# Online Learning for the Black-Box Optimization of Wireless Networks

Anthony Bardou



THESE

en vue de l'obtention du grade de Docteur, délivré par

l'ECOLE NORMALE SUPERIEURE DE LYON

**Ecole Doctorale** N° 512
Informatique et Mathématiques de Lyon (InfoMaths)

**Discipline** : Informatique

Soutenue publiquement le 07/09/2023, par :

**Anthony BARDOU**

---

# Online Learning for the Black-Box Optimization of Wireless Networks

Apprentissage séquentiel pour l'optimisation de boîtes noires dans les réseaux sans-fil

---

Devant le jury composé de :

| | | | |
|---|---|---|---|
| LOISEAU Patrick | Chargé de Recherche HDR | Inria/École Polytechnique/ENSAE | Rapporteur |
| NEGLIA Giovanni | Directeur de Recherche | Inria Côte d'Azur | Rapporteur |
| RALAIVOLA Liva | Professeur des Universités | Aix-Marseille Université/Critéo | Examinateur |
| VERNADE Claire | Chercheuse | Universität Tübingen | Examinatrice |
| BEGIN Thomas | Professeur des Universités | Université Claude Bernard Lyon I | Directeur de thèse |

Laboratoire d'Informatique du Parallélisme
École Normale Supérieure de Lyon
46 allée d'Italie
69007 Lyon

École doctorale n° 512
Bâtiment Blaise Pascal
7 avenue Jean Capelle
69100 Villeurbanne

# REMERCIEMENTS

Cette thèse est le fruit de trois années de travail, rendues possibles par de nombreuses personnes. Je veux ici remercier toutes celles et ceux qui m'ont accompagné et soutenu tout au long de cette aventure.

Mes premiers remerciements vont à mon directeur de thèse, Thomas Begin. À ton contact, j'ai beaucoup appris sur la production d'un travail de recherche. Je te suis profondément reconnaissant pour ton encadrement et tes innombrables conseils. Merci pour ton enthousiasme, pour ta rigueur, pour la confiance que tu m'as accordée et l'autonomie que tu m'as laissée, qui m'ont permis de m'approprier et de mener à bien ce travail de recherche. Enfin, merci pour ton engagement dans ce travail, qui n'aurait pas pu voir le jour sans toi.

Je souhaite également remercier Patrick Thiran. Merci pour ton accueil à l'EPFL et ton regard aiguisé sur mon travail. Ta rigueur et tes précieux conseils m'ont beaucoup apporté. J'ai hâte de continuer à travailler et à apprendre à ton contact après l'obtention de mon doctorat.

Je remercie vivement Anthony Busson et Jean-Marie Gorce, pour avoir apporté leur expertise à cette thèse au travers de nombreuses semaines de collaboration. Merci pour vos conseils, qui m'ont permis d'avancer avec rigueur et confiance dans ce travail.

J'exprime toute ma gratitude à Patrick Loiseau, Giovanni Neglia, Liva Ralaivola et Claire Vernade pour avoir accepté d'évaluer ce travail. Merci en particulier à Liva Ralaivola pour ses retours lors de mes comités de suivi, ainsi qu'à Patrick Loiseau et Giovanni Neglia pour avoir accepté de rapporter cette thèse.

Un immense merci aux membres des équipes Ockham et HoWNet, qui ont grandement contribué à faire de l'ENS de Lyon un environnement de travail riche et agréable. Merci à Antoine, Badr, Clément, Esther, Guillaume, Johann, Léon, Mathieu, Meriem, Théotime, Tung et Youssouph, ainsi qu'à tous ceux que je ne mentionne pas ici, pour votre bonne humeur, pour les rires et les riches discussions que nous avons eus autour d'un bon plat, d'un verre ou d'un café. J'adresse une pensée particulière, pleine d'amitié et de chaleur à Samir, mon cher co-bureau, et à ces centaines de journées passées ensemble. Merci pour ta présence complice tout au long de cette aventure.

Je remercie mes amis, et tout particulièrement les lyonnais : Aloïs, Hugo, Sami et Tristan, merci pour ces nombreux moments en votre compagnie qui m'ont permis de m'échapper quand le travail devenait trop pesant.

J'ai une pensée pleine de reconnaissance pour ma famille et ma belle-famille. Merci pour votre chaleur et les moments si précieux

que nous partageons. Je veux tout particulièrement remercier mes parents, pour leur soutien sans faille dans tout ce que j'entreprends.

Finalement, merci à toi Manon pour m'avoir accompagné chaque jour de ce voyage. Merci d'avoir partagé avec moi le soleil corse, les bouchons lyonnais et les paysages suisses. Merci de m'avoir permis de prendre du recul, parfois contre mon gré. C'est avant tout grâce à toi que ces dernières années ne se résument pas seulement à ces centaines de feuilles de papier noircies et à ces milliers d'heures passées sur mon ordinateur. Merci de me remplir continuellement le coeur de ta chaleur, de tes rires et de ta tendresse.

## ABSTRACT

Nowadays, wireless networks (WNs) are ubiquitous and have to increase the density of their deployment to meet our important connectivity needs. Paradoxically, this deteriorates their quality of service because they have to deal with a larger amount of interference. To manage this unprecedented complexity, wireless standards, such as Wi-Fi or 5G, are becoming more flexible by introducing new degrees of freedom. Unfortunately, the correct exploitation of these parameters is not trivial because it requires to quickly identify an efficient configuration in a high-dimensional space. Moreover, the wide variety of use cases of WNs makes it difficult to model their behaviors, and consequently to find an optimal configuration analytically.

In this thesis, we propose bandit and Bayesian optimization methods able to discover, through trial and error, an efficient configuration of WNs regardless of their deployments. These sequential learning methods, called "online", seek to optimize the performance of a WN by considering it as a black box. In this thesis, we illustrate the capabilities of the proposed methods on the challenging problem of spatial reuse optimization in Wi-Fi networks, and on the power control in 5G networks. Finally, we propose a new asymptotically optimal Bayesian optimization algorithm, able to optimize a high-dimensional black box function in a decentralized fashion. This last contribution could allow the implementation of more efficient protocols in WNs but also in other technological contexts.



## RÉSUMÉ EN FRANÇAIS

De nos jours, les réseaux sans fil (RSF) sont omniprésents et sont amenés, pour répondre à nos besoins importants de connectivité, à augmenter la densité de leurs déploiement et à supporter un grand nombre d'interférences. Pour gérer cette complexité inédite, les standards (Wi-Fi, 5G...) deviennent plus adaptables, en s'octroyant de nouveaux degrés de liberté. Malheureusement, l'exploitation correcte de ces paramètres n'est pas triviale car elle demande d'identifier rapidement une configuration efficace dans un espace de grande dimension. De plus, la grande diversité d'utilisation des RSF rend difficile leur modélisation, et par conséquent la découverte analytique d'une configuration optimale.

Dans cette thèse, nous proposons des méthodes issues des problèmes de bandits et de l'optimisation bayésienne, capables de découvrir, par essai-erreur, une configuration efficace des RSF quel que soit

le scénario dans lequel ils se trouvent. Ces méthodes d'apprentissage séquentiel, dites "en ligne", cherchent à optimiser les performance d'un RSF en le considérant comme une boîte noire. Au cours de la thèse, nous illustrons les capacités des méthodes proposées sur le problème difficile de la réutilisation spatiale dans les réseaux Wi-Fi, puis sur le contrôle de puissance dans les réseaux 5G. Pour finir, nous nous détachons des RSF pour proposer un nouvel algorithme d'optimisation bayésienne, que nous démontrons asymptotiquement optimal, capable de maximiser de façon décentralisée une boîte noire de grande dimension. Cette dernière contribution pourrait permettre l'implémentation de protocoles plus performants dans les RSF mais aussi dans d'autres contextes technologiques.

# LIST OF PUBLICATIONS

Some ideas and figures have appeared previously in the following publications.

### PEER-REVIEWED JOURNALS AND INTERNATIONAL CONFERENCES

[1] Anthony Bardou and Thomas Begin. "INSPIRE: Distributed Bayesian Optimization for ImproviNg SPatIal REuse in Dense WLANs." In: *MSWiM '22: Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. Ed. by ACM. 2022.

[2] Anthony Bardou, Thomas Begin, and Anthony Busson. "Improving the Spatial Reuse in IEEE 802.11ax WLANs: A Multi-Armed Bandit Approach." In: *MSWiM '21: Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. Ed. by ACM. 2021.

[3] Anthony Bardou, Thomas Begin, and Anthony Busson. "Analysis of a Multi-Armed Bandit solution to improve the spatial reuse of next-generation WLANs." In: *Computer Communications* 193 (2022), pp. 279–292.

[4] Anthony Bardou, Thomas Begin, and Anthony Busson. "Mitigating starvation in dense WLANs: A multi-armed Bandit Solution." In: *Ad Hoc Networks* 138 (2023).

[5] Anthony Bardou and Thomas Begin. "Analysis of a decentralized Bayesian optimization algorithm for improving spatial reuse in dense WLANs." In: *Computer Communications* (2023).

### NATIONAL CONFERENCES

[1] Anthony Bardou and Thomas Begin. "INSPIRE: Optimisation bayésienne distribuée pour l'amélioration de la réutilisation spatiale des WLANs denses." In: *AlgoTel 2022-24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*. 2022.

[2] Anthony Bardou, Thomas Begin, and Anthony Busson. "Multi-Armed Bandit Algorithm for Spatial Reuse in WLANs: Minimizing Stations in Starvation." In: *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*. 2022.

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## LIST OF ALGORITHMS



## LIST OF ACRONYMS

3GPP 3rd Generation Partnership Project

ADMM Alternating Directions Method of Multipliers

ANN Artificial Neural Network

AP Access Point

ARCEP Autorité de Régulation des Communications Électroniques, des Postes et de la distribution de la presse

BS Base Station

BBO Black-Box Optimization

BO Bayesian Optimization

CCA Clear Channel Assessment

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

EI Expected Improvement

EMA Exponential Moving Average

FDMA Frequency Division Multiple Access

FER Frame Error Rate
FFR Fractional Frequency Reuse
GM Gaussian Mixture
GP Gaussian Process
GSM Global System of Mobile Communications
ICI Inter-Cell Interference
ICIC Inter-Cell Interference Coordination
IEEE Institute of Electrical and Electronics Engineers
IMAB Infinitely Many-Armed Bandit
LTE Long-Term Evolution
MAB Multi-Armed Bandit
MCMC Monte-Carlo Markov Chain
MCS Modulation Coding Scheme
MFS Maximum Factor Size
ML Machine Learning
MLE Maximum Likelihood Estimation
MVN Multivariate Normal
NOMA Non-Orthogonal Multiple Access
OBSS/PD Overlapping Basic Service Set/Preamble-Detection
OFDMA Orthogonal Frequency Division Multiple Access
OMA Orthogonal Multiple Access
PF Proportional Fairness
QoS Quality of Service
RSS Received Signal Strength
SIC Successive Interference Cancellation
SINR Signal to Interference plus Noise Ratio
SNR Signal to Noise Ratio
STA Station
TDMA Time Division Multiple Access
TS Thompson Sampling
UCB Upper Confidence Bound
WLAN Wireless Local Area Network

# 1 INTRODUCTION

## 1.1 MOTIVATIONS

### 1.1.1 *Wireless Networks are Ubiquitous but Misunderstood*

Having a reliable Internet connection has become an almost constant need for a large part of the population in modern societies. Whether it is to work, entertain, get information, socialize or monitor, a growing number of Internet-based services are created and adopted every day. To meet the different needs of an increasing demand, engineers and researchers have developed a wide variety of access networks. Among them, wireless networks are undoubtedly the most preferred networks, because they are the most convenient. Based on radio transmissions rather than wires, they provide connectivity everywhere within their coverage areas and require lighter infrastructure than wired networks. According to Cisco's forecasts [1], wireless networks accounted for 79 % of the Internet traffic in 2022, when wired networks represented the remaining 21 %.

*[1] Cisco (2019), "Cisco visual networking index: Forecast and trends, 2017–2022 white paper"*

Each wireless technology rigorously obeys to a standard. A standard thoroughly provides all the technical details regarding how the technology is (or should be) implemented, and how devices implementing this technology should interact within the network. This is particularly important to ensure inter-operability but also, sometimes, a given Quality of Service (QoS). This may pertain to achieving some predefined throughput, over a particular area, given some requirements. Therefore, standards are essential, as they make the technological context independent from the manufacturer of the devices. Many of these standards are branded under commercial names and are well-known from the general public. Figure 1.1 depicts some of such standards.

