strategy grows according to how much better it fared against the other strategies. Also, for smaller than average payoffs those strategies get shrunk. It is easy to see that strategies that are strictly dominated will inevitably vanish.

2.5.1 Equilibria

Taking a closer look at eq. 2.17 and 2.18 it is easy to see that a Nash Equilibrium is a rest point for this process. In [56], the following regarding RD and NE, are proven

- All NE are rest points of the RD.
- Pure NE are asymptotically stable.
- Every stable rest point is a NE.

Having proven the above points and with the help of RD equation, one can come to interesting results regarding the equilibria of games. Specifically, the authors of [56] prove that there is at least one NE in any game and that the number of NE appearing in a game are odd.

Breaking down a Nash Equilibrium we have strategies (S_i) that are strictly dominated and, as such, yield lower payoff, so we assume $p_{i,t} \to 0$, as $t \to \infty$. The rest should have the same payoff, which is equal to the average payoff. Thus, $\frac{dp_i}{dt} = 0$.

Moreover, rest points for this equation, are all pure strategies and if there exists a pure Nash equilibrium, then, that is asymptotically stable.

So, if agents in a game start from a Nash Equilibrium then, under the replicator equation they will indefinitely stay at that point.

2.5.2 Discrete and Continuous Dynamics

In this section we will prove that, for the full information setting and two players the Replicator Dynamics is equivalent to the update rule of exp3 algorithm. We will start from the discrete equation and end up in the continuous equation.

$$\dot{\mathbf{p}} = \lim_{\tau \to 0} \frac{\mathbf{p}(t+\tau) - \mathbf{p}(t)}{\tau} \tag{2.19}$$

$$= \lim_{\tau \to 0} \frac{1}{\tau} \left(\frac{\exp\left\{\text{diag}(\mathbf{E}(U)_t)\tau\right\}\mathbf{p}(t)}{\mathcal{Z}(\tau)} - \mathbf{p}(t) \right) \tag{2.20}$$

where, $\mathcal{Z}(\tau)$ is the normalization factor,

$$\mathcal{Z}(\tau) = \sum_{k=1}^{n} \exp\{\mathbf{E}\{U_k\}\tau] p_k.$$

Since the limit produces a $\frac{0}{0}$ we have to take the De L'Hospital rule, so we need to compute

$$\lim_{\tau \to 0} \mathcal{Z}(\tau) = 1$$

and

$$\begin{aligned} \left.\frac{d\mathcal{Z}(\tau)}{d\tau}\right|_{\tau=0} &= \frac{d}{d\tau}\left[\sum_{k=1}^{n} \exp\{\mathbf{E}\{U_k\}\tau\} p_k\right]_{\tau=0} \\ &= \sum_{k=1}^{n} \mathbf{E}\{U_k\} p_k \\ &= \mathbf{p}^T U \mathbf{q} < \infty. \end{aligned}$$

Applying those to (2.20), we get

$$\begin{aligned} \dot{\mathbf{p}} &= \text{diag}\{\mathbf{E}\{U\}(t)\}\mathbf{p}(t) - \mathbf{p}(t)\mathbf{p}^T(t)\mathbf{E}\{U(t)\} \\ &= \text{diag}[U\mathbf{q}(t)]\mathbf{p}(t) - \mathbf{p}(t)\mathbf{p}^T(t)U\mathbf{q}(t) \\ &= \mathbf{p}(t)(\text{diag}[U\mathbf{q}(t)] - \mathbf{p}^T(t)U\mathbf{q}(t)) \end{aligned} \tag{2.21}$$

Properties of Discrete and Continuous Time Dynamics

Even though we have seen that the two dynamics can be derived from each other, when applying those dynamics in games we expect some differences.

In [64] one can found some key points of how differently the dynamics are changing

- Both dynamics eliminate any strictly dominated strategy
- The discrete-time RD may fail to eliminate a pure strategy strictly dominated only by a mixed strategy

Other discrete-time RD

Although we picked the exponential weighted equation as the discrete time replicator dynamics, there is a more general structure for the discrete-time case of RD, which is of the following form,

$$p_{i,t+1} = \frac{\alpha + f(u_{i,t})}{\alpha + f(\bar{u})} p_{i,t}, \tag{2.22}$$

where α is an appropriately chosen variable and f a function. In our case $\alpha = 0$ and $f(x) = exp(\eta x)$. Other candidates for the function are $f(x) = x$ and the INF seen in 1.2.2, i.e. $f(x) = \left(\frac{\eta}{-x}\right)^q + \frac{\gamma}{K}$. For an analysis on the equivalence of eq. 2.22 and continuous RD see [65].

2.5.3 Stability of RD

The notion of stability under the RD is crucial. There are two ways for a game played under the RD to be stable. The

first one is for an ESS to exist. The other is for the RD to be permanent.

Although we have not delved into the issue of stability of the dynamics under time delay, for the case of RD the interested reader can consult [67–70], which, in short, show that RD are stable under time delays.

2.5.4 Replicator Dynamics in Action

As we have seen, Replicator dynamics is played in continuous time. Since our interest lies in settings with single feedback and discrete structure, we will modify the algorithm to suit those needs.

As a result, in the simulations of Chapter 4 we will use the discrete algorithm along with the estimators found in 1.2.1. Of course, this setting has no difference than the exp3 and its variants, but this shows how the areas of MAB and EGT are overlapping when tackling problems of the same nature.

As a reminder, the estimators used, along with their appropriate parameters are

1. $\tilde{g}_{i,t} = \frac{g_{i,t}}{p_{i,t}} \mathbb{1}_{I_t=i}$
2. $\tilde{g}_{i,t} = \frac{g_{I_t,t} \mathbb{1}_{I_t=i} + \beta}{p_{i,t}}$
3. $\tilde{g}_{i,t} = \frac{g_{I_t,t} \mathbb{1}_{I_t=i}}{p_{i,t} + \gamma}$

2.6 Final Remarks

In this chapter we discussed decentralized algorithms that could be used for learning in games. In conclusion, one cannot expect to find algorithms that can reach an NE for

all types of games. In part, this is due to the notion of Nash Equilibrium. That is, a NE may arise naturally as a concept of stability in static games, but in a dynamic environment there is no argument if it can occur, [72].

As a result, other concepts of equilibria have emerged, such as the Evolutionary Stable Strategy and the Correlated Equilibirum, see Appendix A.

Designing Games In the designing of a game, the interest lies in guaranteeing that a learning process will reach a NE, or some other calculable quantity, and, of course these points should have interesting properties relevant to the problem. Towards this goal, when designing a game we should either build it with a special structure, or make certain that an evolutionary stable strategy is existent.

One of the structures talked above, is Potential Games first introduced by Shapley in [73]. A potential game is one that admits a function ϕ such that

$$\mathbf{p}U\mathbf{q} - \mathbf{p}'U\mathbf{q} = \phi(\mathbf{p},\mathbf{q}) - \phi(\mathbf{p}',\mathbf{q}), \forall \mathbf{p},\mathbf{p}',\mathbf{q},\mathbf{q}' \tag{2.23}$$

The fact that these games have this property helps the convergence, even in incomplete information games, even when using simple update rules, see for example [74, 75].

In wireless telecommunications there are works that have taken advantage of this structure to design games that can converge easily to a desired equilibrium, see [76–78].

In Chapter 4 we will use the game of [77] to compare the convergence but also the results to our proposed game.

Chapter 3

Game-theoretic Feedback In Telecom

As we have already seen, the game theoretic algorithms are more suitable for a full information environment, while multi-armed bandits provide an extension[1] for the case of imperfect information. As a result we are keen to treat both as one area.

In this chapter we want to address the various ways a user can receive feedback and the significance of that feedback. Primarily, the quantity of feedback required is mostly dependent on the form of the utility function. For example, we can consider the following utility functions and check their requirements,

1. $$u_i = \Theta(\text{SINR}_i - \gamma) - \frac{p_i}{P_{\max}}, \tag{3.1}$$

   with γ being the SINR threshold, Θ the step function and $P_{\max}$ the maximum power available.

2. $$u_i = \frac{\log(1 + \text{SINR}_i)}{1 + \exp(-\text{SINR}_i)} \frac{1}{p_i} - \beta \cdot p_i, \tag{3.2}$$

[1] Since we have seen the connections between update algorithms from GT and MAB

with β an appropriately chosen parameter.

3.
$$u_i = \log(1 + \text{SINR}_i) - \gamma p_i^2, \tag{3.3}$$
with γ being the SINR threshold.

In the first case, the feedback received can be as little as one bit. Using that knowledge, the transmitter is able to update the utility function. If the channel conditions allow for a higher feedback rate, the receiver could communicate more bits, specifically as much as $\lceil \log K \rceil$ bits are sufficient in order to inform the transmitter about which action would yield the highest payoff and, thus, forming a full information environment.

Considering the second and third equations, at a first glance the feedback needed seems to be infinite. But if, transmitter and receiver have agreed upon a set of predetermined values, the feedback could take the form of a Channel Quality Information (CQI) as done in cellular systems, [79–81], i.e. a limited feedback could be almost as helpful as the analogue one. Due to the required information to update the utility function, this game can be considered a full information setting, since all actions can be updated, using this feedback.

3.1 Feedback Strategies

In what follows, we will list the various feedback strategies and their uses in telecommunications.

Full information setting The first feedback we will consider is to receive the full information vector, i.e. for all actions, hence, called the full information setting. Taking advantage

of this structure, one has the ability to compare all strategies to the highest payoff, contrary to the expectation, as is done in the limited information setting. The appropriate telecommunication problem, is one that can provide enough information as feedback so the user can update all the available actions. Examples range from the exact channel conditions, something that is easily done in TDD [82], or, in some cases, communicating the most appropriate choice can provide enough information, e.g. the game presented above.

Incomplete Information Setting Although a full information setting provides many benefits, feedback is hard to get and, there is a possibility of delayed, erroneous or imperfect feedback. As a result, there is the possibility of absence of feedback. From the above examples, we have seen that simpler utility functions are more reliable in low feedback settings.

If that is the case employing the help of the gain estimators is necessary, see section 2.5.4.

Multiple Actions Feedback Another bandits' setting is the Combinatorial Bandits where, in each round, a subset of the actions is chosen and is played. As a result, the limited information combinatorial bandits problem provides payoffs only for the actions played. The goal of the bandit is to optimize the actions' subset that is selected in each round. This bandits problem is particularly useful for the case of a MIMO setting, where one has multiple antennas and needs to decide how to allocate resources among them. For a review on combinatorial bandits see [84], while for applications in telecom problems see [85–88].

Chapter 4

Simulating Power Control Games

This chapter is dedicated to the comparison between the algorithms/learning techniques that are found in the previous chapters.

The main motivation is to see whether the algorithms presented in this thesis can get close to the available equilibria, which, in turn, would be a strong indicator that using NE for the analysis of games is a good choice.

To achieve the above, we will perform simulations for the following two problems.

1. A power control game designed for sensor networks and analyzed in [89]. The main focus is on achieving the highest rate possible, while lowering the power levels.

2. A power control game with application to an adhoc environment. This game's equilibria have been analyzed in a previous work, for a two-player game. The results show either a pure NE, which is the optimal point shown in figure 4.1. If that point is not in the achievable region of P_{max}, then there appears a mixed NE which forces users to randomize in a way that balances the time each one communicates.

The first game has the following utility function

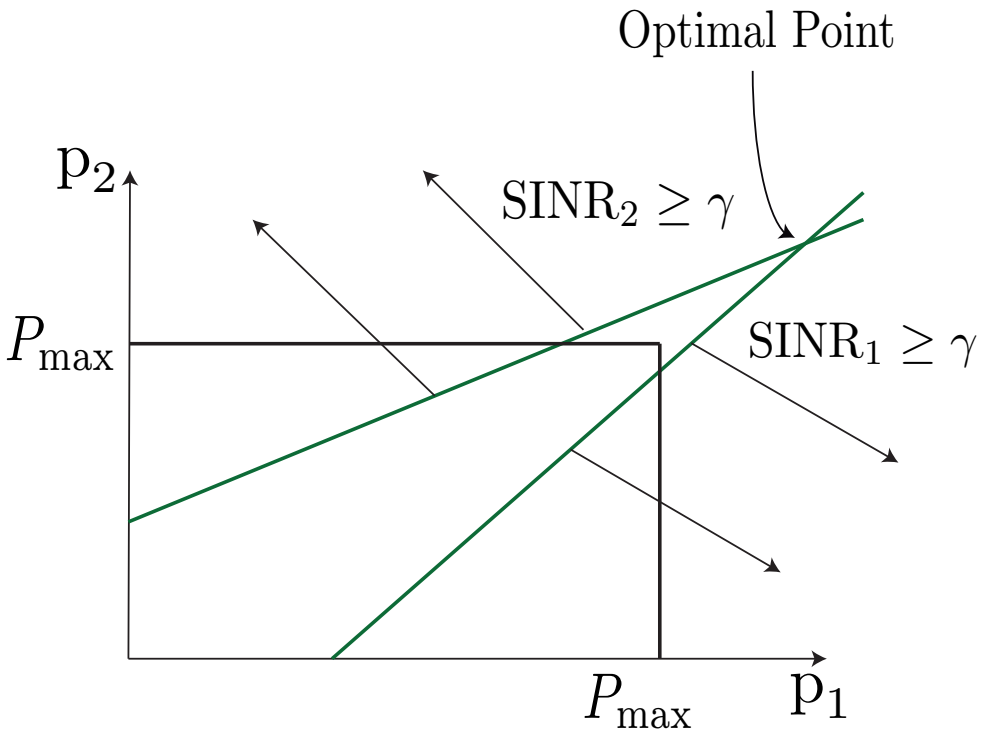
$$u_k = \frac{\log_2(1+\text{SINR}) - \log_2(1+\gamma)}{p_k}, \tag{4.1}$$

while the other has the following

$$u_k = \Theta(SINR - \gamma) - \frac{p_k}{P_{max}} \tag{4.2}$$

Due to the form of the utilities we will call, from now on, the first game as Rate Power Control (RPC game), while the second as Connectivity Power Control (CPC game).

