Collaborative Spatial Reuse in Wireless Networks via Selfish Multi-Armed Bandits

Francesc Wilhelmi✉, Cristina Cano, Gergely Neu, Boris Bellalta, Anders Jonsson and Sergio Barrachina-Muñoz

Abstract—Next-generation wireless deployments are characterized by being dense and uncoordinated, which often leads to inefficient use of resources and poor performance. To solve this, we envision the utilization of completely decentralized mechanisms to enable Spatial Reuse (SR). In particular, we concentrate in Reinforcement Learning (RL), and more specifically, in Multi-Armed Bandits (MABs), to allow networks to modify both their transmission power and channel based on their experienced throughput. In this work, we study the exploration-exploitation trade-off by means of the ε-greedy, EXP3, UCB and Thompson sampling action-selection strategies. Our results show that optimal proportional fairness can be achieved, even if no information about neighboring networks is available to the learners and WNs operate selfishly. However, there is high temporal variability in the throughput experienced by the individual networks, specially for ε-greedy and EXP3. We identify the cause of this variability to be the adversarial setting of our setup in which the set of most played actions provide intermittent good/poor performance depending on the neighboring decisions. We also show that this variability is reduced using UCB and Thompson sampling, which are parameter-free policies that perform exploration according to the reward distribution of each action.

Index Terms—High-Density Wireless Networks, Interference, Decentralized learning, Multi Armed Bandits, Resource Allocation.

1 Introduction

Due to the growing popularity of wireless deployments, especially the ones based on the IEEE 802.11 standard (i.e., Wi-Fi), it is very common to find independent overlapping Wireless Networks (WNs) sharing the same channel resources. Due to the decentralized nature of such kind of deployments, the lack of organization and/or agreement on sharing policies leads to inefficient resources allocation, both in spectral and spatial domains. An illustrative example about the inefficiency noticed in WNs regarding channel sharing can be found in [1], where the authors show that the power level used by wireless devices is typically set, by default, to the maximum, regardless of the distance between communicating nodes, and of the channel occupancy [1]. In decentralized environments, providing a model to obtain the optimal solution can be extremely complex. In WNs, the interactions between devices depend on many features (such as position, environment, transmit power) and are hard to derive. On top of this, network dynamics, in terms of user arrivals and departures, makes the resource allocation problem even harder.

In order to address the SR problem in decentralized WNs, we focus attention on Reinforcement Learning (RL), which has recently become very popular to solve many well-known problems in wireless communications. Some examples can be found for packet routing [2], Access Point (AP) selection [3], [4], optimal rate sampling [5], or energy harvesting in heterogeneous networks [6]. Using online learning, a learner (or agent) obtains data in a sequential order and uses it to predict future good-performing actions. Online learning is, therefore, useful to cope with complex and dynamic environments. This background encourages us to approach a solution for the decentralized SR problem in WNs through online learning techniques. In particular, we exploit dynamic channel selection and transmit power adjustment for dealing with the performance issues resulting from uncoordinated resource sharing in WNs. While a proper frequency planning allows to reduce the interference between wireless devices, tuning the transmit power adds an extra level of SR that can result in improved throughput and fairness.

From the family of online algorithms, we propose the utilization of Multi-Armed Bandits (MABs) [7], which is a well-known model in the online learning literature for solving resource allocation problems. In MABs, a given agent seeks to learn a hidden reward distribution while maximizing the gains. This is known as the exploration-exploitation trade-off. While exploitation selects actions with the goal of maximizing long-term reward given the current estimate, exploration selects actions with the goal of improving the estimate. Unlike for classical RL, MABs do not consider states[1] in general, which can be hard define for the decentralized SR problem. On the one hand, spatial interference cannot be binary treated, thus leading to complex interactions between nodes. On the other hand, the adversarial setting unleashed by decentralized deployments increases the , since the state not only depends on the actions taken by a given node, but also on the adversaries behavior.

This work extends our previous results presented in [8], which were obtained by applying a stateless variation of Q-learning to the SR problem in WNs. Here we generalize the contributions done by implementing several action-selection strategies to allow WNs choose the channel and transmit power to use based on the resulting performance. In this paper, then, we provide a tutorial-based applica-

1. A state refers to a particular situation experienced by a given agent, which is defined by a set of conditions. By having an accurate knowledge on its current situation, an agent can define state-specific strategies that maximize its profits.

tion of MABs to the decentralized SR problem, in order to maximize performance while observing the result from using different SR settings. According to that, we aim to illustrate the main benefits and drawbacks of applying different well-known learning techniques in a selfish way, i.e., independent WNs base their strategy on their own experienced performance. On the one hand, we evaluate the impact of varying parameters intrinsic to the proposed algorithms on the resulting throughput and fairness. In addition, we analyze the effects of learning selfishly, in order to shed some light on the future of decentralized approaches. Notably, we observe that even though players act selfishly, there are algorithms learn to play actions that enhance the overall performance, some times at the cost of high temporal variability. Considering selfish WNs and still obtaining collaborative behaviors is appealing to typical chaotic and dynamic deployments as we get rid of the need for message passing, or inference of neighboring conditions.

The main contributions of this work are summarized below:

- We devise the feasibility of applying MAB algorithms as defined in the online learning literature to solve the resource allocation problem in WNs.
- We study the impact of different parameters intrinsic to the action-selection strategies considered (e.g., exploration coefficients, learning rates) on network performance.
- We show that there are algorithms learn to play collaborative actions even though the WNs act selfishly, which is appealing to practical application in chaotic and dynamic environments. In addition, we shed some light on the root causes of this phenomena.

The remaining of this document is structured as follows: Section 2 outlines relevant related work. Section 3 introduces the proposed learning algorithms and their practical implementation for the resource allocation problem in WNs. Then, Section 4 presents the simulation scenarios and the considerations taken into account. The simulation results are later presented in Section 5. Finally, Section 6 provides some final remarks.

2 Related Work

In this paper we approach the decentralized SR problem through Dynamic Channel Allocation (DCA) and Transmit Power Control (TPC). While the former aims to allocate the available frequency range among potentially overlapping WNs, the latter attempts to increase the number of parallel transmissions by adjusting the power transmitted by each WN. While DCA operate at the frequency level, TPC deals with more complex spatial interactions. For that, an approach that dynamically tunes the transmit power may severely affect higher communication layers such as network (routing) or transport (congestion). TPC directly affects the transmission range, the data rate and the energy consumption, and it may create unidirectional links that unleash fairness issues. In fact, acknowledgments (ACKs) and RTS/CTS frames assume bidirectionality.

DCA has been extensively exploited from the centralized perspective, especially through techniques based on graph coloring [9], [10]. Despite these kind of approaches allow to effectively reduce the interference between WNs, a certain degree of communication is required. Regarding decentralized methods, the authors in [11] propose a very simple approach in which each AP maintains an interference map of their neighbors, so that channel assignment is done through interference minimization. Unfortunately, the interactions between APs due to decentralization are not studied. Separately, [12] proposes two decentralized approaches that rely in the interference measured at both APs and STAs to calculate the best frequency channels for dynamic channel allocation. To do so, a WN, in addition to the interference sensed by its associated devices, considers other metrics such as the amount of traffic, so that some kind of coordination is required at the neighbor level (periodic reporting).

For the case of TPC, one of the first works in studying power management is [13], which shows that large improvements can be achieved if selecting the appropriate transmit power level in ad-hoc WNs. A centralized approach for TPC is presented in [14], which performs power control and Rate Adaptation (RA) in subgroups of Wireless Local Area Networks (WLANs). The creation of clusters allows defining independent power levels between devices in the same group, which are useful to avoid asymmetric links. However, to represent all the possible combinations, graphs can become very large, specially in high-density deployments. Furthermore, we find a decentralized approach that relies on real-time channel measurements [15]. The proposed mechanism (so called Dynamic Transmission Power Control) is based on a set of triggered thresholds that increase/decrease the transmit power according to the situation. The main problem is that thresholds are set empirically (based on simulations), so the mechanism may not always improve performance.

2.1 Solving WNs Resource Allocation Problems via Bandits

In the online learning literature, several MAB settings have been considered such as stochastic bandits [16], [17], [18], adversarial bandits [19], [20], restless bandits [21], contextual bandits [22] and linear bandits [23], [24], and numerous exploration-exploitation strategies have been proposed such as *ε-greedy* [18], [25], *upper confidence bound* (UCB) [17], [18], [26], [27], *exponential weight algorithm for exploration and exploitation* (EXP3) [18], [19] and *Thompson sampling* [16]. The classical multi-armed bandit problem models a sequential interaction scheme between a learner and an environment. The learner sequentially selects one out of K actions (often called *arms* in this context) and earns some rewards determined by the chosen action and also influenced by the environment. Formally, the problem is defined as a repeated game where the following steps are repeated in each round $t = 1, 2, \ldots, T$:

1) The environment fixes an assignment of rewards $r_{a,t}$ for each action $a \in [K] \stackrel{\text{def}}{=} \{1, 2, \ldots, K\}$,
2) the learner chooses action $a_t \in [K]$,
3) the learner obtains and observes reward $r_{a_t,t}$.

