Analysis of a Decentralized Bayesian Optimization Algorithm for Improving Spatial Reuse in Dense WLANs

Anthony Bardou[a], Thomas Begin[a]

[a] *Univ. Lyon, ENS de Lyon, UCBL, CNRS, LIP, 42 allee d'Italie, Lyon, 69007, France*

Abstract

Despite representing the prominent means of accessing the Internet, WLANs remain subject to performance issues, which may be mitigated through more efficient spatial reuse of radio channels. In this perspective, the IEEE 802.11ax amendment enables the dynamic update of two key parameters in wireless transmission: the transmission power (`TX_POWER`) and the sensitivity threshold (`OBSS_PD`). In this paper, we present `INSPIRE`, a distributed online learning solution performing local Bayesian optimizations based on Gaussian processes to improve spatial reuse in WLANs. `INSPIRE` makes no explicit assumptions on the WLANs' topology and favors altruistic behaviors of the access points in their search for adequate configurations of their `TX_POWER` and `OBSS_PD` parameters. `INSPIRE` can easily be extended to work with a limited number of observations to throttle its computational complexity. We demonstrate the superiority of `INSPIRE` over other state-of-the-art strategies using the ns-3 simulator and two examples inspired by real-life deployments of dense WLANs. Our results show that, in only a few seconds, `INSPIRE` is able to drastically increase the quality of service of operational WLANs by improving their fairness and throughput. Finally, we discuss the configurations recommended by `INSPIRE`. We show that they comply with an 802.11ax empirical recommendation, and we correlate their values with some graph-based metrics of the WLAN topologies.

Keywords: Machine Learning, Bayesian Optimization, Spatial Reuse, IEEE 802.11, Power Control

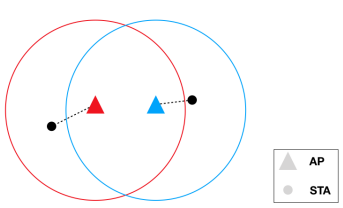
1. Introduction

Wireless networks have become ubiquitous and part of our everyday life in modern societies. According to Cisco's forecasts [1], Wireless Local Area Networks (WLANs) accounted for 57% of the Internet traffic in 2022 when mobile and wired networks represented 22% and 21%, respectively. WLANs typically follow the guidelines of the IEEE standard 802.11 (commercially known as Wi-Fi). They comprise one or multiple Access Points (APs) that relay back-and-forth frames to a set of associated wireless stations (STAs). In addition to their wireless interface, APs also embed a wired interface to provide access to the Internet.

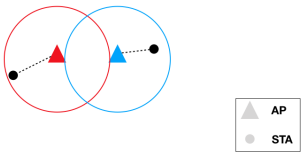
The spatial reuse of radio frequency resources, together with improvements in physical layers and allocation of new radio frequency bands, has been and remains a prominent leverage to increase the efficiency of wireless networks. This problem of assigning radio resources is eased by the breaking of radio bands into separate channels. Indeed, APs assigned to different orthogonal channels can transmit simultaneously without interfering with each other. Furthermore, APs assigned to the same channel can transmit concurrently and successfully if their distance makes the power attenuation on their signal strong enough.

However, from a higher perspective, the spatial reuse of WLANs is somewhat less advanced than that of mobile networks. In the latter networks, this concept, often coined as "small cell" or "cell densification", has led, over the years, the cell radius to decrease from 35 km for early 2G systems to 5 km for 3G systems, 100 m for 4G systems, and about 25 m in 5G [2]. Conversely, the radio range of APs in WLANs has remained constant since the inception of 802.11 in the 1990s with the transmission power of APs set to 100 mW (20 dBm). But this has changed with the recent release of the IEEE 802.11ax amendment [3], opening the path to major gains through more effective spatial reuse in WLANs.

The 802.11ax amendment, released by IEEE in 2021 and commercially branded by the Wi-Fi Alliance as Wi-Fi 6, enables APs to dynamically adjust/tune two key parameters at each AP: `TX_PWR` and `OBSS_PD`. The former parameter specifies the power level (in dBm) at which the AP transmits its data. The latter parameter defines the sensitivity threshold (in dBm). If the energy received is below this level, this indicates to the AP that the radio channel is clear and thus available for transmission. Otherwise, the AP must defer its transmissions. While prior amendments to 802.11 held these `TX_PWR`



(a) With the default configuration of TX_PWR and OBSS_PD, the two APs are within each other's detection range so that they cannot transmit simultaneously.



(b) The value of TX_PWR is reduced on both AP so that they do not belong to each other's detection range. Under this configuration, concurrent transmissions from the two AP may occur at the same time.

Figure 1: Adequately configuring the TX_PWR parameter of APs can significantly improve the spatial reuse of radio channels in WLANs.

and OBSS_PD parameters constant (typically 20 dBm and -82 dBm respectively), 802.11ax has made them dynamic with their values spanning from 1 to 21 dBm for the former and from -82 to -62 dBm for the latter. Adjusting the configurations of TX_PWR and OBSS_PD can help overcome the limitations of spatial reuse in dense environments by allowing APs that are close to each other to transmit on the same channel.

Figure 1 depicts a simple example of two APs operating on the same radio channel and illustrates how different configurations of the TX_PWR parameter can lead to different performance. Note that in this simple example, concurrent transmissions of the two APs could also be attained by increasing OBSS_PD at each AP (and keeping TX_PWR at their default value). Although the two options produce similar effects here, in general, reducing TX_PWR and increasing OBSS_PD may affect the WLANs' performance differently (see Table 2 of [4] for more details).

Despite the potential of 802.11ax to improve the spatial reuse of radio channels, finding an adequate configuration of TX_PWR and OBSS_PD for the APs in a WLAN is a complex problem. First, an adequate configuration is very topology-specific. In other words, knowing an efficient configuration for a given scenario is of (almost) no value for another scenario. Second, a distributed solution would be more appreciated than a centralized solution. Not only does this avoid the search in an otherwise very high dimensional space but this avoids the assumption of having a centralized entity (e.g., a controller) deciding the configurations of all APs. Additionally, a centralized entity can be an acceptable assumption if all the interfering APs belong to

the same WLAN but unrealistic if they belong to concurrent WLANs. Third, forecasting the performance of WLANs with an analytical model that can subsequently help establish optimal configurations is difficult. The degree of details in the models will be either too coarse and thus inapplicable, or adequate but unscalable when scenarios involve multiple APs and STAs. An alternative to this centralized, analytical strategy is a decentralized, learning strategy where APs can apply new configurations of their parameters, measure the effect of these changes on their performance, and exchange their experience with surrounding APs. This paves the way for the use of online and reinforcement learning techniques in a distributed manner.

In this paper, we present a decentralized online learning strategy, based on Bayesian optimization and Gaussian Processes (GPs) to efficiently address the issue of improving the spatial reuse of a radio channel in WLANs. This leads us to define a distributed algorithm, known as `INSPIRE`, which makes no explicit assumptions about the topology of WLANs or the radio environments and thus can apply to any WLANs (including when APs belong to different WLANs). The current paper extends our previous work presented in [5]. Overall, the contributions of the paper are as follows:

- We demonstrate the ability of GPs at approximating the reward function that reflects the performance of WLANs, and at exploring efficient AP configurations;
- We establish the superiority of a divide-and-conquer approach to handle the complex problem of setting the `TX_PWR` and `OBSS_PD` parameters at each AP;
- We introduce `INSPIRE` a distributed solution that lets the APs of concurrent WLANs automatically adapt their internal parameters' setting in their own interest as well as in the interest of obtaining more efficient spatial reuse of radio channels;
- We propose a lightweight version of `INSPIRE`, referred to as `INSPIRE_LIM`, which can be of practical interest when limited computing resources are available to conduct the search for an adequate setting of `TX_PWR` and `OBSS_PD` parameters.
- We evaluate the efficiency of `INSPIRE` on real-life inspired case studies using a detailed network discrete-event simulator and compare its

performance with several state-of-the-art solutions (centralized or distributed).

- We correlate the values found for TX_PWR and OBSS_PD by INSPIRE with metrics related to the network topology under study.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. We use Section 3 to present our proposed strategy and the main theoretical results to handle the issue of spatial reuse of a radio channel in WLANs. The performance evaluation of our strategy is reported in Section 4. We use Section 5 to deepen the understanding of the numerical results as well as the parameter settings made by INSPIRE. Section 6 concludes this paper.

2. Background

2.1. Spatial Reuse with TX_PWR and OBSS_PD

The release of the IEEE 802.11ax amendment [6] in late 2021 marks a new era for the spatial reuse of radio channels of WLANs: Nodes can dynamically adjust their transmission power (TX_PWR) and sensitivity threshold (OBSS_PD) parameters. For a detailed explanation of how this new feature is implemented, we refer the interested reader to [7], which also provides simple scenarios to illustrate its potential benefits.

Years before IEEE released the 802.11ax amendment, the idea of dynamically updating TX_PWR and OBSS_PD has been explored by some researchers. The pioneering work of [8] presents an analytical model that, based on the current radio channel conditions, dynamically configures OBSS_PD on each node of a Wi-Fi-based mesh network. Concurrently, [9] established that adapting TX_PWR can lead to increased throughput and reduced energy consumption. More recently, in 2020, [10] casts the issues of positioning the APs of a WLAN and choosing their TX_PWR as an optimization problem. The authors provides a solution to this problem that delivers a static configuration of TX_PWR for a WLAN. But their solution does no account for the number of STAs nor the type of traffic in the WLAN.

The difficulty of accurately modeling the dependency between the configuration parameters of a large WLAN and its performance is a strong hurdle to the development of spatial reuse strategies based on analytical models. As a result, most of the proposed strategies are data-driven. Adaptive by

construction, they constitute promising candidates in the search for configurations that improve the spatial reuse of a radio channel and thus, the performance of WLANs.

Machine learning (ML) techniques are natural candidates for addressing problems requiring a data-driven approach, and the spatial reuse problem is no exception to that rule. [11] addresses the problem of configuring `TX_PWR` and `OBSS_PD` with a two-scale solution using artificial neural networks (ANN). In their strategy, STAs and APs first adjust their value of `OBSS_PD` to minimize interference. Then, an ANN, which was trained offline through simulation, is used to increase the fairness between STAs in terms of attained throughput. However, given the vast diversity of WLAN topologies, the offline learning of the ANN appears as a clear limitation to the generalization of this strategy. An online learning procedure is proposed by [12], which uses reinforcement learning and more precisely the Multi-Armed Bandit (MAB) framework to find the optimal configuration of `TX_PWR` and `OBSS_PD` in a WLAN. The approach comprises two agents with one sampling promising configurations through a multivariate normal distribution, and the other identifying the best configuration among those already sampled with Thompson sampling and Normal-Gamma priors. A similar strategy is proposed by [13], which recommends the use of a mixture of hyperspheres to subsample the configuration space. These three ML solutions [11, 12, 13] are tested on the network simulator ns-3 and lead to significant WLAN improvements. However, in order to perform their optimization, they all assume the presence of a central controller that has access and control over all the APs in the WLANs. By construction, these approaches are centralized, and hence cannot be applied to cases where concurrent WLANs managed by different owners interfere with others.