Figure 4.1: Points of Interest in a 2-user setting



We can see that the two functions are inherently different. The CPC is extremely simple and requires only 1-bit feedback from the receiver, while RPC requires precise information regarding the SINR received. This difference will

help us explore how feedback affects the outcome. In particular, when simulating the RPC game, the available feedback at the transmitter (CSIT) will be limited to n-bits. Since we don't know the bounds of SINR, we will assume the most easy to obtain, i.e. $\text{SINR}_{\text{min}} = 0$ and $\text{SINR}_{\text{max}} = P_{\text{max}}$, while the feedback received will provide a quantization of the SINR received.

Moreover, both games will be illustrated in an interference channel, for two or more users.

4.1 Methodology

In order to help our analysis, we will use - for a two user setting - a figure similar to 4.1. As we can see, if the optimal point is inside the P_{max} region, then the desirable property of any algorithm applied to the utility of CPC, is the convergence to that point. On the other hand, the CPC utility function should force users inside the cone that is formed by the following equations

$$\text{SINR}_1 \geq \gamma \tag{4.3}$$

$$\text{SINR}_2 \geq \gamma. \tag{4.4}$$

This chapter will help in the comparison of the algorithms presented in previous chapters, by using simulations. The quantities that we will focus on are

- the empirical distributions of actions, given by

$$\hat{p}_{i,T} = \sum_{t=1}^{T} \mathbb{1}_{I_t=i}, \ \forall i \tag{4.5}$$

- the strategy profiles of users,
- the average connectivity and power, and

- the connectivity percentage of each user.

4.1.1 Simulating exp3

The algorithms that will be used are from the exp3 algorithmic family. As a reminder, vanilla exp3 and exp3.IX update strategies by, calculating the weights, while exp3.P uses a mixture of weight-calculation comnined with a uniform distribution. The estimators for each case are

- Vanilla exp3 estimator

$$\tilde{g}_{i,t} = \frac{g_{i,t}}{p_{i,t}} \mathbb{1}_{I_t=i} \tag{4.6}$$

- exp3.P estimator

$$\tilde{g}_{i,t} = \frac{g_{I_t,t} \mathbb{1}_{I_t=i} + \beta}{p_{i,t}} \tag{4.7}$$

- exp3.IX estimator

$$\tilde{g}_{i,t} = \frac{g_{I_t,t} \mathbb{1}_{I_t=i}}{p_{i,t} + \gamma} \tag{4.8}$$

Some learning rules have been omitted from this chapter for the following reasons. From the EGT chapter

- the simple algorithms, since we have shown that are not capable of converging even in much simpler games[1]
- the Best Response and the Smoothed BR, since, either we have argued about specific problems that arise from their use, or, because they need knowledge of the whole game (utility matrix) in order to update the strategies.

[1] The most simple games known are the 2×2 games.

- Additionally, the Logit dynamics and Replicator Dynamics can be seen as special cases of the exp3 with the use of the above estimators.

Moreover, from the MAB chapter

- The INF (Implicit Normalized Forcaster) since the authors of [37] don't provide with the appropriate variables that achieve those resutls, and most importantly due to the peculiar logarithm in their estimator, which needs extra care in selecting the above parameters.

Simulations

This section is dedicated to the presentation of the results of the simulations. The first part is dedicated to the comparison between the learning methods, which we will apply on the CPC utility function, in, both a pure NE setting and in a mixed NE setting.

Then we will compare the two utility functions under the same learning algorithm. Actually, this part is further broken down to two parts the 2-user case and the many user case. Since the two user case is more easily analyzed, it will allow us to examine how the two utility functions distribute the available resources. On the other hand, we will also focus on the performance of the utility functions in the case when the number of users is ~ 10 and, as such, the interference received is large.

4.2 Comparison of Learning Algorithms

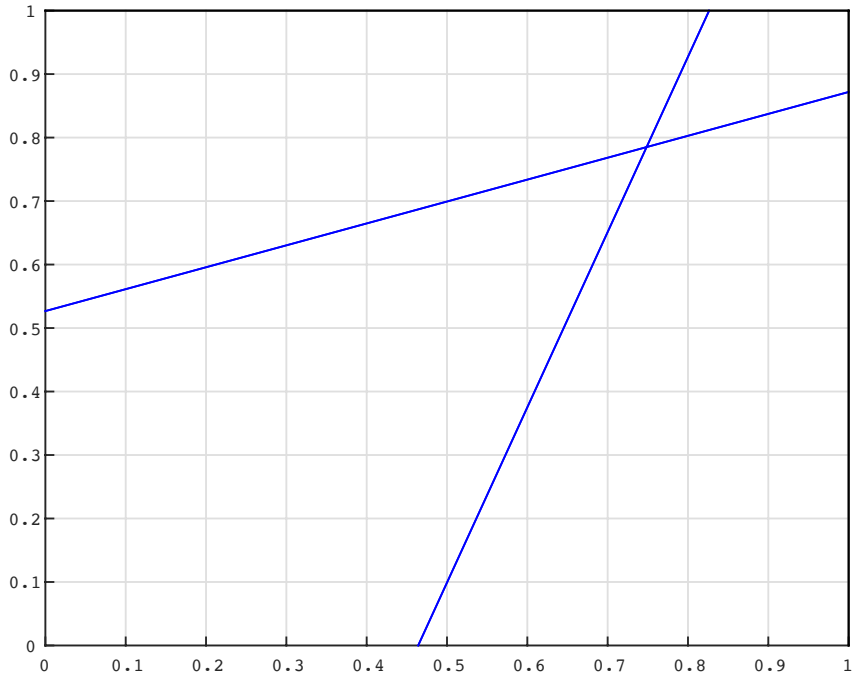
4.2.1 The pure NE case

In this part we will compare side-by-side the learning algorithms presented in this thesis. The utility function that will

be used is the CPC, for which a thorough analysis has been performed, and, thus, allowing for a good understanding of what to expect.

Firstly, we will be concerned with the pure NE, and specifically with the setting of fig 4.2. Ideally, one would expect all algorithms to converge in some point inside the cone (and, of course, inside the region given by the maximum available Power).

Figure 4.2: The SINR region for the 2-player game with a pure NE



The connectivity achieved by users as a whole can be seen in fig 4.3.

The first observation is that exp3.P achieves lower connectivity than the other two algorithms. Regarding the performance of the other two algorithms, we can see that, both achieve high connectivity, close to the desired point.

Figure 4.3: Connectivity for the game of figure 4.2

Figure 4.4: Vanilla Exp3

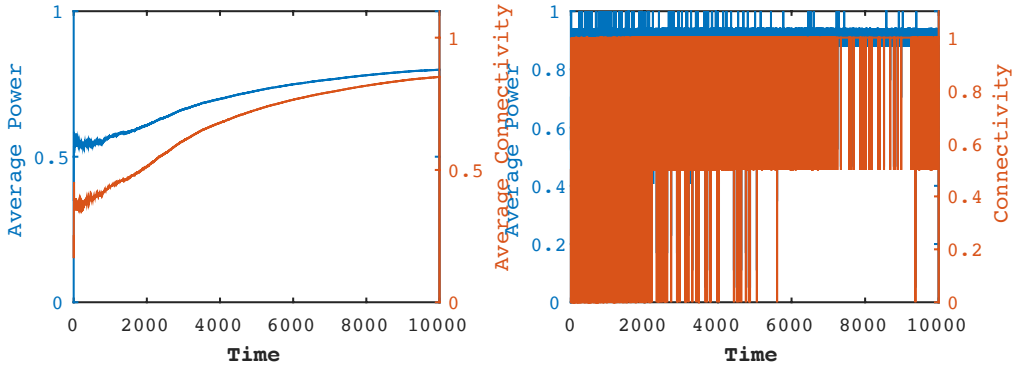


Figure 4.5: exp3.P

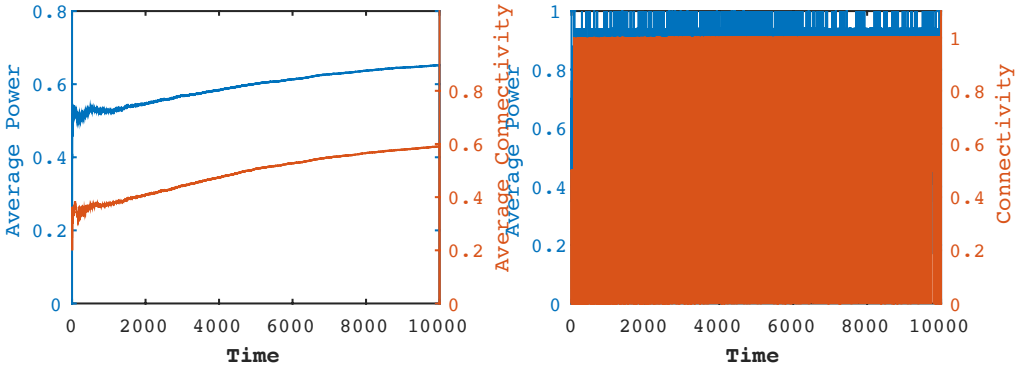
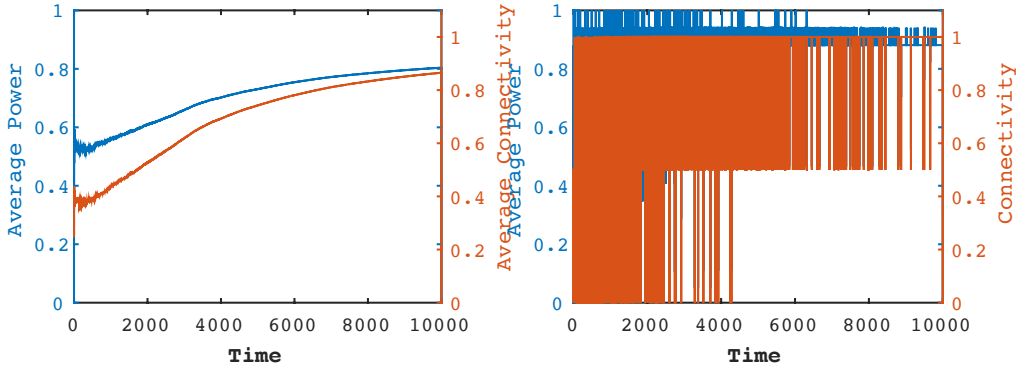


Figure 4.6: exp3.IX



4.2.2 The case of mixed NE

This setting can be seen in figure 4.7. A characteristic of this is that even though player 1 could choose $P \approx 0.9$ and always be connected, the CPC utility function should drive both to randomize, as can be seen by our analysis. This effect is due to player 2's utility being negative and, thus, forcing player 2 to change to $P = 0$ and, as a result player 1 would have a better option, i.e. less costly in terms of power, while maintaining connectivity.

As such, we are interested in learning algorithms that could reach this point and not just stick to $\mathbf{P} = (1, 0.9)$.