The bandit literature largely focuses on the perspective of the learner with the objective of coming up with learning

algorithms that attempt to maximize the sum of the rewards gathered during the whole procedure (either with finite or infinite horizon). As noted above, this problem has been studied under various assumptions made on the environment and the structure of the arms. The most important basic cases are the *stochastic* bandit problem where, for each particular arm a, the rewards are i.i.d. realizations of random variables from a fixed (but unknown) distribution ν_a, and the *non-stochastic* (or *adversarial*) bandit problem where the rewards are chosen arbitrarily by the environment. In both cases, the main challenge for the learner is the *partial observability* of the rewards: the learner only gets to observe the reward associated with the chosen action a_t, but never observes the rewards realized for the other actions.

Let $\mathrm{r}_{a^*,t}$ and $\mathrm{r}_{a,t}$ be the rewards obtained at time t from choosing actions a^* (optimal) and a, respectively. Then, the performance of learning algorithms is typically measured by the *total expected regret* defined as

$$R_T = \sum_{t=0}^{T} \mathbb{E}\left[(\mathrm{r}_{a^*,t} - \mathrm{r}_{a,t})\right].$$

An algorithm is said to *learn* if it guarantees that the regret grows sublinearly in T, that is, if $R_T = o(T)$ is guaranteed as T grows large, or, equivalently, that the average regret R_T/T converges to zero. Intuitively, sublinear regret means that the learner eventually identifies the action with the highest long-term payoff. Note, as well, that the optimal action a^* is the same across all the rounds. Most bandit algorithms come with some sort of a guaranteed upper bound on R_T which allows for a principled comparison between various methods.

To the best of our knowledge, there is very little related work on applying multi-armed bandit techniques to the problem of resource allocation in WNs. In [28], the authors propose modeling a resource allocation problem in LTE networks through MABs. In particular, a set of Base Stations (BS) learn the best configuration of Resource Blocks (RBs) in a decentralized way. For that purpose, a variation of EXP3 (so-called Q-EXP3) is proposed, which allows to reduce the strategy set. Despite a regret bound is provided, it is subject to the fact that a perfect resource allocation exists. In addition, a large number of iterations is required to find the optimal solution in a relatively small scenario, thus revealing the difficulties shown by decentralized settings.

With a higher degree of relation to the problem proposed here, the authors in [29] show a channel selection and power control approach in infrastructureless networks, which is modeled through bandits. In particular, two different strategies are provided to improve the performance of two Device to Device (D2D) users (each one composed by a transmitter and a receiver), which must learn the best channel and transmit power to be selected. Similarly to our problem, users do not have any knowledge on the channel or the other's configuration, so they rely on the experienced performance in order to find the best configuration. An extension of [29] is provided by the same authors in [30], which includes a calibrated predictor (referred in the work as *forecaster*) to infer the behavior of the other devices in order to counter act their actions. In each agent, the information of the forecaster is used to choose the highest-rewarding action with a certain probability, while the rest of actions are randomly selected. Henceforth, assuming that all the networks use a given strategy $\mathcal{X}$, fast convergence is ensured. Results show that channel resources are optimally distributed in a very short time frame through a fully decentralized algorithm that does not require any kind of coordination. Both aforementioned works rely in the existence of a unique Nash Equilibrium, which favors convergence. In contrast, in this article we aim to extend Bandits utilization to denser deployments, and, what is more important, to scenarios with limited available resources in which there is not a unique Nash Equilibrium (NE) that allows fast-convergence. Thus, we aim to capture the effects of applying selfish strategies in a decentralized way (i.e., agent i follows a strategy $\mathcal{X}_i$ that does not consider the strategies of the others) and we also provide insight about the importance of past information for learning in dense WNs, which has not been studied before.

3 Multi-Armed Bandits for Improving Spatial Reuse in WNs

The decentralized resource allocation problem in WNs we aim to solve in this work can be modeled as an approximation of contextual bandits. Originally, the contextual bandits problem considers the existence of a context (or a state) that influences the action-selection process. As a consequence, the set of available strategies varies with the context and the probability distribution of a given reward depends both on the chosen actions and also on the context. In the resource allocation problem at hand, the context could include information such as the number of neighboring WNs, their position, their current channel selection, the transmission power they use as well as their action-selection strategies, among other. However, we are interested in the case where no context can be inferred given that such information is hard and costly to obtain in practice, especially in dynamic scenarios. Thus, we model the decentralized SR problem in WNs as an adversarial bandit problem, in which the actions of a given WN affect the pay-off distributions of the others' actions. Such model requires the existence of a NE with a pure strategy,[2] which does not always hold for unplanned deployments due to the scarcity of the available resources and the number of overlapping WNs. Understanding the implications derived from such an adversarial setting in the absence of a NE is one the main goals of this paper, which, to the best of our knowledge, has been barely considered in the previous literature.

We model this adversarial problem as follows. Let arm $a \in \mathcal{A}$ (we denote the size of $\mathcal{A}$ with K) be a configuration that a WN may choose. Each configuration is a combination of channel and transmit power (e.g., a_1 = {Channel: 1, TPC: 5 dBm}), and grants a reward that depends on the others' configuration. Let $\Gamma_{i,t}$ be the throughput experienced by WN_i at time t, and Γ_i^* the maximum achievable throughput by WN_i in case of isolation (i.e., no interference is experienced in the selected channel). We then define the reward $r_{i,t}$ experienced by WN_i at time t as:

2. A pure strategy NE is conformed by a set of strategies and payoffs, so that no player can obtain further benefits by deviating from its strategy.

$$r_{i,t} = \frac{\Gamma_{i,t}}{\Gamma_i^*} \leq 1,$$

For simplicity, we consider that each WN is composed by an AP and a single STA. Note that in typical uncoordinated wireless deployments (e.g., residential buildings), STAs are typically close to the AP to which they are associated. Thus, having several STAs associated to the same AP does not significant impact to the inter-WNs interference that is aimed to be studied in this work. According to that, Γ_i^* is computed as $B \cdot \log_2(1+\text{SNR}_i)$, where B is the bandwidth, and SNR_i is the Signal-to-Noise ratio experienced by the STA of WN_i. We consider this model because it allows to properly capture the interference generated between WNs, which is sought to be minimized.

We have considered the *ε-greedy*, *EXP3*, *UCB* and *Thompson sampling* action-selection strategies, which are described next in this section. While ε-greedy and EXP3 explicitly include the concepts of *exploration coefficient* and *learning rate*, respectively, UCB and Thompson sampling are parameter-free policies that extend the concept of exploration (actions are explored according to their estimated value and not by commitment). Henceforth, we are able to study the effects of these different kind of policies on the WNs behavior. Moreover, we chose the aforementioned policies because they are widely spread and considered of remarkable importance in the MABs literature.

Furthermore, in order to provide a fair comparison, we have considered that all the policies use the same reward. Regarding the learning process, we assume that WNs select actions during the same time slot. The reward experienced by the WNs is computed once all of them have chosen their action for the current iteration. In practice the action-selection process at WNs will most probably not be synchronized but we leave the study of any possible effects of that desynchronization to future work.

3.1 ε-greedy

The *ε-greedy* policy [18], [25] is arguably the simplest learning algorithm attempting to deal with exploration-exploitation trade-offs. In each round t, the ε-greedy algorithm explicitly decides whether to explore or exploit: with probability ε, the algorithm picks an arm uniformly at random (exploration), and otherwise it plays the arm with the highest empirical return $\hat{r}_{k,t}$ (exploitation).

In case ε is fixed for the entire process, the expected regret is obviously going to grow linearly as $\Omega(\varepsilon T)$ in general. Therefore, in order to obtain a sublinear regret guarantee (and thus an asymptotically optimal growth rate for the total rewards), it is critical to properly adjust the exploration coefficient. Thus, in our implementation of the ε-greedy algorithm, we use a time-dependent exploration rate of $\varepsilon_t = \varepsilon_0/\sqrt{t}$, as suggested in the literature [18]. The adaptation of this policy to our setting is shown as Algorithm 1.

Algorithm 1 Implementation of Multi-Armed Bandits (ε-greedy) in a WN. $\mathcal{U}(1,K)$ is a uniform distribution that randomly chooses from 1 to K.

Input: SNR: information about the Signal-to-Noise Ratio received at the STA, $\mathcal{A}$: set of possible actions in $\{a_1, ..., a_K\}$
Initialize: $t = 0$, $\varepsilon_t = \varepsilon_0$, $r_k = 0, \forall a_k \in \mathcal{A}$
while *active* **do**
  Select a_k $\begin{cases} \underset{k=1,...,K}{\text{argmax}}\, r_{k,t}, & \text{with prob. } 1-\varepsilon \\ k \sim \mathcal{U}(1,K), & \text{otherwise} \end{cases}$
  Observe the throughput experienced Γ_t
  Compute the reward $r_{k,t} = \frac{\Gamma_t}{\Gamma^*}$, where $\Gamma^* = B\log_2(1+\text{SNR})$
  $\varepsilon_t \leftarrow \varepsilon_0/\sqrt{t}$
  $t \leftarrow t+1$
end

3.2 EXP3

The EXP3 algorithm [19], [20] is an adaptation of the weighted majority algorithm of [31], [32] to the non-stochastic bandit problem. EXP3 maintains a set of non-negative weights assigned to each arm and picks the actions randomly with a probability proportional to their respective weights (initialized to 1 for all arms). The aim of EXP3 is to provide higher weights to the best actions as the learning procedure proceeds.