Each wireless standard aims at addressing specific use cases and involves parameters that can be tuned to adapt the wireless network to a particular scenario, with the ultimate goal of increasing the network performance. Although an intuitive understanding of the relationship between a parameter and a performance metric can be easily inferred, accurately capturing it in an analytical model remains a notoriously hard problem for many wireless technologies. This can be explained by the numerous exogenous factors that impact the quality of a radio transmission, such as the spatial distribution of devices, the workload of the network or the amount of concurrent radio transmissions in the same electromagnetic frequency band. This implies that a given

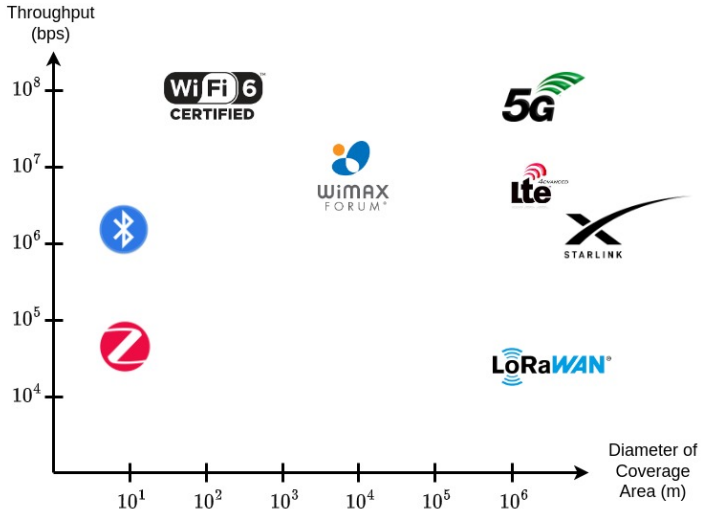



Figure 1.1: Some wireless networks standards arranged by their throughput and the diameter of their coverage areas.

wireless network configuration that achieves good results in a specific deployment can lead to poor performance if used in another, yet similar deployment. As for now, this prevents most wireless networks to be optimally configured, and generally forces them to rely on simple algorithms and / or on human network administrators to find a good configuration of their parameters.

Unfortunately, finding such good configurations is made more difficult by the growing demand of dense sets of devices requiring a wireless connection,[1] which calls for denser deployments of wireless technologies. To better adapt to these difficult scenarios, a number of wireless standards have introduced new degrees of freedom for networks to exploit. This in turn exacerbates the complexity of optimizing the wireless networks (e.g. maximizing their throughput or minimizing their energy consumption) by introducing additional dimensions to the configuration space.

[1] *Especially in dense urban areas.*

Overall, to cope with this complexity, wireless standards require scalable algorithms able to derive an optimal network configuration while being agnostic regarding the actual wireless network deployment. The research of such algorithms remains a major hot topic in the computer networks community nowadays [2].

*[2] Chen et al. (2021), "Distributed learning in wireless networks: Recent progress and future challenges"*

### 1.1.2 *An In-Depth Example: the Spatial Reuse of a Wireless Local Area Network*

A Wireless Local Area Network (WLAN) typically covers an area from a dozen to a hundred meters in diameter. As medium-sized networks easy to deploy, WLANs have become ubiquitous: they are found in homes, offices and public spaces. Although WLANs carry most of the Internet traffic (57 % in 2022 according to Cisco [1]), they suffer from

a whole set of problems exacerbated by dense deployments. In this section, we detail one of them: the scarcity of the frequency band. A significant fraction of this thesis aims at addressing this problem.

#### 1.1.2.1 *Architecture of WLANs*

The vast majority of WLANs follow the guidelines of the Institute of Electrical and Electronics Engineers (IEEE) standard 802.11, commercially known as Wi-Fi, that describes the association and the communication processes between two types of devices:

- the Station (STA), that is the end-user wireless device requiring access to the Internet and,
- the Access Point (AP), that is the device that serves as a gateway to the Internet.

An 802.11 network comprises multiple APs that relay back-and-forth frames to a set of associated STAs. In addition to their wireless interface, APs also embed a wired interface to provide access to the Internet. Note that a STA is associated with a single AP, however an AP generally serves multiple associated STAs. Figure 1.2 illustrates a possible spatial distribution of 802.11 devices along with the existing associations between APs and STAs. The spatial arrangement of the APs and STAs is commonly referred to as the *WLAN topology*. In the remaining of this section, we put a strong focus on how 802.11 share the frequency band among the wireless devices. For a more complete, technical overview of the technology, we refer the interested reader to [3].

*[3] Gast (2005),* 802.11 wireless networks: the definitive guide

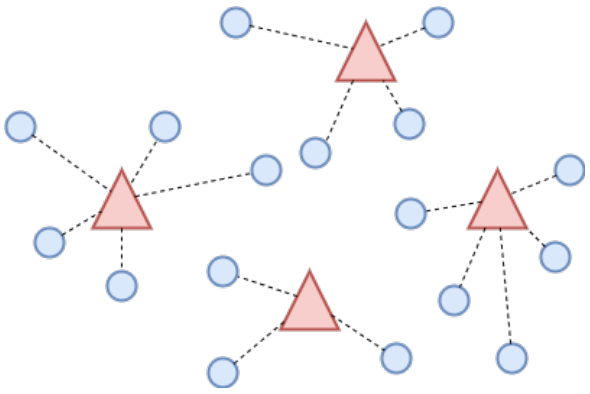


Figure 1.2: An 802.11 network topology. APs are depicted with red triangles, STAs with blue circles and associations between APs and STAs with dashed lines.

#### 1.1.2.2 *Sharing the Frequency Band*

Like many wireless technologies, 802.11 devices use an omnidirectional[2] antenna to transmit data using radio transmissions in an al-

[2] *Emitting with equal power in all directions.*

located frequency band.[3] When two devices are close enough - both in the frequency and in the spatial domains - their radio transmissions interfere and the overall transmission quality is deteriorated. This makes the allocated frequency band a scarce resource that needs to be shared efficiently and with respect to the needs of the transmitting devices. Conversely, when two wireless devices are sufficiently far (geographically) from each other, they can reuse the same frequency band without interfering.[4] This is called *the spatial reuse* of the frequency band, and it drastically increases the performance of the WLAN. Note that, nowadays, spatial reuse optimization in WLANs remains a notoriously hard problem.

To limit interference, 802.11 splits the frequency band into partially-overlapping chunks called *radio channels*.[5] When two 802.11 devices belong to orthogonal channels, they can transmit simultaneously without experiencing interference. Therefore, allocating different radio channels to neighboring wireless devices (as illustrated by Figure 1.3) drastically increases the individual throughput of the 802.11 network. One can refer to [4] for a recent survey of the many efficient channel allocation techniques that exist.

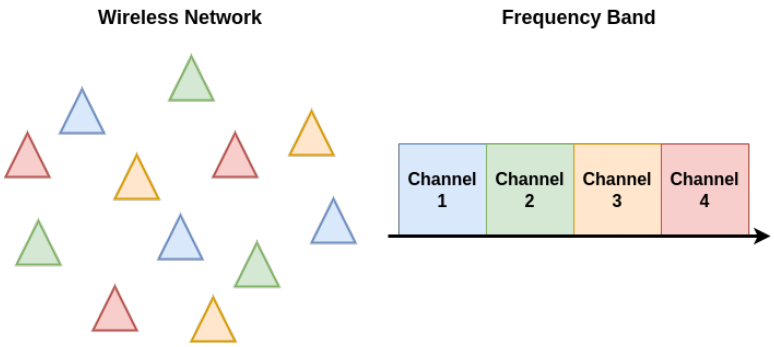



Figure 1.3: A channel allocation in an 802.11 network topology. APs are depicted with triangles, and the four radio channels are shown with four different colors.

Although interference can be significantly reduced with an efficient channel allocation, it can still represent a problem for radio channels with a high spatial density of APs. To further reduce interference within a given radio channel, 802.11 introduces another mechanism: the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. Each device, before transmitting, uses a Clear Channel Assessment (CCA) to make sure that the radio frequency medium is not already busy. To do so, it measures the level of energy in its radio channel for a short period of time and compares it to a threshold,[6] called the *sensitivity threshold* or the Overlapping Basic Service Set/Preamble-Detection (OBSS/PD) threshold, and denoted `OBSS_PD` in this thesis. When the measured level of energy is greater than `OBSS_PD`, the 802.11 device defers its transmission during a random period of time called a *backoff*, to avoid creating interference for

[3] *Either the $2.45 \pm 0.05$ GHz or the $5.8 \pm 0.075$ GHz band.*

[4] *Because of the natural attenuation of the electromagnetic signals.*

[5] *A channel can be 20, 40, 80 or 160 MHz wide.*

[4] *Iacoboaiea et al. (2021), "Real-time channel management in WLANs: Deep reinforcement learning versus heuristics"*

[6] *Specified in decibel-milliwatts (dBm).*

the ongoing transmission. As long as the channel is not sensed idle, the transmission is postponed during a random backoff. When the channel is eventually sensed idle (either immediately or when the last backoff has come to zero), the device is allowed to transmit its data. Historically, OBSS_PD and the transmission power of each device,[7] denoted TX_PWR in this thesis, were considered static in the standard. Up until recently, they were set at -82 dBm and 20 dBm respectively [5].

In this thesis, we call two devices *in conflict* with each other if they cannot transmit data at the same time because of the CSMA/CA protocol. Note that, among all the conflicts existing in WLANs,[8] those occurring between APs (the AP-AP conflicts) are the most troublesome as the traffic is mostly downstream[9] [6]. Therefore, the number of conflicts between APs within the radio channels must be minimized in order to optimize the performance of 802.11 networks. Nowadays, 802.11 networks with a large spatial density of APs are deployed within public spaces (e.g. train stations, malls), companies or residential buildings. As a consequence, the number of conflicts between APs grows despite an efficient channel allocation. At the end of the day, the performance of the dense networks is reduced and the end-user experience is significantly degraded.

[7] *Also specified in dBm.*

[5] *Gast (2012),* 802.11n: the survival guide

[8] *AP-AP, AP-STA and STA-STA.*

[9] *From the AP to the associated STAs.*

[6] *Adeyemi et al. (2018), "Exploration of daily Internet data traffic generated in a smart university campus"*

#### 1.1.2.3 *Recently Introduced Spatial Reuse Mechanisms*

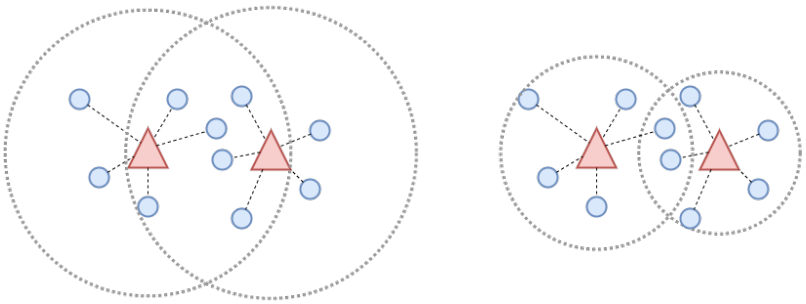
Due to new use-cases of WLANs, IEEE recently released the 802.11ax amendment (commercially branded by the Wi-Fi Alliance as Wi-Fi 6) that modifies the 802.11 standard. It aims at improving many aspects of 802.11 networks, such as reducing their energy consumption or allowing them to guarantee deterministic transmission delays. 802.11ax also aims at improving the spatial reuse of the radio channels, opening the path to major gains in performance. To do so, it enables each AP to dynamically adjust two of their key parameters [7]: OBSS_PD and TX_PWR. While prior amendments to 802.11 held TX_PWR and OBSS_PD constant, 802.11ax authorizes their values to dynamically vary from 1 to 21 dBm for the former and from -82 to -62 dBm for the latter. The two parameters must also comply with the following constraint

$$\texttt{OBSS\_PD} \leq \max(-82, \min(-62, -82 + (20 - \texttt{TX\_PWR}))). \tag{1.1}$$

The constraint (1.1) ensures that a device cannot have a high sensitivity threshold (e.g. -62 dBm) while simultaneously having a high transmission power (e.g. 20 dBm). We provide below the application of the constraint (1.1) for three values of TX_PWR:

- if $\texttt{TX\_PWR} = 20$ dBm, $\texttt{OBSS\_PD} \leq -82$ dBm,
- if $\texttt{TX\_PWR} = 10$ dBm, $\texttt{OBSS\_PD} \leq -72$ dBm,
- if $\texttt{TX\_PWR} = 1$ dBm, $\texttt{OBSS\_PD} \leq -63$ dBm.

[7] *IEEE (2021), "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"*



(a) The default configuration of TX_PWR puts the two APs within each other's detection range, they cannot transmit simultaneously.

(b) The value of TX_PWR can be tuned and adapt to this deployment. Under this new configuration, the APs can transmit simultaneously.

Figure 1.4: Adequately configuring the TX_PWR parameter can significantly improve the spatial reuse of radio channels in WLANs by removing conflicts between APs belonging to the same radio channel.

Figure 1.4 depicts a simple example of two APs operating on the same radio channel and illustrates how two 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 [8] for more details).

*[8] Wilhelmi et al. (2019), "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs"*

Although 802.11ax has the potential to improve the spatial reuse of radio channels, the standard does not provide any algorithm to find an adequate configuration of TX_PWR and OBSS_PD for the APs in a WLAN. As stated before, this is mainly due to the difficulty to capture the relationship between a configuration of TX_PWR and OBSS_PD and the performance metrics of a WLAN in an analytical model, especially in large, dense WLANs. The lack of a scalable analytical model prevents the emergence of a consensual way of finding the optimal configuration. As such, it is up to the manufacturers and, more globally, to the research community to solve this problem.

A significant portion of this thesis tackles this specific issue.