Figure 4.7: The SINR region for the 2-player game with mixed NE

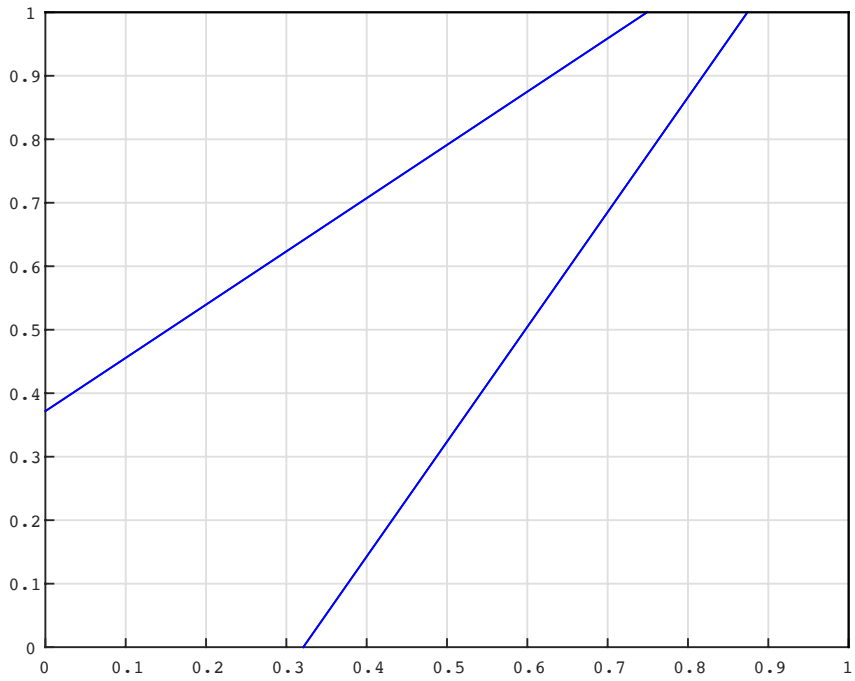


Figure 4.8: Empirical Distributions for the game of figure 4.7

Figure 4.9: Vanilla exp3

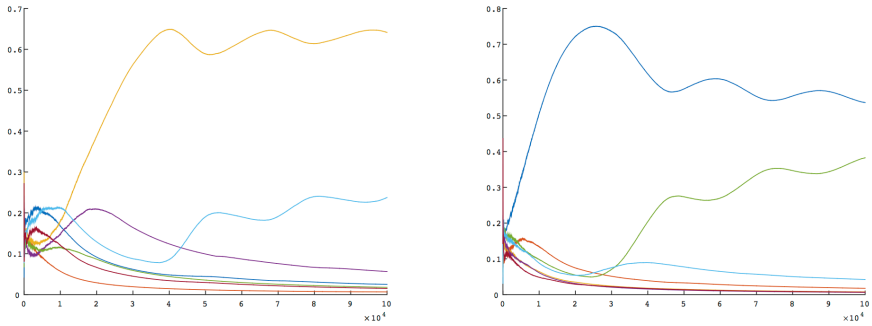


Figure 4.10: exp3.P

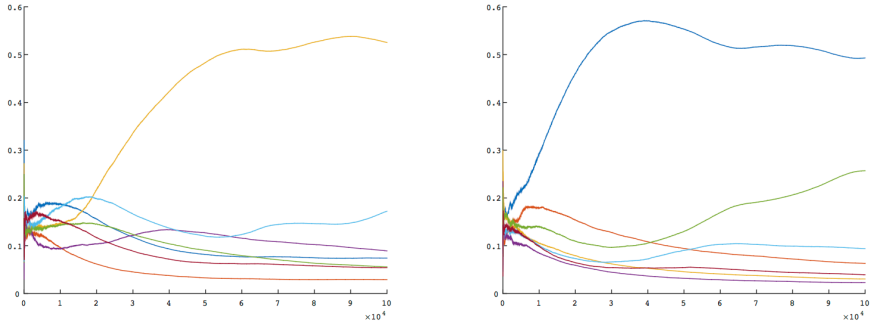
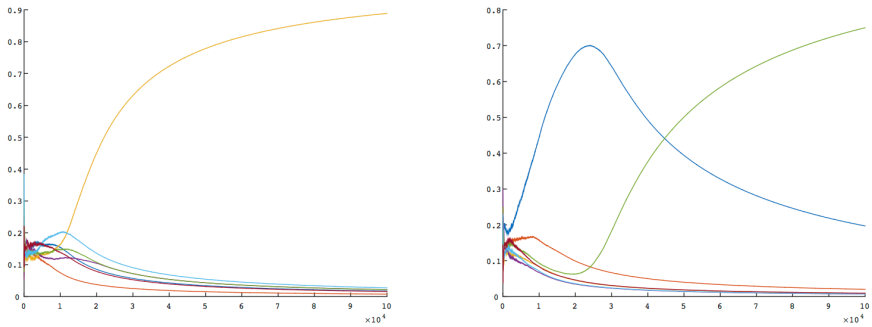


Figure 4.11: Exp3.IX



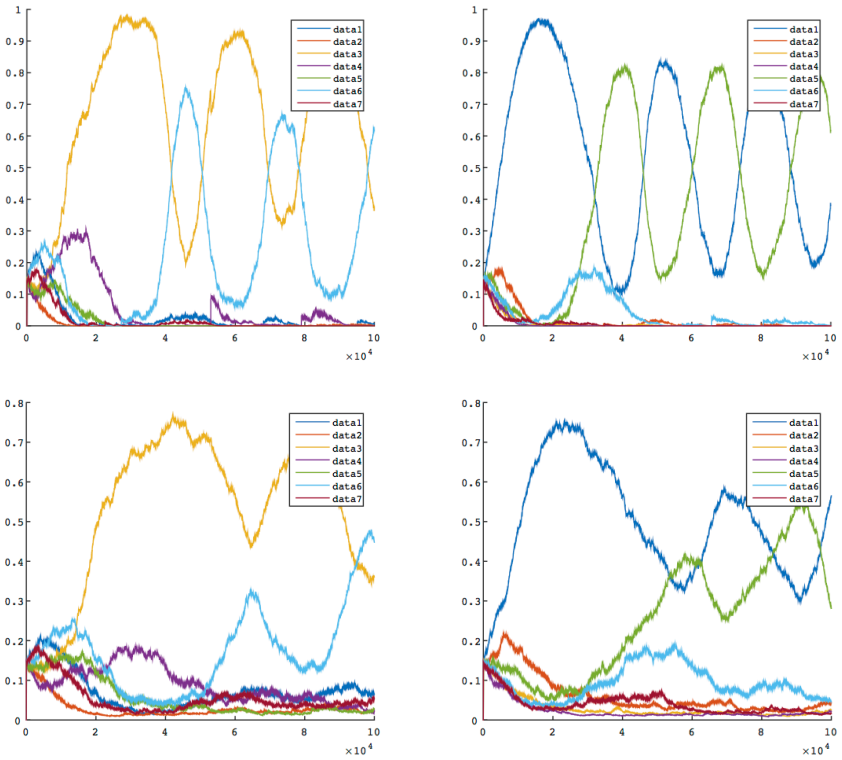
By examining the above results, a first observation, is

that exp3.IX algorithm is yielding the exact opposite result from what one wants, i.e., converging to a pure equilibrium.

In this example, we will employ the empirical distributions in order to get a picture of what players end up choosing and compare that with the dynamics of the strategies of the two players, fig 4.12. This result is extremely powerful, since we observe that, although the strategies converge only on average on the desired mixed NE, in reality, players end up playing the mixed NE, with some small perturbation.

Moreover, For the case of vanilla exp3 there is a bigger variation, compared to exp3.P. This is expected if one recalls the analysis from MAB chapter regarding the variance of these algorithms.

Figure 4.12: Actual Distributions of players for the mixed game of fig. 4.7. Algorithms vanilla exp3 and exp3.P



From the above figures we can deduce that both vanilla exp3 and exp3.P have the ability to converge on average in a mixed NE, and, actually, engage in it, if we consider the empirical distributions.

Figure 4.13: Overall Connectivity for Vanilla exp3 & exp3.P

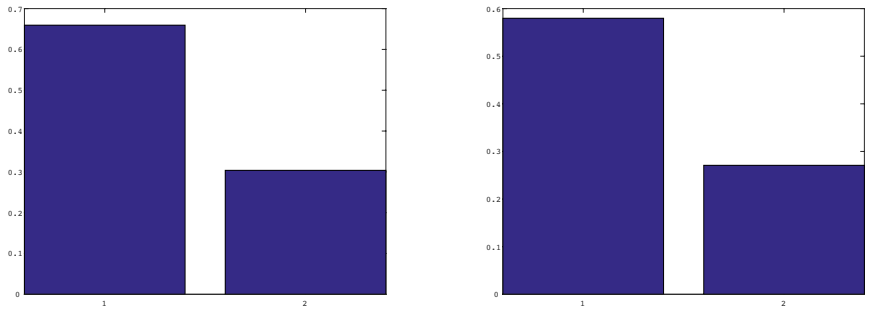
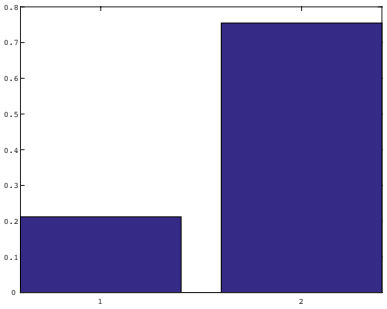


Figure 4.14: Overall Connectivity for exp3.IX



4.3 Two Players - Comparison of Utility Functions

As promised, this section is dedicated in comparing the two utility functions in various settings and exploring what is the part of the quantity of feedback in the convergence of these algorithms.

4.3.1 Pure Equilibrium

The setting of interest in this subsection can be seen in figure 4.15, and is similar to the one in fig 4.2.

Having compared the various learning algorithms, in this section we choose the original exp3 algorithm to perform the comparison between the two utility functions, under various scenarios. This selection is due to the performance of the algorithm both in the pure and the mixed NE.

Figure 4.15: The SINR region for the 2-player game with a pure NE

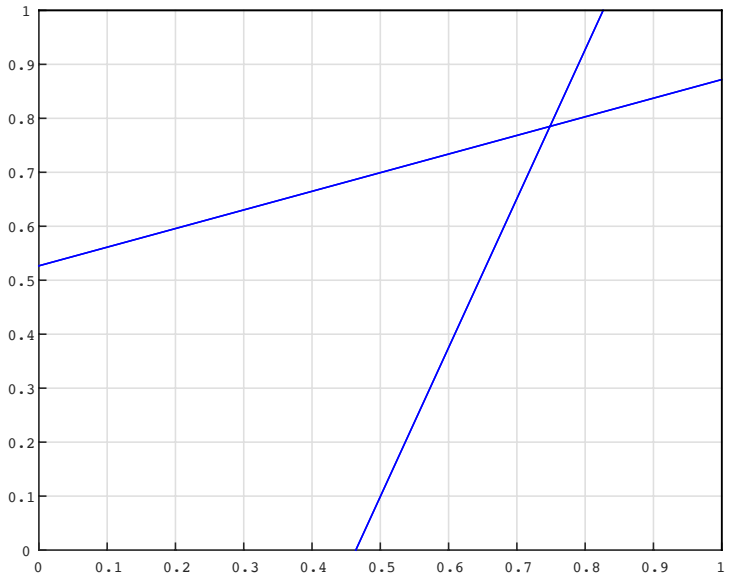
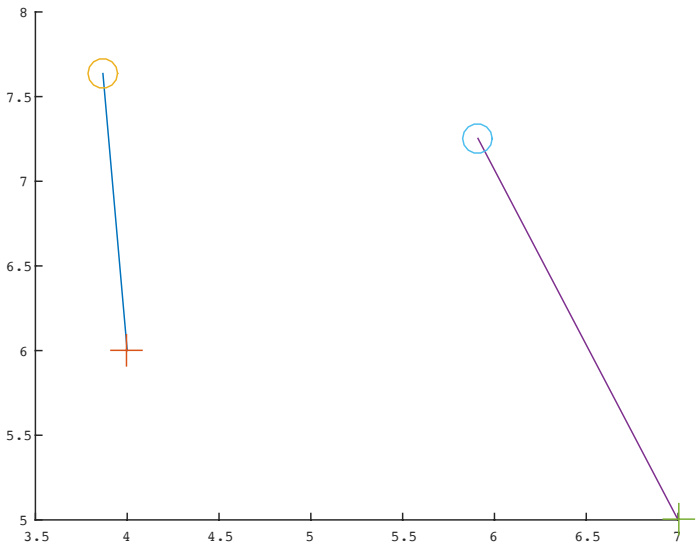


Figure 4.16: The two pairs of the above game in the plane



From figures 4.17-4.22 we can see that as the quantity of feedback grows, the final result converges faster to the optimal point[2]. Also, we can obserce that even with 1-bit feedback, in the case of the CPC utility function, the NE can be achieved, and that, actually, the 1-bit utility function achieves lower power usage, something that is more easily seen in the figures of the empirical distributions.

It can be easily assumed that feedback should be a function of P_{max}, in the following sense,

$$FB(bits) = fb_0 + \log_2 P_{\max} \tag{4.9}$$

where fb_0 is the quantity of reference in the case of $P = 1$, which could be in the range $4 - 7$ bits. Of course this modification holds only for the two person game. Adding more users should require higher precision in the SINR estimation, although the correlation isn't as straightforward. For

[2]The optimal point is easily detected, as it yields a connectivity of 1.

that reason we will have to examine this relationship further in the respective section, i.e. in the many players section.