Distributed approaches are undisputedly better fit than centralized approaches to handle cases with a set of concurrent WLANs. [14] introduces a distributed algorithm named Dynamic Sensitive Control which is run on every STAs of a WLAN. In short, each STA tries to dynamically reduce its value of `OBSS_PD` to favor concurrent transmissions while keeping it high enough to ensure a high quality signal reception. Similarly, [15] proposes Link-aware Spatial Reuse (LSR), a distributed algorithm designed for the APs. In LSR, each AP chooses another AP, which is allowed to transmit concurrently, and then prescribes a value of `TX_PWR` for the selected AP. These two algorithms rely on a single measurement metric reflecting the quality of the received signal, namely the Received Signal Strength, to choose the nodes configura-

tion. More recently, strategies using distributed MAB approaches have been proposed [16, 4]. They both use Thompson sampling with Gaussian priors to find the best couple of TX_PWR and OBSS_PD at each AP. In [16], each AP seeks to maximize the throughput of its associated STAs. On the other hand, in [4], the authors assume that every AP has access to the performance of all other APs in the WLAN; then each AP attempts to maximize a global reward that takes into account the performance of all the other nodes. Both strategies [16, 4] solutions were evaluated in a self-made simulator with simple random scenarios.

Table 1 summarizes the main characteristics of the data-driven strategies discussed above. It shows that, out of the eight considered strategies, two (namely [14, 11]) only focus on the configuration of the OBSS_PD parameter (keeping the TX_PWR parameter fixed). To help in the comparison of the different strategies, we introduce two concepts: "pull area" and "push area". The pull area indicates the area on which each node is assumed to obtain information (this typically includes parameter configurations and performance measurements). Depending on the strategy being considered, the pull area can include just the node itself, the surrounding nodes, or the whole set of nodes in the WLANs. The push area designates the area which each AP can influence typically through the prescription of parameter configurations. In the case of centralized strategies (e.g., [11, 12, 13]), the pull and push areas naturally cover the whole set of APs. We distinguish partially distributed strategies (e.g., [4]) wherein either the pull or push area includes the whole set of APs with fully distributed strategies (e.g., [14, 15, 16]) in which both the pull and push areas differ from the whole set of APs. We observe in Table 1 that only three out of the eight state-of-the-art strategies can be considered as fully distributed. The last four columns of Table 1 pertain to the performance evaluation used to validate each of these strategies. It appears that most strategies were evaluated without considering the dynamical selection of the Modulation Coding Scheme (MCS) for the speed of the wireless links, nor bidirectional (with upstream and downstream) traffic. This can be seen as a strong limitation since this overlooks some associated trade-offs. For instance, increasing the value of TX_PWR certainly enables the communication to operate with a faster data rate (larger MCS), but at the same time, it increases the level of interference with surrounding APs. Additionally, most strategies were evaluated on relatively simple scenarios (with a few APs and a limited number of radio channels), often using a self-made network simulator.

Table 1: Comparison of the state-of-the-art data-driven strategies. The last column refers to the size of the scenarios involved in the performance evaluation of the strategy. For instance, 216/18 means the evaluation comprises 216 APs distributed over 18 radio channels.

Proposed solution	Tuning of `TX_PWR`	Degree of centralization		Dynamic MCS	Traffic Up/Down	Simulator	APs / channels
		Pull area	Push area				
WCNC'15 [14]		Associated STAs	Associated STAs		Up	Self-made	100/3
WCNC'21 [15]	✓	AP itself	AP itself	✓	Down	ns-3	6/1
Globecom'20 [11]		All APs	All APs		Up/Down	ns-3	3/1
ADHOC'19 [16]	✓	AP itself	AP itself		Down	Self-made	8/1
JNCA'19 [4]	✓	All APs	AP itself		Down	Self-made	8/1
MSWiM'21 [12]	✓	All APs	All APs		Down	ns-3	10/1
ADHOC'23 [13]	✓	All APs	All APs	✓	Up/Down	ns-3	216/18
`INSPIRE`	✓	Surrounding APs	Surrounding APs	✓	Up/Down	ns-3	216/18

This paper is an extension of the previous work [5] that proposes a fully distributed strategy to address the problem of the spatial reuse of radio channels in WLANs. The proposed strategy, named `INSPIRE`, can be applied to any arrangement of WLANs and its novelties are mostly twofold. First, to the best of our knowledge, it is the first strategy making use of Gaussian Processes to explore promising WLAN configurations in the quest of discovering the optimal one. Gaussian processes are recognized tools to deal with the exploration vs. exploitation dilemma (see [17, 18]) which is at the center of the spatial reuse problem. Second, unlike the existing fully distributed strategies, `INSPIRE` allows each AP to account for its surroundings thanks to pull and push areas broader than a single node. Through the use of a simple consensus method, APs of the WLANs achieve to behave altruistically selecting configurations for the "greater good" of the WLANs. We also introduce realistic scenarios, inspired by real-life WLANs, with dynamic MCS and bidirectionnal traffic, to evaluate and compare the efficiency of all the considered strategies.

2.2. Bayesian Optimization with Gaussian Processes

Before describing our solution, let us introduce Bayesian Optimization (BO) and Gaussian Processes (GPs) in this section, as they play a key role in our proposed solution to the spatial reuse problem in WLANs.

Gaussian Process. A GP is a stochastic process, that is, a collection of random variables $\{Y(x)\}_{x\in\mathcal{C}}$ indexed by a set $\mathcal{C}$. The specificity that makes a *Gaussian* process is that any finite set of random variables $\{Y(x_1), \cdots, Y(x_n)\}$ has a joint multivariate Gaussian distribution. It can be fully specified by its mean function μ and its covariance function Σ

$$\mu(x) = \mathbb{E}\left[Y(x)\right] \tag{1}$$

$$\Sigma\left(x, x'\right) = \mathbb{E}\left[\left(Y(x) - \mu(x)\right)\left(Y(x') - \mu(x')\right)\right]. \tag{2}$$

In most cases, a BO algorithm rely on the pioneering work [19], that proposes a unified framework for regression tasks using GPs. We describe this approach in a few lines and we illustrate it with Figure 2. A GP is considered as a surrogate model for a black-box objective function f. It requires a prior GP, describing the random variables $\{f(x)\}_{x\in\mathcal{C}}$, by a prior mean function $\mu_0(x)$ and a covariance function $k(x, x')$. The vast majority of BO algorithms consider, without loss of generality, that $\mu_0(x) = 0$. For

any point $x \in \mathcal{C}$, it is known by construction that $f(x) \sim \mathcal{N}(\mu_0(x), \sigma_0^2(x))$, with $\sigma_0^2(x) = \Sigma(x, x)$. Given t observations, described by the $t \times d$ matrix of datapoints $X = (x_i)_{i\in[1,t]}^\top$ and by the corresponding t-dimensional vector of outputs $y = (y_i)_{i\in[1,t]}^\top$, the posterior distribution of $f(x)$ is obtained from the $t+1$-dimensional joint Gaussian distribution of the t observations along with the queried $f(x)$, by conditioning on the observed data. Then, a posteriori, we have $f(x) \sim \mathcal{N}(\mu_t(x), \sigma_t^2(x))$ with

$$\mu_t(x) = \Sigma(x, X)^\top K^{-1} y \tag{3}$$

$$\sigma_t^2(x) = \sigma_0^2(x) - \Sigma(x, X)^\top K^{-1} \Sigma(x, X) \tag{4}$$

with $\Sigma(x, X) = (\Sigma(x, x_i))_{i\in[1,t]}$ the t-dimensional vector built by applying the covariance function to x and each datapoint in X and $K = (\Sigma(x_i, x_j))_{i,j\in[1,t]}$ the $t \times t$ matrix built by applying the covariance function to each possible couple of datapoints in X.

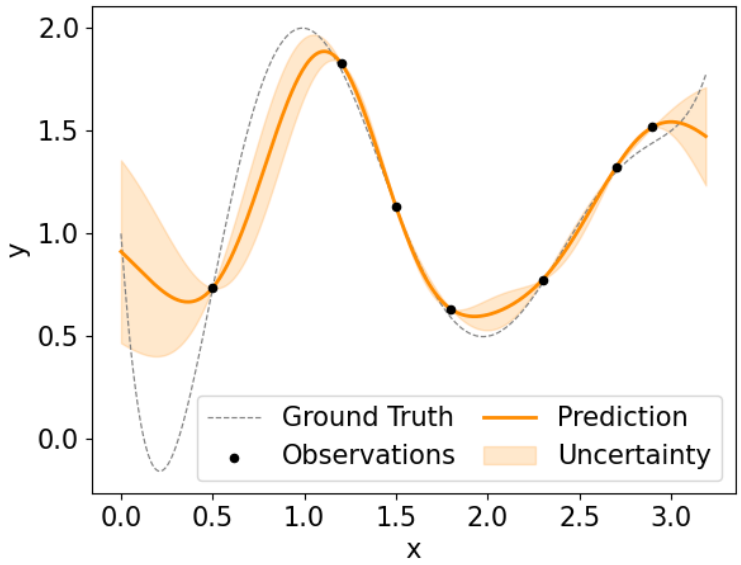


Figure 2: Gaussian Process Regression. The ground truth $f(x)$ is depicted with a grey dashed line and the noisy observations (X, y) with black dots. The posterior mean $\mu_t(x)$ is shown with a solid orange line and the posterior standard deviation $\sigma_t(x)$ as an orange interval.

A BO algorithm needs to provide, at each time step t, a query x^t that achieves a trade-off between exploration and exploitation. The BO algorithm

typically bases its querying policy on the maximization of an acquisition function that quantifies the benefits of observing $f(x)$. Common acquisition functions include Probability of Improvement [20], Expected Improvement [21] and GP-UCB [22]. Further details are provided about the choice of the acquisition function in our case and its maximization in Section 3.3. The use of an efficient acquisition function allows us to sequentially extend our observed data set with promising points that improve the quality of the GP regression and ultimately get closer and closer to the optimal configuration.