### 1.1.3 *The Framework of Black-Box Optimization*

As described in Section 1.1.1 and illustrated with Section 1.1.2, optimizing the performance of wireless networks often involves finding a configuration of parameters without precise prior understanding of the influence of the parameters on the performance metrics. This makes it difficult, if not impossible, to derive a closed form for the optimal configuration of the wireless network parameters. There-

fore, this thesis does not seek to provide a unified framework able to provide an analytical solution to an arbitrary optimization task on wireless networks.[10] Instead, it proposes data-driven methods that exploit sequential feedback provided by the wireless network to *discover and simultaneously optimize* the function $f$ describing a given performance metric with respect to the wireless network parameters. These data-driven methods belong to a special subset of mathematical optimization called Black-Box Optimization (BBO). In this section, we introduce the BBO framework and how promising it is regarding performance optimization in wireless networks.

[10] *One can legitimately doubt that such a tool will ever exist.*

Many real-world applications involve the optimization of an unknown $d$-dimensional objective function $f : \mathcal{C} \subseteq \mathbb{R}^d \to \mathbb{R}$ that may be noisy and costly to evaluate. Examples of such tasks include robotics [9], computational biology [10] and, of course, networking [11]. In such applications, $f$ is only defined by its inputs and outputs and hence, can be considered as a black box. Due to the lack of information, $f$ must be queried at specific inputs to be discovered, in an iterative fashion. Thus, BBO is a form of *online* learning, as opposed to *offline* learning where all the data used in the learning task has been collected beforehand and is fully accessible.

[9] *Lizotte et al. (2007), "Automatic Gait Optimization with Gaussian Process Regression"*

[10] *González et al. (2014), "Bayesian Optimization for Synthetic Gene Design"*

[11] *Hornby et al. (2006), "Automated Antenna Design with Evolutionary Algorithms"*

Evaluation and comparison of BBO algorithms is often done in the output space, through the notion of *regret*.

**Definition 1.1** (Instantaneous Regret)**.** *The instantaneous regret at time $t$ of a BBO algorithm regarding the optimization of the function $f$ is*

$$r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t) \tag{1.2}$$

*with $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$ and $\boldsymbol{x}_t$ the query of the BBO algorithm at time $t$.*

Instantaneous regrets can be aggregated to provide a cumulative evaluation metric up to a time $t$.

**Definition 1.2** (Cumulative Regret)**.** *The cumulative regret up to time $T$ of a BBO algorithm regarding the optimization of the function $f$ is*

$$R_T = \sum_{t=1}^{T} f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t) \tag{1.3}$$

*with $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$ and $\boldsymbol{x}_t$ the query of the BBO algorithm at time $t$.*

A common goal for BBO algorithms is to minimize their cumulative regrets, as defined by Definition 1.2. To do so, they all follow the same generic online procedure, described in Algorithm 1.1. The main loop is described from line 4 to line 10, but note that line 6 describes the observation of the output of $f$ at a given query $\boldsymbol{x}_t$ perturbed by some noise,[11] while lines 7-9 simply describe data management

[11] *From an arbitrary distribution $\mathcal{F}$ with parameters $\boldsymbol{\theta}$.*

steps. Actually, all the uniqueness of a BBO algorithm resides in how it implements line 5, that is, how it queries $f$ using the previously collected data $(\boldsymbol{X}_t, \boldsymbol{y}_t)$.

**Algorithm 1.1** Generic Online Optimization Loop

**Input**: objective function $f$.

1: $t = 0$
2: $\boldsymbol{X}_0 = \emptyset$
3: $\boldsymbol{y}_0 = \emptyset$
4: **while true do**
5: Determine $\boldsymbol{x}_t$ using $(\boldsymbol{X}_t, \boldsymbol{y}_t)$ so that $f(\boldsymbol{x}_t)$ brings information about $\arg\max_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$
6: Observe $y_t = f(\boldsymbol{x}_t) + \epsilon$, with $\epsilon \sim \mathcal{F}(\boldsymbol{\theta})$
7: $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t \cup \{\boldsymbol{x}_t\}$
8: $\boldsymbol{y}_{t+1} = \boldsymbol{y}_t \cup \{y_t\}$
9: $t = t + 1$
10: **end while**

Since $f$ is a black-box, the query $\boldsymbol{x}_t$ must be determined at time $t$ by meeting two objectives:

- *Exploration*: querying $f(\boldsymbol{x}_t)$ should help to better understand the black-box function $f$,
- *Exploitation*: to keep a low regret, $f(\boldsymbol{x}_t)$ should be as large as possible.

Each of these objectives often lead in different querying policy, both of them being suboptimal: exploring too much leads to query uninteresting subsets of the function domain,[12] exploiting too much carries a high risk of missing the optimal configuration $\boldsymbol{x}^*$. Here, a fundamental problem in online data-driven decision-making processes facing uncertainty is unveiled: *the exploration-exploitation dilemma* [12]. There is no single, consensual solution to achieve a trade-off between the two objectives, and every BBO algorithm achieves it in a different way through a different implementation of line 5 in Algorithm 1.1.

Figure 1.5 illustrates the benefits of compromising between exploration and exploitation. Recall that a BBO algorithm tries to discover and optimize the depicted function $f$ by successive queries. In this figure, we use a Gaussian Process (GP) as a surrogate model for $f$ (as thoroughly described in the next chapter, Section 2.2.4). The best guess of the surrogate model is depicted as a solid line and the uncertainty associated with its predictions as an orange area. The candidates for the next query according to exploration and exploitation are shown with crosses. The candidate for exploitation is obtained by maximizing the current prediction about $f$.[13] Conversely, the candidate for exploration is obtained by maximizing the uncertainty about the prediction.[14,15] Because the objective of the BBO algorithm is to

[12] *It is known a priori that $\boldsymbol{x}^*$ is not in these subsets.*

[12] *Berger-Tal et al. (2014), "The exploration-exploitation dilemma: a multidisciplinary framework"*

[13] *i.e. maximizing the solid orange line.*

[14] *i.e. finding the largest shaded interval.*

[15] *Note that this is a way of figuring out the "most surprising" output of $f$, in order to learn as much as possible in a single observation.*

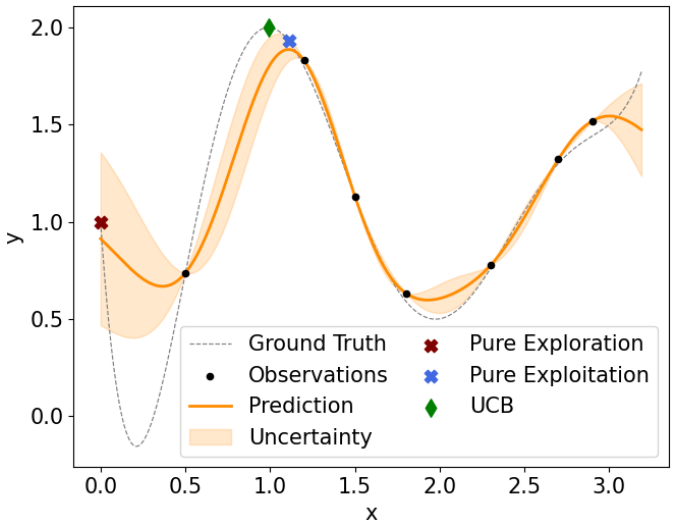



Figure 1.5: An illustration of the exploration-exploitation dilemma. The function $f$ is shown with a gray dashed line, the observations $(\boldsymbol{X}_t, \boldsymbol{y}_t)$ with black dots. Candidates for the next query $\boldsymbol{x}_t$ can be observed with colored markers: pure exploration (maroon), pure exploitation (blue) and a trade-off called Upper Confidence Bound (UCB) (green).

minimize its cumulative regret $R_T$ by finding the maximum of $f$ as quickly as possible, note that neither exploration nor exploitation yield the optimal query. However, as depicted by the green diamond, trading off some exploitation for some exploration allows to find a query much closer to the maximal argument of $f$. This particular trade-off is called UCB [13] and is discussed in details (along with others) in Section 2.2.

*[13] Auer (2002), "Using confidence bounds for exploitation-exploration trade-offs"*

Successfully achieving an exploration-exploitation trade-off at each iteration is key to obtain good empirical performance. In fact, some BBO algorithms also provide theoretical guarantees, by ensuring that the objective function will be optimized in a finite or an infinite amount of time. The latter guarantee, also called *asymptotic optimality*, is defined as follows.

**Definition 1.3** (Asymptotic Optimality)**.** *A BBO algorithm is said asymptotically optimal if*

$$\lim_{T \to +\infty} \frac{R_T}{T} = 0 \tag{1.4}$$

*with $R_T$ the cumulative regret defined in Definition 1.2.*

An asymptotically optimal BBO algorithm ensures *no-regret* performance. That is, it guarantees that its cumulative regret will grow sublinearly over time. Consequently, in the long run, such a BBO algorithm necessarily reaches the global optimum of any objective function $f$ that satisfies its introduced assumptions. Clearly, such a guar-

antee is desirable in many optimization tasks, because it removes the risk of being trapped at a local optimum.

## 1.2 THESIS OUTLINE

In this thesis, we propose to apply the black-box optimization framework to performance optimization in wireless networks, and especially to the spatial reuse problem in 802.11ax wireless networks (which was detailed in Section 1.1.2). The organization of the thesis is as follows:

- Chapter 2 is two-fold. First, it makes an extensive review of all the proposed approaches to solve the spatial reuse problem in WLANs based on the 802.11ax amendment. Second, it discusses the advantages and drawbacks of online BBO algorithms applicable to this problem, namely global optimization, heuristics, Multi-Armed Bandit (MAB) and Bayesian Optimization (BO).
- Chapter 3 describes two ad-hoc methods developed with the MAB framework to address the spatial reuse problem in 802.11ax, in a centralized fashion.
- Chapter 4 focuses on a decentralized BO algorithm that addresses the spatial reuse problem in 802.11ax and significantly outperforms the state-of-the-art methods (including those developed in Chapter 3). Note that the proposed algorithm is decentralized, which is a desirable feature since most algorithms in the 802.11 ecosystem are also decentralized.
- Chapter 5 considers a different technological context: the cellular networks. After briefly describing the technological context and the relevant mechanisms for our study, we assess the benefits of a new multiple access approach envisioned for 6G, referred to as NOMA. Simultaneously, we demonstrate that the algorithm proposed in Chapter 4 can be very useful in this new technological context, as our study relies on it to tune the transmission powers of cellular antennas.
- Chapter 6 is dedicated to an asymptotically optimal algorithm that improves the state-of-the-art of decentralized BO for high-dimensional, noisy black-box functions. Note that this algorithm can address any BBO problem, including of course some problems arising in the optimization of wireless networks.

A graphical outline of this thesis is depicted in Figure 1.6. Each chapter is projected in a 2-dimensional space according to the degree of theoretical guarantees it provides and how much it relies on the

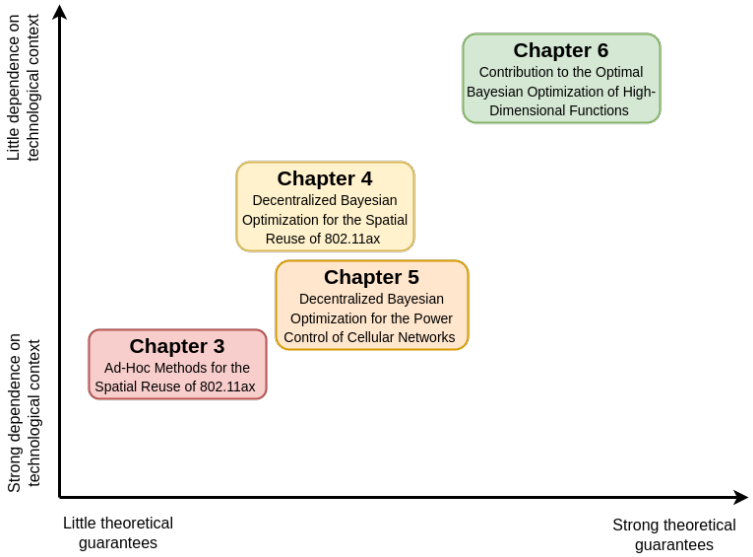



Figure 1.6: Graphical outline of the thesis.

technological context. This figure depicts the evolution of the contributions defended in this thesis and exhibit a clear, bottom-up, long-term scientific direction: the development of a black box optimization method providing strong theoretical guarantees and applicable in multiple technological contexts.

Each chapter is written to be self-contained and can be read independently from the others, although Chapter 4 is complementary to Chapter 5, and Chapter 2 is relevant to all the others. At the end of each chapter, we provide a summary of the proposed contributions, along with their strengths and weaknesses and we cite our related scientific publications when applicable. The chapters are arranged so that the questions raised at the end of a chapter are often answered in the next chapters.

## 1.3 CONTRIBUTIONS

This thesis comprises two main kinds of contributions:

- Algorithmic contributions:
  - We propose two centralized ad-hoc algorithms to address the spatial reuse problem in 802.11ax networks.
  - We propose INSPIRE, a decentralized Bayesian optimization algorithm to address the spatial reuse problem in 802.11ax networks.
  - We show the efficiency of INSPIRE in another technological context by proposing a decentralized algorithm for the power control in cellular networks.[16]

[16] *Joint work with Jean-Marie Gorce, INSA Lyon.*

  - We design DuMBO, a novel decentralized algorithm dedicated to the Bayesian optimization of high-dimensional black-box function.[17]

[17] *Joint work with Patrick Thiran, EPFL.*

- Theoretical contributions:
  - We propose a minimax optimal consensus in a decentralized system, exploited by INSPIRE.
  - We propose a better approximation of the exploration term for Bayesian optimization algorithms in decentralized systems, exploited by DuMBO.
  - We demonstrate the asymptotic optimality of DuMBO.
  - We demonstrate the minimax optimality of an early-stopped version of DuMBO.