Figure 4.17: Power & Connectivity for the game of figure 4.2 with the vanilla exp3 algorithm

Figure 4.18: The “simple” power control problem with 1-bit feedback

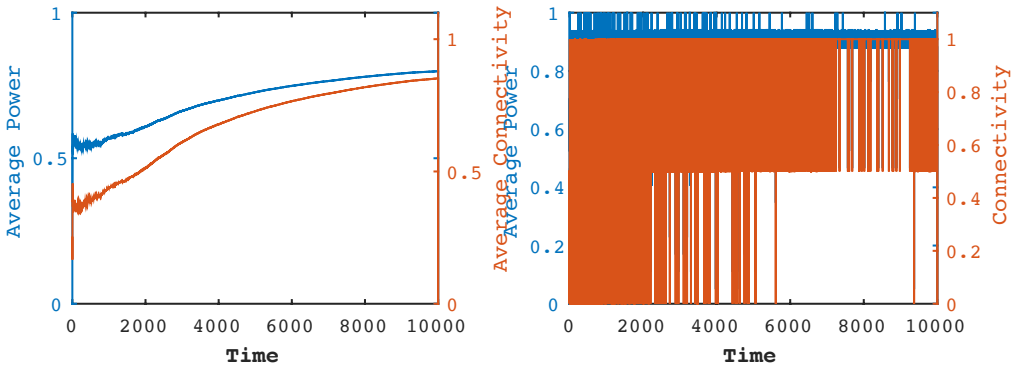


Figure 4.19: The 2-bit feedback power control

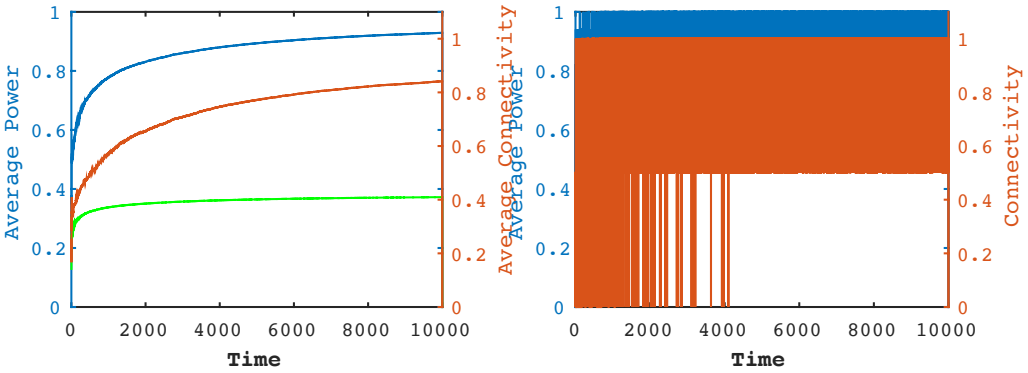


Figure 4.20: The 4-bit feedback power control

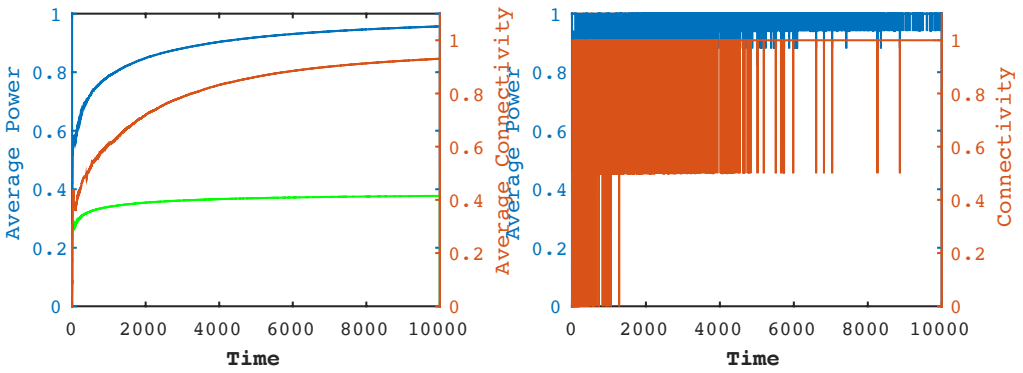


Figure 4.21: The 7-bit feedback power control

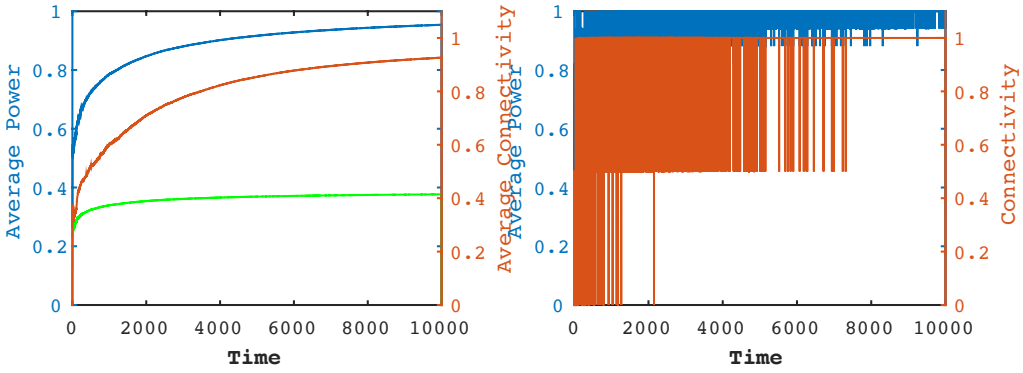
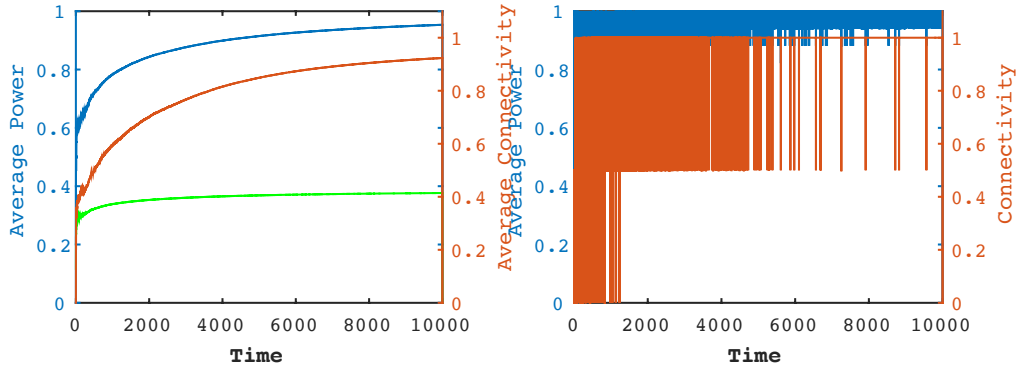


Figure 4.22: The 10-bit feedback power control



The following figures present the empirical distributions of the players. One thing that we can observe is that about 4 bits would yield the same result to a more feedback rich environment.

Figure 4.23: Empirical Distributions for the game of figure 4.15 with the vanilla exp3 algorithm

Figure 4.24: Empirical Distributions for the 1-bit game

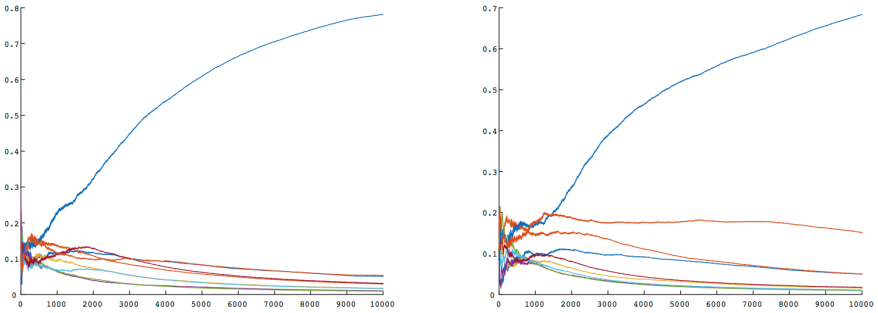


Figure 4.25: Empirical Distributions for the 2-bits game

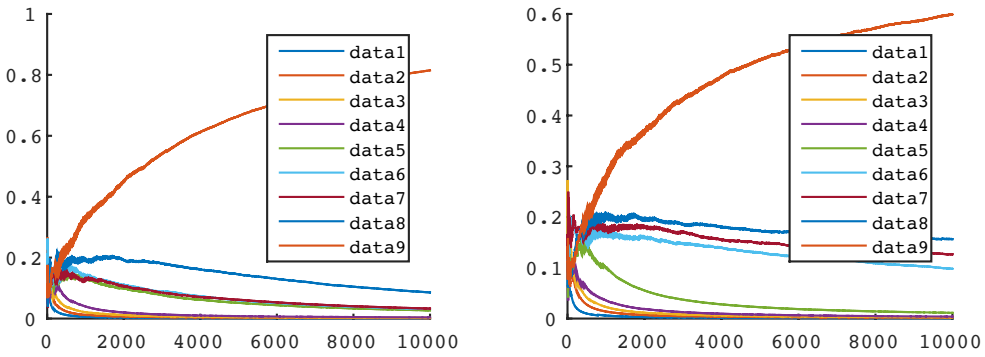


Figure 4.26: Empirical Distributions for the 4-bits game

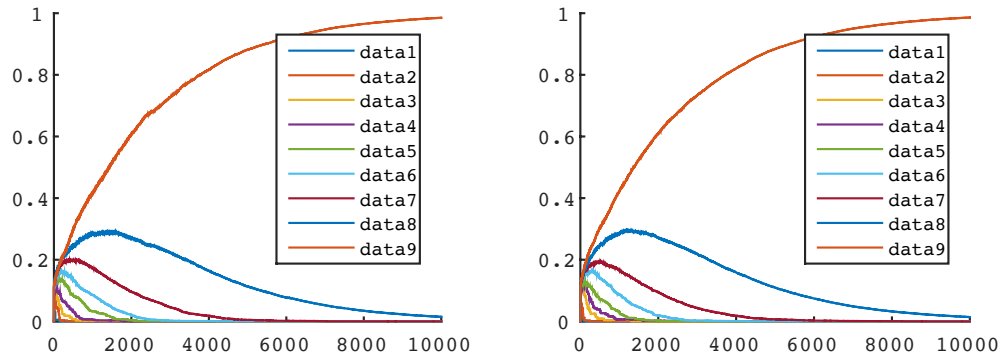


Figure 4.27: Empirical Distributions for the 7-bits game

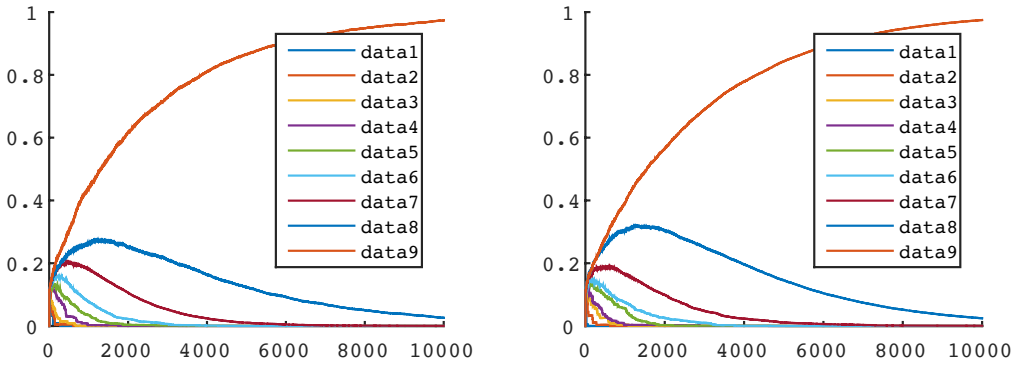
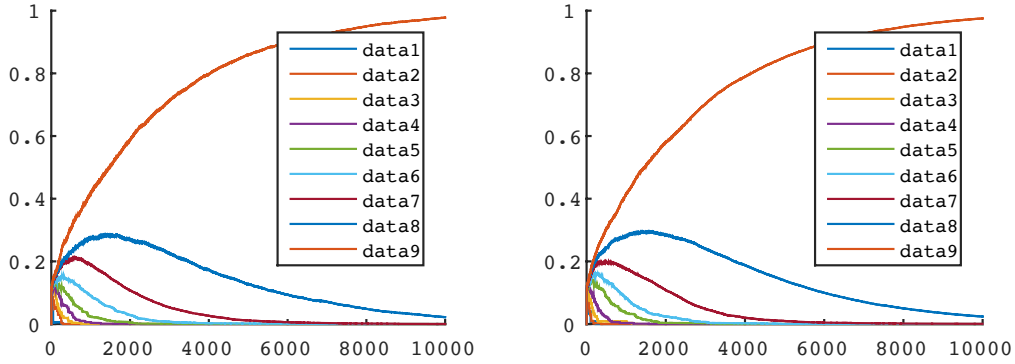


Figure 4.28: Empirical Distributions for the 10-bits game



We can see from the empirical distributions that, even 4 bits are enough for the system of the second model to reach the optimal point.