More formally, letting $w_{k,t}$ be the weight of arm k at time $t \in \{1, 2...\}$, EXP3 computes the probability $p_{k,t}$ of choosing arm k in round t as

$$p_{k,t} = (1-\gamma)\frac{w_{k,t}}{\sum_{i=1}^{\text{K}} w_{i,t}} + \frac{\gamma}{K},$$

where $\gamma \in [0,1]$ is a parameter controlling the rate of exploration. Having selected arm a_t, the learner observes the generated pay-off $r_{a_t,t}$ and computes the importance-weighted reward estimates

$$\widehat{r}_{k,t} = \frac{\mathbb{I}_{\{I_t=k\}} r_{k,t}}{p_{k,t}}$$

for all $k \in [K]$, where $\mathbb{I}_{\{A\}}$ denoting the indicator function of the event A taking a value of 1 if A is true and 0 otherwise. Finally, the weight of arm k is updated as a function of the estimated reward:

$$w_{k,t+1} = w_{k,t} e^{\frac{\eta \cdot \hat{r}_{k,t}}{K}},$$

where $\eta > 0$ is a parameter of the algorithm often called the *learning rate*. Intuitively, η regulates the rate in which the algorithm incorporates new observations with large values of η corresponding to more confident updates and smaller values leading to more conservative behavior. As we did for the exploration coefficient in *ε-greedy*, we use a time-dependent learning rate of $\eta_t = \eta_0/\sqrt{t}$ [18]. Our implementation of EXP3 to the WN problem is detailed in Algorithm 2.

Algorithm 2 Implementation of Multi-Armed Bandits (EXP3) in a WN

Input: SNR: information about the Signal-to-Noise Ratio received at the STA, $\mathcal{A}$: set of possible actions in $\{a_1, ..., a_K\}$

Initialize: $t = 0$, $\eta_t = \eta_0$, $w_{k,t} = 0, \forall a_k \in \mathcal{A}$

while *active* **do**

$p_{k,t} \leftarrow (1-\gamma)\frac{w_{k,t}}{\sum_{i=1}^{K} w_{i,t}} + \frac{\gamma}{K}$

Draw $a_k \sim p_{k,t} = (p_{1,t}, p_{2,t}, ..., p_{K,t})$

Observe the throughput experienced Γ_t

Compute the reward $r_{k,t} = \frac{\Gamma_t}{\Gamma^*}$, where $\Gamma^* = B\log_2(1+\text{SNR})$

$\widehat{r}_{k,t} \leftarrow \frac{r_{k,t}}{p_{k,t}}$

$w_{k,t} \leftarrow w_{k,t-1}^{\frac{\eta_t}{\eta_{t-1}}} \cdot e^{\eta_t \cdot \widehat{r}_{k,t}}$

$w_{t,k'} \leftarrow w_{t-1,k'}^{\eta_t/\eta_{t-1}}, \forall k' \neq k$

$\eta_t \leftarrow \frac{\eta_0}{\sqrt{t}}$

$t \leftarrow t+1$

end

3.3 UCB

The *upper confidence bound* (UCB) action-selection strategy [18], [26], [27] is based on the principle of *optimism in face of uncertainty*: in each round, UCB selects the arm with the highest statistically feasible mean reward given the past observations. Statistical feasibility here is represented by an upper confidence bound on the mean rewards which shrinks around the empirical rewards as the number of observations increases. Intuitively, UCB trades off exploration and exploitation very effectively, as upon every time a suboptimal arm is chosen, the corresponding confidence bound will shrink significantly, thus quickly decreasing the probability of drawing this arm in the future. The width of the confidence intervals is chosen carefully so that the true best arm never gets discarded accidentally by the algorithm, yet suboptimal arms are not drawn as few times as possible. To obtain the first estimates, each arm is played once at the initialization.

Formally, let n_k be the number of times that arm k has been played, and $\Gamma_{k,t}$ the throughput obtained by playing arm k at time t. The average experienced reward $\overline{r}_{k,t}$ by arm k at time t is therefore given by:

$$\overline{r}_{k,t} = \frac{1}{n_k}\sum_{s=1}^{n_k} r_{k,s}$$

Based on these average rewards, UCB selects the action that maximizes $\overline{r}_{k,t} + \sqrt{\frac{2\ln(t)}{n_k}}$. By doing so, UCB implicitly balances exploration and exploitation, as it focuses efforts on the arms that are $i)$ the most promising (with large estimated rewards) or $ii)$ not explored enough (with small n_k). Our implementation of UCB to the WN problem is detailed in Algorithm 3.

3.4 Thompson sampling

Thompson sampling [16] is a well-studied action-selection technique that had been known for its excellent empirical performance [33] and was recently proven to achieve strong performance guarantees, often better than those warranted by UCB [34], [35], [36]. Thompson sampling is a Bayesian algorithm: it constructs a probabilistic model of the rewards and assumes a prior distribution of the parameters of said model. Given the data collected during the learning procedure, this policy keeps track of the posterior distribution of the rewards, and pulls arms randomly in a way that the drawing probability of each arm matches the probability of the particular arm being optimal. In practice, this is implemented by sampling the parameter corresponding to each arm from the posterior distribution, and pulling the arm yielding the maximal expected reward under the sampled parameter value.

For the sake of practicality, we aim to apply Thompson sampling using a Gaussian model for the rewards with a standard Gaussian prior as suggested in [37]. By standard calculations, it can be verified that the posterior distribution of the rewards under this model is Gaussian with mean

$$\hat{r}_k(t) = \frac{\sum_{w=1:k}^{t-1} r_k(t)}{n_k(t)+1}$$

and variance $\sigma_k^2(t) = \frac{1}{n_k+1}$, where n_k is the number of times that arm k was drawn until the beginning of round t. Thus, implementing Thompson sampling in this model amounts to sampling a parameter θ_k from the Gaussian distribution $\mathcal{N}\left(\hat{r}_k(t), \sigma_k^2(t)\right)$ and choosing the action with the maximal parameter. Our implementation of Thompson sampling to the WN problem is detailed in Algorithm 4.

Algorithm 3 Implementation of Multi-Armed Bandits (UCB) in a WN

Input: SNR: information about the Signal-to-Noise Ratio received at the STA, $\mathcal{A}$: set of possible actions in $\{a_1, ..., a_K\}$

Initialize: $t = 0$, play each arm $a_k \in \mathcal{A}$ once

while *active* **do**

Draw $a_k = \underset{k=1,...,K}{\text{argmax}}\ \overline{r}_k + \sqrt{\frac{2ln(t)}{n_k}}$

Observe the throughput experienced Γ_t

Compute the reward $r_{k,t} = \frac{\Gamma_t}{\Gamma^*}$, where $\Gamma^* = B\log_2(1+\text{SNR})$

$n_k \leftarrow n_k + 1$

$\overline{r}_k \leftarrow \frac{1}{n_k}\sum_{s=1}^{n_k} r_{k,s}$

$t \leftarrow t+1$

end

4 System model

For the remainder of this work, we consider several WNs placed in a 3-D scenario in order to study the effects of applying MABs in a decentralized manner (with parameters described later in Section 4.3). For simplicity, we consider WNs to be composed by an Access Point (AP) transmitting to a single Station (STA) in a downlink manner. We consider as well that the number of channels is equal to half the number of coexisting WNs, so that SR is challenging.

4.1 Channel modeling

Path-loss and shadowing effects are modeled using the log-distance model for indoor communications. The path-loss

Algorithm 4 Implementation of Multi-Armed Bandits (Thompson sampling) in a WN

Input: SNR: information about the Signal-to-Noise Ratio received at the STA, $\mathcal{A}$: set of possible actions in $\{a_1, ..., a_K\}$

Initialize: $t = 0$, for each arm $a_k \in \mathcal{A}$, set $\hat{r}_k = 0$ and $n_k = 0$

while *active* **do**

For each arm $a_k \in \mathcal{A}$, sample $\theta_k(t)$ from normal distribution $\mathcal{N}(\hat{r}_k, \frac{1}{n_k+1})$

Play arm $a_k = \underset{k=1,...,K}{\operatorname{argmax}}\ \theta_k(t)$

Observe the throughput experienced Γ_t

Compute the reward $r_{k,t} = \frac{\Gamma_t}{\Gamma^*}$, where $\Gamma^* = B\log_2(1+\text{SNR})$

$\hat{r}_{k,t} \leftarrow \frac{\hat{r}_{k,t} n_{k,t} + r_{k,t}}{n_{k,t}+2}$

$n_{k,t} \leftarrow n_{k,t} + 1$

$t \leftarrow t + 1$

end

between WN i and WN j is given by:

$$\mathrm{PL}_{i,j} = \mathrm{P}_{\mathrm{tx},i} - \mathrm{P}_{\mathrm{rx},j} = \mathrm{PL}_0 + 10\alpha\log_{10}(d_{i,j}) + \mathrm{G_s} + \frac{d_{i,j}}{d_{\mathrm{obs}}}\mathrm{G_o},$$

where $\mathrm{P}_{\mathrm{tx},i}$ is the transmitted power in dBm by the AP in WN_i, α is the path-loss exponent, $\mathrm{P}_{\mathrm{rx},j}$ is the power in dBm received at the STA in WN_j, PL_0 is the path-loss at one meter in dB, $d_{i,j}$ is the distance between the transmitter and the receiver in meters, $\mathrm{G_s}$ is the log-normal shadowing loss in dB, and $\mathrm{G_o}$ is the obstacles loss in dB. Note that we include the factor d_{obs}, which is the average distance between two obstacles in meters.