Under some assumptions, BO offers optimality guarantees regarding the optimization of an arbitrary objective function f, and it has obtained excellent empirical performance in a variety of black-box optimization tasks [23, 24, 25]. Since the spatial reuse of a WLAN can be seen as a black-box objective function, we propose to use BO to optimize the performance of WLANs. We are now ready to introduce our proposed solution, called `INSPIRE` and described in the next section.

2.3. Positioning

The current paper extends our previous work presented in [5] in several ways: (i) It studies the response time of the `INSPIRE` algorithm; (ii) It introduces a lightweight version of `INSPIRE` where only the latest observations (50 in our case) of the WLAN performance are considered; (iii) It sheds light on why seemingly similar network topologies may actually result into optimization problems of different complexity; (iv) It correlates the best values found for `TX_PWR` and `OBSS_PD` at each AP with metrics related to the network topology.

3. Proposed solution

3.1. Assumptions on the WLANs under study

Let $\mathcal{W}$ denote the set of concurrent WLANs under study, each of which being comprised of one or more APs. We let V be the set of APs in $\mathcal{W}$ that operate on the radio channel of interest. We denote by N the number of APs, by s_i the set of STAs associated with AP i and by S the total number of STAs in the considered radio channel of $\mathcal{W}$. Thus, we have: $S = \sum_{i=1}^{N} |s_i|$. Finally, we use $\mathcal{N}_i$ to designate the set of APs that are within the communication range of AP i (when every AP is under the default configuration of the `TX_PWR` and `OBSS_PD` parameters). Note that AP i itself belongs to $\mathcal{N}_i$. We refer to the APs in $\mathcal{N}_i$ as the surroundings of AP i.

We make no assumptions on $\mathcal{W}$, including on the specific arrangement of its APs and STAs, other than the three detailed below.

First, we assume that every AP i is able to exchange control frames (possibly through its beacon frames) with its surrounding APs (*i.e.*, the ones in $\mathcal{N}_i$). By the same token, we suppose that at least one AP i has another AP in its communication range (*i.e.*, $\exists i \in [\![1, N]\!], \mathcal{N}_i \neq \{i\}$), otherwise the spatial reuse of the radio channel would already be at its apex.

Second, we assume that the N APs have their `TX_PWR` and `OBSS_PD` parameters configurable (as defined by the 802.11ax amendment). We use x_i^t to denote the configuration of AP i with regards to its two `TX_PWR` and `OBSS_PD` parameters at time t. Analogously, x^t represents the configuration of the N APs from $\mathcal{W}$ at time t. Thus, we have: $x_i^t \in C = [\![-82, -62]\!] \times [\![1, 21]\!]$ dBm and $x^t \in C^N$.

Lastly, we assume that each AP in $\mathcal{W}$ can periodically run performance tests and obtain, in return, the mean throughput attained by each of its STAs over a short time interval Δt. More formally, we use the vector $T^t \in \mathbb{R}^{+S}$ to denote the throughput attained by the S STAs of $\mathcal{W}$ given the WLAN configuration x^t at time t. Throughout this paper, we sometimes refer to T^t as $T(x^t)$ to explicitly exhibit the dependency between the STAs' throughputs and APs' configurations.

In this work, we seek to discover an adequate configuration x^* of the N APs composing $\mathcal{W}$ that improves the collective experience of the S STAs through a better reuse of their radio channel. We address this problem as a reinforcement learning task in which, at regular time intervals t, the APs collect measurements T^t associated to their current configuration x^t, and need to decide their next configuration x^{t+1}. The obstacles towards that objective are mostly threefold. (i) We need to define a meaningful objective function that APs will attempt to optimize collectively; (ii) We are facing the well-known exploration vs. exploitation dilemma since the search for an adequate configuration of the WLANs should be as seamless as possible (without disrupting the STAs). This leads us to cast the problem as a MAB problem where the arms are the WLANs' configurations. Following the MAB terminology, we refer to the objective function as the reward function; (iii) We are looking for a strategy that can be applied in a distributed way since it would be in general unrealistic to assume that (concurrent) APs have a fine knowledge beyond their surroundings.

3.2. Reward function

We need to define a reward function R that appraises the "goodness" of a configuration x with regards to the WLANs performance. Because multiple criteria may be considered in the definition of R, there is no universal definition. However, assessing the quality of a configuration x can be derived from the STAs throughputs $T(x)$ obtained with APs configured with x. Among all easily computable reward functions, $R(x) = \prod_{T_i \in T(x)} T_i$ is called the proportional fairness (PF) and provides a convenient trade-off between fairness and cumulated throughput. However, PF is often criticized for its high variability, since $\frac{\partial R}{\partial T_i} = \prod_{T_j \in T(x), j \neq i} T_j$.

To overcome this drawback, we consider the logarithm of PF. This lowers its variability, which becomes: $\frac{\partial R}{\partial T_i} = \frac{1}{T_i}$ (note that T_i is typically much larger than 1). This also emphasizes the contribution of STAs with low throughputs in the computation of R and provides a pleasant closed-form to optimize. For an arbitrary set of APs X, we can define $R_X = \log \prod_{\substack{i \in X \\ j \in s_i}} T_j(x)$. Then, our *global* reward function R is:

$$\begin{aligned} R(x) = R_V(x) &= \log \prod_{\substack{i \in V \\ j \in s_i}} T_j(x) \\ &= \sum_{\substack{i \in V \\ j \in s_i}} \log T_j(x) \end{aligned} \tag{5}$$

However, to compute Equation 5, an AP must have a complete knowledge of the performance attained by the STAs of all APs or, at least, be able to communicate with all the APs in $\mathcal{W}$. This is in contradiction with our assumption that APs only have a partial knowledge of $\mathcal{W}$, limited to their surrounding APs. To design a reward function compatible with the distributed case, we proceed as follows. Each AP i applies Equation 5 but restricted to the set of its associated STAs and obtains in return a "selfish" reward denoted by $R_{\{i\}}$. Previous work [16] have showed that considering such selfish rewards may have a positive but limited impact on the WLANs performance. Therefore, we introduce a more altruistic reward, denoted by R_i that accounts not only for the "selfish" reward of AP i (*i.e.*, $R_{\{i\}}$) but also for the rewards of the surrounding APs (*i.e.*, the ones in $\mathcal{N}_i$). The "altruistic" local reward of AP i is computed as:

$$R_i(x) = \sum_{j \in \mathcal{N}_i} \frac{R_{\{j\}}(x)}{|\mathcal{N}_j|} \tag{6}$$

Note that R_i defined with Equation 6 ensures that a configuration x maximizing all the local rewards is also a maximum for the global reward R since we have: $\sum_{i=1}^{N} R_i(x) = R(x)$.

Proof. This is a straightforward property since

$$\sum_{i=1}^{N} R_i(x) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{R_{\{j\}}(x)}{|\mathcal{N}_j|} \tag{7}$$

By noticing that $i \in \mathcal{N}_j \iff j \in \mathcal{N}_i$, we can permute the indices and use the definition of $R_{\{j\}}$:

$$\sum_{j=1}^{N} \frac{R_{\{j\}}(x)}{|\mathcal{N}_j|} \sum_{i \in \mathcal{N}_j} 1 = \sum_{j=1}^{N} \sum_{k \in s_j} \log T_k(x) = R(x) \tag{8}$$

□

3.3. Local reward maximization

For the sake of clarity and since all variables in this section are relative to an AP i, we often omit the subscript i in the notations.

Now that each AP i has its own local reward function, we need a model of the knowledge of AP i about R_i in order to find its argmax. We represent the beliefs of AP i about R_i by defining a prior distribution on the reward function space with a GP.

In our case, a GP can be defined as a collection of random variables indexed by configurations of APs in $\mathcal{N}_i$: $\{Y_c; c \in C^{|\mathcal{N}_i|}\}$ such that every finite collection $(Y_{c_1}, \cdots, Y_{c_n}) \sim \mathcal{N}(\mu, \Sigma)$. Without loss of generality, we assume the GP to have zero mean so that it is entirely determined by its covariance function $\Sigma : C^{|\mathcal{N}_k|} \times C^{|\mathcal{N}_k|} \to \mathbb{R}^+$. We use X_t to denote the $t \times 2|\mathcal{N}_k|$ features matrix gathering the tested configurations $(x^1, \cdots, x^t)^T$ and Y_t to denote the $t \times 1$ label vector gathering the corresponding local reward values $(R_i(x^1), \cdots, R_i(x^t))^T$. Given X_t and Y_t, we can use Equations 3 and 4 to infer the distribution of the reward value for an arbitrary configuration x, $R_i(x)$.

Since GPs can be used as a prior on a function space, they are useful to solve regression problems as well as maximization tasks. In our case, the AP i uses a GP to model R_i and to assist the exploration of promising configurations of the APs in $\mathcal{N}_i$, maximizing R_i in a Bayesian way.

Choosing the covariance function Σ is a critical step when designing a GP as it determines some key features such as its isotropy and smoothness. Since the reward function, which quantifies the quality of spatial reuse in $\mathcal{N}_i$, is likely to exhibit threshold effects, we choose a covariance function that decreases rapidly as the distance between two considered configurations increases. This allows the GP to model the true function instead of considering a smooth approximation of it. Because we have no incentive to prefer any particular direction over another, we let the covariance function $\Sigma(x, x')$ depend only on $||x - x'||$ to ensure the isotropy of the GP. This leads us to use a Matérn kernel [26] with parameter $\nu = \frac{3}{2}$, which is defined as

$$\Sigma(x, x') = s^2 \left(1 + \frac{\sqrt{3}||x - x'||}{\rho}\right) e^{-\frac{\sqrt{3}||x-x'||}{\rho}} \tag{9}$$

where s^2 and ρ are two hyperparameters whose values are approximated by maximizing the likelihood of Y_t (which is Gaussian) during the learning process.