# 2 RELATED WORKS

**Outline.** This chapter is two-fold. First, Section 2.1 makes an extensive review of the solutions proposed by the research community to address the spatial reuse problem in a Wireless Local Area Network (WLAN) through the control of the transmission power (`TX_PWR`) and the sensitivity threshold (`OBSS_PD`) of devices. Then, Section 2.2 provides an in-depth description of the relevant black-box optimization approaches for the spatial reuse problem in WLANs and the theoretical guarantees they provide.

## 2.1 POWER AND SENSITIVITY CONTROL IN WLANS

Recall that, in a WLAN, a wireless device (such as a phone or a computer) is called a Station (STA) and is associated to an Access Point (AP) that serves as a gateway to the Internet. As described in details in Section 1.1.2, the spatial reuse problem in modern WLANs is addressed in two steps. First, the frequency band is split into orthogonal chunks called radio channels, and different radio channels are allocated to neighboring devices so they can transmit data without interfering with each other. This problem is typically addressed as a graph-coloring problem [14], although more recent 802.11 amendments provided new degrees of freedom and allowed the emergence of new techniques such as channel bonding [15]. Nowadays, channel allocation is performed using heuristics [16] or Machine Learning (ML) techniques [17].

However, even with an optimal channel allocation, too many conflicts may persist between APs in case of dense WLANs, which may result into poor performance for the end-users. To fix this, the recent 802.11 amendment (802.11ax or Wi-Fi 6) allows the dynamical update of `TX_PWR` and `OBSS_PD`. This new feature allows APs and STAs to further adapt to their specific radio environment within their channels but at the same time calls for algorithms able to derive promising values for `TX_PWR` and `OBSS_PD` within a radio channel. For a detailed explanation of how the Institute of Electrical and Electronics Engineers (IEEE) 802.11ax standard implements the adaptation of `TX_PWR` and `OBSS_PD`, we refer the reader to [18].

The problem of tuning of `TX_PWR` and `OBSS_PD` has been explored many years before the introduction of 802.11ax. The solutions proposed by the research community can be organized into two different categories, depending on whether they are based on an analytical

*[14] Mishra et al. (2006), "A Client-Driven Approach for Channel Management in Wireless LANs"*

*[15] Herzen et al. (2013), "Distributed spectrum assignment for home WLANs"*

*[16] Barrachina-Muñoz et al. (2019), "Online primary channel selection for dynamic channel bonding in high-density WLANs"*

*[17] López-Raventós et al. (2020), "Concurrent decentralized channel allocation and access point selection using multi-armed bandits in multi BSS WLANs"*

*[18] Wilhelmi et al. (2021), "Spatial reuse in IEEE 802.11 ax WLANs"*

model or mostly data-driven. We review the two categories in this section.

### 2.1.1 *Analytical Approaches*

Analytical approaches propose mathematical models of the WLAN. Based on these models, they derive the optimal configuration of TX_PWR and / or OBSS_PD.

*[19] Zhu et al. (2004), "Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks"*

*[20] Kim et al. (2004), "SP-TPC: a self-protective energy efficient communication strategy for IEEE 802.11 WLANs"*

*[21] Qiu et al. (2020), "Joint access point placement and power-channel-resource-unit assignment for 802.11 ax-based dense WiFi with QoS requirements"*

Pioneering efforts were made in 2004 by [19], that presents an analytical model to derive the optimal value for OBSS_PD in 802.11-based mesh networks. To do so, OBSS_PD is tuned dynamically on each node as a function of the radio channel conditions. The TX_PWR parameter, however, is not considered. The same year, [20] proposed to tune the other parameter, TX_PWR. The authors showed that tuning the transmission power of the devices increases the throughput of the WLAN and simultaneously reduces its energy consumption. More recently, [21] cast the issue of AP positioning and their power allocation as a single optimization problem. Their solution addresses the initial positioning of APs, the channel allocation and the configuration of the TX_PWR parameter on each AP. However it delivers a single, static configuration that does not account for the number of STAs associated with each AP, nor for the amount of traffic exchanged between STAs and APs.

Analytical approaches do not constitute the majority of the solutions to the tuning problem of TX_PWR and OBSS_PD. This is because, in practice, the large number of parameters and the complexity of the physical layer hinder the use of complex analytical model-based solutions. Furthermore, they often do not scale with the number of APs and STAs in the WLAN [22] and when they do, they are too coarse-grained to provide interesting configurations [23, 24]. Instead, data-driven techniques, which are more practical and adaptive by construction, appear as natural candidates to this adaptation problem.

*[22] Bianchi (2000), "Performance analysis of the IEEE 802.11 distributed coordination function"*

*[23] Laufer et al. (2015), "The capacity of wireless CSMA/CA networks"*

*[24] Stojanova et al. (2019), "Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach"*

### 2.1.2 *Data-Driven Approaches*

Data-driven techniques are intrinsically iterative. Instead of considering the whole WLAN model and immediately derive the optimal configuration, they explore multiple configurations to gain knowledge about the WLAN and converge towards an efficient configuration. Although often lacking the theoretical guarantees of their analytical counterparts, they rely on a smaller set of assumptions, making them more applicable. They also often benefit from a lower computational complexity.

*[25] Afaqui et al. (2015), "Evaluation of dynamic sensitivity control algorithm for IEEE 802.11 ax"*

Among the first data-driven approaches is the Dynamic Sensitive Control (DSC) decentralized algorithm, proposed by [25] and dedicated to the tuning of the OBSS_PD parameter on STAs. With DSC,

STAs periodically measure the Received Signal Strength (RSS) of beacons[18] received by their associated AP. They gradually decrease their OBSS_PD to favor concurrent transmissions from neighboring devices, while keeping this value high enough to ensure the proper reception of beacons. Another decentralized algorithm is proposed by [26], referred to as Link-aware Spatial Reuse (LSR). Like DSC, its decisions are based on the RSS. More precisely, each AP chooses its concurrent AP as the one having the lowest RSS. Then, the prescribed value for TX_PWR is based on the Frame Error Rate (FER) when the AP and its concurrent AP transmit simultaneously. Overall, both DSC and LSR algorithms [25, 26] represent practical and effective solutions that can typically increase the throughput of a WLAN by a factor of up 20-30%. However, none of them fully exploit the potential benefits offered by the 802.11ax standard. DSC only modifies the value of the OBSS_PD parameter for STAs so that performance improvements mostly apply to the small portion of traffic that goes upstream. In the case of LSR, the solution applies to APs and downstream traffic, but it limits to one the number of concurrent APs allowed to transmit simultaneously. In practice, this limitation is further increased since concurrent APs do not always have frames to send at once.

[18] *Small messages periodically broadcast by APs to advertise their WLAN.*

*[26] Lee et al. (2021), "LSR: link-aware spatial reuse in IEEE 802.11 ax WLANs"*

Recently, ML has impacted many fields, and computer networks are no exception. To optimize the performance of WLANs, recent surveys such as [27] list hundreds of contributions. Among them, several made use of ML techniques to address the issue of tuning the TX_PWR and OBSS_PD parameters in WLANs. In 2020, a two-steps solution relying on an Artificial Neural Network (ANN) was proposed by [28]. In their framework, STAs and APs first locally adjust their value of OBSS_PD to decrease interference. Then, thanks to an ANN whose parameters have been set using offline simulations, they mitigate potential unfairness among STAs in terms of attained throughput, which may otherwise occur due to the various locations of STAs. The authors use the ns-3 simulator to show that their solution can improve the throughput and fairness of WLANs up to 36% and 82%, respectively. Although the empirical improvement is significant, this method suffers from a major drawback. Indeed, ANNs comprise numerous parameters that require a large amount of data to be set properly. As far as we know, there is no dataset recognized by the networking community whose content captures the large diversity of scenarios that can be encountered in WLANs. Because of this lack of representative dataset, the predefined parameters of the ANN may not be accurate for all WLAN topologies. The same can be said about the federated approach used in [29], since it relies on ANNs trained on synthetic data generated from an homemade simulator. More generally, the ability of the proposed offline methods to handle diverse WLAN topologies remains to be properly demonstrated.

*[27] Szott et al. (2022), "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning"*

*[28] Ak et al. (2020), "FSC: Two-scale AI-driven fair sensitivity control for 802.11 ax networks"*

*[29] Wilhelmi et al. (2022), "Federated spatial reuse optimization in next-generation decentralized IEEE 802.11 WLANs"*

This calls for *online* data-driven approaches, not only able to learn from previously collected data, but also to build their own representative dataset given the specific scenario they are deployed on. In [8, 30], the authors formalize the problem of setting the `TX_PWR` and `OBSS_PD` parameters of each AP as a Multi-Armed Bandit (MAB) problem. In both cases, the authors describe a decentralized MAB algorithm that aims to maximize a reward function defined at the AP level. The two solutions mostly differ in terms of the reward definition. In the first solution, the reward function at each AP corresponds to its throughput. This can be described as a "selfish" solution since each AP tries to optimize its own reward regardless of its impact on the neighboring devices. Conversely, the second reward revolves around a max-min normalized function of the throughput of the current AP and of its direct neighboring devices. Using a home-made simulator, the authors show that their solution significantly outperforms the default configuration of the WLAN and that the selfish reward may lead to unfair situations between APs. In fact, this was expected since the MAB algorithms are run concurrently, without any consensus mechanism guaranteeing that the sum of the individual reward functions is maximized.

*[30] Wilhelmi et al. (2019), "Collaborative spatial reuse in wireless networks via selfish multi-armed bandits"*

Table 2.1 summarizes the differences between the considered state-of-the-art approaches in this section. It considers the tuning of the two parameters `TX_PWR` and `OBSS_PD`, as well as the use of dynamic Modulation Coding Scheme (MCS),[19] the type of traffic considered, the simulator used for the presented experiments and the size of the largest network topology on which the solutions were evaluated. As the table shows, only a handful of solutions [8, 26, 30] address the joint tuning of `TX_PWR` and `OBSS_PD`. These strategies are evaluated on "vanilla topologies",[20] composed of at most 8 APs operating on a single radio channel. Furthermore, among them, only one [26] is evaluated with dynamic MCS on a recognized network simulator (ns-3 [31]). This calls for a solution addressing the joint tuning problem of our two parameters, rigorously evaluated on a complex scenario (many APs distributed on many radio channels, bidirectional traffic and dynamic MCS).

[19] *Determining in particular the modulation type for bit encoding and the coding rate.*

[20] *That is, naive topologies that are somewhat failing to reflect the complexity of real WLAN deployments.*

*[31] The ns3 Project (2020),* The Network Simulator ns-3

Overall, the majority of state-of-the-art solutions to the tuning problem of `TX_PWR` and `OBSS_PD` are data-driven. Although better empirical performance has been obtained, these methods suffer from either the lack of datasets recognized as representative by the networking community (which would strengthen the adaptability of offline methods) or alternatively, the lack of theoretical guarantees. Furthermore, they are often evaluated on homemade simulators and vanilla scenarios that do not reflect properly the complexity of real WLAN deployments. Based on these observations, this thesis aims to provide an online, data-driven solution demonstrating good empirical performance supported by strong theoretical guarantees and a more thorough per-

Table 2.1: Comparison of the state-of-the-art approaches for the spatial reuse problem in WLANs. The last column refers to the number of APs and the number of orthogonal radio channels used in the performance evaluation.

| Proposed solutions | Tuning of OBSS_PD | Tuning of TX_PWR | Dynamic MCS | Traffic Up/Down | Simulator | APs / channels |
|---|---|---|---|---|---|---|
| WCMC'04 [19] | ✓ | | | Up/Down | Opnet | 100/1 |
| VTC'04 [20] | | ✓ | | Up | Self-made | 8/1 |
| Infocom'20 [21] | | ✓ | | Up/Down | Self-made | 100/11 |
| WCNC'15 [25] | ✓ | | | Up | Self-made | 100/3 |
| WCNC'21 [26] | ✓ | ✓ | ✓ | Down | ns-3 | 6/1 |
| Globecom'20 [28] | ✓ | | | Up/Down | ns-3 | 3/1 |
| ADHOC'19 [30] | ✓ | ✓ | | Down | Self-made | 8/1 |
| JNCA'19 [8] | ✓ | ✓ | | Down | Self-made | 8/1 |

formance evaluation. To achieve this goal, we first make a review of promising online Black-Box Optimization (BBO) techniques and their theoretical guarantees in the next section.

## 2.2 BLACK-BOX OPTIMIZATION

As described in Section 1.1.3, BBO refers to the optimization of an objective function $f : \mathcal{C} \subseteq \mathbb{R}^d \to \mathbb{R}$ defined solely by its inputs and outputs. A BBO algorithm must query $f$ in such a way that it achieves a trade-off between exploration and exploitation. We review in this section the solutions to achieve this trade-off.