4.2 Throughput calculation

As previously mentioned, we use the power received and the interference to calculate the maximum theoretical throughput of each WN i at time t by using the Shannon Capacity,

$$\Gamma_{i,t} = B\log_2(1 + \mathrm{SINR}_{i,t}),$$

where B is the channel width and the experienced Signal to Interference plus Noise Ratio (SINR) is given by:

$$\mathrm{SINR}_{i,t} = \frac{\mathrm{P}_{i,t}}{\mathrm{I}_{i,t} + \mathrm{N}},$$

where $\mathrm{P}_{i,t}$ and $\mathrm{I}_{i,t}$ are the received power and the sum of the interference at WN i at time t, respectively, and N is the floor noise power. For each WN, we consider the interference to be the total power received from all the APs in the same channel. For capturing the worst-case scenario, we consider that all WNs are continuously transmitting (i.e., we do not consider empty channel periods left by the medium access procedure). Adjacent channel interference is also considered in $\mathrm{I}_{i,t}$, so that the transmitted power leaked to adjacent channels is 20 dBm lower for each extra channel separation.

4.3 Simulation Parameters

According to [38], which provides an overview of the IEEE 802.11ax-2019 standard, a typical high-density scenario for residential buildings contains $0.0033\mathrm{APs/m}^3$ (i.e., 100 APs in a $100 \times 20 \times 15$ m area). Accordingly, for simulation purposes, we define a map scenario with dimensions $10 \times 5 \times 10$ m, containing from 2 to 8 APs. In addition, for the first part of the simulations, we consider a setting containing 4 WNs that form a grid topology. In it, STAs are placed at the maximum possible distance from the other networks. The 4-grid scenario is illustrated in Figure 1. Table 1 details the parameters used.

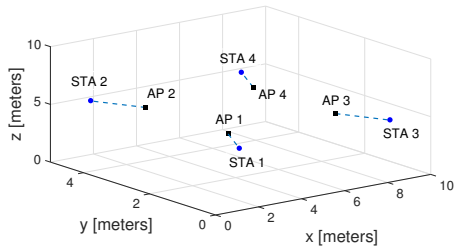


Fig. 1: Grid scenario containing 4 WNs, each one composed by an AP and a STA.

Parameter	Value
Map size (m)	$10 \times 5 \times 10$
Number of coexistent WNs	{2, 4, 6, 8}
APs/STAs per WN	1 / 1
Distance AP-STA (m)	$\sqrt{2}$
Number of Channels	2
Channel Bandwidth (MHz)	20
Initial channel selection model	Uniformly distributed
TPC Values (dBm)	{5, 10, 15, 20}
PL_0 (dB)	5
G_s (dB)	Normally distributed with mean 9.5
G_o (dB)	Uniformly distributed with mean 30
d_{obs} (meters between two obstacles)	5
Noise level (dBm)	-100
Traffic model	Full buffer (downlink)
Number of learning iterations	10,000

TABLE 1: Simulation parameters

5 Performance Evaluation

In this Section, we evaluate the performance of each action-selection strategy presented in Section 3 when applied to the resource allocation problem in WNs.[3] For that purpose, we first evaluate the ε-greedy, EXP3, UCB and Thompson sampling policies in a controlled environment (i.e., the scenario illustrated in Figure 1). Without loss of generality, we consider a symmetric configuration in this scenario in order to simplify evaluation. Then, in Section 5.2, we provide a performance comparison of the aforementioned scenarios with different densities and with randomly located WNs.

5.1 Toy Grid Scenario

5.1.1 *Optimal Solution*

Before presenting the simulation results of applying a particular action-selection policy, we introduce the optimal solutions that: *a)* maximize the aggregate throughput and

3. All of the source code used in this work open [39], encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others under the GNU General Public License v3.0

b) correspond to proportional fairness for the considered grid scenario, shown in Figure 1. The proportional fairness solutions satisfy $\max_{a_k \in \mathcal{A}} \sum_{i \in \mathcal{WN}} \log(\Gamma_{i,a_k})$.

Let actions range from a_1 to a_8, and map to {channel number, transmit power (dBm)}: a_1 = {1, 5}, a_2 = {2, 5}, a_3 = {1, 10}, a_4 = {2, 10}, a_5 = {1, 15}, a_6 = {2, 15}, a_7 = {1, 20} and a_8 = {2, 20}. According to our model, the action selection of the 4 WNs that grants the maximum aggregate throughput is any consecutive combination of {a_1, a_1, a_7, a_8}. Despite the scenario is symmetric, the distance between WNs is not the same in all the cases. Henceforth, the optimal solution in terms of aggregate throughput requires that WNs using the maximum transmit power are located as closely as possible. In the case of proportional fairness, the optimal selection changes to any combination of {a_7, a_8} in which WNs using the same channel are located each other at the farthest possible distance.

The optimal aggregate throughput when throughput is maximized is 1124 Mbps. This is achieved when two of the WNs sacrifice themselves by choosing a lower transmit power (action a_1). On the other hand, when considering proportional fairness, the best configuration grants an aggregate throughput of 891 Mbps, which is quite lower than in the former case, but allows experiencing a fairer configuration. Note that all WNs obtain the same throughput in this case as frequency reuse is achieved (closer WNs choose different channels), but power is kept equal for each WN and despite experiencing higher interference, none of the WNs sacrifice themselves. The aim of the following performance evaluation is to gain insight into whether a collaborative result, close to this equal division of resources given by the proportional fair action selection, is likely to be learned by the algorithms and if so, under which conditions.

5.1.2 Performance of the ε-greedy policy

We first study the performance of the ε-greedy policy when applied to the WNs problem, which allows us to study the purest form of the exploitation-exploration trade-off. First, we show the impact of modifying the initial exploration coefficient, ε_0 (recall that we have considered $\varepsilon_t = \varepsilon_0/\sqrt{t}$), on the average throughput experienced by the overlapping WNs. Figure 2 shows the aggregate throughput obtained in the grid scenario when applying ε-greedy during 10,000 iterations, and for each ε_0 value between 0 and 1 in 0.1 steps. The average and standard deviation of the throughput from 100 simulation runs are shown, which are compared with the optimal throughput presented in Section 5.1.1. As shown in the figure, the aggregate throughput obtained in average is quite similar for all the provided values of ε_0, except for the complete random case where no exploration is done ($\varepsilon_0 = 0$). The lower the ε_0 parameter, the less exploration is performed. Consequently, for low ε_0 the average throughput is highly dependent on how good/bad were the actions taken at the beginning of the learning process, which results in a higher standard deviation as ε_0 goes to 0.

Now, in order to provide a deeper analysis on the WNs behavior when implementing the ε-greedy policy, we focus on the actions probability and the temporal evolution of throughput derived from a single simulation run. To this aim, we use $\varepsilon_0 = \{0.1, 0.5, 1\}$, which correspond to three representative cases (low exploration, middle degree of exploration and intensive exploration).

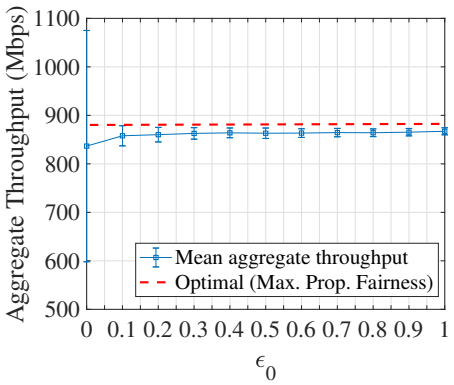


Fig. 2: Average aggregate throughput and standard deviation obtained for each ε_0 value in ε-greedy. Results are from 100 simulations lasting 10,000 iterations each. The proportional fair solution is also shown (red dashed line).

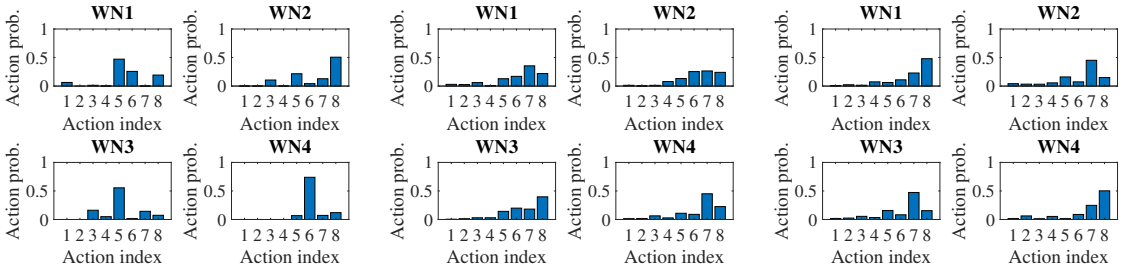
Figure 3 shows the action probability distribution at each WN at the end of the 10,000 iteration-long simulation, and for each of the selected ε_0 values.

We observe in this figure that more aggressive actions (i.e., the ones that use the maximum transmit power) are chosen with a higher rate in all the WNs as ε_0 increases. A higher degree of exploration allows WNs to observe the effects of the actions taken by the others, thus acting in consequence. Accordingly, WNs alternate good/poor performance depending on the actions of the others, which at the same time are also selected intermittently. Despite using a high transmit power allows obtaining a better SNR at the receivers, the generated interference at times becomes counter-productive in terms of individual and aggregate throughput.

On the other hand, as ε_0 decreases (refer to Figure 3(a)), WNs are prone to exploit the same action more frequently, which are the ones that provide the highest known performance. This low degree of exploration results also in a smaller usage of suboptimal actions.