As discussed before, each AP i faces the exploitation vs exploration dilemma in its attempt to find the optimal configuration. A common way in the MAB framework to appraise a given strategy π is then to consider the cumulative regret $\Gamma(\pi)$. In our problem, $\Gamma(\pi)$ is expressed with Equation 10 for an episode of D steps, since it is expressed as the cumulative sum of the differences between the best reward that the AP can get and $R_i(\pi(t))$, which is the actual reward obtained at time t for the strategy π.

$$\Gamma(\pi) = \sum_{t=1}^{D} \max_{x \in C^{|\mathcal{N}_i|}} R_i(x) - R_i(\pi(t)) \tag{10}$$

Minimizing the cumulative regret with GP models is usually performed by defining a strategy π that derives from the maximization of an acquisition function A: $\pi(t) = \arg\max_{x \in C^{|\mathcal{N}_i|}} A_t(x)$. However, this assumes that our search space $C^{|\mathcal{N}_i|}$ is continuous. Since each AP i deals with discrete configurations of APs in $\mathcal{N}_i$, we systematically round the recommendation of the GP to the nearest valid WLAN configuration. Many acquisition functions exist, such as Knowledge Gradient (KG) [27], GP-UCB [22] or the Expected Improvement (EI) [21]. We choose EI over KG (whose computational cost can rapidly become prohibitive) and GP-UCB (which was found to be less efficient in our experiments). The EI acquisition function is expressed

as $A_t(x) = \mathbb{E}\left[(\mu_{t+1}(x) - \max_{1\leq k\leq t} R_i(x_k))^+\right]$ given that $X_{t+1} = (X_t, x)$, $Y_{t+1} = (Y_t, R_i(x))$. Since $R_i(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$, we can derive a convenient closed-form for EI, as shown in Equation 11.

$$EI(x) = (\mu(x) - R^*_{i,t})\Phi(Z) + \sigma(x)\phi(Z) \tag{11}$$

with $R^*_{i,t} = \max_{1\leq k\leq t} R_i(x_k)$, $Z = \frac{\mu(x)-R^*_{i,t}}{\sigma(x)}$, Φ and ϕ being respectively the CDF and the PDF of a standard Gaussian distribution.

Then, the AP i can try to maximize Equation 11 by differentiating it and performing a gradient ascent. By applying its strategy $\pi_i(t) = \arg\max_{x\in C^{|\mathcal{N}_i|}} EI(x)$ and classical gradient ascent techniques on Equation 11, AP i provides promising configurations for its surrounding APs in $\mathcal{N}_i$.

3.4. Aggregation of local prescriptions

In the previous sections, we have described how each AP i computes its local reward and relies on its model $\mathcal{GP}_i$ to explore promising configurations for the APs in $\mathcal{N}_i$.

However, more coordination between APs is required. By construction, the collection $\mathcal{F} = (\mathcal{N}_k)_{1\leq k\leq N}$ is a cover of the set of APs in $\mathcal{W}$ but not a partition. In fact, if $\mathcal{F}$ had only null intersections (*i.e.*, $\forall j, k, \mathcal{N}_j \cap \mathcal{N}_k = \emptyset$), then the spatial reuse of the radio channel would already be at its apex and there is no need for improvement. Figure 3 illustrates an example with 5 APs in which the collection $\mathcal{F} = (\mathcal{N}_1, \cdots, \mathcal{N}_5)$ exhibits multiple non-null intersections. As a result, most APs will receive *a set* of *different* prescriptions for the configuration of their `TX_PWR` and `OBSS_PD` parameters at their next iteration. For instance, AP 1 will receive prescriptions from APs 2 and 4 in addition to its own prescription. Since APs can only test one configuration at a time, one of those prescriptions must be chosen, or preferably, a consensus between them must be reached.

In general, independently maximizing each local reward function is very likely to lead to a sub-optimal situation since, for non-linear optimization problems, individual interests are often not aligned with the global interest (e.g., the famous Tragedy of the Commons [28]). Without more information on the relation between the configuration of the APs and the measured throughputs of STAs, it seems difficult to provide an expression for the maximal argument of the global reward function R given the maximal argument of the local reward functions R_i. However, recall that $\sum_{i=1}^{N} R_i(x) = R(x)$. We can leverage this property to provide guarantees.

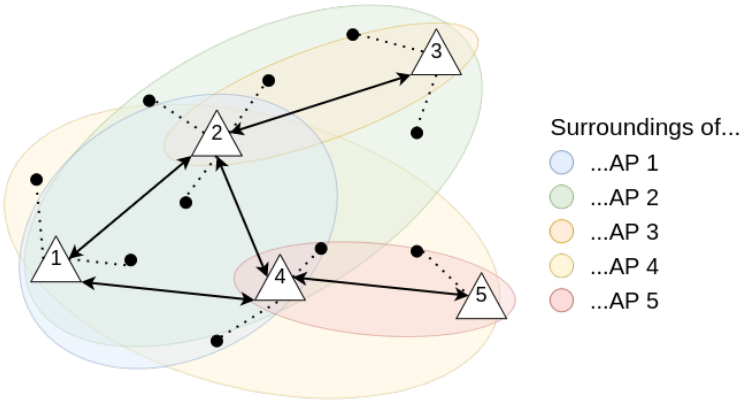


Figure 3: A WLAN represented by a graph with APs depicted as labelled triangles and STAs as black dots. An edge exists between two APs when they are in the communication range of each other. We use different colors to illustrate the surroundings of each AP in $\mathcal{F}$.

Theorem 1. *Let $\{R_i\}_{1\leq i\leq N}$ be a set of Lipschitzian functions, $w \in [0,1]^N, ||w||_1 = 1$ be a weight vector and $\{x_j^i\}_{i\in\mathcal{N}_j}$ be the prescriptions received by the AP j, with $x^i = \arg\max_{x\in C^{|\mathcal{N}_i|}} R_i(x)$, then the weighted marginal median of the received sets of prescriptions is a maximin optimum $\tilde{x}$ of $\sum_{i=1}^N w_i R_i(\tilde{x}_{\mathcal{N}_i})$:*

$$\tilde{x}_j = median\left(\{\left(x_j^i, w_i\right)\}_{i\in\mathcal{N}_j}\right) \tag{12}$$

Proof. Let x^i be the prescription of AP i, with x_j^i the prescription of AP i for AP j and $w \in [0,1]^N$ a normalized weight vector ($||w||_1 = 1$). Consider the quantities $R^* = \sum_{i=1}^N w_i R_i(x^i)$ and $\tilde{R} = \sum_{i=1}^N w_i R_i(\tilde{x}_{\mathcal{N}_i})$ for a given consensus $\tilde{x} \in C^N$. If $x^i = \arg\max_{x\in C^{|\mathcal{N}_i|}} R_i(x)$, then it follows that $\forall \tilde{x} \in C^N, R^* - \tilde{R} > 0$. We want to minimize this difference. If all the functions R_i are λ-Lipschitzian, we have:

$$
\begin{aligned}
R^* - \tilde{R} &= \sum_{i=1}^{N} w_i R_i(x^i) - \sum_{i=1}^{N} w_i R_i(\tilde{x}_{\mathcal{N}_i}) \\
&= \sum_{i=1}^{N} w_i (R_i(x^i) - R_i(\tilde{x}_{\mathcal{N}_i})) \\
&\leq \lambda \sum_{i=1}^{N} w_i ||x^i - \tilde{x}_{\mathcal{N}_i}||_1 \\
&\leq \lambda \sum_{i=1}^{N} w_i \sum_{j \in \mathcal{N}_i} \sum_{d=1}^{D} |x^i_{j,d} - \tilde{x}_{j,d}| = \Psi(\tilde{x})
\end{aligned}
\tag{13}
$$

by expliciting the L_1 norm over the D dimensions of C. By rearranging the indices and splitting the absolute values, we have:

$$
\begin{aligned}
\Psi(\tilde{x}) &= \lambda \sum_{d=1}^{D} \sum_{j=1}^{N} \sum_{i \in \mathcal{N}_j} w_i |x^i_{j,d} - \tilde{x}_{j,d}| \\
&= \lambda \sum_{d=1}^{D} \sum_{j=1}^{N} \left(\sum_{\substack{i \in \mathcal{N}_j \\ x^i_{j,d} < \tilde{x}_{j,d}}} w_i (\tilde{x}_{j,d} - x^i_{j,d}) + \sum_{\substack{i \in \mathcal{N}_j \\ x^i_{j,d} \geq \tilde{x}_{j,d}}} w_i (x^i_{j,d} - \tilde{x}_{j,d}) \right)
\end{aligned}
\tag{14}
$$

We want to minimize $\Psi(\tilde{x})$. This is equivalent to finding $\tilde{x}$ so that $\nabla\Psi$ is 0, which can be written as, $\forall (j,d) \in [\![1, N]\!] \times [\![1, D]\!]$:

$$
\frac{\partial \Psi}{\partial \tilde{x}_{j,d}} = \lambda \left(\sum_{\substack{i \in \mathcal{N}_j \\ x^i_{j,d} < \tilde{x}_{j,d}}} w_i - \sum_{\substack{i \in \mathcal{N}_j \\ x^i_{j,d} \geq \tilde{x}_{j,d}}} w_i \right) = 0 \tag{15}
$$

The only value of $\tilde{x}_{j,d}$ which ensures this relation is the weighted median of the sample $\{x^i_{j,d}\}_{i \in \mathcal{N}_j}$, provided that the weights cancel out. It remains to verify that this critical point is a minimum by considering the coefficients of the Hessian matrix. To this aim, we can rewrite the partial derivative of Ψ:

$$\frac{\partial \Psi}{\partial \tilde{x}_{j,d}} = -\lambda \sum_{i \in \mathcal{N}_j} w_i \text{sign}\left(x^i_{j,d} - \tilde{x}_{j,d}\right)$$
$$= -\lambda \sum_{i \in \mathcal{N}_j} w_i \left(2H_{\frac{1}{2}}\left(x^i_{j,d} - \tilde{x}_{j,d}\right) - 1\right)$$

with $H_{\frac{1}{2}}$ the Heaviside function. Then, we obtain the Hessian matrix coefficients:

$$\frac{\partial \Psi}{\partial \tilde{x}_{j,d} \partial \tilde{x}_{j',d'}} = \begin{cases} 0 & \text{if } (j,d) \neq (j',d'), \\ 2\lambda \sum_{i \in \mathcal{N}_j} w_i \delta\left(x^i_{j,d} - \tilde{x}_{j,d}\right) & \text{otherwise.} \end{cases} \tag{16}$$

with δ the Dirac impulsion. These coefficients define a positive definite matrix if $\tilde{x}_{j,d} \in \{x^i_j\}_{i \in \mathcal{N}_j}$, which is necessarily the case if $\tilde{x}_{j,d}$ is a weighted median of the sample. Note that we must not create new data: if two elements are eligible to be the weighted median, we must select one of them and not taking the average between the two.

Since the weighted marginal median of the prescriptions minimizes the upper bound of $R^* - \tilde{R}$, it is a maximin optimum for $\tilde{R} = \sum_{i=1}^N w_i R_i(\tilde{x}_{\mathcal{N}_i})$. □

Since, by definition of the local reward functions, $\sum_{i=1}^N R_i(x) = R(x)$ (see Eq. 8), we can apply Theorem 1 with, $\forall i \in [\![1, N]\!], w_i = \frac{1}{N}$ and state that taking the marginal median of the prescriptions is a good way to reach high values of the global reward R.