Figure 4.29: Overall connectivity using vanilla exp3 in the pure NE game, 4.15

Figure 4.30: Overall Connectivity for the 1-bit game

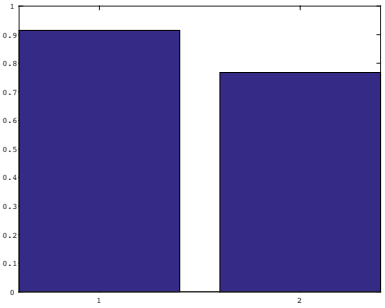


Figure 4.31: Overall Connectivity for the 2-bit & 4-bits games

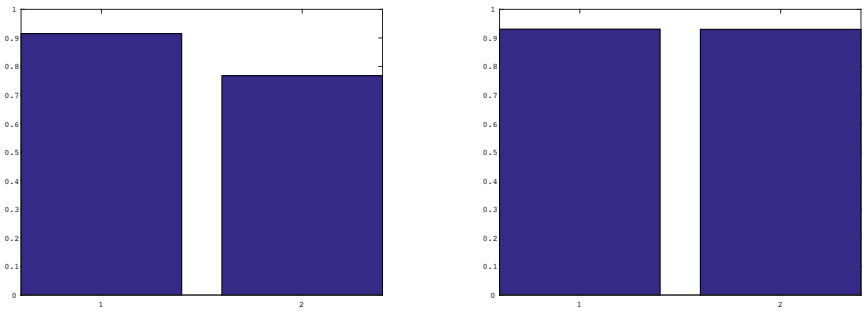
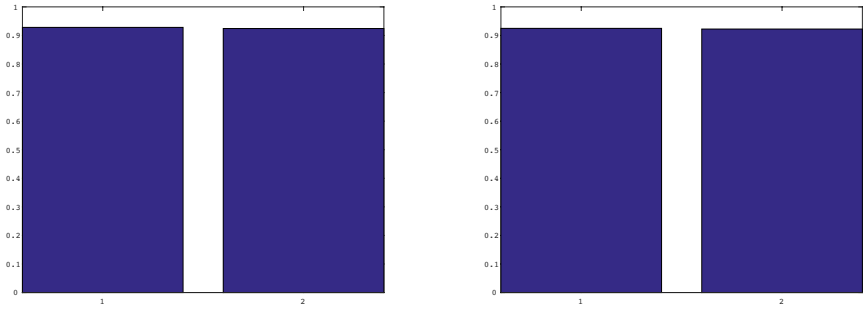


Figure 4.32: Overall Connectivity for the 7-bit & 10-bits games

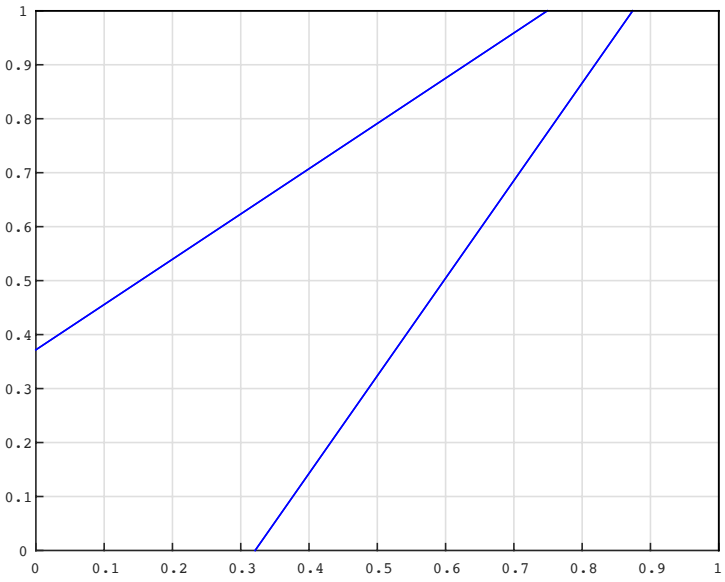


As expected, observing fig 4.39 we can deduce that both algorithms can achieve high connectivity, and, also, that as the quantity of feedback increases so does the performance of the RPC utility function, i.e. reaching the optimal point faster.

Finally, comparing the two utilities the CPC achieves the same connectivity with the RPC, with the sole difference of lower power.

4.3.2 Mixed Equilibrium

Figure 4.33: The SINR region for the 2-player game with mixed NE



Given the above setting we are interested in how these two players change their actions and adapt their strategies.

Figure 4.34: Connectivity using exp3.P in the mixed NE game, 4.2

Figure 4.35: Connectivity for the 1-bit game

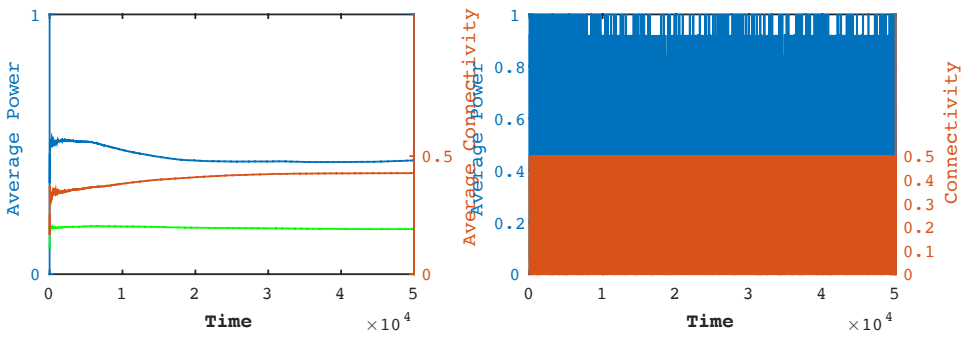


Figure 4.36: Connectivity for the 2-bit game

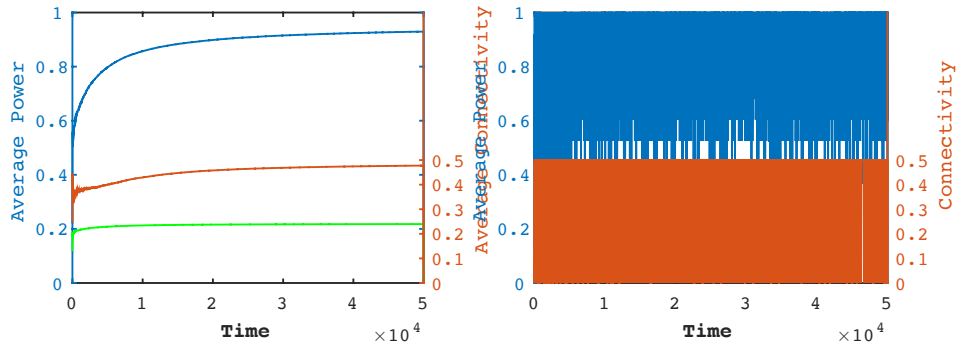


Figure 4.37: Connectivity for the 4-bits game

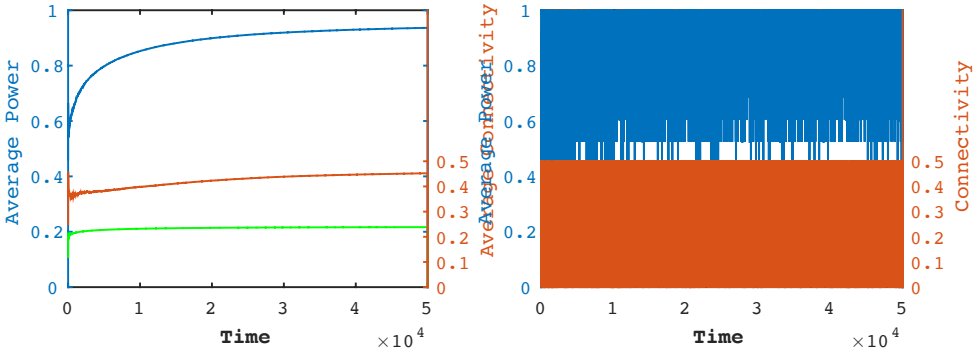


Figure 4.38: Connectivity for the 10-bits game

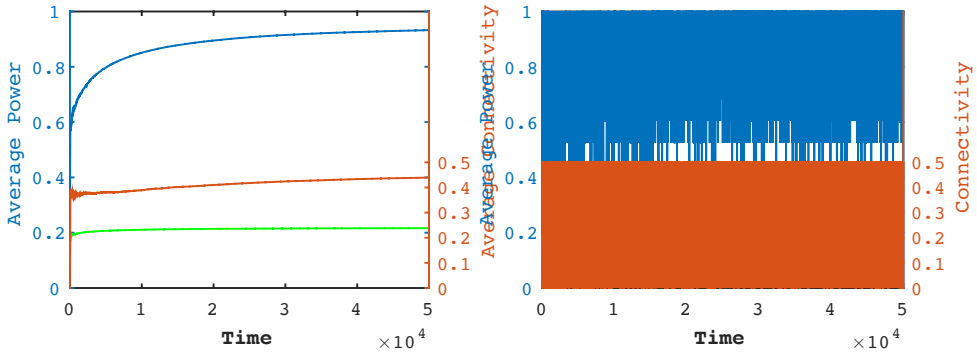


Figure 4.39: Overall connectivity using vanilla exp3 in the pure NE game, 4.2

Figure 4.40: Overall Connectivity for the 1-bit & 2-bits games

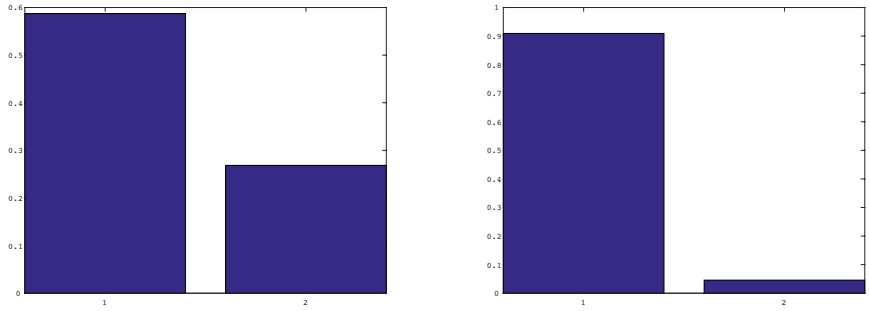
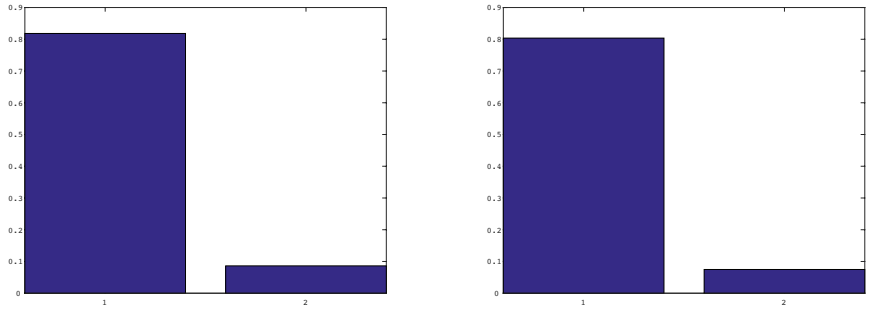


Figure 4.41: Overall Connectivity for the 4-bits & 10-bits games



From the overall connectivity we can see how the CPC utility provides a more balanced connectivity between users, while keeping the power in lower levels.

Contrary to that, using the RPC utility, we can see that it drives both users to consume as much power as possible, even though one of them cannot establish a stable link.

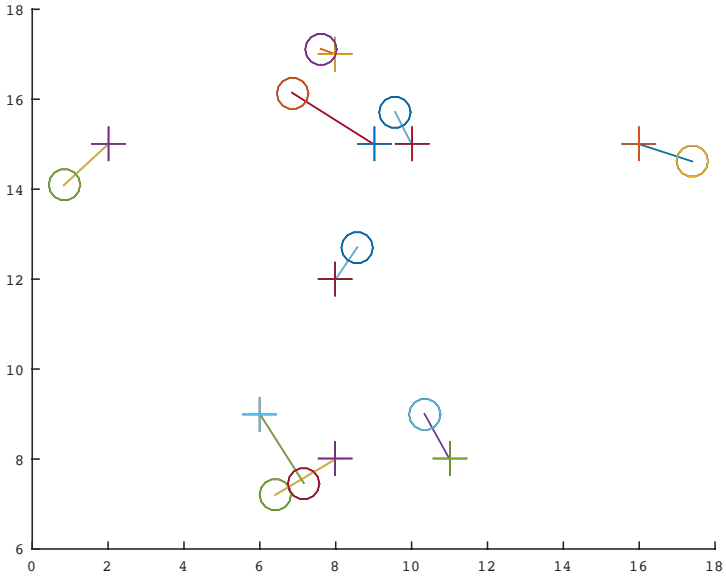
4.4 Many Players

This section is dedicated to a multi-player scenario. Even though an analysis is performed in a two user setting, for

the easiness of calculations, one cannot expect a two player scenario to be the most common, or to be that adversarial as to result in mixed NE.

Motivated by that, we use the setting presented in figure 4.42, having 9 pairs to draw some results on a more realistic environment.

Figure 4.42: 9 Pairs in the plane



This setting is of interest because we have all types of users, ones with achievable pure NE, others that experience mild interference and have to randomize their actions, and, finally, others who experience high interference and, due to the utility, shut down transmission.