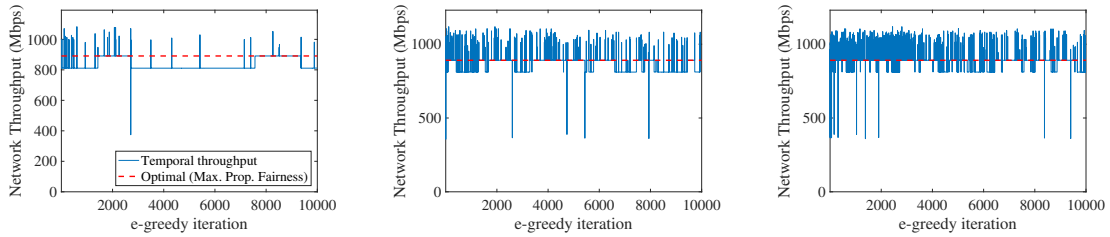
Figure 4 shows the aggregate throughput evolution, which suffers from higher temporal variability as ε_0 increases. The same observation can be done from the individual throughput evolution (shown in Figure 5), which in addition shows larger differences between the individual patterns for lower ε_0 values. These differences correspond to unfair performance resulting from a low degree of exploration.

To gain further insight into fairness, we depict in Figure 6 the average and standard deviation of the throughput at each WN. We have now concentrated in the last 5,000 iterations of the simulation, disregarding the instability provoked during the transitory regime, so that we get a better picture of the long-term behavior. We see that as ε_0 increases, the experienced fairness becomes higher due to the temporal fluctuation that occurs as a consequence of performing more exploratory actions. Moreover, we also observe that the standard deviation of the throughput is similar at each WN.



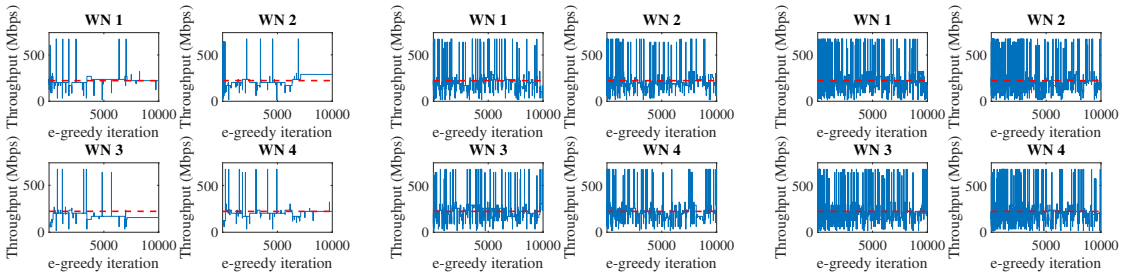
(a) $\varepsilon_0 = 0.1$ (b) $\varepsilon_0 = 0.5$ (c) $\varepsilon_0 = 1$

Fig. 3: Probability of taking a given action in ε-greedy for different ε_0 values after a simulation of 10,000 iterations.



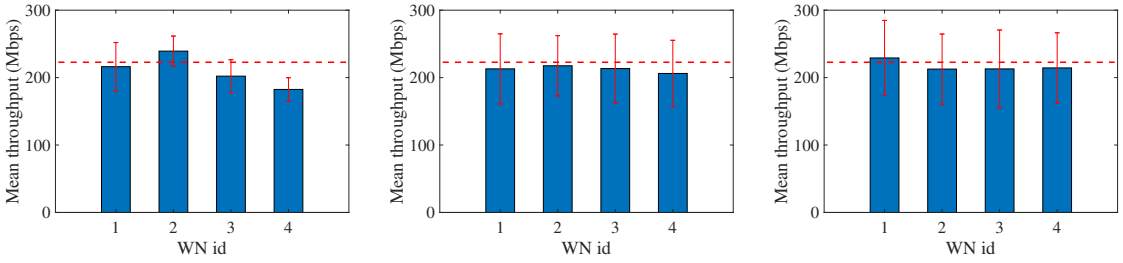
(a) $\varepsilon_0 = 0.1$ (b) $\varepsilon_0 = 0.5$ (c) $\varepsilon_0 = 1$

Fig. 4: Aggregate throughput evolution for a single ε-greedy 10,000-iterations simulation and for different ε_0 values. The proportional fair result is also shown (red dashed line).



(a) $\varepsilon_0 = 0.1$ (b) $\varepsilon_0 = 0.5$ (c) $\varepsilon_0 = 1$

Fig. 5: Individual throughput evolution for a single ε-greedy 10,000-iterations simulation and for different ε_0 values. The proportional fair result is also shown (red dashed line).



(a) $\varepsilon_0 = 0.1$ (b) $\varepsilon_0 = 0.5$ (c) $\varepsilon_0 = 1$

Fig. 6: Average throughput and standard deviation experienced per WN in the last 5,000 iterations of a single ε-greedy 10,000-iterations simulation run for different ε_0 values. The proportional fair result is also shown (red dashed line).

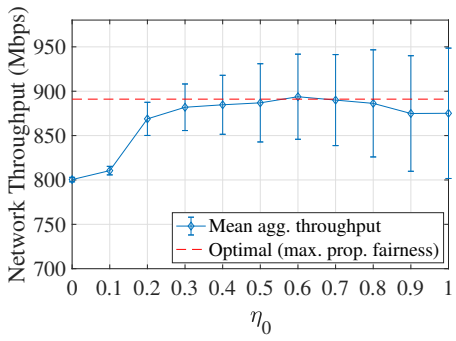


Fig. 7: Average aggregate throughput and standard deviation obtained for each η_0 value in MAB (EXP3). The aggregate throughput of the proportional fair solution is also shown (red dashed line). Results are from 100 simulations lasting 10,000 iterations each.

To sum up, we have seen that tuning ε_0 clearly entails a trade-off between the fairness and the temporal variation of throughput. Choosing a low ε_0 value grants low variability, but WNs are prone to fall into local maximums, as the experienced performance is just a consequence of the randomness in taking particular actions. This also generates a fairness problem, since the unlucky WNs end up experiencing lower performance, as well as they are not able to discover the best throughput actions. On the other side, choosing a high ε_0 value results in higher throughput and better long-term fairness but at the expense of intermittent good/poor performance.

5.1.3 Performance of the EXP3 policy

We now evaluate the trade-off between fairness and temporal throughput variability in the EXP3 policy, which includes the learning rate η, a parameter that controls how fast old beliefs are replaced by newer ones. In EXP3 we also find the γ parameter, which regulates explicit exploration by tuning the importance of weights in the action-selection procedure. Setting $\gamma = 1$ results in completely neglecting weights (actions have the same probability to be chosen). On the other side, setting $\gamma = 0$, the effect of weights are at its highest. Thus, in order to see clearly the effects of the EXP3 weights, which directly depend on η, we fix γ to 0.

As we did before for ε-greedy, we start analyzing the impact of modifying the input parameters of EXP3 on the average aggregate throughput (Figure 7). We vary the initial learning rate, η_0, between 0 and 1 in 0.1 steps. As it can be observed, setting η_0 to 0 results in the lowest average throughput, and a low variability is obtained. In this case, action weights are never updated and the arms are randomly chosen with the same probability. As η_0 increases, the proportional fair solution is approximated at the expense of a higher variability between experiments, since less exploration is performed when η_0 increases. Note, as well, that the maximum average network throughput is reached for $\eta_0 = 0.6$.

Now, for a single 10,000-iteration simulation, we set $\eta_0 = \{0.1, 0.5, 1\}$ to closely study the implications of modifying the initial learning rate. Figure 8 compares the resulting actions probability for these low, medium and high η_0 values.

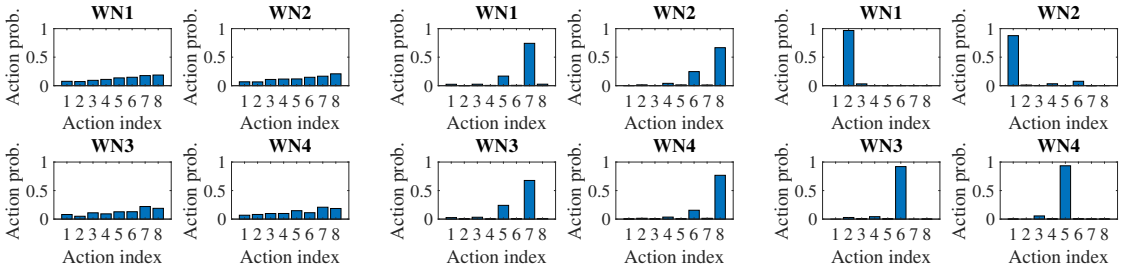
We see that actions are tried with similar probability for $\eta_0 = 0.1$, since the usage of a low learning rate entails a high similarity between weights. This situation is significantly improved for $\eta_0 = 0.5$, in which few actions are selected most of the times. The increased learning rate allows to rapidly capture the effects of the others' actions, since more exploration is carried out. As a consequence, actions using the highest transmit power are clearly preferred to overcome the adversarial setting. Note, as well, that optimal channel allocation is learned by WNs despite competing with each other. Finally, for $\eta_0 = 1$, a single action is clearly preferred by each WN. However, unfair situations eventually occur due to the lack of exploration. In particular, we observe that WN_1 and WN_3, which share the same channel, use very different power levels in their favorite actions (5 and 15 dBm, respectively). This, of course, generates a fairness imbalance that favors WN_3, which enjoys a higher SINR. As mentioned, this behavior occurs eventually, since the variability in the output can be understood as a consequence of a fast convergence achieved during the transitory learning phase. So, results from different simulations will significantly vary.