3.5. Algorithm and complexity

Algorithm 1 summarizes the main steps of our proposed strategy `INSPIRE` run on each AP of the WLANs.

Contrary to what one might think, the most resource intensive operation in Algorithm 1 is not the inversion of the $\Sigma(X_t, X_t)$ matrix. Since, at step t, we already know the Cholesky decomposition of $\Sigma(X_{t-1}, X_{t-1}) = LL^T$, the Cholesky decomposition of $\Sigma(X_t, X_t)$ is easily obtained. The most resource-intensive operation is the maximization of the acquisition function through gradient ascent. Since this requires computing many matrix-vector multiplications for at most m steps, the computational complexity of Algorithm 1 at time t is $O(mt^2)$.

Algorithm 1 `INSPIRE` run at each AP i

Input: subset $\mathcal{N}_i$ of APs

1: Initialize the Gaussian Process $\mathcal{GP}_i$
2: **while true do**
3: Find a prescription $x^i = \arg\max_{x \in C^{|\mathcal{N}_i|}} EI_i^t(x)$ by gradient ascent
4: Broadcast x^i to APs in $\mathcal{N}_i$
5: Receive the prescriptions x_i^j from AP $j, j \neq i, j \in \mathcal{N}_i$
6: Compute the consensus $\tilde{x}_i^{t+1}$ with Equation 12
7: Test x_i^{t+1} for Δt seconds and compute its selfish reward $R_{\{i\}}$ with Equation 5 applied only to AP i
8: Broadcast $R_{\{i\}}$, $|\mathcal{N}_i|$ and $\tilde{x}_i^{t+1}$ to APs in $\mathcal{N}_i$
9: Receive $R_{\{j\}}$, $|\mathcal{N}_j|$ and $\tilde{x}_j^{t+1}$ from AP $j, j \neq i, j \in \mathcal{N}_i$
10: Compute the local reward R_i with Equation 6 and the local configuration $\tilde{x}_{\mathcal{N}_i}^{t+1}$
11: Add the pattern $\left(\tilde{x}_{\mathcal{N}_i}^{t+1}, R_i\right)$ to $\mathcal{GP}_i$
12: **end while**

It is worth noting that the dimensionality of the problem (*i.e.*, $\dim(C^{|\mathcal{N}_i|}) = |\mathcal{N}_i| \dim C$) does not appear in the expression of the asymptotic computational complexity of `INSPIRE`. This interesting property results from the use of a kernel function by GPs to compare WLANs configurations. This gives `INSPIRE` the ability to handle arbitrarily dense WLANs, or to optimize more parameters than just `TX_PWR` and `OBSS_PD`, without taking a hefty toll on its execution time. In fact, the real burden to the execution time of `INSPIRE` is the optimization step t. This compels us to bound the size of X_t and to find a balance between the amount of collected data on the WLANs' performance and configuration and a quick execution time.

In practice, to limit the computational burden, we can apply a windowing method (*e.g.* sliding window) to bound the size of X_t and so the computational complexity of `INSPIRE`. We denote this new version by `INSPIRE_LIM` and we evaluate the impact of limiting the size of the data set on the empirical performance of the proposed solution in the following section.

4. Performance Evaluation

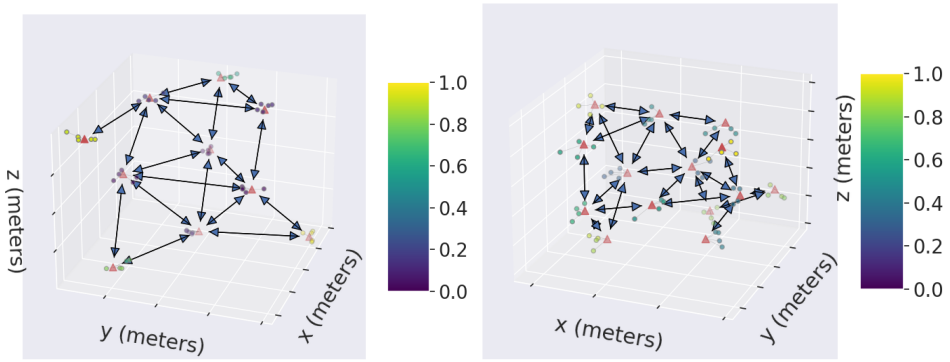
4.1. Experimental settings

To evaluate the ability of `INSPIRE` at improving the spatial reuse of a radio channel through the configuration of the `TX_PWR` and `OBSS_PD` parameters, we consider two distinct scenarios.

The first scenario is inspired by the WLAN deployment of Cisco in their offices in San Francisco. In [29], Cisco provides the location of 60 APs that together deliver wireless connectivity to their employees on a floor. To account for the WLANs' activity from other floors, we consider a three-floor building and we replicate on each floor the same arrangement of APs as in Cisco's offices. This leads us to a total number of 180 APs spanned over three floors. Assuming 18 independent radio channels, we run a radio channel allocation algorithm to determine the radio channel used by each AP. For our first scenario, we consider the subgraph resulting from the channel allocation with the highest density. We use **T1** to refer to this topology (*i.e.*, arrangements of APs and STAs), which is illustrated in Figure 4a. **T1** exhibits a total of 10 APs and we associate a number of 5 STAs per AP.

The second scenario addresses the case of many single-AP WLANs deployed and operated independently in a relatively limited area. This is typically the case in housing units where each apartment is equipped with its own AP so that the APs are often only a few meters away from a number of others. More specifically, we consider a nine-story building with 216 apartments of 25 m^2 each. We randomly position an AP within each apartment as well as 4 STAs per AP. Then, similarly to the first scenario, we apply a radio channel allocation algorithm given a total of 18 radio channels, to obtain the topology of interest denoted by **T2**. Note that **T2** consists of 14 APs and 56 STAs. Figure 4b depicts the topology **T2**.

For each scenario, we consider heavily loaded conditions. APs attempt to transmit frames to each of their associated STAs at a rate of 50 Mbps while the latter attempt to send their frames to the AP at a lower rate of 3.33 Mbps. These assumptions are in line with the downstream traffic largely exceeding the upstream traffic in WLANs. Given the speed of wireless links in 802.11ax, the buffers of the APs will always be full of frames waiting to be sent. More generally, considering APs in saturation represents undoubtedly the most difficult case when dealing with the spatial reuse of a radio channel. Therefore, if `INSPIRE` manages to significantly improve the WLANs' performance under these circumstances, then it can only do better under normal



(a) Topology **T1**.

(b) Topology **T2**.

Figure 4: The two considered topologies. APs are shown as red triangles, connected by a arrow if they lie in each other's communication range. Associated STAs are shown as dots colored according to their throughputs: warm, yellowish colors indicate that the STA has enough throughput most of the time while, on the contrary, cool, blueish colors indicate that the STA has mostly not enough throughput under the default configuration of 802.11: 20 dBm for `TX_PWR` and -82 dBm for `OBSS_PD`.

conditions.

To better appraise the quality of `INSPIRE`, we also consider a "control strategy", several state-of-the-art solutions which were discussed in Section 2 as well as a version of `INSPIRE` that caps the amount of collected data with a windowing method. All the considered approaches are briefly summarized below:

- `DEFAULT`: Every AP keeps its default configuration for the `TX_PWR` and `OBSS_PD` parameters (*i.e.*, $(-82, 20)$ dBm);
- `WCNC'15`: Each AP implements a simple distributed algorithm to dynamically update its `OBSS_PD` parameter [14];
- `JNCA'19`: Each AP solves a MAB problem using Thompson sampling to dynamically update their `TX_PWR` and `OBSS_PD` parameters [4];
- `MSWiM'21`: Similar to `JNCA'19`, except that the sampling of new configurations is performed through a multivariate Gaussian mixture, and that the solution is centralized [12];
- `ADHOC'23`: Similar to `MSWiM'21`, except that the sampling of new configurations is performed through a mixture of hyperspheres [12];

- `INSPIRE_LIM`: Similar to `INSPIRE`, except that only the last 50 observations are considered to make predictions.

We implemented `INSPIRE` (based on the open-source Gaussian process library LibGP [30]) as well as the six strategies described above in the open-source network simulator ns-3 [31]. ns-3 is a well-established realistic discrete-event simulator that implements most of the network protocols involved in the WLANs communication from the Physical up to the Application layer. We report in Table 2 the simulation parameters used in the rest of this section. Unlike previous works (e.g., [12, 14, 11, 16, 4])) with the exception of [15], our simulations incorporate the mechanism of rate adaptation that let APs and STAs dynamically vary the speed of their wireless links (through the use of different Modulation Coding Scheme (MCS)) in response to the quality of the received signal. This is particularly important for the sake of our study since changing the value of `TX_PWR` necessarily affects the quality of the received signal and thus the MCS. Since our simulated WLANs take place in buildings, we choose an appropriate path loss by combining the models `ItuR1238` and `InternalWallsLoss`, both implemented by ns-3. With these propagation models, the signal is decreased by an additional attenuation coefficient each time it goes through a floor or a wall. The attenuation coefficients are respectively -4 dBm (which is the default value in `ItuR1238`) and -8 dBm.

We instrumented ns-3 to collect and compute a number of performance metrics. At the end of each iteration, the quality of the spatial reuse is assessed with Equation 5, although distributed strategies internally use the local reward defined in Equation 6. Then, we compute the classical metric to analyze the efficiency of a strategy at dealing with a MAB problem: (i) The cumulative regret (with Equation 10 using a normalized version of the global reward in Equation 5). We also collect the following performance metrics to reflect the effect of each strategy on the behavior of the WLANs and of their STAs: (ii) The execution time of the solution in seconds, on a logarithmic scale, (iii) The number of starving STAs, which we define as STAs experiencing a very low throughput (namely, less than 10% of their attainable throughput) and (iv) The cumulated throughput, which simply sums all STAs' throughput.