### 2.2.1 *Direct Search Methods*

We start our reviewing of BBO algorithms by introducing a famous direct search method. For a complete review of direct search methods and their applications, one can refer to [32, 33]. Unlike classic optimization methods that use first-order information (e.g. gradient descent) or second-order information (e.g. Newton's method), direct search methods do not assume their availability nor try to approximate them. A case in point in this domain is the DIRECT (DIviding RECTangle) algorithm [34]. It proposes to solve the problem[21]

$$\begin{aligned} &\underset{\boldsymbol{x}\in\mathcal{C}}{\text{maximize}}\ f(\boldsymbol{x}) \\ &\text{subject to } l_i \leq x_i \leq u_i, i \in [1, d]. \end{aligned} \tag{2.1}$$

Because each variable $x_i$ has a lower bound $l_i$ and an upper bound $u_i$, it can be normalized to $[0, 1]$ so that the search space becomes the unit hypercube. To query $f$, DIRECT proceeds by recursively trisecting the search space. At the first iteration, the search space is trisected

*[32] Lewis et al. (2000), "Direct search methods: then and now"*

*[33] Alarie et al. (2021), "Two decades of blackbox optimization applications"*

*[34] Jones et al. (1993), "Lipschitzian optimization without the Lipschitz constant"*

[21] *For consistency purposes, each optimization task is reformulated as a maximization task.*

into (hyper)rectangles, and the points at the center of each rectangle are queried. Then, at each iteration, DIRECT selects a rectangle (with a procedure detailed in the following), trisects it into smaller rectangles and queries $f$ at the center of each rectangle. An illustration of the execution of DIRECT on a 2-dimensional search space is provided in Figure 2.1.

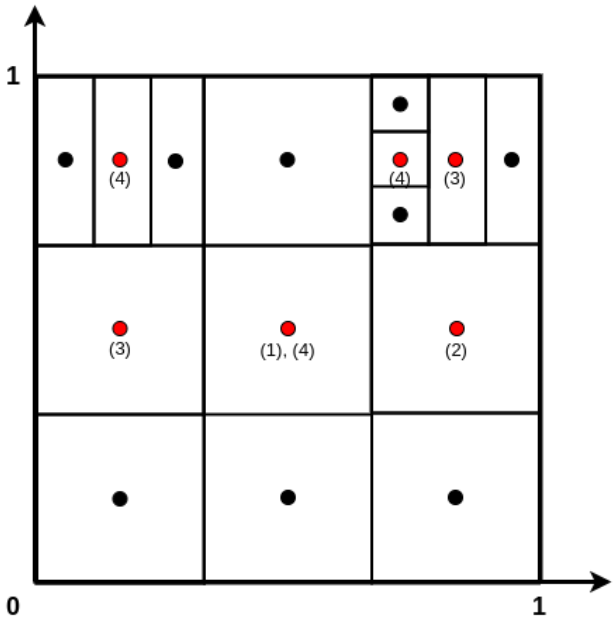



Figure 2.1: Illustration of a possible execution of DIRECT after 4 iterations in a 2-dimensional search space. The center of each chosen rectangle is colored in red. The iteration step appears next to each red dot.

DIRECT assumes that $f$ is Lipschitz-continuous, that is,

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}, |f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L||\boldsymbol{x} - \boldsymbol{y}|| \tag{2.2}$$

with $L > 0$ and $||\cdot||$ an arbitrary norm.

To achieve an exploration-exploitation trade-off, DIRECT tries to identify the optimal rectangle $R$ to trisect using an upper bound on the value of $f$ contained in each rectangle, denoted $U_f(\boldsymbol{c}_i, d_i)$, with $\boldsymbol{c}_i$ the center of the rectangle $R_i$, $d_i = \max_{\boldsymbol{x} \in R_i} ||\boldsymbol{c}_i - \boldsymbol{x}||$ and $||\cdot||$ an arbitrary norm:

$$U_f(\boldsymbol{c}_i, d_i) = f(\boldsymbol{c}_i) + Ld_i. \tag{2.3}$$

Naturally, the objective of DIRECT is to trisect the rectangle that provides the largest upper bound (2.3). We retrieve here an addition of two terms: an exploitation term $f(\boldsymbol{c}_i)$, and an exploration term $Ld_i$. This is an early application of the *optimism in the face of uncertainty* principle, that encourages the algorithm to explore unknown regions of the search space by making an optimistic guess about what lies in it.

Note that, contrary to what one might think, the Lipschitz constant $L$ is *not* an hyperparameter of the algorithm. In fact, DIRECT manages to identify a subset of potentially optimal rectangles $\mathcal{P}$, by using multiple estimations of $L$ through a procedure described in [34]. During an iteration, DIRECT trisects all the potentially optimal rectangles in $\mathcal{P}$. The trisection is systematically carried out on the longest side of each rectangle.

By construction, DIRECT is guaranteed to find the global optimum of $f$ asymptotically (if $f$ is Lipschitz-continuous). It has inspired many research works on direct search methods and continues to do so [35]. In practice, it is known to quickly find the attraction basin of the optimum of $f$. However, as for most direct search methods, it can take a long time to actually find the optimum and it is very sensitive to the "curse of the dimensionality" [35]. Additionally, as far as we know, DIRECT does not provide a cumulative regret upper bound (or *regret bound* for short) ensuring its good performance. Therefore, it may not be a good fit to find rapidly a good configuration in a dense WLAN.

*[35] Jones et al. (2021), "The DIRECT algorithm: 25 years Later"*

### 2.2.2 *Metaheuristics*

Other promising candidates for BBO in WLANs are metaheuristics. In this section, we present a famous evolution strategy and we discuss its application to the spatial reuse problem. For a comprehensive listing of metaheuristics, we refer the reader to [36].

*[36] Gendreau et al. (2010),* Handbook of metaheuristics

Within the class of metaheuristics is a subclass of stochastic methods called *evolutionary algorithms* that are broadly based on the principle of biological evolution. At each iteration (also called *generation*), an evolutionary algorithm generates candidate solutions based on the solutions of the previous iteration (called the *parents*), applying some kind of stochastic variations, analogous to mutations in biological systems. Then, the most promising candidates (*i.e.* those with the better $f$-values) are selected to become the parents of the next generation.

The Covariance Matrix Adaptation-Evolution Strategy (CMA-ES), introduced by [37], is an evolutionary algorithm that has demonstrated excellent empirical performance for the optimization of continuous, nonlinear, nonconvex functions [38]. It proposes to sample the candidate solutions from a Multivariate Normal (MVN) distribution (denoted $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ at iteration $k$) on the input space $\mathcal{C}$. The $f$-values of the candidates are then evaluated, and the parameters $\boldsymbol{\mu}_{k+1}$ and $\boldsymbol{\Sigma}_{k+1}$ of the MVN at the iteration $k+1$ are computed in order to maximize the likelihood of the best sampled candidates. For more details regarding the implementation or the derivation of the updating formulas, we refer the reader to [39, 40].

*[37] Hansen et al. (1996), "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation"*

*[38] Auger et al. (2005), "Performance evaluation of an advanced local search evolutionary algorithm"*

*[39] Hansen et al. (2011), "CMA-ES: evolution strategies and covariance matrix adaptation"*

*[40] Hansen (2016), "The CMA evolution strategy: A tutorial"*

CMA-ES can be seen as a natural gradient descent algorithm, although it does not require any first-order information about the ob-

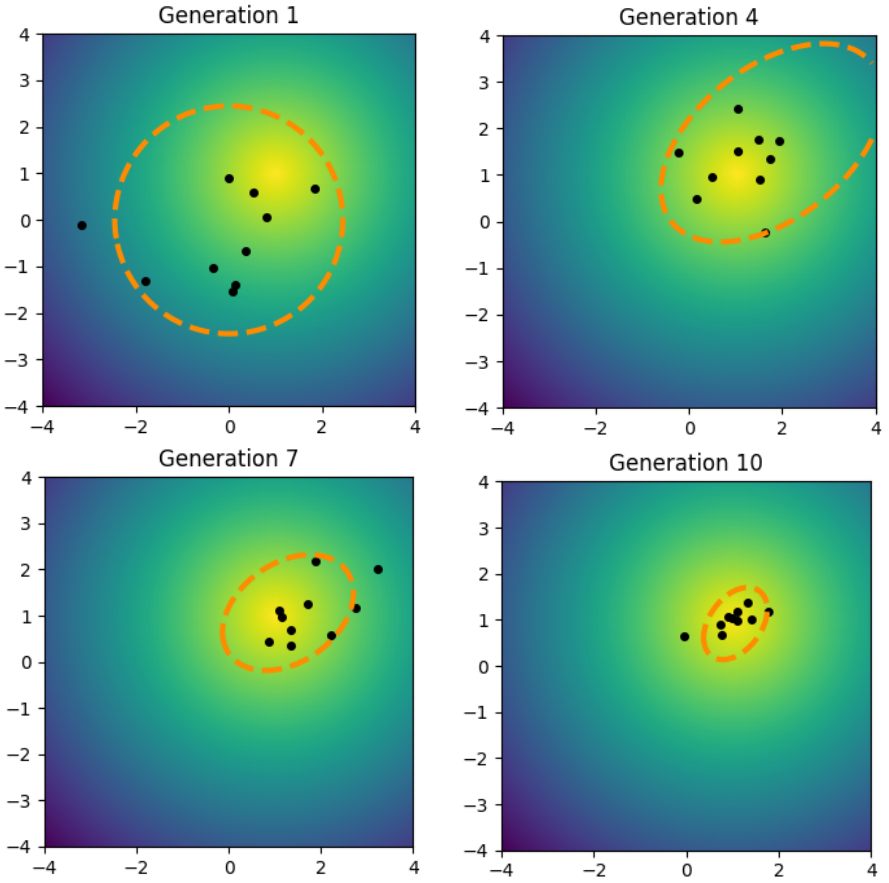



Figure 2.2: Execution of CMA-ES on $f(\boldsymbol{x}) = ||\boldsymbol{x} - \mathbf{1}||_2$, depicted with a color gradient. The sampled candidates are shown with black dots and the MVN density with a single orange dashed contour line.

jective function $f$. To illustrate this, Figure 2.2 depicts an execution of CMA-ES on the minimization of the convex function $f(\boldsymbol{x}) = ||\boldsymbol{x} - \mathbf{1}||_2$. Throughout the generations, the MVN shifts its mean $\boldsymbol{\mu}_k$ towards the global minimum of $f$ (*i.e.* $(1, 1)$) and reduces the norm of its covariance matrix $\boldsymbol{\Sigma}_k$ so that, after the 10th generation, the vast majority of the sampled candidates constitutes a set of high-quality solutions.

CMA-ES is less sensible to the curse of dimensionality than DIRECT, since it performs well on problems comprising up to a thousand dimensions [41]. That is more than enough for the spatial reuse optimization of the vast majority of WLANs. However, sampling multiple candidates at each generation is a significant drawback of the algorithm especially when $f$ is expensive to evaluate, as it is often the case in WLANs. Moreover, even though it performs well on non-convex, nonsmooth examples, CMA-ES does not offer any guarantee regarding the global optimization of $f$, as it could be trapped in the basin of attraction of a local optimum. As such, it may not be the best fit for the spatial reuse problem in WLANs.

[41] *Omidvar et al. (2011), "A comparative study of CMA-ES on large scale global optimisation"*

### 2.2.3 *Multi-Armed Bandit*

In the MAB framework, one considers a set of arms $\mathcal{A}$. At each time step $t$, an arm must be pulled to collect a random reward $y_t$. Each arm $k$ has its own probability distribution $\mathcal{R}_k$ used to sample its returned reward when pulled. The objective of a MAB algorithm is to maximize the sum of the rewards obtained from a sequence of arm pulls. Since the algorithm does not have any prior knowledge about the arms, its objective boils down to discovering as quickly as possible the arm that has the largest expected reward and pulling it as much as possible.

More formally, denoting by $\mu_k$ the expectation of $\mathcal{R}_k$, a MAB algorithm defines a policy $\pi : \mathbb{N} \to \mathcal{A}$ to maximize $\sum_{t=1}^{T} \mu_{\pi(t)}$, given a (potentially infinite) time budget $T$. Using this framework, it is easy to formulate the notion of immediate regret introduced by Definition 1.1. Denoting $k^* = \arg\max_{k \in \mathcal{A}} \mu_k$, the immediate regret of a MAB algorithm at time $t$ is

$$r_t = \mu_{k^*} - \mu_{\pi(t)}. \tag{2.4}$$

The cumulative regret $R_t$ (Definition 1.2) as well as the notion of asymptotic optimality (Definition 1.3) follow from this reformulation of the immediate regret.

The spatial reuse problem in WLANs can be formulated as a MAB problem, with each configuration $\boldsymbol{x} \in \mathcal{C}$ of a WLAN considered as an arm. Modeling the outputs of the black-box objective function $f$ as stochastic rewards makes sense, since $f$ can be perturbed by some observational noise. Due to the large cardinality of $\mathcal{C}$, some variants of the MAB may be more suitable than its classical formulation, such as the Infinitely Many-Armed Bandit (IMAB) where the number of arms is considered larger than the time budget $T$ [42], or the Continuum-Armed Bandit [43], where the arms are chosen within a real field. We now review the classical solutions to the MAB problem that are relevant to this thesis. Their adaptation to the spatial reuse problem is discussed in great length in Chapter 3.