Figure 9 shows the temporal aggregate throughput, which suffers from a higher variability when η is lower. Regarding individual performance, the same behavior can be observed in the temporal throughput evolution (Figure 10). In this case, setting $\eta_0 = 1$ implies trusting best-performing actions for larger periods, i.e., higher weights are assigned even if actions are suboptimal. Therefore, as η_0 decreases, the temporal variability increases because the action-selection process becomes uniformly random. However, this variability does not entail that the proportional fair solution is approached. Thus, η_0 must be carefully set in order to provide a low temporal variability, at the same time that enough exploration is carried out for reaching the optimal fairness solution.

One can notice a fairness imbalance by observing the individual throughput for $\eta_0 = 1$ in Figure 10, which is further illustrated in Figure 11 (again, we have considered the last 5,000 iterations to avoid capturing the transitory phase). The average throughput is more evenly distributed for $\eta_0 = 0.1$ and $\eta_0 = 0.5$, but at the expense of the experienced throughput variability seen above. The unfairness suffered for $\eta_0 = 0.1$ can be understood as a lack of exploration that prevents finding a fair resource distribution. Thus, we conclude that a similar trade-off experienced in ε-greedy appears in EXP3 with parameter η_0. The main difference is that, in this case, a random action-selection process is approximated as η_0 decreases.

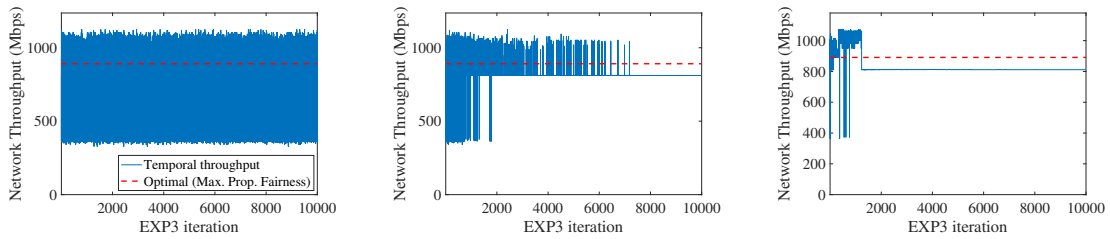
5.1.4 Performance of the UCB policy

Unlike ε-greedy and EXP3, the basic form of UCB does not include any input parameter to regulate variables such as the learning rate or the exploration coefficient. On the contrary, the action-selection process relies in the generated rewards only. We will study in this section how the previously seen trade-off between fair throughput and variability results for this parameter-free policy.



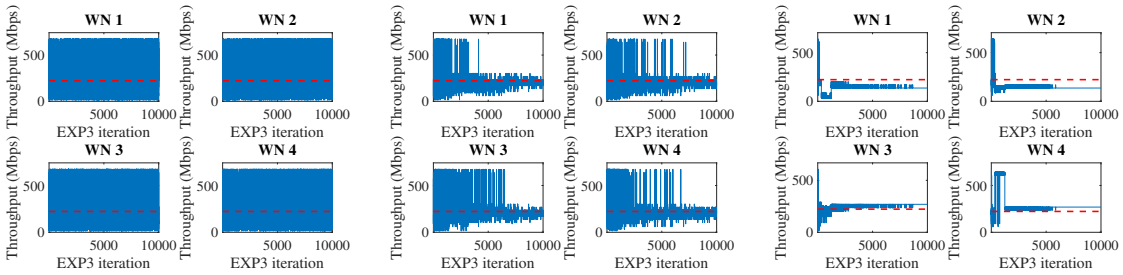
(a) $\eta_0 = 0.1$ (b) $\eta_0 = 0.5$ (c) $\eta_0 = 1$

Fig. 8: Probability of taking a given action in EXP3 for different η_0 values after a simulation of 10,000 iterations.



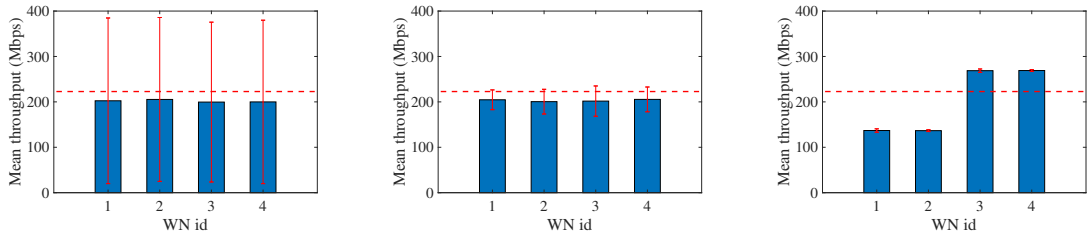
(a) $\eta_0 = 0.1$ (b) $\eta_0 = 0.5$ (c) $\eta_0 = 1$

Fig. 9: Aggregate throughput evolution for a single EXP3 10000-iterations simulation and for different η_0 values. The proportional fair result is also shown (red dashed line).



(a) $\eta_0 = 0.1$ (b) $\eta_0 = 0.5$ (c) $\eta_0 = 1$

Fig. 10: Individual throughput evolution for a single EXP3 10000-iterations simulation and for different η_0 values. The proportional fair result is also shown (red dashed line).



(a) $\eta_0 = 0.1$ (b) $\eta_0 = 0.5$ (c) $\eta_0 = 1$

Fig. 11: Average throughput and standard deviation experienced per WN in the last 5,000 iterations of a single EXP3 10000-iterations simulation for different η_0 values. The proportional fair result is also shown (red dashed line).

We start now by showing the action probability distribution (shown in Figure 12) derived from the UCB implementation. We observe that WNs clearly prefer playing the actions corresponding to proportional fairness (i.e., the actions with highest transmit power and that result in frequency reuse).

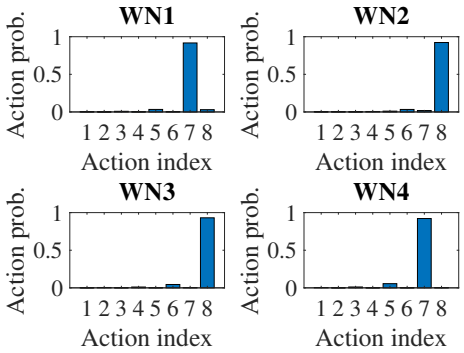


Fig. 12: Probability of taking a given action in UCB for a simulation of 10,000 iterations.

Now, we concentrate in the temporal aggregate throughput (shown in Figure 13). The results show a very high variability, which slowly decreases as the number of iterations increases. The same observations can be done in the individual throughput evolution (Figure 14). As a result, UCB provides long-term fairness but at the expense of a high temporal variability caused by exploration.

In Figure 15 we show the average throughput experienced by each WN, as well as the standard deviation indicating the suffered variability (only for the last 5,000 iterations). As seen before, one can observe really high fairness but considerable variability due to the temporal evolution of throughput.

Compared to ε-greedy and EXP3, UCB shows a clearer preference for selecting the proportional fair solutions. However, the intermittent good/poor performance of the actions due to the adversarial setting still keeps the degree of exploration very high, resulting in high temporal variability.

5.1.5 *Performance of the Thompson sampling policy*

Similarly to UCB, in the basic form of Thompson sampling we do not have any tunable parameter. As with the other policies, we use a single 10,000-iteration simulation to show in detail the WNs behavior when applying Thompson sampling. We first show the actions probability in Figure 16. Note the well-pronounced preference for the proportional fair actions, which indicates fast convergence and little exploratory operations.

The temporal aggregate throughput is shown in Figure 17. As it can be observed, Thompson sampling achieves a lower variability in the temporal throughput with respect to UCB. The same observations can be done in the individual throughput evolution (Figure 18).

The abovementioned low temporal variability is highlighted in Figure 19, which shows that the average experienced throughput enjoys a low standard deviation. Moreover, high fairness among WNs can be observed.

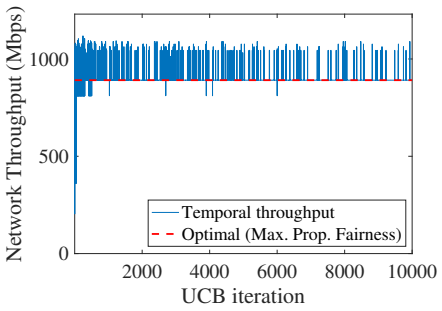


Fig. 13: Aggregate throughput evolution for a single UCB 10,000-iterations simulation. The proportional fair result is also shown (red dashed line).

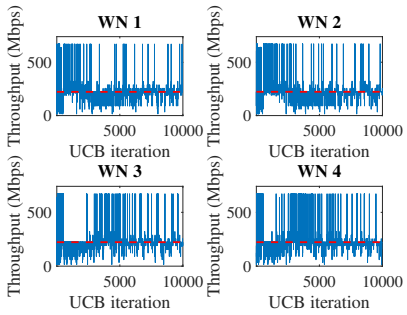


Fig. 14: Individual throughput evolution for a single UCB 10,000-iterations simulation. The proportional fair result is also shown (red dashed line).