Each simulation lasts 30 seconds and we replicated them independently 22 times to obtain and visualize their first, second, and third quartiles. When the quartiles of a performance metric vary too much within a single simu-

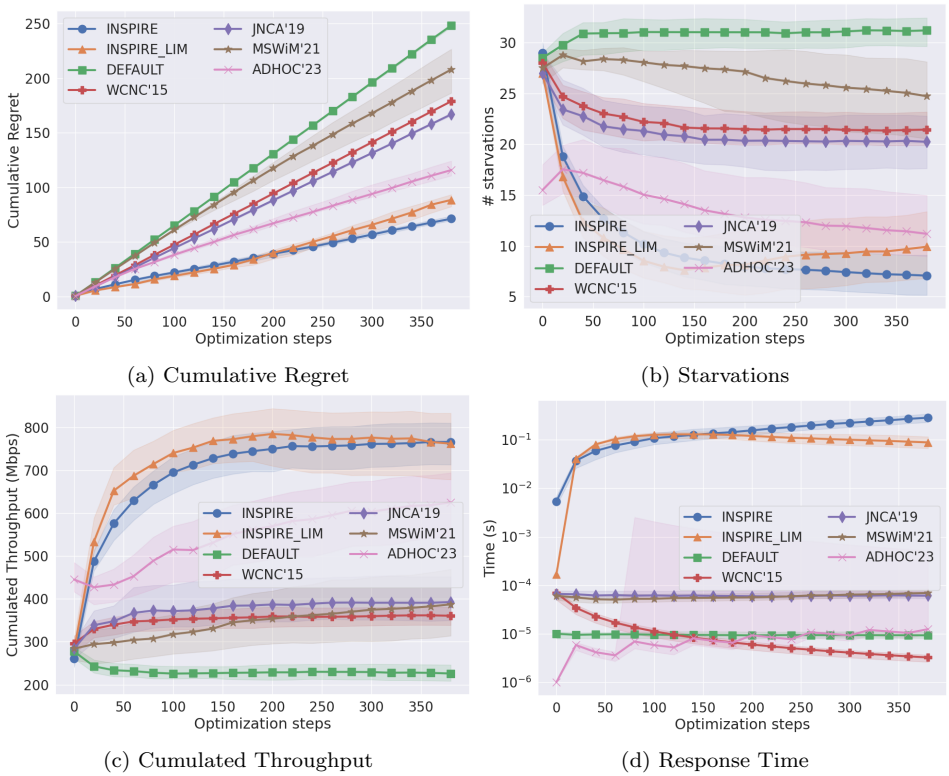
Table 2: ns-3 parameters.

Parameter	**Value**
ns-3 version	3.31
Number of repetitions	22
Simulation duration	30 s
Duration of an iteration (Δt)	75 ms
Packet size	1,464 bytes
Downlink traffic	50.0 Mbps
Uplink traffic	3.33 Mbps
Channel size	20 MHz
Frequency band	5 GHz
A-MDPU Aggregation	4
Path loss	`HybridBuildings` (`ItuR1238` + `InternalWallsLoss`)
Wi-Fi Manager	`IdealWifiManager`

lation, we apply an exponential moving average (with $\alpha = 0.04$) to extract the underlying trends of the quartiles sequences. The metrics are collected throughout the whole duration of the simulation. At the end of each iteration, we compute all the performance metrics and then we refer to the current strategy to decide what will be the next configuration of the WLANs. Since an iteration lasts $\Delta t = 75$ ms and a simulation lasts 30 seconds, the quality of each solution is assessed over 400 iterations.

4.2. Numerical results

Figure 5 illustrates the performance metrics delivered by the ns-3 simulator for each strategy in the case of topology **T1**. The cumulative regret, represented in Figure 5a, indicates which strategy has performed the best at any time of the simulation. `INSPIRE` is found to be the most efficient strategy, reducing the cumulative regret by 70% more than `DEFAULT` and by over 45% than `ADHOC'23`, which happens to be the most efficient state-of-the-art strategy. We now look at the other performance metrics to better understand how much `INSPIRE` is able to improve the behavior of the WLAN and of its STAs. Taking `DEFAULT` as a baseline, Figure 5b shows that `INSPIRE` reduces the number of STAs in starvation by 80% while Figure 5c demonstrates that our proposed solution manages to increase the cumulated throughput (+400%). Considering Figure 5d, which reports the response time of the



(a) Cumulative Regret

(b) Starvations

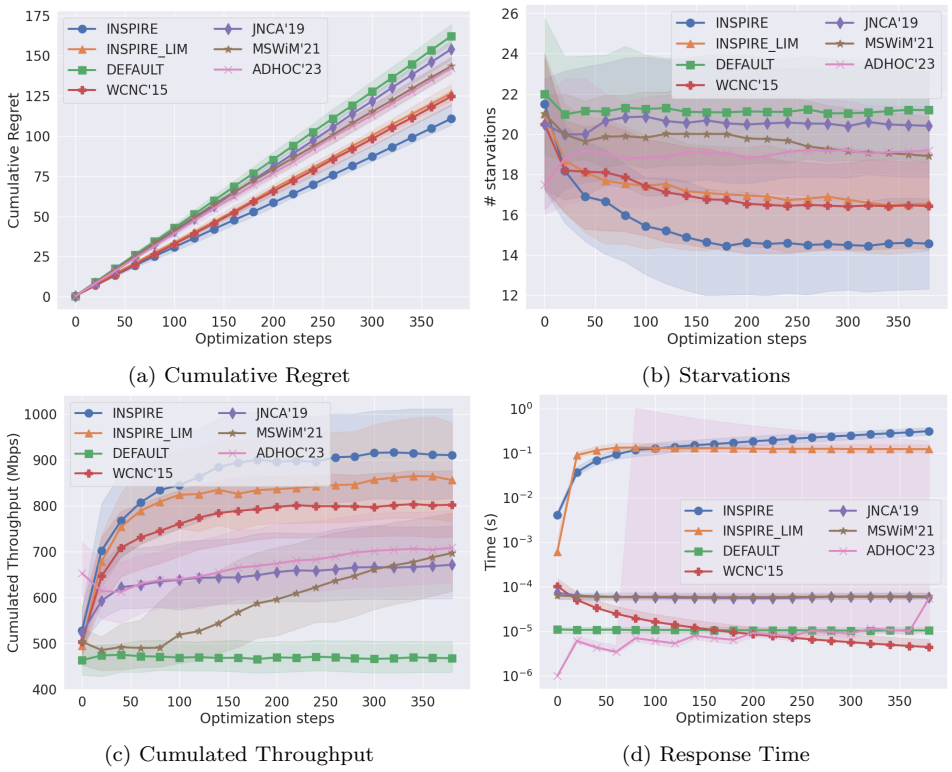
(c) Cumulated Throughput

(d) Response Time

Figure 5: Performance metrics delivered on topology **T1** by each strategy.

considered strategies, we observe that the increase in the performance metrics comes at a computational cost. However, we argue that a response time of around a tenth of a second remains acceptable when making decisions on WLANs. Comparing the response time of `INSPIRE` and `INSPIRE_LIM`, we see that capping the size of the dataset also caps the response time, which is in line with the explanations detailed in Section 3.5. Overall, although its performance are slightly below those of `INSPIRE`, `INSPIRE_LIM` constitutes a significant improvement over the considered state-of-the-art strategies. Hence, it is a good alternative to `INSPIRE` if one accepts to trade off some empirical performance for a constant response time.

We now turn to the case of topology **T2**. First, we observe in Figure 6a that among the six considered strategies, `INSPIRE` is the one that manages to decrease the most the cumulative regret with a decline of about 36% compared to the `DEFAULT` configuration at the end of the simulation. The proposed solution also outperforms `WCNC'15`, which is found to be the



(a) Cumulative Regret

(b) Starvations

(c) Cumulated Throughput

(d) Response Time

Figure 6: Performance metrics delivered on topology **T2** by each strategy.

best state-of-the-art strategy on this topology, by a margin of 14%. Looking at the performance of WLANs and of their STAs, Figure 6b shows that `INSPIRE` is able to limit the number of STAs starving from throughput by a degree of 36% when compared to the `DEFAULT` configuration. Similarly, the cumulated throughput of STAs have their value increased by 28% and nearly doubled with `INSPIRE` (see Figure 6c). Although the dimensionality of a configuration is larger for **T2** than for **T1** (28 against 20), the response time of `INSPIRE` is virtually equivalent for the two topologies and remains around a tenth of a second. This is because the additional time required to compare higher-dimensional configurations is marginal compared to the additional time required to make inferences on a larger dataset (see Section 3.5). Eventually, note that `INSPIRE_LIM` is slightly outperformed by `INSPIRE`, but remains roughly equivalent to the best considered state-of-the-art strategy.

Overall, through the study of topologies **T1** and **T2**, `INSPIRE` demonstrates its superiority over the other state-of-the-art strategies. Its caped

version, INSPIRE_LIM, also demonstrates its empirical competitiveness. The significant improvements brought by our proposed solution on all performance metrics are permanently obtained after 100 iterations only (corresponding to 7.5 seconds of simulated time). In other words, in less than 10 seconds, INSPIRE manages to significantly improve the behavior of the WLANs and of the associated STAs thanks to a better spatial reuse of the radio channel. This efficiency in searching and finding an adequate configuration of the TX_PWR and OBSS_PD parameters at each AP of the WLANs mostly results from the distributed, altruistic use of Gaussian processes which we further discuss in the next section.

5. Discussion

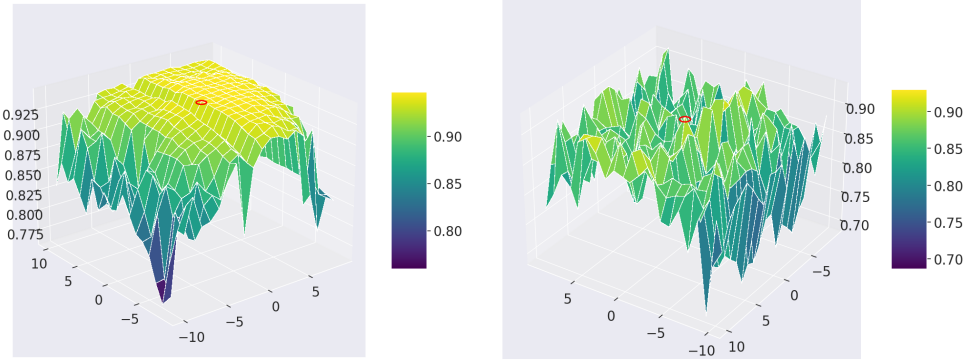
5.1. Seemingly similar problems with vastly different complexity

The topologies **T1** and **T2** may look similar, but INSPIRE performed differently on each of them. By the end of the optimization process (*i.e.*, 400 steps), the performance metrics for **T1** were improved by at least 70% from their initial values under the DEFAULT configuration, and only 7 STAs (representing 14% of the STAs) were still starving from throughput. In case of **T2**, the progress was less with 14 STAs (representing 25% of the STAs) remaining in starvation. This difference results from the location of STAs relatively to the APs. Looking at Figure 4, it appears that STAs in **T2** are further from their associated AP than the ones in **T1**. As a consequence, STAs are also closer to a concurrent AP in **T2** than in **T1**. Indeed, while STAs in **T1** are on average 10 times closer to their associated AP than to a concurrent AP, this ratio drops to an average value of 4 for STAs on **T2**. With STAs closer to concurrent APs, the spatial reuse problem becomes more difficult. As a matter of fact, to reach its associated STA, the AP must transmit at a greater power, increasing its chance to cause interference to the surrounding APs. Similarly, STAs that are far away from their AP are significantly affected by the transmissions of concurrent APs.