*[42] Wang et al. (2008), "Algorithms for infinitely many-armed bandits"*

*[43] Agrawal (1995), "The continuum-armed bandit problem"*

To solve the MAB problem, one could naively propose a greedy approach: the arm with the best empirical mean $\hat{\mu}_k^t$ is chosen at each iteration. Obviously, the empirical mean of an arm $k$ at time $t$ is

$$\hat{\mu}_k^t = \frac{1}{n_k^t} \sum_{i=1}^{t} y_t \mathbb{1}_{\pi(i)=k} \tag{2.5}$$

with $n_k^t$ counting the number of pulls on arm $k$ up to time $t$ and $\mathbb{1}_{\pi(t)=k}$ the indicator function indicating if the arm $k$ was pulled at time $t$.

This corresponds to a very simple policy that initially samples each arm once before always choosing exploitation over exploration. As such, it is unable to find the optimal arm since the policy is entirely based on a small set of observations, too small for the empirical means $\{\hat{\mu}_k^t\}_{k \in \mathcal{A}}$ to be close to the true expected rewards $\{\mu_k\}_{k \in \mathcal{A}}$.

This leads to a cumulative regret to be in $\mathcal{O}(T)$.[22] However, the greedy approach can be made significantly better by allowing it to pull a uniformly drawn arm with probability $\epsilon > 0$. This is called the $\epsilon$-greedy approach (see [44] for more details). If $\epsilon$ is kept constant, the cumulative regret remains in $\mathcal{O}(\epsilon T)$. However, when it is decreased at an appropriate rate (see [45], which sets $\epsilon \propto \frac{1}{T}$), the cumulative regret becomes sublinear asymptotically. Therefore, this variant of $\epsilon$-greedy can be shown asymptotically optimal. Nevertheless, note that [46] showed the cumulative regret bound of an optimal bandit algorithm to be $\Theta(\log T)$ in the case of finite variance for the reward distribution of each arm. Therefore, this variant of $\epsilon$-greedy does not match the known optimal cumulative regret bounds. This indicates that there are better approaches than $\epsilon$-greedy to solve the MAB problem.

Another bandit approach is Thompson Sampling (TS), a natural Bayesian algorithm based on [47] and named after its author. As a Bayesian approach, it requires some prior knowledge about the arms. Namely, it needs

(i) $p(y|k, \boldsymbol{\theta})$ the probability distribution function (p.d.f.) of the reward $y$ given the arm $k$, parameterized by some $\boldsymbol{\theta} \in \Theta$,

(ii) $p(\boldsymbol{\theta})$ the a priori p.d.f. of the parameters $\boldsymbol{\theta}$, and

(iii) $p(\boldsymbol{\theta}|\mathcal{D})$ the a posteriori p.d.f. of the parameters $\boldsymbol{\theta}$, conditioned by the observations in $\mathcal{D}$.

Given this information, TS proposes to pull the next arm by sampling a probability distribution defined on the set of arms $\mathcal{A}$. The probability $p_k$ of pulling the arm $k$ corresponds to the probability of $k$ being the optimal arm, that is

$$p_k = \int_\Theta \mathbb{1}_{\mathbb{E}[y|k,\boldsymbol{\theta}]=\arg\max_{k' \in \mathcal{A}} \mathbb{E}[y|k',\boldsymbol{\theta}]} p(\boldsymbol{\theta}|\mathcal{D}) \, \mathrm{d}\boldsymbol{\theta} \tag{2.6}$$

with $\mathbb{1}$ the indicator function indicating whether the arm $k$ is the optimal arm under the current parameters $\boldsymbol{\theta}$.

Thompson sampling, through (2.6), addresses the exploration-exploitation dilemma in an elegant fashion. Although being one of the oldest heuristics for the MAB problem, TS still belongs to the state-of-the-art nowadays. It has near-optimal regret bounds, as demonstrated by [48, 49], even matching the lower regret bounds given by [46] in some specific settings [50]. Thus, it represents an excellent candidate for solving the spatial reuse problem in WLANs under the MAB framework.

Eventually, we consider Upper Confidence Bound (UCB) (introduced by [13]) to address the MAB problem. It is a perfect instance of the principle of optimism in the face of uncertainty since it defines a policy based on an optimistic estimation (called an upper confidence bound)

[22] *That is, it grows linearly with the time $T$.*

*[44] Sutton et al. (2018), Reinforcement learning: An introduction*

*[45] Auer et al. (2002), "Finite-time analysis of the multiarmed bandit problem"*

*[46] Lai et al. (1985), "Asymptotically efficient adaptive allocation rules"*

*[47] Thompson (1933), "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples"*

*[48] Agrawal et al. (2012), "Analysis of thompson sampling for the multi-armed bandit problem"*

*[49] Agrawal et al. (2013), "Further Optimal Regret Bounds For Thompson Sampling"*

*[50] Kaufmann et al. (2012), "Thompson sampling: An asymptotically optimal finite-time analysis"*

of each arm's expected reward. In its most generic form, the upper confidence bound $u_k^t$ for an arm $k$ at time $t$ is

$$u_k^t = \hat{\mu}_k^t + \sigma_k^t \tag{2.7}$$

with $\sigma_k^t$ chosen so that $\hat{\mu}_k^t - \sigma_k^t \leq \mu_k \leq \hat{\mu}_k^t + \sigma_k^t$ with high probability.

Then, UCB proposes $\pi(t) = \arg\max_{k \in \mathcal{A}} u_k^t$. Note that the exploration-exploitation trade-off is naturally addressed since an arm $k$ can be chosen even with a low $\hat{\mu}_k^t$ if it has a large $\sigma_k^t$, indicating by construction that the arm has not been explored enough to properly estimate its expected reward. Since $\sigma_k^t$ decreases rapidly with the number of pulls, the most pulled arms quickly emerge as those which exhibit a large, properly estimated expected reward $\hat{\mu}_k^t$.

Since its introduction in 2002, UCB inspired decades of research for addressing the MAB problem and beyond. The UCB approaches mainly differ in the assumptions they make and consequently, in their definitions of the upper confidence bound. To name a few, KL-UCB [51] define their upper confidence bound based on the Kullback-Leibler divergence for the bounded bandit problem[23] while LinUCB [52] assumes a linear dependency between a context vector and the expected reward of an arm for the contextual bandit problem.[24] These methods match the logarithmic regret bounds of [46] and have the advantage of being easier to understand theoretically than TS. As such, UCB also deserves our attention in addressing the spatial reuse problem in WLANs.

*[51] Garivier et al. (2011), "The KL-UCB algorithm for bounded stochastic bandits and beyond"*

[23] *All $\mathcal{R}_k$ are bounded distributions.*

*[52] Chu et al. (2011), "Contextual bandits with linear payoff functions"*

[24] *Additional information about the arms is embedded into a context vector that is observable.*

### 2.2.4 *Bayesian Optimization*

Eventually, we consider Bayesian Optimization (BO) as a candidate for addressing the spatial reuse problem in WLANs. The vast majority of BO algorithms rely on Gaussian processes, so let us introduce them with our notations.

**Gaussian Process.** A Gaussian Process (GP) is a stochastic process, that is, a collection of random variables $\{Y(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{C}}$ indexed by a set $\mathcal{C}$. The specificity of a *Gaussian* process is that any finite set of random variables $\{Y(\boldsymbol{x}_1), \cdots, Y(\boldsymbol{x}_n)\}$ has a joint multivariate Gaussian distribution. It can be fully specified by its mean function $\mu$ and its covariance function $k$

$$\mu(\boldsymbol{x}) = \mathbb{E}\left[Y(\boldsymbol{x})\right] \tag{2.8}$$

$$k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbb{E}\left[\left(Y(\boldsymbol{x}) - \mu(\boldsymbol{x})\right)\left(Y(\boldsymbol{x}') - \mu(\boldsymbol{x}')\right)\right]. \tag{2.9}$$

In most cases, a BO algorithm relies on the pioneering work [53] that proposes a unified framework for regression tasks using GPs. We describe the approach in a few lines and we illustrate it with Figure 2.3. A GP serves as a surrogate model for the black-box objec-

*[53] Williams et al. (1995), "Gaussian Processes for Regression"*

tive function $f$. It requires a prior GP, describing the random variables $\{f(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{C}}$, a prior mean function $\mu_0(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x},\boldsymbol{x}')$. Throughout this thesis, we assume without loss of generality, as the vast majority of BO algorithms does, that $\mu_0(\boldsymbol{x}) = 0$. For any point $\boldsymbol{x} \in \mathcal{C}$, it is known by construction that $f(\boldsymbol{x}) \sim \mathcal{N}\left(\mu_0(\boldsymbol{x}), \sigma_0^2(\boldsymbol{x})\right)$, with $\sigma_0^2(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x})$. Given $t$ observations, described by the $t \times d$ matrix of datapoints $\boldsymbol{X} = (\boldsymbol{x}_i)_{i\in[1,t]}^\top$ and by the corresponding $t$-dimensional vector of outputs $\boldsymbol{y} = (y_i)_{i\in[1,t]}^\top$, the posterior distribution of $f(\boldsymbol{x})$ is obtained from the $t+1$-dimensional joint Gaussian distribution of the $t$ observations along with the queried $f(\boldsymbol{x})$, by conditioning on the observed data. Then, a posteriori, $f(\boldsymbol{x}) \sim \mathcal{N}\left(\mu_t(\boldsymbol{x}), \sigma_t^2(\boldsymbol{x})\right)$ with

$$\mu_t(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x},\boldsymbol{X})^\top \boldsymbol{K}^{-1}\boldsymbol{y} \tag{2.10}$$

$$\sigma_t^2(\boldsymbol{x}) = \sigma_0^2(\boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x},\boldsymbol{X})^\top \boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x},\boldsymbol{X}) \tag{2.11}$$

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

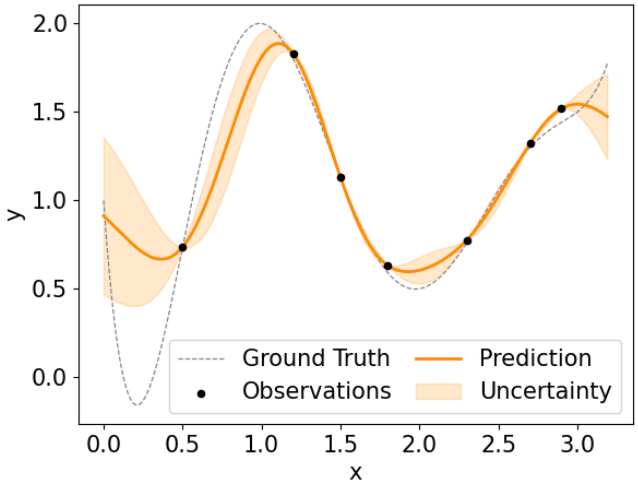



Figure 2.3: Gaussian Process Regression. The ground truth $f(\boldsymbol{x})$ is depicted with a grey dashed line and the noisy observations $(\boldsymbol{X},\boldsymbol{y})$ with black dots. The posterior mean $\mu_t(\boldsymbol{x})$ is shown with a solid orange line and the posterior standard deviation $\sigma_t(\boldsymbol{x})$ as an orange interval.

As for any BBO algorithm, a BO algorithm needs to provide, at each time step $t$, a query $\boldsymbol{x}^t$ that achieves a trade-off between exploration and exploitation. The BO algorithm typically bases its querying policy on the maximization of an acquisition function $\varphi_t(\boldsymbol{x})$ that quantifies the benefits of observing $f(\boldsymbol{x})$ in terms of exploration and

exploitation. Common acquisition functions include Knowledge Gradient [54], Probability of Improvement [55], Expected Improvement (EI) [56] and, as mentionned before, UCB [13]. Each of them captures a different definition of what is a good compromise between exploration and exploitation. They also vary widely in terms of computational complexity. The latter (UCB), is one of the least complex acquisition functions and leads to an asymptotically optimal application to GPs, with GP-UCB [57], defined as

$$\varphi_t(\boldsymbol{x}) = \mu_t(\boldsymbol{x}) + \beta_t^{\frac{1}{2}} \sigma_t(\boldsymbol{x}). \tag{2.12}$$

One could recognize the general formulation of UCB, introduced in (2.7). Intuitively, $\mu_t(\boldsymbol{x})$, as a best-guess estimation of $f(\boldsymbol{x})$, is the exploitation term. Conversely, $\sigma_t(\boldsymbol{x})$ describes the uncertainty of the prediction and is a suitable exploration term. As for $\beta_t^{1/2}$, it is used to control the contribution of $\sigma_t(\boldsymbol{x})$ to $\varphi_t(\boldsymbol{x})$, ensuring that $\mu_t(\boldsymbol{x}) - \beta_t^{1/2}\sigma_t(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq \mu_t(\boldsymbol{x}) + \beta_t^{1/2}\sigma_t(\boldsymbol{x})$ with high probability. Note that other acquisition functions can exhibit no-regret performance. As an example, under simple modifications of the initial algorithm introduced in [56], EI is also asymptotically optimal [58].

At each time $t$, the BO algorithm must find $\boldsymbol{x}^t = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \varphi_t(\boldsymbol{x})$ to query it. However, finding the maximum of $\varphi_t(\boldsymbol{x})$ is a challenge in itself and represents the most expensive task in an iteration. To address it, the vanilla BO algorithm uses a grid-search or a global optimization algorithm such as DIRECT (see Section 2.2.1).