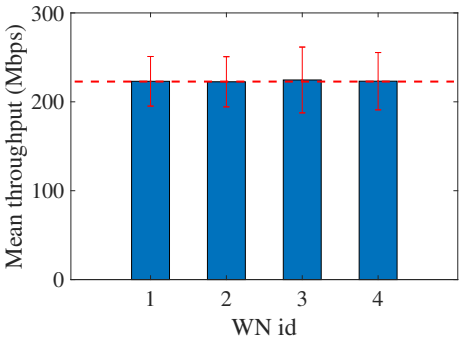


Fig. 15: Average throughput and standard deviation experienced per WN in the last 5,000 iterations of a single UCB 10,000-iterations simulation. The proportional fair result is also shown (red dashed line).

These results show that even though the rewards encourage selfish behavior, the adversarial setting makes Thompson sampling to converge to collaborative actions while keeping the degree of exploration low. While the excellent performance we observe echoes other published results that show the superiority of Thompson sampling in various settings [22], we highlight that this outstanding empirical performance is actually rather unexpected in our *non-stochastic* setting. Indeed, existing theoretical results only explain the excellent performance of Thompson sampling in *stochastic* bandit problems. On the other hand, the randomization employed by Thompson sampling is at least superficially related to techniques used in game-theoretic online learning

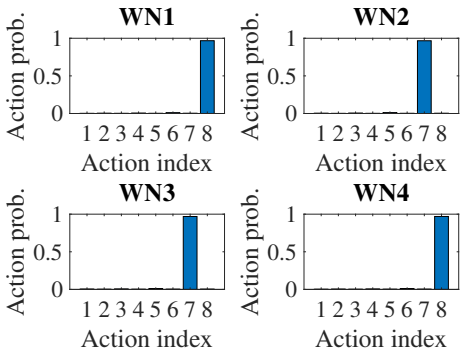


Fig. 16: Probability of taking a given action in Thompson sampling for a simulation of 10,000 iterations.

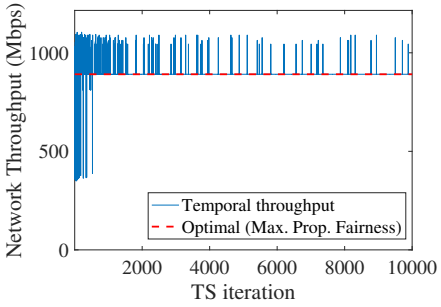


Fig. 17: Aggregate throughput evolution for a single Thompson sampling 10,000-iterations simulation. The proportional fair result is also shown (red dashed line).

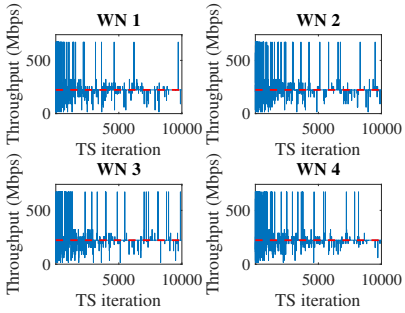


Fig. 18: Individual throughput evolution for a single Thompson sampling 10,000-iterations simulation. The proportional fair result is also shown (red dashed line)

[32], [40], with the connection made clear only in a special case [41]. The behavior observed in our experiments may indicate a more profound relationship with such game-theoretic algorithms. We leave the investigation of this matter as an exciting direction for future work.

5.2 Random Scenarios

After studying the WNs interactions in the grid scenario, we now evaluate whether the previous conclusions generalize to random scenarios with an arbitrary number of WNs. To this aim, we use the same $10 \times 5 \times 10$ m scenario and randomly allocate $N = \{2, 4, 6, 8\}$ WNs.

We set $\varepsilon_0 = 1$ in ε-greedy, which is the value that granted the highest aggregate throughput in the experiment shown

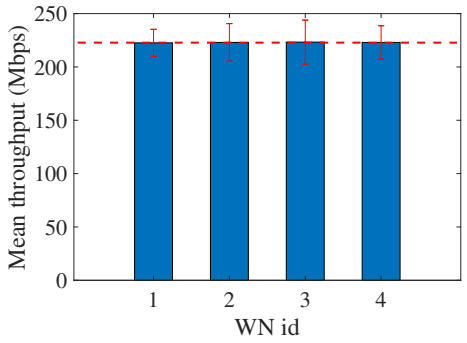


Fig. 19: Average throughput and standard deviation experienced per WN in the last 5,000 iterations of a single Thompson sampling 10,000-iterations simulation. The proportional fair result is also shown (red dashed line).

in Figure 2. We do the same for EXP3 and choose $\eta_0 = 0.6$, which grants the highest aggregate throughput in average (refer to Figure 7).

In Figure 20 we show, for each number of networks and action-selection strategy, an histogram of the mean throughput experienced by each WN during the last 5,000 iterations, and for each of the 100 repetitions.

As we can observe, histograms of UCB and Thompson sampling are slightly narrower in almost all the cases, which indicates a higher fairness experienced by the WNs with respect to the other policies. Note, as well, that for the $N = 2$, results are similar for all the policies because finding a good-performing configuration in low-density environments is more likely to occur. Thus, fewer exploration is required than for higher densities.

We now evaluate temporal variability. In Table 2 we show the average standard deviation of the throughput experienced by each WN during the last 5,000 iterations (again, we concentrate on the regime after the initial learning phase). We consider the average results from 100 repetitions. With this, we are able to evaluate, for each network density, the variability in the temporal throughput provided by each of the policies in random topologies. As previously

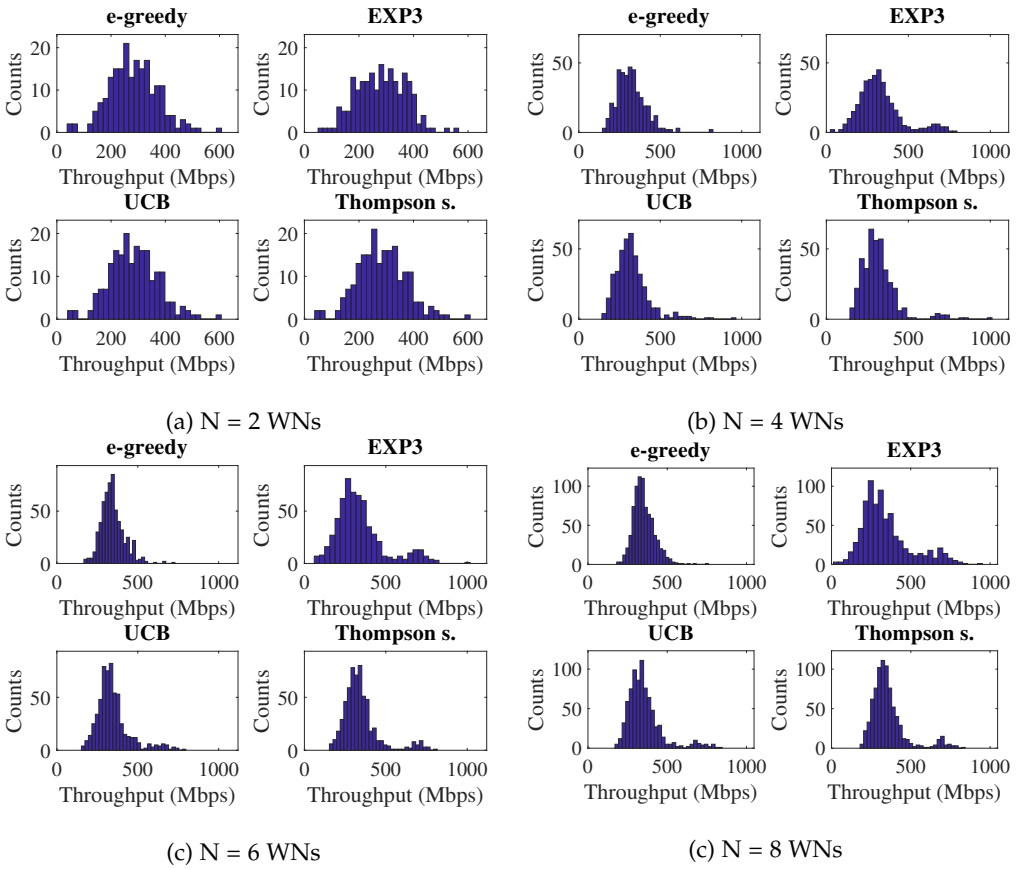
Number of WNs	$\overline{\sigma}(\Gamma_{i \in N})$ (Mbps)			
	ε-greedy	EXP3	UCB	Thompson s.
2	12.1314	31.2897	31.9422	12.2079
4	83.7318	95.4976	73.9136	50.6985
6	120.8570	81.3259	67.6045	62.8171
8	130.0322	73.5171	63.2272	68.8547

TABLE 2: Mean standard deviation of the throughput experienced $\Gamma_{i \in N}$ by each WN_i during the last 5,000 iterations. 100 repetitions are considered for averaging purposes.

seen in the toy grid scenario introduced in Section 5.1, both UCB and Thompson sampling provide the best results in terms of temporal throughput variability. Despite ε-greedy provides very good results for $N = 2$, it does not properly scale up with the number of WNs.