Figure 4.43: Connectivity using vanilla exp3 in the many users scenario, 4.42

Figure 4.44: Connectivity for the 1-bit game

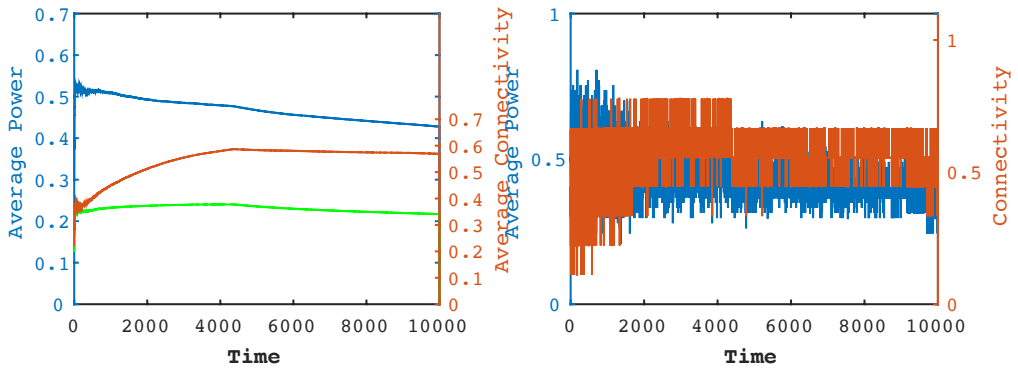


Figure 4.45: Connectivity for the 2-bit game

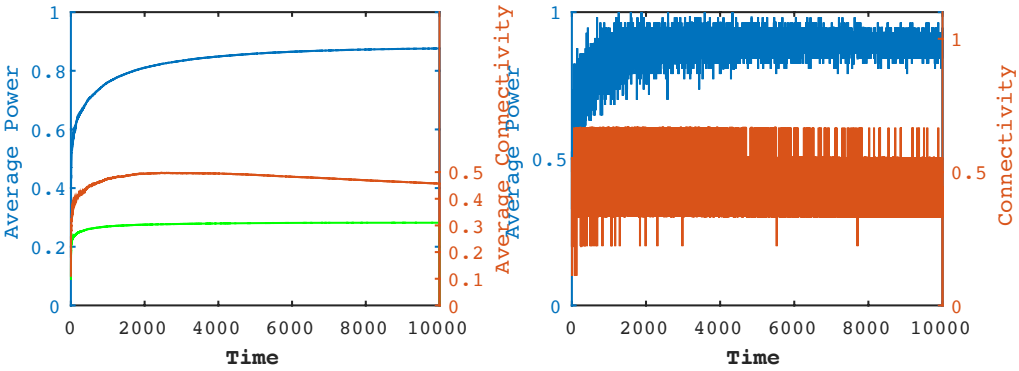


Figure 4.46: Connectivity for the 4-bit game

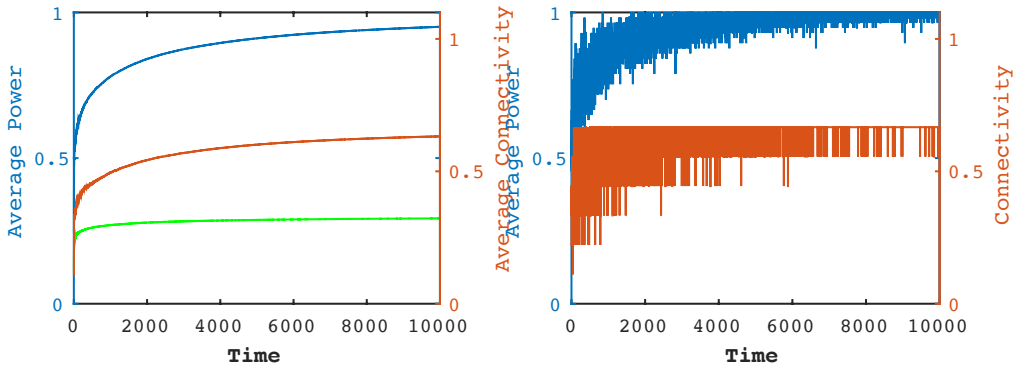


Figure 4.47: Connectivity for the 7-bit game

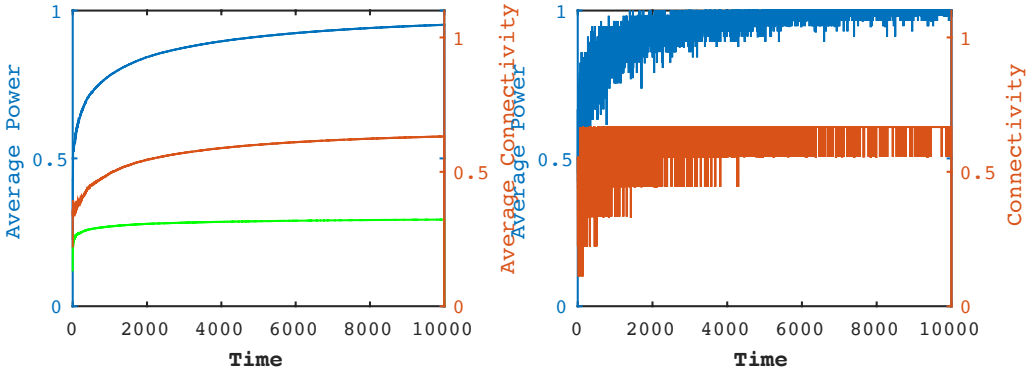
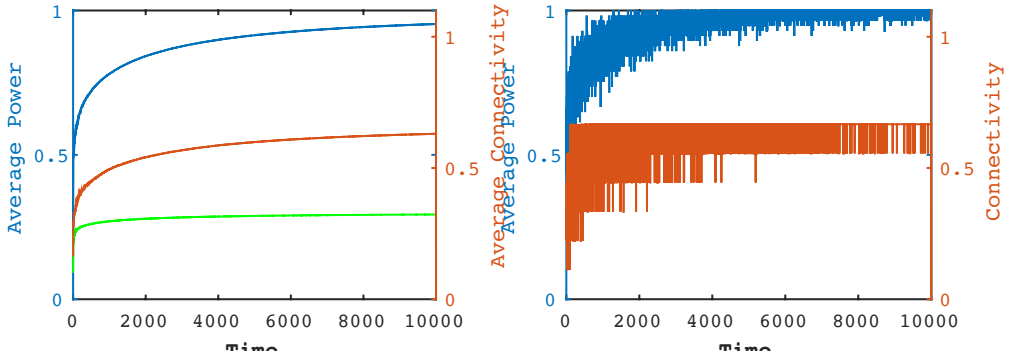


Figure 4.48: Connectivity for the 10-bit game



Many interesting results can be drawn from the above figures

- The CPC game has as sole purpose to reduce power, while maintaining, or even increasing the connectivity of the system.
- The RPC utility focuses on increasing the rates, even if that means reaching P_{max}, which it does, rather quickly.
- An adequate quantity of feedback is ~ 4-bits, even in the case of many users.
- The RPC game provides power control only in the case of a pure NE (optimal point). If that point is not achievable, then power is maximized.

Figure 4.49: Overall Connectivity using vanilla exp3 in the many users scenario, 4.42

Figure 4.50: Overall Connectivity for the 1-bit game

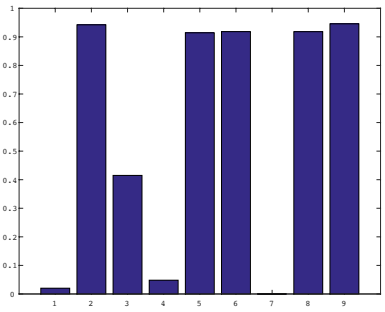
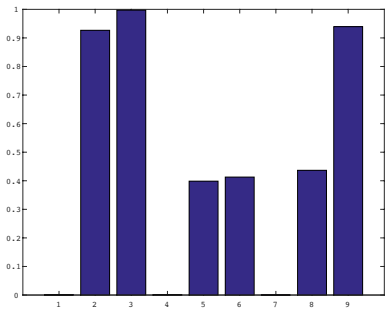


Figure 4.51: Overall Connectivity for the 2-bit & 4-bit games



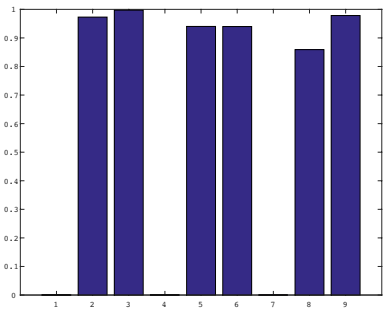
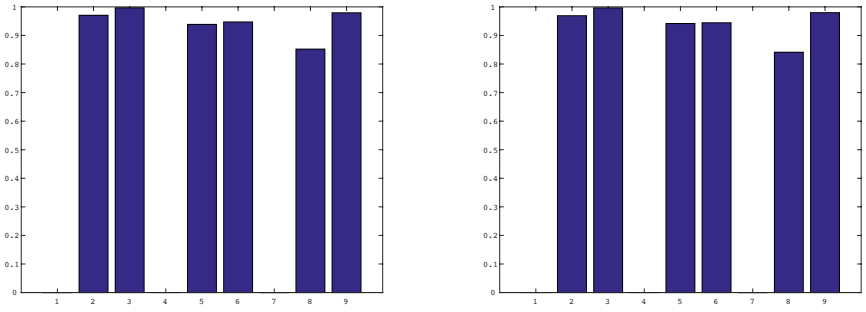


Figure 4.52: Overall Connectivity for the 7-bit & 10-bit games



Moreover, taking into account the overall connectivity, one can see that the RPC utility function allows more users to communicate, in the long run, although the average connectivity stays the same.

4.5 Final Remarks on Learning

The purpose of this chapter was two-fold. On the one hand, the focus was on exploring how the learning algorithms of the previous chapters compare to one another. What we have seen is that the two algorithms, vanilla exp3 and exp3.P, have the most favourable results.

Another observation is that the above algorithms can reach, not only a pure NE, but most importantly converge to a mixed equilibrium. As a result, analysing a game through Nash equilibria is an appropriate approach from a learning theory perspective.

On the other hand, we compared the utility functions and, specifically, the results under various scenarios. The conclusions we can draw are that if a utility is a function of SINR, then it needs a higher amount of feedback, which is in the range of $4-7$ bits. That quantity increases linearly

to $\log_2 P_{\max}$, but remains unaffected by the increase of interference. That means that a quantity of $4-7$ bits (P$=1$) is adequate to describe the interference received.

Moreover, when comparing the two power control methods, we can see that, the CPC focuses on the power control aspect more than the other. Also, the CPC function is more appropriate for adversarial environments and when feedback is hard to get. If one has an environment that is feedback-rich, while most users can communicate, then the RPC utility drives the system to rate maximisation.

About the design of a utility function. When designing the simulations a recurring problem that arose was the difficulty of learning algorithms to adapt in utility functions that are not restricted in the $[0,1]$ range. Also, if the values produced by the utility are not distributed in a, somewhat, evenly manner among those values the learning algorithms would not be able to discern them. As an example, we can take a look at the following utility, appearing in [90]

$$u_k = \frac{\log_2(1+\text{SINR}_k)}{1+\exp(-\text{SINR}_k)} \cdot \frac{1}{p_k} - p_k \tag{4.10}$$

When applying the learning methods to this function one has to have clear bounds that are the same regardless of the environment, something that is not easy to do, since SINR could yield extreme values. If, for example, the environment has high interference, it will yield utility values that are close to each other and, as such it may take very long to convergence.

Additionally, this function, when considered in a low interference environment will produce values with big differences from each other which would result in jumps in the

strategies. Since, the results are environment-dependent, adjusting this utility to specific bounds is not a trivial task, thus requiring modification of the utility.

Finally, one can compare that to the utility functions we used, which, regardless of the environment, could be bounded.

Appendix A

A Short Review of Equilibria

This appendix is concerned with the analysis of the equilibria that appeared in Chapter 2. Although there is a plethora of equilibria appearing in the literature, this appendices goal is to give a better insight in the ones that appear in the former problems.

A.1 Nash Equilibrium

The most famous equilibrium was introduced by Nash in [91]. The notion of a NE is that given rational agents engaged in a game, none would prefer to deviate from this strategy, since the expected payoff would be smaller.

For a 2-player game, Nash equilibrium can be stated as

$$\begin{aligned} \mathbf{p}^\star U_1 \mathbf{q}^\star &\geq \mathbf{p} U_1 \mathbf{q}^\star \\ \mathbf{q}^\star U_2 \mathbf{p}^\star &\geq \mathbf{q} U_2 \mathbf{p}^\star \end{aligned} \tag{A.1}$$

which is, for the pair of strategies $\mathbf{p}^\star, \mathbf{q}^\star$ none can be better off by unilaterally changing his strategy.

A.2 Correlated Equilibrium

Another refinement of Nash Equilibrium is the Correlated equilibrium (CE), which was introduced in [92]. While each NE in a game is a distribution over players' available actions, which is called a strategy, a CE is a distribution over strategies, which is called a correlated strategy. As such, a NE is always a CE and, thus, every game has at least a NE.

Aumann, who proposed the use of CE instead of NE, argues in [93] about some of the problems of NE. His basic points are:

1. Although the notion of NE is consistent to itself, i.e., if any player's best response to all other players' strategies is equal to his current strategy then that is a NE, it creates a chicken and egg situation, i.e., of which of the two came first.

2. Assuming a game with only one equilibrium, one cannot guarantee that all other players will strategize in the same way and play the equilibrium.

3. When there are more than one equilibria in game, how can a player know which one to play so as to align with other players' decisions?

4. A NE makes sense if a player knows other players' strategies, which is rather difficult.

A.3 Evolutionary Stable Strategy

The ESS is a stricter concept than NE, it was introduced in [43] and adds the restriction of stability to a Nash equilibrium. To make the concept more easily understood, we

will assume that a game is played by two populations, where each population is extremely large and is divided among the available actions of the player it belongs to. In each step each member is interacting with the other population by playing his particular action and receiving a payoff. Then, a strategy is ES if

- The population of every action is constant (NE requirement)
- By inserting a small group (mutants) in the population the system will, eventually move to the prior strategy

Formally on top of eq. A.1 there is,

$$\begin{aligned} ((1-\epsilon)\cdot \mathbf{p}^\star + \epsilon \cdot \mathbf{p}) U_1 \mathbf{q}^\star > \mathbf{p}^\star U_1 \mathbf{q}^\star \\ ((1-\epsilon)\cdot \mathbf{q}^\star + \epsilon \cdot \mathbf{q}) U_2 \mathbf{p}^\star > \mathbf{q}^\star U_2 \mathbf{p}^\star \end{aligned} \tag{A.2}$$

In other words, there is a neighbourhood of an ESS which is asymptotically stable.