Although classical BO offers no-regret guarantees and has demonstrated excellent empirical performance in a variety of domains [10, 11, 59], it suffers from the same drawbacks as the grid-search or DIRECT. In particular, it struggles with high-dimensional objective functions, such as those typically found in the optimization of the spatial reuse in WLANs. Nevertheless, in this thesis, we explore its application to the spatial reuse problem in Chapter 4 and we propose a novel way to extend BO to high-dimensional spaces in Chapter 6.

[54] *Gupta et al. (1996), "Bayesian look ahead one-stage sampling allocations for selection of the best population"*

[55] *Jones et al. (1998), "Efficient global optimization of expensive black-box functions"*

[56] *Mockus (1994), "Application of Bayesian approach to numerical methods of global and stochastic optimization"*

[57] *Srinivas et al. (2012), "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting"*

[58] *Qin et al. (2017), "Improving the expected improvement algorithm"*

[59] *Bergstra et al. (2013), "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures"*

## 2.3 SUMMARY

In this section, we summarize this chapter in a few points:

- The spatial reuse problem in WLANs is a high-dimensional problem that analytical approaches do not address properly in practice.
- Data-driven approaches obtain the best empirical performance in this problem so far, but they do not offer any theoretical guarantee and are evaluated on vanilla scenarios.
- Regarding BBO techniques, BO and the MAB framework stand out as great candidates in our quest for a theoretically supported solution to the spatial reuse problem in WLANs.

- In the remaining of this thesis, we explore the application of the MAB framework and BO to the spatial reuse problem (Chapters 3 and 4), to the power control of cellular networks (Chapter 5) and we contribute to the extension of BO to high-dimensional spaces (Chapter 6).

# 3 MULTI-ARMED BANDIT APPROACHES FOR 802.11

**Outline.** We propose to address the spatial reuse problem in a Wireless Local Area Network (WLAN) (see Section 1.1.2) with the Multi-Armed Bandit (MAB) framework (see Section 2.2.3). We describe two similar, centralized MAB approaches that dynamically adapt the transmission power (`TX_PWR`) and the sensitivity threshold (`OBSS_PD`) of each Access Point (AP) in a WLAN. They rely on different subsampling strategies of the configuration space $\mathcal{C}$, under a regularity assumption of the objective function $f$. In this chapter, $f$ is an ad-hoc performance metric designed to keep each Station (STA) above a given throughput threshold. To demonstrate their empirical effectiveness, the two proposed solutions are evaluated against state-of-the-art strategies by simulating increasingly complex scenarios, using a well-recognized network simulator.

## 3.1 SYSTEM UNDER STUDY

We consider a WLAN comprising multiple APs and stationary STAs as well as a controller that configures and manages the WLAN. STAs are associated to the AP with the strongest signal strength. To access the radio channel, APs and STAs use a listen-before-talk scheme referred to as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Clear Channel Assessment (CCA), as described in Section 1.1.2. The WLAN implements the 802.11ax amendment, and the parameters `TX_PWR` and `OBSS_PD` meet the constraint (1.1). Additionally, for the most part of this chapter, the WLAN workload is considered at saturation. That is, the APs always have data to transmit to their associated STAs. Variations of the workload are discussed and explored in Section 3.4.3.

Several performance metrics are considered worthy of interest to evaluate the efficiency of a WLAN at providing wireless access to its STAs. First, the aggregate throughput (also known as system throughput or cumulative throughput) represents the sum of the throughputs of all individual STAs in the WLAN. Second, the fairness in the distribution of access to the radio channel among STAs is another critical factor. Measures of fairness such as Jain's index [60] are common means to determine whether certain STAs are receiving a disproportionate share of the radio resource at the expense of other STAs. Indeed, certain STAs may struggle to access the radio channel due to an unfavorable location in the WLAN. These STAs are said to be *in starvation of throughput* and they represent a major issue for network

*[60] Jain et al. (1984), "A quantitative measure of fairness and discrimination"*

administrators. In this chapter, a STA is considered to be starving if it cannot obtain at least a percentage $\alpha = 10\%$ of the throughput they would have in the absence of other STAs. Third, Frame Error Rate (FER) of each STA, which indicates the percentage of frames lost due to collisions and poor channel condition, can also be worthy of interest to network administrators. As discussed earlier in Chapter 1, the dynamical setting of `TX_PWR` and `OBSS_PD` on each AP can significantly change these performance metrics.

The controller is the perfect device to implement our optimization algorithm, as it is assumed to have access to every other device in the WLAN. It can observe the throughput of each STA, and set the parameters of each AP. It is also assumed to have enough computing power to run complex solutions in an acceptable amount of time, unlike the average WLAN device that favors simplicity and power saving. Controllers satisfying these assumptions ease the management and the configuration of the WLAN devices. They are typically found in WLANs comprising a large fleet of APs and operated by a single entity (e.g. in malls, train stations or open spaces). However, assuming the existence of a controller is less realistic when one considers a collection of WLANs, each operated by a different entity.

## 3.2 PROBLEM FORMULATION

We are looking for a correct network configuration $\boldsymbol{x} \in \mathcal{C}$ of a WLAN comprising $n$ APs. As stated in Section 3.1, $\mathcal{C}$ is defined as the set of all the WLAN configurations that satisfy the constraint (1.1), that is $\mathcal{C} = \{\boldsymbol{x} | \boldsymbol{x} \in \{-82, -62\} \times \{1, 21\} \text{ and } \boldsymbol{x} \text{ satisfies (1.1)}\}^n$. More intuitively, $\mathcal{C}$ can be described as the set resulting from the Cartesian product of $n$ triangular, uniform 2-dimensional grids.

To find a good configuration in $\mathcal{C}$, we use a Black-Box Optimization (BBO) algorithm located in the WLAN controller. We assume that the controller receives regular reports on the individual throughputs of the $n_S$ STAs $\boldsymbol{T}(\boldsymbol{x}) = (T_1(\boldsymbol{x}), \cdots, T_{n_S}(\boldsymbol{x}))$ under any configuration $\boldsymbol{x} \in \mathcal{C}$. As for now, note that, for the sake of simplicity, the argument of the throughputs will be omitted, such that they will be denoted $\boldsymbol{T} = (T_1, \cdots, T_{n_S})$. The controller uses the MAB framework, and consider that each network configuration $\boldsymbol{x}_i \in \mathcal{C}$ represents an arm that can be pulled at any time $t$ to obtain a noisy objective value (a.k.a. a *reward*) $y_i^t$. Note that $y_i^t$ is drawn from a probability distribution $\mathcal{R}(\boldsymbol{\theta}_i)$ whose parameters $\boldsymbol{\theta}_i$ are unknown from the agent but invariant in time since the network under study is assumed to be stationary. The expected reward of a configuration $\boldsymbol{x}_i$ is, as one could expect, the objective value $f(\boldsymbol{x}_i)$, with $f$ remaining to be defined. Our objective boils down to find an efficient policy $\pi : \mathbb{N} \to \mathcal{C}$ that determines which arm to pull (*i.e.* which WLAN configuration in $\mathcal{C}$ to test) at each time step $t$ given the previously collected data $\mathcal{D} = \left\{ \left( \boldsymbol{x}_{\pi(i)}, y^i_{\pi(i)} \right) \right\}_{i \in [1, t-1]}$.

Given the exponential growth of the cardinality of the configuration space $\mathcal{C}$ with the number $n$ of APs (bounded by $21^{2n}$ since an AP has two tunable parameters, each taking at most 21 values), our problem is more precisely framed as an Infinitely Many-Armed Bandit (IMAB). Thus, in practice, the WLAN controller cannot explore the whole set of arms in a reasonable amount of time and must instead work on a subset of $\mathcal{C}$, referred to as *the reservoir*.

In the remaining of this section, we describe how the objective function is designed and we give some insights about how $\mathcal{C}$ is subsampled to build the reservoir. We propose two subsampling strategies described in Sections 3.4 and 3.5.

### 3.2.1 *Objective Function*

The design of the black-box objective function $f$ is a critical step as it can deeply influence the outcome of the optimization process. In the case of a WLAN, the reward function aims at quantifying the quality of a network configuration in terms of spatial reuse through multiple performance metrics, as discussed in Section 3.1. There exist different ways of combining these performance metrics to obtain a scalar objective function to optimize. In this chapter, we adopt the standpoint of a network administrator that considers a WLAN configuration favorable if it ensures a fair share of throughput among the STAs. More precisely, we enumerate, by order of importance, three criteria to take into account:

(i) the number of STAs that are starving for throughput should be minimized,

(ii) the fairness between the STAs should be maximized,

(iii) the aggregate throughput of the WLAN should be maximized.

Satisfying criteria (ii) and (iii) at the same time is challenging. Indeed, in most WLAN topologies, increasing the fairness between the STAs is made at the expense of a lower aggregate throughput. Conversely, increasing the network aggregate throughput often implies a decrease in fairness. To address this problem, we consider the Proportional Fairness (PF) of the throughputs of the STAs. PF is widely used in computer networks to reach a natural trade-off between a high sum and a low variance in a sample (e.g. see [61, 62]). We focus on a normalized version of the PF of the STAs throughputs $\boldsymbol{T} = (T_1, \cdots, T_{n_S})$, given by

$$\text{PF}\left(\boldsymbol{T}, \boldsymbol{T}^*\right) = \prod_{i=1}^{n_S} \frac{T_i}{T_i^*} \tag{3.1}$$

with $\boldsymbol{T}^* = \left(T_1^*, \cdots, T_{n_S}^*\right)$ the attainable throughputs of the STAs, defined as the throughput that each STA would obtain without any inter-

*[61] Jiang et al. (2005), "Proportional fairness in wireless LANs and ad hoc networks"*

*[62] Li et al. (2008), "Proportional fairness in multi-rate wireless LANs"*

ference. This information is easily obtained through simple analytical models or simulations.

Regarding criterion (i), recall that we consider that a STA is in starvation if its effective throughput is lower than $\alpha = 10$ % of its attainable throughput, that is $T_i < \alpha T_i^*$. We account for criterion (i) by enforcing the following constraint on the objective function $f$:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{C}, f(\boldsymbol{x}) \geq f(\boldsymbol{x}') \Leftrightarrow S(\boldsymbol{x}) \leq S(\boldsymbol{x}') \tag{3.2}$$

with $S(\boldsymbol{x}) = \sum_{i=1}^{n_S} \mathbb{1}_{T_i < \alpha T_i^*}$ counting the number of STAs in starvation under the configuration $\boldsymbol{x}$.

To enforce the constraint (3.2), the indices of STAs are partitioned into two subsets: $\mathcal{T}^- = \{i | T_i < \alpha T_i^*\}$, which includes the indices off all starving STAs, and its complementary $\mathcal{T}^+ = \{i | T_i \geq \alpha T_i^*\}$. Then, the ad-hoc objective function $f$ is defined as

$$f(\boldsymbol{x}) = \frac{|\mathcal{T}^-| \mathrm{PF}\left(\boldsymbol{T}_{\mathcal{T}^-}, \boldsymbol{T}^*_{\mathcal{T}^-}\right) + |\mathcal{T}^+| \left(n_S + \mathrm{PF}\left(\boldsymbol{T}_{\mathcal{T}^+}, \boldsymbol{T}^*_{\mathcal{T}^+}\right)\right)}{n_S(n_S+1)}. \tag{3.3}$$

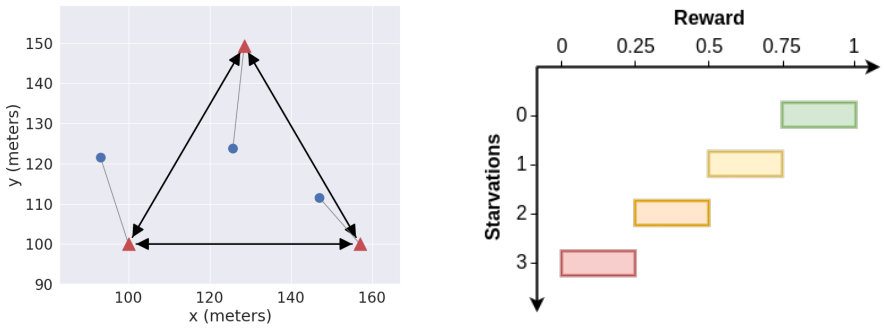



Figure 3.1: Ad-hoc objective function intervals w.r.t. the number of starving STAs in a toy example WLAN topology. The APs are depicted as red triangles, the associated STAs as blue circles and the conflicts between APs as black arrows.

By construction, the objective function $f$ returns a value within $[0, 1]$ and meets Property (3.2). That is, with the objective function (3.3), a configuration $\boldsymbol{x}$ with a given number of starving STAs will always be better valued than another configuration $\boldsymbol{x}'$ with a larger number of starving STAs. Figure 3.1 depicts the reward intervals with respect to the number of STAs in starvation in a toy WLAN topology. As an aside, note that due to the importance of conflicts in the understanding of a WLAN behavior, it is a common practice to represent WLANs by their conflict graph between APs (e.g. [63, 64]). Conflicts between STAs are typically not represented in conflict graphs as the vast majority of traffic in WLANs is downstream. The toy WLAN comprises $n_S = 3$ STAs, and $n_S$ directly determines the number of reward intervals ($n_S + 1 = 4$ intervals in Figure 3.1). The number of starving STAs $S(\boldsymbol{x})$ under a configuration $\boldsymbol{x}$ indicates the reward interval within which lies $f(\boldsymbol{x})$.