6 Conclusions

We have shown that decentralized learning allows improving SR in dense WN, so that collaborative results,



(a) N = 2 WNs

(b) N = 4 WNs

(c) N = 6 WNs

(c) N = 8 WNs

Fig. 20: Histogram of the average throughput experienced by each WN during the last 5,000 iterations. Results from 100 repetitions are considered.

sometimes close to optimal proportional fairness can be achieved. This result is achieved even though WNs act selfishly, aiming to maximize their own throughput. These collaborative actions are, at times, accompanied by high temporal throughput variability, which can be understood as a consequence of the rate at which networks change their configuration in response of the opponents behavior. A high temporal variability may provoke negative issues in a node's performance, as its effects may be propagated to higher layers of the protocol stack. For instance, a high throughput fluctuation may entail behavioral anomalies in protocols such as Transmission Control Protocol (TCP). We have studied this trade-off between fair resource allocation and high temporal throughput variability in ε-greedy, EXP3, UCB and Thompson sampling action-selection strategies. Our results show that while this trade-off is hard to regulate via the learning parameters in ε-greedy and EXP3, UCB and, especially, Thompson sampling are able to achieve fairness at a reduced temporal variability. We identify the root cause of this phenomena to the fact that both UCB and Thompson sampling consider the probability distribution of the rewards, and not only their magnitude.

We left as future work to further study the MABs application to WNs through distributed (with message passing) and centralized (with complete information) approaches with shared reward. Furthermore, we would like to extend this work to enhance both throughput and stability by inferring the actions of the opponents and acting in consequence. Defining the resource allocation problem as an adversarial game is one possibility to do so. Finally, we intend to particularize to IEEE 802.11 WLANs to study the effects of applying RL when using a medium access protocol such as CSMA. In fact, this would let us consider dynamic Carrier Sense Threshold (CST) to further improve SR.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), by the European Regional Development Fund under grant TEC2015-71303-R (MINECO/FEDER), and by a Gift from CISCO University Research Program (CG#890107) & Silicon Valley Community Foundation.

References

[1] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-management in chaotic wireless deployments. *Wireless Networks*, 13(6):737–755, 2007.

[2] Michael Littman and Justin Boyan. A distributed reinforcement learning scheme for network routing. In *Proceedings of the international workshop on applications of neural networks to telecommunications*, pages 45–51. Psychology Press, 1993.

[3] Biljana Bojovic, Nicola Baldo, Jaume Nin-Guerrero, and Paolo Dini. A supervised learning approach to cognitive access point selection. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1100–1105. IEEE, 2011.

[4] Biljana Bojovic, Nicola Baldo, and Paolo Dini. A neural network based cognitive engine for ieee 802.11 WLAN access point selection. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 864–868. IEEE, 2012.

[5] Richard Combes, Alexandre Proutiere, Donggyu Yun, Jungseul Ok, and Yung Yi. Optimal rate sampling in 802.11 systems. In *INFOCOM, 2014 Proceedings IEEE*, pages 2760–2767. IEEE, 2014.

[6] Marco Miozzo, Lorenza Giupponi, Michele Rossi, and Paolo Dini. Distributed q-learning for energy harvesting heterogeneous networks. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 2006–2011. IEEE, 2015.

[7] Sébastien Bubeck and Nicoló Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[8] Francesc Wilhelmi, Boris Bellalta, Cristina Cano, and Anders Jonsson. Implications of decentralized q-learning resource allocation in wireless networks. *arXiv preprint arXiv:1705.10508*, 2017.

[9] Janne Riihijarvi, Marina Petrova, and Petri Mahonen. Frequency allocation for WLANs using graph colouring techniques. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 216–222. IEEE, 2005.

[10] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for WLANs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):19–31, 2005.

[11] Robert Akl and Anurag Arepally. Dynamic channel assignment in IEEE 802.11 networks. In *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, pages 1–5. IEEE, 2007.

[12] Jeremy K Chen, Gustavo De Veciana, and Theodore S Rappaport. Improved measurement-based frequency allocation algorithms for wireless networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 4790–4795. IEEE, 2007.

[13] Tamer A ElBatt, Srikanth V Krishnamurthy, Dennis Connors, and Son Dao. Power management for throughput enhancement in wireless ad-hoc networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 3, pages 1506–1513. IEEE, 2000.

[14] Suhua Tang, Hiroyuki Yomo, Akio Hasegawa, Tatsuo Shibata, and Masayoshi Ohashi. Joint transmit power control and rate adaptation for wireless LANs. *Wireless personal communications*, 74(2):469–486, 2014.

[15] Carlos Gandarillas, Carlos Martín-Engeños, Héctor López Pombo, and Antonio G Marques. Dynamic transmit-power control for WiFi access points based on wireless link occupancy. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 1093–1098. IEEE, 2014.

[16] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[17] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[18] Peter Auer, Nicoló Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[19] Peter Auer, Nicoló Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

[20] Peter Auer, Nicoló Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[21] Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.

[22] L. Li, W. Chu, J. Langford, and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

[23] Naoki Abe, Alan W Biermann, and Philip M Long. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4):263–293, 2003.

[24] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[25] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[26] R. Agrawal. Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Mathematics*, 27:1054–1078, 1995.

[27] A.N. Burnetas and M.N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17:122–142, 1996.

[28] Pierre Coucheney, Kinda Khawam, and Johanne Cohen. Multi-armed bandit for distributed inter-cell interference coordination. In *ICC*, pages 3323–3328, 2015.

[29] Setareh Maghsudi and Sławomir Stańczak. Joint channel selection and power control in infrastructureless wireless networks: A multiplayer multiarmed bandit framework. *IEEE Transactions on Vehicular Technology*, 64(10):4565–4578, 2015.

[30] Setareh Maghsudi and Sławomir Stańczak. Channel selection for network-assisted D2D communication via no-regret bandit learning with calibrated forecasting. *IEEE Transactions on Wireless Communications*, 14(3):1309–1322, 2015.

[31] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

[32] 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*, 55:119–139, 1997.

[33] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.

[34] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[35] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[36] Nathaniel Korda, Emilie Kaufmann, and Remi Munos. Thompson sampling for 1-dimensional exponential family bandits. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1448–1456. Curran Associates, Inc., 2013.

[37] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107, 2013.

[38] Boris Bellalta. IEEE 802.11 ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, 2016.

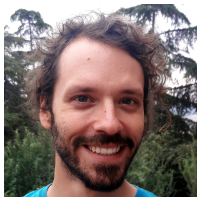
[39] Francesc Wilhelmi. Collaborative spatial reuse in wireless networks via selfih mulit-armed bandits. https://github.com/wn-upf/Collaborative_SR_in_WNs_via_Selfish_MABs Commit: ab5442c48cccf8c770ad709826b4ab93bdcbfa56, 2017.

[40] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[41] Aditya Gopalan. Thompson sampling for online learning with linear experts. *arXiv preprint arXiv:1311.0468*, 2013.

Francesc Wilhelmi holds a B.Sc. degree in Telematics Engineering (2015) and a M.Sc. in Intelligent and Interactive Systems in (2016), both from Universitat Pompeu Fabra (UPF). He is currently a Ph.D Student in the Wireless Networking Research Group (WNRG) in the Department of Information and Communication Technologies (DTIC) at the UPF. His interests are related to online learning for enhancing Spatial Reuse (SR) in high-density wireless networks.

Cristina Cano holds a Ph.D. (2011) in Information, Communication and Audiovisual Media Technologies from the Universitat Pompeu Fabra (UPF). She has been a research fellow in the Hamilton Institute of the National University of Ireland, Maynooth (2012-2014), in Trinity College Dublin (2015-2016) and in Inria- Lille in France (first half of 2016). Currently, she is with the WINE research group of the Universitat Oberta de Catalunya (UOC). Her research interests include coexistence of wireless heterogeneous networks, distributed resource allocation and online optimisation.



Gergely Neu received his M.Sc. degree in Electrical Engineering and his Ph.D. degree in Computer Science from the Budapest University of Technology and Ecomomics (Hungary) in 2008 and 2013, respectively. He has worked as a postdoctoral researcher at the SequeL team of INRIA Lille – Nord Europe (Lille, France, 2013–2015), and is currently a postdoctoral researcher at the Artificial Intelligence and Machine Learning Research group at Universitat Pompeu Fabra (Barcelona, Spain, 2013–). His main research interests are in machine learning theory, including reinforcement learning and online learning with limited feedback and/or very large action sets.

Boris Bellalta is an Associate Professor in the Department of Information and Communication Technologies (DTIC) at Universitat Pompeu Fabra (UPF). He is the head of the Wireless Networking research group at DTIC/UPF. His research interests are in the area of wireless networks, with emphasis on the design and performance evaluation of new architectures and protocols.

Anders Jonsson is the director of the Artificial Intelligence and Machine Learning group at Universitat Pompeu Fabra (UPF) in Barcelona, Spain. He received his Ph.D in computer science in 2005 from the University of Massachusetts Amherst, USA, and has been at UPF ever since. Anders' main research focus is sequential decision making, formulated both as AI planning and as reinforcement learning, but he also conducts research in other areas of AI and machine learning. He is interested in many extensions of the problem of sequential decision making, such as regularization and exploration, hierarchical representations of decision strategies, problems involving multiple agents, and adapting existing techniques to lifelong learning, allowing a system to learn and explore during extended periods of time.

Sergio Barrachina-Muñoz obtained his B.Sc. degree in Telematics Engineering and his M.Sc. in Intelligent Interactive Systems in 2015 and 2016, respectively, both from Universitat Pompeu Fabra (UPF), Barcelona. Currently, he is a PhD student and teacher assistant in the Department of Information and Communication Technologies (DTIC) at Universitat Pompeu Fabra (UPF). His main research interests are focused on developing autonomous learning methods and techniques for improving the performance of next-generation wireless networks.