A remark on Equilibria

Though there are different kinds of equilibria, one should ask for which type of equilibrium one should design a game and then how these can be achieved.

Although NE is an extremely important notion, on the other hand it has been seen that other concepts of equilibria can replace it depending on the application. For example, we have seen that in EGT the concept of ESS is more appropriate since a NE is always stable, but, not always asymptotically stable.

On the other hand, there are arguments that since NE is a PPAD problem, then one cannot expect the agents to be always able to reach it.

Finally, comparing CE and NE, we can say that even though a NE appears to be natural and the idea of randomization among strategies may be odd, the results of learning dynamics show that a CE occurs often.

Appendix B

Power Control's Matlab Code

```
function[]=powerControlSimulator()
%% this enlists many functions for simulation
%% at this point the exp3 & its variants and logit dynamics
clc; clear all; close all;
global N g eta delta beta estimator
if exist('ConfigurationVfeedback.mat', 'file') == 2
    load('ConfigurationVfeedback');
    disp('Caution: Loaded ConfigurationVfeedback');
    disp('If you want a new one, delete ConfigurationVfeedback.mat file');
else
    %% point 1 of variable decisions
    N=9; % number of pairs (Tx-Rx)
    a=9;   % available actions (helps to be less than 10, else increase time)
    Pmax=1; % maximum Power available
    gamma=0.3; % gamma variable - minimum connectivity

%% variables of not much important %% keep them as are
lambda=10;
sigmaFactor=1/9;
positions=poissrnd(lambda, [2*N,2]);
for i=2:2:2*N
    positions(i,:)=positions(i-1,:)+normrnd(0, lambda*sigmaFactor, [1,2]);
end
%% determining H matrix
% not that good a model, but, if you think about it
% it doesn't really matter
x=positions;
h=zeros(N,N);
for k=1:N
    for m=1:N
        h(k,m)=1/sqrt( (x(2*k-1,1)-x(2*m,1))^2+ (x(2*k-1,2)-x(2*m,2))^2 );
```

```
32      end
33  end
34  save('ConfigurationVfeedback')
35  end
36  % if ~isempty(positions)
37
38  figure(100)
39  hold on
40  for i=1:N
41      plot(positions((2*i-1):2*i,1), positions((2*i-1):2*i,2)) ;
42      plot(positions(2*i-1,1), positions(2*i-1,2), '+', 'MarkerSize',20) ;
43      plot(positions(2*i,1), positions(2*i,2), 'o', 'MarkerSize',20) ;
44  end
45  drawnow
46  hold off
47  % end
48
49  %% Point 2 of variable decisions (Final)
50  %Modify these variables to fit your purpose
51  %keep in mind that these are independent from the model stored in the file!
52  tau=.1e5; %%% The wisest councelor to all
53  %%% choose estimator wisely
54  estimator=3; % select estimator, 1 vanilla exp3, 2 exp3P, 3 exp3.IX
55  method='logit' % options: exp3, logit (logit uses an estimator too)
56  % in case of feedback %% also, overrides feedback of loaded file!!
57  bits=10
58
59  %% Initialize
60  AvPow=zeros(1,N);
61  AvCon=zeros(1,N);
62  AvThr=zeros(1,N);
63  pr=1/a*ones(a,N);   %probability vector (initially uniform)
64  prT=zeros(a,N,tau); %store probabilities for all T
65  k=zeros(1,N);   %action chosen for each player
66  P=zeros(1,N);   %power level chosen for each player
67  kappa=zeros(tau,3); % Average Power, Average Connectivity, Average Throughput
68  lambda=0;
69  kappa2=kappa;
70  App=zeros(a,N,tau); % Appearences of an action
71  App(:,:,1)=ones(a,N);
72  cumCon=zeros(tau, N);
73  s=zeros(N,1); %%% store rate for each user
74  Pow=linspace(Pmax/20,Pmax,a); % Power vector starting from Pmax/20
75  if (estimator==1 || estimator==3)
76      eta=sqrt(2*log(a)/a/tau); g=sqrt(log(a)/(2*a*tau));
77  else %exp3.P parameters (estimator 2)
78      eta=0.95*sqrt(log(a)/tau/a); delta=1e-2;
79      g=1.05*sqrt(a*log(a)/tau); beta=sqrt(log(a/delta)/tau/a);
80  end
```

```
81 if N==2
82     plotArchitecture(N, gamma, h, Pmax)
83     disp('If happy with this configuration press any key to simulate')
84 pause
85 end
86
87
88
89
90 %% main algorithm
91 for t=2:tau
92     try
93     for i=1:N
94         k(i)=mnrnd(1,pr(:,i),1)*(1:a)';
95         P(i)=Pow(k(i));
96         App(:, i, t)=App(:, i, t-1);
97         App(k(i),i, t)=App(k(i),i, t-1)+1; % cumulative sum of appearences
98         prT(:,i,t)=pr(:,i); % all probability vectors
99     end
100    catch %usually only logit could present some problem with NaN
101        % if so change estimator
102       disp('An error occurred');    disp('Execution will continue.');
103       pr
104       sum(pr,1)
105       break;
106    end
107    Con=0;
108    for j=1:N
109        % average sinr for rate calculcation
110        d=sinr(h(j,:)', P, j);  %insert channel, powers and player
111        s(j)=d+s(j); % towards the average
112        u=utility(d, P(j), Pmax, bits, gamma, j); % self explaining
113
114        if strcmp('exp3', method)
115            if (estimator==1 || estimator==3)
116                pr(:,j)=exp3(a, k(j), u, pr(:,j), j, 1);
117            else
118                pr(:,j)=exp3(a, k(j), u, pr(:,j), j, 0);
119            end
120        % if NO uniform distribution last exp3 arg == 1
121        % else whatever
122        elseif strcmp('logit', method)
123            pr(:,j)=otherFuncs(k(j), u, pr(:,j), 4);
124        end
125
126 %% Update Av Power, Connectivity
127        AvPow(j)=AvPow(j)+P(j);  % actual power
128        AvCon(j)=AvCon(j)+(d>=gamma); % actual connectivity
129        Con=Con+(d>=gamma); % total connectivity
```

```
130             cumCon(t,j)=cumCon(t-1,j)+(d>=gamma);
131             AvThr(j)=AvThr(j)+log(1+d);
132         end
133 %% update Average Power, Average Connectivity
134             lambda=lambda+1;
135             kappa(lambda,:)=[sum(AvPow/t), sum(AvCon/t), sum(AvThr)/t];
136             kappa2(lambda,:)=[sum(P), Con, sum(AvThr)];
137 end
138
139 figure(10)
140 bar(cumCon(end,:)./(tau*ones(1,N)))
141 plotArchitecture(N, gamma, h, Pmax)
142
143 plotArchitec(N, prT, t, a, App);
144 %     errors
145 plotResults(kappa, kappa2, N, t);
146 end
147
148 function[u]=utility(sinr, p, Pmax, n, gamma, i)
149 % using n-bit feedback for the SINR
150 %%% in case one needs a quantized SINR due to limited feedback
151 s=nBitSINR(sinr, n, Pmax);
152
153 %% then, write or choose the utility function
154
155 u=((sinr>=gamma)-p/Pmax+1)/2; %%%% transformed to [0,1)
156
157 % u=(log2(1+s)-log2(1+gamma))/p;
158
159
160
161 end
162
163 function[p]=exp3(K, choice, reward, pr, i, j)
164     global eta estimator N g % K is for arms (actions)
165     persistent weights;
166     if isempty(weights)
167         weights=ones(K,N); %weights
168         wValue=1; %use to initialize weights to wValue
169         weights=wValue*weights;
170     end
171     gamma=g;
172
173     rewards=updateWeights(pr, reward, choice, estimator, K);
174
175     weights(:,i)=weights(:,i).*exp(eta*rewards);
176     if j==1
177         p=weights(:,i)/sum(weights(:,i));
178     else       p=(1-gamma)*weights(:,i)./sum(weights(:,i))+gamma/K;
```

```
179     end
180
181 end
182
183 function[p]=otherFuncs(choice, reward, pr, i)
184 global estimator
185     if i==4 % logit dynamics with estimators 1,3
186         beta=1.5; % beta should be chosen initially
187 %         % need beta, estimator, pr, and payoff
188         d=updateWeights(pr, reward, choice, estimator, length(pr));
189         p=exp(beta* (max(d)-d) ); %regularized to avoid overflow
190         p=p/sum(p);
191     else
192         disp('incomlpete Yet');
193     end
194
195 end
196
197 function[]=plotArchitecture(N, gamma, h, Pmax)
198     if N==2
199         figure(1)
200         P0=gamma./[h(1,1),h(2,2)]   ;
201
202         P1=min([Pmax,                  Pmax;
203             P0(1)*(1+h(1,2)*Pmax)   (Pmax-P0(2))/(P0(2)*h(2,1))]);
204         P2=min([Pmax,                  Pmax;
205             (Pmax-P0(1))/(P0(1)*h(1,2))  P0(2)*(1+h(2,1)*Pmax)]);
206         grid on
207         hold on
208         plot([P0(1),P1(1)],[0,P2(1)],'b')
209         plot([0,P1(2)],[P0(2),P2(2)],'b')
210
211         plot([0,Pmax],[Pmax,Pmax],'k')
212         plot([Pmax,Pmax],[0,Pmax],'k')
213     end
214 end
215
216 function[]=plotArchitec(N, prPrime, tau, a, App)
217     for j=1:N
218         figure(2)
219         subplot(ceil(sqrt(N)) ,ceil(sqrt(N)) ,j)
220         for i=1:a
221             hold all
222             pfds=squeeze(prPrime(i,j,1:tau));
223             plot(1:tau,pfds);
224         end
225         if N<=3
226             legend('show')
227         end
```

```
228     end
229
230     for j=1:N
231         figure(3)
232         subplot(ceil(sqrt(N)) ,ceil(sqrt(N)) ,j)
233         for i=1:a
234             hold all
235             app=squeeze(App(i,j,1:tau)./sum(App(:,j,1:tau)) ); % appearences
236             plot(1:tau,app);
237         end
238         if N<=3
239             legend('show')
240         end
241     end
242 end
243
244 function[]=plotResults(kappa, kappa2, N, tau)
245     figure(4)
246     subplot(2,2,1);
247     [ax1,hLine1,hLine2]=plotyy(1:1:tau,kappa(:,1)/N,1:1:tau,kappa(:,2)/N);
248     hold on
249     plot(1:1:tau,kappa(:,3)/N, 'g');
250     hold off
251     ylabel(ax1(1), 'Average Power');
252     ylabel(ax1(2), 'Average Connectivity');
253     set(ax1(2), 'ylim', [0 1.1]);
254     xlabel('\bf{Time}');
255
256     subplot(2,2,2);
257     [ax2, hLine3, hLine4]=plotyy(1:1:tau,kappa2(:,1)/N, 1:1:tau,kappa2(:,2)/N);
258     ylabel(ax2(1), 'Average Power');
259     ylabel(ax2(2), 'Connectivity');
260     xlabel('\bf{Time}');
261     set(ax2(2), 'ylim', [0 1.1]);
262
263 end
264
265 function[rewards]=updateWeights(pr, reward, choice, rule, K)
266     %choose between update rules / estimators
267     global beta g
268     if rule==1 % exp3
269         rewards=zeros(K,1);
270         rewards(choice)=reward/pr(choice);
271     elseif rule==2 % exp3P
272         rewards=beta./pr;
273         rewards(choice)=rewards(choice)+reward/pr(choice);
274     elseif rule==3 %exp3.IX
275         rewards=zeros(K,1);
276         rewards(choice)=reward/(pr(choice)+g);
```

```
277       elseif rule==4 % implicit normalized forecaster
278           % use only if you insert the appropriate variables
279           rewards=zeros(K,1);
280           rewards(choice)=-log(1-(beta*reward/pr(choice)))/beta;
281       else disp('Wrong rule')
282       end
283   end
284
285   function[s]=sinr(H, P, i)
286       hi=H(i)*P(i);
287       s=hi/(P*H-hi+1);
288   end
289
290   function[s_quant]=nBitSINR(s, n, Pmax)
291
292   % linear quantization of the sinr
293   % for use in case of limited feedback
294   d=1/2^n;
295   x=s/(Pmax);
296   s_quant=Pmax*min(max( (floor(x/d)+1/2)*d, 0), 1);
297   end
```

Bibliography

[1] A. Dytso, "Survey of interference channel,"

[2] O. El Ayach, A. Lozano, and R. Heath, "On the overhead of interference alignment: Training, feedback, and cooperation," *Wireless Communications, IEEE Transactions on*, vol. 11, pp. 4192–4203, November 2012.

[3] Y. Ma, H. Chen, Z. Lin, B. Vucetic, and X. Li, "Spectrum sharing in rf-powered cognitive radio networks using game theory," *arXiv preprint arXiv:1510.02851*, 2015.