To verify that **T2** constitutes a more complex example than **T1**, we examine the shape of the reward function in both cases. Because of the high dimensionality the reward function and the lack of a closed-form expression, we resort to a slicing technique to provide a visualization of the reward function in Equation 5. We postpone to Appendix A the details of this slicing. Figure 7 illustrates the obtained random slices in case of **T1** and **T2**. Figure 7a suggests a relatively smooth reward function in **T1**. On the other

hand, the reward function in the case of **T2** is much more erratic, featuring a lot of local maxima as shown by Figure 7b.

Interestingly, Figure 7a shows that `INSPIRE` succeeded to find a configuration that is maximal in this (random) slice for the case of **T1**. We also notice that many configurations of equivalent efficiency exist, which also tends to ease the search for an adequate configuration. Conversely, in the case of **T2**, `INSPIRE` does not find the best configuration since the slice of Figure 7b shows that, at least, a 6% better configuration exist. Nonetheless, `INSPIRE` was able to find an efficient configuration.



(a) Slice for **T1**, with the best solution found by `INSPIRE` at (0, 0) and two scaled random unit vectors. The maximum of the slice is shown with a red circle.

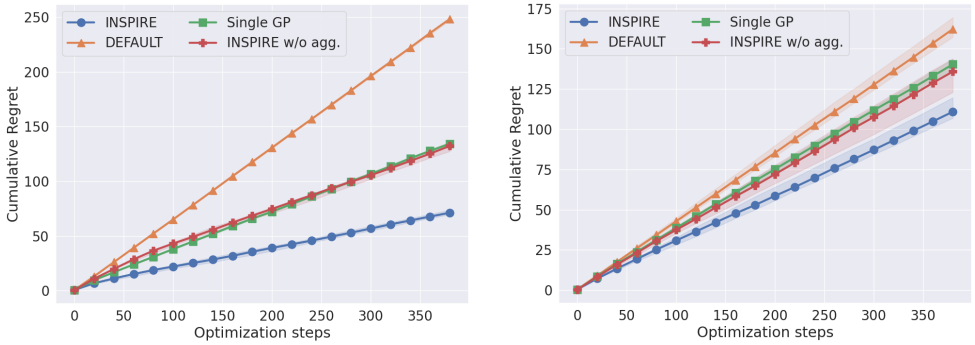
(b) Slice for **T2**, with the best solution found by `INSPIRE` at (0, 0) and two scaled random unit vectors. The maximum of the slice is shown with a red circle.

Figure 7: Random slices of the global reward function for the topologies **T1** and **T2**.

5.2. Benefits of distributed prescriptions

Eventually, to justify our choice of letting APs exploit only local information and prescribe network configurations to their surrounding APs, we consider alternative versions of `INSPIRE`: (i) `GPs w/o agg.` where each AP keeps using the local, altruistic reward of Equation 6 but does not aggregate local prescriptions, and prescribes only for its own configuration and (ii) `Single GP`, a centralized version of `INSPIRE` where a single GP has a complete knowledge of the WLANs and decides on the configuration of every AP. We compare these alternative strategies with `INSPIRE` and the `DEFAULT` strategy by considering their cumulative regret on topologies **T1** and **T2** in Figure 8. Figure 8a shows that, on **T1**, the alternative strategies have a cumulative regret 46% lower than `DEFAULT`. However, `INSPIRE` manages to

reduce its cumulative regret by an extra 25%. Despite the greater complexity of the function in **T2**, this extra reduction factor persists but at a value of 13% as shown in Figure 8b. Given the significant gap between `INSPIRE` and `GPs w/o agg.`, it is clear that prescribing for surrounding APs and aggregating those prescriptions leads to a more altruistic behaviour, which in turn brings additional benefits at the scale of the WLANs. More surprisingly, `INSPIRE` outperforms its centralized counterpart `Single GP`. At first glance, this is counter-intuitive since `Single GP` has a complete knowledge and control over the APs of the WLANs. However, `Single GP` involves a single agent to optimize a function of high complexity, which has to deal with $K|C|$ variables. Decentralization breaks down this task into simpler local optimization problems of lower dimension. With `INSPIRE`, each AP k only deals with $|\mathcal{N}_k||C|$ variables, which is significantly less than $K|C|$ for large WLANs. Overall, APs run by `INSPIRE` solve simpler optimization problems and behave altruistically by ensuring a consensus with surrounding APs. By doing so, they manage to improve the spatial reuse of the radio channel at the scale of the WLANs.



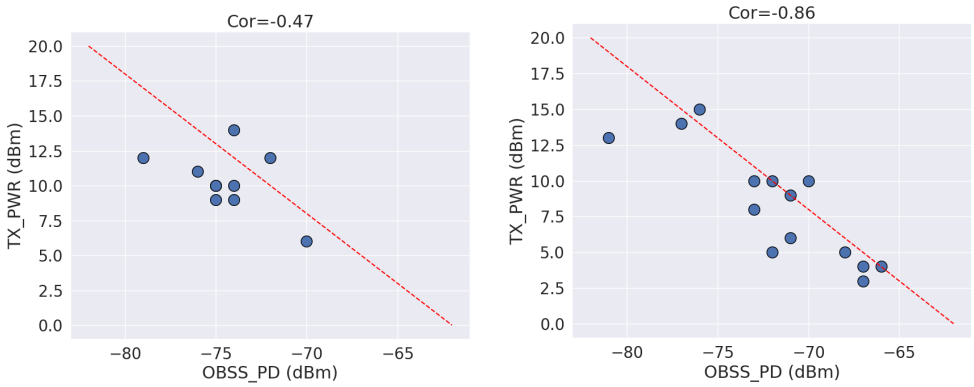
(a) Cumulative regret on topology **T1**.

(b) Cumulative regret on topology **T2**.

Figure 8: Comparison of the cumulative regrets of the different versions of `INSPIRE` on **T1** and **T2**.

5.3. Interpretation of the recommended configurations

In this section, we provide insights about the configurations (i.e., `TX_PWR` and `OBSS_PD`) recommended by `INSPIRE` on **T1** and **T2**, trying to bridge the gap between the found values for `TX_PWR` and `OBSS_PD` and graph-based metrics related to the WLAN topology.



(a) Recommended configuration for **T1**.

(b) Recommended configuration for **T2**.

Figure 9: Recommended configurations according to `INSPIRE` for each AP in topologies **T1** and **T2**. Each dot represents an AP. An AP violates the constraint described in Equation 17 when it is located above the dashed red line.

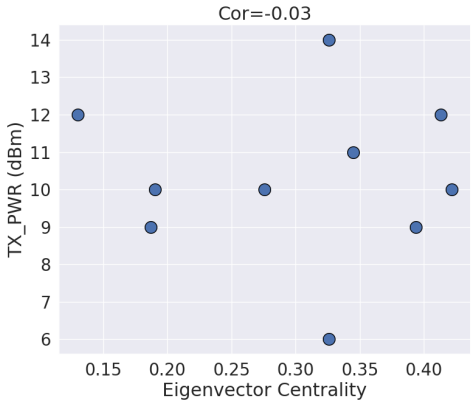
We start by providing a visualization for the configurations found by `INSPIRE` for each AP in **T1** and **T2** in Figure 9. We observe that, for a large majority of APs, the larger `OBSS_PD`, the lower `TX_PWR` (with a correlation coefficient of -0.47 and -0.86 for **T1** and **T2**, respectively). Interestingly, this is somehow in agreement with a constraint in the 802.11ax amendment [3] that enforces the following relation.

$$\texttt{OBSS\_PD} \leq \max(-82, \min(-62, -82 + (20 - \texttt{TX\_PWR}))). \tag{17}$$

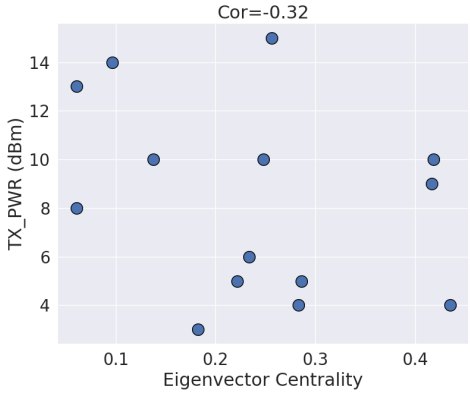
The boundary of the constraint described in Equation 17 is depicted with a red dashed line on Figure 9. Interestingly, although Equation 17 was not included as a constraint in our optimization task, the vast majority of APs configured with the configurations recommended by `INSPIRE` satisfy it.

We now study the relationships between the centrality of an AP in WLAN topology and its recommended configuration for `TX_PWR` and `OBSS_PD`. We consider the *eigenvector centrality* [32], defined as the normalized eigenvector associated with the largest eigenvalue of the adjacency matrix of the conflict graph of the considered topology (displayed in Figure 4). An AP with an eigenvector centrality close to 1 is in conflict with a lot of APs, which are themselves in conflict with many APs. Conversely, an AP with an eigenvector centrality close to 0 means little to no conflicts with APs which are themselves isolated.

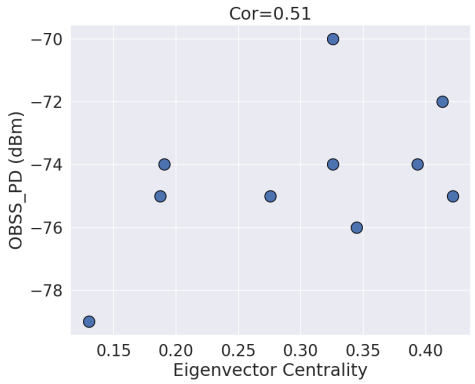
Figure 10 reports the results. We start by considering the `TX_PWR` of the



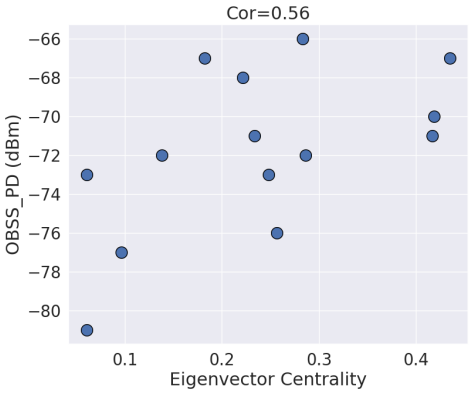
(a) `TX_PWR` for each AP w.r.t. its eigenvector centrality in **T1**.



(b) `TX_PWR` for each AP w.r.t. its eigenvector centrality in **T2**.



(c) `OBSS_PD` for each AP w.r.t. its eigenvector centrality in **T1**.



(d) `OBSS_PD` for each AP w.r.t. its eigenvector centrality in **T2**.

Figure 10: Visualization of the recommended configurations w.r.t. the eigenvector centrality of the APs in topologies **T1** and **T2**. Each dot represents an AP.

APs with respect to their eigenvector centrality for **T1** and **T2**, in Figures 10a and 10b respectively. It appears that the more an AP is central, the lower its `TX_PWR`. However, the low magnitude of the correlation coefficients (-0.03 and -0.32 for **T1** and **T2**, respectively) indicates that there is maybe more to understand about the found values for `TX_PWR` of the APs. Conversely, regarding the `OBSS_PD` reported in Figures 10c and 10d, a stronger relationship can be observed with the eigenvector centrality of an AP. There is a clear positive correlation between the two quantities (0.51 and 0.56 for **T1** and **T2**, respectively), meaning that the more an AP is central, the larger its `OBSS_PD`.

Overall, these results confirm that the optimization of the AP configuration is not an easy problem since no systematic, strong (*i.e.* with a magnitude close to 1) correlations between the optimal values of TX_PWR, OBSS_PD and the considered graph properties of the WLAN topology seem to exist. Note that we considered other definitions of centrality, which ended up exhibiting lower correlations with the recommended configurations. To conclude, as suggested by Figure 10, our results indicate that, in general, the more central an AP, the higher its OBSS_PD and, to a lesser extent, the lower its TX_PWR.

6. Conclusions

In this work, we have presented INSPIRE, a bayesian optimization method to improve the spatial reuse of radio channels in WLANs by configuring two parameters of APs: the transmission power (TX_PWR) and the sensitivity threshold (OBSS_PD), which can be dynamically configured with the latest Wi-Fi amendments. To address the difficult problem of sharing efficiently and fairly the resource of a radio channel, INSPIRE works as a distributed solution where each AP solves a local Multi-Armed Bandit problem with the help of information and actions limited to its surrounding APs (*i.e.* within its communication range). The development of the solution includes (i) an intuitive quantification (based on STAs throughputs) of the "goodness" of a configuration of TX_PWR and OBSS_PD for concurrent APs in WLANs, both at local and global scales, (ii) the use of an acquisition function and Gaussian processes to find local configurations that maximize approximations of local reward functions, and (iii) an altruistic behavior facilitated by prescriptions to surrounding APs along with a consensus method which aggregates the prescriptions of surrounding APs for the "greater good" of the WLANs.

We show that INSPIRE can be easily extended to a version working with a limited number of observations (through the use of a sliding window) to limit its computational complexity while only slightly affecting its ability to find an efficient AP configuration.

INSPIRE has been evaluated and compared with other state-of-the-art strategies addressing the same problem, using the open-source network simulator ns-3 that implements all the layers of the network stack. The different strategies were compared on two examples inspired by real-life deployments of dense WLANs in both professional and domestic environments, representing two optimization tasks of different complexities. INSPIRE was found to outperform other state-of-the-art strategies by significantly reducing the

number of STAs in starvation and increasing the cumulated throughput of the WLANs in only a few seconds. Interestingly, it has also demonstrated significantly better empirical performance than its centralized counterpart, which brings another practical example of the superiority of a distributed approach over a centralized approach when addressing a complex issue.

As future work, we plan to assess the quality of `INSPIRE` for a specific class of WLANs where STAs are mobile (e.g., customers in a shopping mall). Another natural follow-up would be to experiment `INSPIRE` with real material on a testbed.

Acknowledgements

This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR).

References

[1] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022 white paper. *Cisco Internet Report*, 17:13, 2019.

[2] Harald Haas. Lifi is a paradigm-shifting 5g technology. *Reviews in Physics*, 3:26–31, 2018.

[3] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Enhancements for high-efficiency wlan. *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, pages 1–767, 2021.

[4] Francesc Wilhelmi, Sergio Barrachina-Muñoz, Boris Bellalta, Cristina Cano, Anders Jonsson, and Gergely Neu. Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in wlans. *Journal of Network and Computer Applications*, 127:26–42, 2019.

[5] Anthony Bardou and Thomas Begin. Inspire: Distributed bayesian optimization for improving spatial reuse in dense wlans. In *Proceedings*

of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems, MSWiM '22, page 133–142, New York, NY, USA, 2022. Association for Computing Machinery.

[6] IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, 2021.

[7] Francesc Wilhelmi, Sergio Barrachina-Muñoz, Cristina Cano, Ioannis Selinis, and Boris Bellalta. Spatial reuse in ieee 802.11ax wlans. *Computer Communications*, 170:65–83, 2021.

[8] Jing Zhu, Xingang Guo, L. Lily Yang, W. Steven Conner, Sumit Roy, and Mousumi M. Hazra. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. *IEEE Wireless Communications and Networking Conference (WCNC'04).*, 2004.

[9] Youngsoo Kim, Jeonggyun Yu, and Sunghyun Choi. SP-TPC: a self-protective energy efficient communication strategy for IEEE 802.11 WLANs. In *IEEE Vehicular Technology Conference (VTC'04)*, 2004.

[10] Shuwei Qiu, Xiaowen Chu, Yiu-Wing Leung, and Joseph Kee-Yin Ng. Joint access point placement and power-channel-resource-unit assignment for 802.11ax-based dense wifi with qos requirements. In *IEEE Conference on Computer Communications (INFOCOM'20).*, pages 2569–2578, 2020.

[11] Elif Ak and Berk Canberk. Fsc: Two-scale ai-driven fair sensitivity control for 802.11ax networks. In *IEEE Global Communications Conference (GLOBECOM'20).*, pages 1–6, 2020.

[12] Anthony Bardou, Thomas Begin, and Anthony Busson. Improving the spatial reuse in ieee 802.11ax wlans: A multi-armed bandit approach. In *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'21)*, 11 2021.

[13] Anthony Bardou, Thomas Begin, and Anthony Busson. Mitigating starvation in dense wlans: A multi-armed bandit solution. *Ad Hoc Networks*, 138:103015, 2023.

[14] Muhammad Shahwaiz Afaqui, Eduard Garcia-Villegas, Elena Lopez-Aguilera, Graham Smith, and Daniel Camps. Evaluation of dynamic sensitivity control algorithm for ieee 802.11ax. In *IEEE Wireless Communications and Networking Conference (WCNC'15).*, pages 1060–1065, 2015.

[15] Hyunjoong Lee, Hyung-Sin Kim, and Saewoong Bahk. Lsr: Link-aware spatial reuse in ieee 802.11ax wlans. In *IEEE Wireless Communications and Networking Conference (WCNC'21)*, pages 1–6, 2021.

[16] Francesc Wilhelmi, Cristina Cano, Gergely Neu, Boris Bellalta, Anders Jonsson, and Sergio Barrachina-Muñoz. Collaborative spatial reuse in wireless networks via selfish multi-armed bandits. *Ad Hoc Networks*, 88:129–141, 2019.

[17] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[18] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

[19] Christopher K. I. Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Conference on Neural Information Processing Systems (NeurIPS'95)*, 1995.

[20] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[21] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[22] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[23] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[24] Daniel Lizotte, Tao Wang, Michael Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 944–949, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[25] Javier González, Joseph Longworth, David C. James, and Neil D. Lawrence. Bayesian optimization for synthetic gene design. In *NIPS Workshop on Bayesian Optimization in Academia and Industry*, 2014.

[26] Marc G. Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of Machine Learning Research*, 2, 3 2002.

[27] Shanti S. Gupta and Klaus J. Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2):229–244, 1996.

[28] Garrett Hardin. The tragedy of the commons. *Journal of Natural Resources Policy Research*, 1(3):243–253, 2009.

[29] Cisco. High density wi-fi deployments. `https://documentation.meraki.com/Architectures_and_Best_Practices/Cisco_Meraki_Best_Practice_Design/Best_Practice_Design_-_MR_Wireless/High_Density_Wi-Fi_Deployments`, 2020. Accessed: 2022-01-21.

[30] Manuel Blum. Libgp. GitHub, https://github.com/mblum/libgp, 2016.

[31] The Network Simulator ns-3. `https://www.nsnam.org/`, 2020. Accessed: 2022-01-21.

[32] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

Appendix A. Building of the reward slices

We detail in this appendix how we build the random slices of the reward function showed in Figure 7. For the sake of completeness, we also provide the numerical values of the unit vectors of our (randomly) selected bases.

For visualization purposes, we build a random vector basis in two dimensions. Considering the best configuration c^* found by `INSPIRE` on a given topology, we uniformly draw two random configurations $\hat{u}, \hat{v} \in C^K$ and use $(c^*, \hat{u}, \hat{v})$ as a vector basis. By plotting the reward of each configuration attainable through this basis (i.e., any valid configuration c that can be expressed as $c = c^* + x\hat{u} + y\hat{v}, (x, y) \in \mathbb{R}^2$), we obtain a 3d plot representing a random slice of the considered reward function, with the best configuration found at the origin of the plot.

Details of the vector basis for topology **T1** (Figure 7a) are as follows:

- c_1^* = (-73, 10, -74, 10, -73, 12, -77, 9, -76, 14, -72, 12, -72, 10, -74, 10, -74, 13, -72, 10)
- $\hat{u_1}$ = (-.29, .53, -.41, .2, -.08, .78, -.16, .33, -.61, .74, -.53, .12, -.24, .78, -.57, .61, -.49, .69, -.16, .82)
- $\hat{v_1}$ = (-.09, .03, -.5, .32, -.58, .53, -.53, .18, 0, .38, -.55, .55, -.47, .15, -.12, .58, -.18, .15, -.5, .38)

Corresponding values for **T2** in Figure 7b are:

- c_2^* = (-76, 9, -76, 11, -67, 3, -70, 8, -75, 9, -74, 8, -71, 5, -66, 6, -73, 9, -68, 10, -65, 3, -66, 12, -78, 6, -76, 16)
- $\hat{u_2}$ = (0, .21, -.06, .14, 0, .15, -.02, .14, -.23, .04, -.18, .18, -.16, .15, -.01, .05, -.11, .23, -.01, .08, -.23, .21, -.02, .11, -.19, .16, -.07, .04)
- $\hat{v_2}$ = (-.02, .15, -.38, 0, -.02, .13, -.42, .15, -.15, .23, -.19, .15, -.02, .25, -.42, .38, -.4, .38, -.31, .15, -.29, .08, -.25, .19, -.04, 0, -.4, .31)