[4] B. Wang, Y. Wu, and K. R. Liu, "Game theory for cognitive radio networks: An overview," *Computer networks*, vol. 54, no. 14, pp. 2537–2561, 2010.

[5] N. Nie and C. Comaniciu, "Adaptive channel allocation spectrum etiquette for cognitive radio networks," *Mobile networks and applications*, vol. 11, no. 6, pp. 779–797, 2006.

[6] D. Niyato and E. Hossain, "Competitive spectrum sharing in cognitive radio networks: a dynamic game approach," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 7, pp. 2651–2660, 2008.

[7] J. Liu, R. Deng, S. Zhou, and Z. Niu, "Seeing the unobservable: Channel learning for wireless communication networks," *CoRR*, vol. abs/1508.01899, 2015.

[8] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in hetnets: Old myths and open problems," *Wireless Communications, IEEE*, vol. 21, no. 2, pp. 18–25, 2014.

[9] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "Rat selection games in hetnets," in *INFOCOM, 2013 Proceedings IEEE*, pp. 998–1006, IEEE, 2013.

[10] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 6, pp. 2706–2716, 2013.

[11] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets wifi in wireless small cell networks," *Communications Magazine, IEEE*, vol. 51, no. 6, pp. 44–50, 2013.

[12] F. Meshkati, A. J. Goldsmith, H. V. Poor, and S. C. Schwartz, "A game-theoretic approach to energy-efficient modulation in CDMA networks with delay qos constraints," *CoRR*, vol. abs/0705.1788, 2007.

[13] H. Shirani-Mehr, H. C. Papadopoulos, S. A. Ramprashad, and G. Caire, "Joint scheduling and ARQ for MU-MIMO downlink in the presence of inter-cell interference," *CoRR*, vol. abs/1001.1187, 2010.

[14] L. Song, Z. Han, Z. Zhang, and B. Jiao, "Non-cooperative feedback-rate control game for channel state information in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 1, pp. 188–197, 2012.

[15] A. Antonopoulos, C. Verikoukis, C. Skianis, and O. B. Akan, "Energy efficient network coding-based mac for cooperative arq wireless networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 190–200, 2013.

[16] L. Mao, S. Xu, T. Fu, and Q. Huang, "Game theory based power allocation algorithm in high-speed mobile environment," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, Sept 2012.

[17] L. Badia, M. Levorato, F. Librino, and M. Zorzi, "Cooperation techniques for wireless systems from a networking perspective," *Wireless Communications, IEEE*, vol. 17, no. 2, pp. 89–96, 2010.

[18] Z. Han, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press, 2012.

[19] K. R. Liu and B. Wang, *Cognitive radio networking and security: A game-theoretic view*. Cambridge University Press, 2010.

[20] T. Alpcan, H. Boche, M. L. Honig, and H. V. Poor, *Mechanisms and games for dynamic spectrum allocation*. Cambridge University Press, 2013.

[21] V. Srivastava, J. O. Neel, A. B. MacKenzie, R. Menon, L. A. DaSilva, J. E. Hicks, J. H. Reed, R. P. Gilles, *et al.*, "Using game theory to analyze wireless ad hoc networks.," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1-4, pp. 46–56, 2005.

[22] X. Chen and X. Deng, "Settling the complexity of two-player nash equilibrium.," in *FOCS*, vol. 6, pp. 261–272, 2006.

[23] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[24] A. Rakhlin, "Online bandit problems." Lecture 26.

[25] C. Tekin and M. Liu, "Adaptive learning of uncontrolled restless bandits with logarithmic regret," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp. 983–990, IEEE, 2011.

[26] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[27] F. Diaz, "Integration of news content into web results," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 182–191, ACM, 2009.

[28] K. Hofmann, S. Whiteson, and M. de Rijke, "Contextual bandits for information retrieval," in *NIPS 2011 Workshop on Bayesian Optimization, Experimental Design, and Bandits, Granada*, vol. 12, p. 2011, 2011.

[29] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," *arXiv preprint arXiv:1402.0555*, 2014.

[30] J. Unnikrishnan and V. V. Veeravalli, "Dynamic spectrum access with learning for cognitive radio," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pp. 103–107, IEEE, 2008.

[31] K. Liu and Q. Zhao, "Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 3010–3013, IEEE, 2010.

[32] L. Lai, H. Jiang, and H. V. Poor, "Medium access in cognitive radio networks: A competitive multi-armed bandit framework," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pp. 98–102, IEEE, 2008.

[33] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[34] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part ii: Markovian rewards," *Automatic Control, IEEE Transactions on*, vol. 32, pp. 977–982, Nov 1987.

[35] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: I.i.d. rewards," *Automatic Control, IEEE Transactions on*, vol. 32, pp. 968–976, Nov 1987.

[36] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi, "Bandits with heavy tail," *ArXiv e-prints*, Sept. 2012.

[37] J.-Y. Audibert and S. Bubeck, "Regret bounds and minimax policies under partial monitoring," *The Journal of Machine Learning Research*, vol. 11, pp. 2785–2836, 2010.

[38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[39] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *CoRR*, vol. abs/1204.5721, 2012.

[40] G. Neu, "Explore no more: improved high-probability regret bounds for non-stochastic bandits," *CoRR*, vol. abs/1506.03271, 2015.

[41] J.-y. Audibert and S. Bubeck, "Minimax policies for bandits games," in *COLT*, 2009.

[42] S. Bubeck and A. Slivkins, "The best of both worlds: Stochastic and adversarial bandits," in *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pp. 42.1–42.23, 2012.

[43] J. M. Smith and G. Price, "lhe logic of animal conflict," *Nature*, vol. 246, p. 15, 1973.

[44] J. G. Cross, "A stochastic learning model of economic behavior," *The Quarterly Journal of Economics*, pp. 239–266, 1973.

[45] T. Börgers and R. Sarin, "Learning through reinforcement and replicator dynamics (mimeo)," 1995.

[46] P. Sastry, V. Phansalkar, and M. Thathachar, "Decentralized learning of nash equilibria in multi-person stochastic games with incomplete information," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 5, pp. 769–777, 1994.

[47] D. Ferraioli, "Logit dynamics for strategic games mixing time and metastability," 2012.

[48] D. Ryabko and B. Ryabko, "Predicting the outcomes of every process for which an asymptotically accurate stationary predictor exists is impossible," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, pp. 1204–1206, IEEE, 2015.

[49] C. Alós-Ferrer and N. Netzer, "The logit-response dynamics," *Games and Economic Behavior*, vol. 68, no. 2, pp. 413–427, 2010.

[50] L. E. Blume, "The statistical mechanics of strategic interaction," *Games and economic behavior*, vol. 5, no. 3, pp. 387–424, 1993.

[51] G. Ostrovski and S. van Strien, "Payoff performance of fictitious play," *CoRR*, vol. abs/1308.4049, 2013.

[52] G. W. Brown, "Iterative solution of games by fictitious play," *Activity analysis of production and allocation*, vol. 13, no. 1, pp. 374–376, 1951.

[53] J. S. Jordan, "Three problems in learning mixed-strategy nash equilibria," *Games and Economic Behavior*, vol. 5, no. 3, pp. 368–386, 1993.

[54] L. S. Shapley *et al.*, "Some topics in two-person games," *Advances in game theory*, vol. 52, pp. 1–29, 1964.

[55] B. J. McGill and J. S. Brown, "Evolutionary game theory and adaptive dynamics of continuous traits," *Annual Review of Ecology, Evolution, and Systematics*, pp. 403–435, 2007.

[56] J. Hofbauer and K. Sigmund, "Evolutionary game dynamics," *Bulletin of the American Mathematical Society*, vol. 40, no. 4, pp. 479–519, 2003.

[57] W. H. Sandholm, *Population games and evolutionary dynamics*. MIT press, 2010.

[58] A. B. d. S. Rocha, A. Laruelle, and P. Zuazo Garín, "Replicator dynamics and evolutionary stable strategies in heterogeneous games," 2011.

[59] P. Mertikopoulos and W. H. Sandholm, "Learning in games via reinforcement and regularization,"

[60] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical biosciences*, vol. 40, no. 1, pp. 145–156, 1978.

[61] Y. Sato, E. Akiyama, and J. D. Farmer, "Chaos in learning a simple two-person game," *Proceedings of the National Academy of Sciences*, vol. 99, no. 7, pp. 4748–4751, 2002.

[62] Y.-W. Cheung and D. Friedman, "A comparison of learning and replicator dynamics using experimental data," *Journal of Economic Behavior & Organization*, vol. 35, no. 3, pp. 263 – 280, 1998.

[63] J. Hofbauer, K. Sigmund, *et al.*, "The theory of evolution and dynamical systems: mathematical aspects of selection," tech. rep., 1988.

[64] W. Albers, W. Güth, P. Hammerstein, B. Moldovanu, and E. van Damme, *Understanding Strategic Interaction: Essays in Honor of Reinhard Selten*. Springer Berlin Heidelberg, 2012.

[65] Z. AlSharawi, J. M. Cushing, and S. Elaydi, *Theory and Applications of Difference Equations and Discrete Dynamical Systems: ICDEA, Muscat, Oman, May 26-30, 2013*, vol. 102. Springer, 2014.

[66] W. H. Sandholm, "Local stability under evolutionary game dynamics," *Theoretical Economics*, vol. 5, no. 1, pp. 27–50, 2010.

[67] J. Alboszta, J. Mie, *et al.*, "Stability of evolutionarily stable strategies in discrete replicator dynamics with time delay," *Journal of theoretical biology*, vol. 231, no. 2, pp. 175–179, 2004.

[68] H. Tembine, E. Altman, and R. El-Azouzi, "Delayed evolutionary game dynamics applied to medium access control," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pp. 1–6, IEEE, 2007.

[69] J. Miekisz, R. Jankowski, and M. Matuszak, "Replicator dynamics with strategy dependent time delays," *preprint*, 2014.

[70] J. Alboszta and J. Miekisz, "Stability of evolutionarily stable strategies in discrete replicator dynamics with time delay," *eprint arXiv:q-bio/0409024*, Sept. 2004.

[71] J. P. Rabanal and D. Friedman, "Incomplete information, dynamic stability and the evolution of preferences: Two examples," *Dynamic Games and Applications*, vol. 4, no. 4, pp. 448–467, 2014.

[72] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[73] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[74] S. Grammatico, F. Parise, M. Colombino, and J. Lygeros, "Decentralized convergence to nash equilibria in constrained mean field control," *CoRR*, vol. abs/1410.4421, 2014.

[75] G. Christodoulou, V. S. Mirrokni, and A. Sidiropoulos, "Convergence and approximation in potential games," in *STACS 2006*, pp. 349–360, Springer, 2006.

[76] S. Lasaulce, M. Debbah, and E. Altman, "Methodologies for analyzing equilibria in wireless games," *arXiv preprint arXiv:0906.0447*, 2009.

[77] G. Scutari, S. Barbarossa, and D. P. Palomar, "Potential games: A framework for vector power control problems with coupled constraints," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 4, pp. IV–IV, IEEE, 2006.

[78] K. Yamamoto, "A comprehensive survey of potential game approaches to wireless networks," *CoRR*, vol. abs/1506.07942, 2015.

[79] H.-S. Yu, H. Oh, H. Lee, G.-K. Choi, Y. Lim, and Y. Moon, "Apparatus and method for transmitting/receiving channel quality information of subcarriers in an orthogonal frequency division multiplexing system," Nov. 19 2004. US Patent App. 10/992,110.

[80] S. Sesia, I. Toufik, and M. Baker, *LTE: the UMTS long term evolution*. Wiley Online Library, 2009.

[81] F. Khan, "Methods of transmitting channel quality information and power allocation in wireless communication systems," Mar. 14 2003. US Patent App. 10/387,866.

[82] D. J. Love, R. W. Heath Jr, V. K. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 8, pp. 1341–1365, 2008.

[83] B. G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[84] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1404–1422, 2012.

[85] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pp. 1–9, IEEE, 2010.

[86] A. Mukherjee and A. Hottinen, "Learning algorithms for energy-efficient mimo antenna subset selection: Multi-armed bandit framework," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 659–663, IEEE, 2012.

[87] X.-Y. Li, "Push the limit of wireless network capacity: a tale of cognitive and coexistence," in *Proceedings of the 1st ACM workshop on Cognitive radio architectures for broadband*, pp. 31–32, ACM, 2013.

[88] N. Gulati and K. R. Dandekar, "Learning state selection for reconfigurable antennas: A multi-armed bandit approach," *Antennas and Propagation, IEEE Transactions on*, vol. 62, no. 3, pp. 1027–1038, 2014.

[89] Z. Liu, Y. Wang, and C. Liang, "A game theory based power control algorithm in wireless sensor network," *Journal of Convergence Information Technology*, vol. 8, no. 9, p. 1119, 2013.

[90] F. Meshkati, M. Chiang, H. V. Poor, and S. C. Schwartz, "A game-theoretic approach to energy-efficient power control in multicarrier cdma systems," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 6, pp. 1115–1129, 2006.

[91] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[92] R. J. Aumann, "Subjectivity and correlation in randomized strategies," *Journal of mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.

[93] R. J. Aumann, "Correlated equilibrium as an expression of bayesian rationality," *Econometrica: Journal of the Econometric Society*, pp. 1–18, 1987.