*[63] Mishra et al. (2006), "A Client-Driven Approach for Channel Management in Wireless LANs"*

*[64] Garetto et al. (2008), "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks"*

### 3.2.2 *Problem Breakdown*

A common practice to circumvent the infinite number of arms in an IMAB problem consists of restricting the exploration to a limited subset of solutions composed of random arms that constitute a reservoir. Typically, without any information indicating which arm to prefer, the arms in the reservoir are drawn uniformly (e.g., [42, 65, 66]). In our case, this approach is not suitable as the vast majority of network configurations leads to poor solutions. However in our case, unlike a typical IMAB problem, in which no hypothesis can be made on the relationship between the arms and their rewards, it is safe to assume some regularity about the objective function $f$.

*[65] David et al. (2014), "Infinitely many-armed bandits with unknown value distribution"*
*[66] Aziz et al. (2018), "Pure exploration in infinitely-armed bandit models with fixed-confidence"*

**Assumption 3.1.** *The objective function $f$ satisfies*

$$\exists L > 0, \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{C}, ||\boldsymbol{x}_i - \boldsymbol{x}_j||_1 = 1 \implies |f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)| < L \quad (3.4)$$

Intuitively, Assumption 3.1 states that the objective values of two neighboring configurations cannot be further apart than $L$. Note that, if the configuration space $\mathcal{C}$ was assumed to be continuous, Assumption 3.1 would translate into assuming that $f$ is $L$-Lipschitz continuous. Note that this assumption is of interest if $L$ is not too large. This property provides useful information that can be leveraged to sample more promising configurations than those provided by a naive uniform sampling. Therefore, our problem breaks down into two subproblems that must be solved concurrently:

(i) leverage Assumption 3.1 to sample promising configurations in $\mathcal{C}$ for the reservoir,

(ii) identify the best arm in the reservoir and pull it as much as possible.

In the remaining of this chapter, we consider two algorithms: *the sampler*, in charge of (i) and *the optimizer*, in charge of (ii). Figure 3.2 summarizes the main principles of our solution: the optimizer requests a new configuration to the sampler, which selects $\boldsymbol{x}_i$ and returns it to the optimizer. The optimizer tests the configuration $\boldsymbol{x}_i$ on the WLAN, obtains a reward $y_i$ and forwards it to the sampler. Both agents update their internal states according to the new observed couple $(\boldsymbol{x}_i, y_i)$. The next sections in this chapter are devoted to the description and evaluation of the optimizer (Section 3.3) and two different samplers (Sections 3.4 and 3.5).

## 3.3 EXPLOITATION IN THE RESERVOIR

The objective of the optimizer is to quickly identify the best configuration $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} f(\boldsymbol{x})$ among the current reservoir of configurations and to pull it as much as possible. The reservoir is periodically

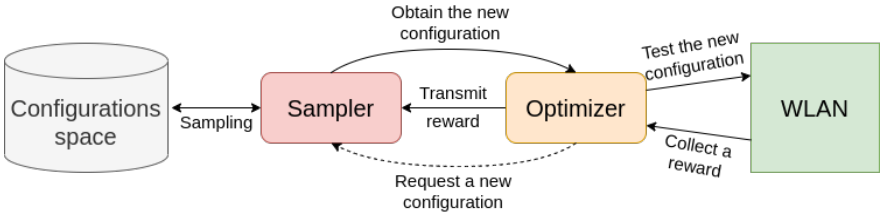



Figure 3.2: Our MAB solution oultine.

updated with a new configuration whenever the optimizer requests it to the sampler.

To quickly identify the best arm in the reservoir, we consider Thompson Sampling (TS), based on [47] and introduced in Section 2.2.3. Recall that TS requires some prior knowledge about the objective values: $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ the distribution of the noisy reward given the configuration $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$, but also $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathcal{D})$, the prior and posterior distributions of the parameters $\boldsymbol{\theta}$ themselves, respectively. TS was used in previous works for the spatial reuse problem in WLANs [8, 30]. In these works, it is assumed that $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ is the p.d.f. of $\mathcal{N}(f(\boldsymbol{x}), 1)$. In other words, the variance is assumed to be fixed at 1, and the parameter $f(\boldsymbol{x})$ has a Gaussian prior, initially set at $\mathcal{N}(0, 1)$.

Although assuming a fixed variance for the distribution of all the arms simplifies the calculations, this limits the domain of applicability of the solution and makes it less effective in practice, as demonstrated later in Section 3.4.2. To prevent this, we design a solution under the following assumptions.

**Assumption 3.2.** *For all $\boldsymbol{x}_i \in \mathcal{C}$, $p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$ is the p.d.f. of $\mathcal{N}\left(f(\boldsymbol{x}_i), \sigma_i^2\right)$, with $\boldsymbol{\theta} = \left(f(\boldsymbol{x}_i), \sigma_i^{-2}\right)$.*

**Assumption 3.3.** *The prior distribution of $\boldsymbol{\theta}$ is a Normal-Gamma.*

Intuitively, Assumption 3.2 specifies the distribution of the noisy rewards (*i.e.* they are assumed to be Gaussian, conditioned by some parameters $\boldsymbol{\theta}$), while Assumption 3.3 describes the uncertainty relative to the parameters $\boldsymbol{\theta}$ themselves. Note that Assumption 3.3 refers to the mean and the precision[25] of the Gaussian rewards distribution. We provide more details regarding the Normal-Gamma distribution in the next paragraph.

[25] *The inverse of the variance $\sigma^2$.*

**Normal-Gamma distribution.** A Normal-Gamma distribution is a bivariate distribution of four parameters $(\mu, \lambda, \alpha, \beta)$. It refers to the distribution of a couple of random variables $(X, Y)$, such that $Y \sim \Gamma(\alpha, \beta)$ and $X|Y \sim \mathcal{N}\left(\mu, \frac{1}{\lambda Y}\right)$. Therefore, sampling from a Normal-Gamma boils down to sampling $Y \sim \Gamma(\alpha, \beta)$, before sampling $X \sim \mathcal{N}\left(\mu, \frac{1}{\lambda Y}\right)$. In this thesis, we denote the Normal-Gamma distribution as $\mathcal{N}\Gamma(\mu, \lambda, \alpha, \beta)$.

Standard Bayesian calculations (e.g. see [67]) show that the Normal-

[67] *Bolstad et al. (2016),* Introduction to Bayesian statistics

Gamma distribution is a conjugate prior for the unknown parameters of a Gaussian distribution. We introduce and apply this concept with our notations. Given a configuration $\boldsymbol{x}_i$, we are looking for the parameters of its reward distribution, that is $\mathcal{N}\left(\mu_i, \sigma_i^2\right)$ according to Assumption 3.2, with $\mu_i = f(\boldsymbol{x}_i)$. Given Assumption 3.3, we have $\left(\mu_i, \sigma_i^{-2}\right) \sim \mathcal{NT}\left(\mu_i^0, \lambda_i^0, \alpha_i^0, \beta_i^0\right)$. After observing an i.i.d. sample $\boldsymbol{Y} = (y_1, \cdots, y_n)$ of size $n$, with $\forall k \in [1, n], y_k \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$, the posterior distribution $\left(\mu_i, \sigma_i^2\right) | \boldsymbol{Y}$ can be computed with Bayes' theorem and yields another Normal-Gamma[26] distribution $\mathcal{NT}\left(\mu_i^n, \lambda_i^n, \alpha_i^n, \beta_i^n\right)$ with parameters

[26] *Hence the denomination "conjugate prior".*

$$\mu_i^n = \frac{\lambda_i^0 \mu_i^0 + n\bar{y}}{\lambda_i^0 + n}, \tag{3.5}$$

$$\lambda_i^n = \lambda_i^0 + n, \tag{3.6}$$

$$\alpha_i^n = \alpha_i^0 + \frac{n}{2}, \tag{3.7}$$

$$\beta_i^n = \beta_i^0 + \frac{1}{2}\left(ns + \frac{\lambda_i^0 n(\bar{y} - \mu_i^0)^2}{\lambda_i^0 + n}\right), \tag{3.8}$$

with $\bar{y}$ the empirical mean of the sample $\boldsymbol{Y}$ and $s$ its empirical variance.

Now that a posterior distribution for the parameters $\boldsymbol{\theta}$ has been found, we describe the optimizer in Algorithm 3.1. The algorithm has two main parameters: the exploration rate $\epsilon \in ]0, 1[$ and the sample size $n \in [2, +\infty[$, that respectively determine the probability of adding a new configuration to the reservoir and how often the Normal-Gamma parameters of a configuration are updated. With line 4, the optimizer determines if the current step is an exploration step (by entering the `if` block) or an exploitation term (by jumping to line 12). If an exploration step is chosen, the optimizer requests a new configuration $\boldsymbol{x}_i$ to the sampler (line 5) and tests it $n$ consecutive times on the environment to collect a whole sample $\boldsymbol{Y}_i$ (line 6). Then, the a priori Normal-Gamma distribution parameters are initialized from simple empirical statistics of the sample $\boldsymbol{Y}_i$ (line 9) and the configuration $\boldsymbol{x}_i$ is added to the reservoir $R$ (line 11).

TS is executed if the optimizer chooses exploitation. Between lines 13-16, the Normal-Gamma distributions of all the configurations in the reservoir are sampled once and the configuration $\boldsymbol{x}_j$ obtaining the best mean parameter $\mu_j$ for this sampling is determined on line 17. Doing so, each configuration has a probability of being selected proportional to its probability of being the optimal configuration[27] in the reservoir, according to the current beliefs of the optimizer. Then, the chosen configuration is tested and the obtained reward is added to the corresponding rewards sample $\boldsymbol{Y}_j$ (line 18). If $\boldsymbol{Y}_j$ contains enough rewards, the parameter of the corresponding Normal-Gamma are updated and $\boldsymbol{Y}_j$ is reset between lines 20-23. Eventually, the data collected during this step is forwarded to the sampler with line 25.

[27] *That is, being the configuration with the largest expected reward.*

**Algorithm 3.1** Optimizer

**Input**: sample size $n$, exploration rate $\epsilon$

$R \leftarrow \emptyset$
$t \leftarrow 0$
**loop**
  **if** $R = \emptyset$ **or** rand() $< \epsilon$ **then**
    Get a new configuration $\boldsymbol{x}_i$ using the sampler
    Test $\boldsymbol{x}_i$ $n$ times on the environment and collect rewards in $\boldsymbol{Y}_i$
    Compute its mean $\bar{y}$ and its variance $s$
    $t \leftarrow t + n$
    $(\mu_i^t, \lambda_i^t, \alpha_i^t, \beta_i^t) \leftarrow \left(\bar{y}, n, \frac{n}{2}, \frac{ns}{2}\right)$
    $\boldsymbol{Y}_i \leftarrow \emptyset$
    $R \leftarrow R \cup \{\boldsymbol{x}_i\}$
  **else**
    **for** $\boldsymbol{x}_i$ **in** $R$ **do**
      Sample $g_i$ from $\Gamma\left(\alpha_i^k, \beta_i^k\right)$
      Sample $\mu_i$ from $\mathcal{N}\left(\mu_i^k, \left(\lambda_i^k g_i\right)^{-1}\right)$
    **end for**
    $j \leftarrow \operatorname{argmax}_i \mu_i$
    Test $\boldsymbol{x}_j$ on the environment and add the reward to $\boldsymbol{Y}_j$
    $t \leftarrow t + 1$
    **if** $|\boldsymbol{Y}_j| = n$ **then**
      Update prior parameters $\left(\mu_j^k, \lambda_j^k, \alpha_j^k, \beta_j^k\right)$ using (3.5)-(3.8)
      $\boldsymbol{Y}_j \leftarrow \emptyset$
    **end if**
  **end if**
  Forward tests and rewards to the sampler
**end loop**

Algorithm 3.1 has a computational complexity proportional to the number of configurations in $R$, since it needs to sample as many distributions as there are configurations in the reservoir. Since the number of configurations in the reservoir depends directly on $\epsilon$ and the time step $t$, the computational complexity of Algorithm 3.1 is in $\mathcal{O}(\epsilon t)$.

## 3.4 EXPLORATION WITH GAUSSIAN MIXTURES

### 3.4.1 *Subsampling with a Gaussian Mixture*

In this section, we describe and evaluate our first proposition to identify promising configurations by subsampling the configuration space $\mathcal{C}$. First, let us leverage Assumption 3.1 by simply extending the property to configurations separated by an arbitrary distance:

**Proposition 3.1.**

$$\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{C}, ||\boldsymbol{x}_i - \boldsymbol{x}_j||_1 = n \implies |f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)| \leq Ln. \tag{3.9}$$

*Proof.* The proof by recurrence is immediate. Assumption 3.1 initializes our recurrence for $n = 1$. Then, assuming that the property is true for an arbitrary distance $n$, let us consider $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{C}$ such that $||\boldsymbol{x}_i - \boldsymbol{x}_j||_1 = n + 1$. We build $\boldsymbol{x}_k$ on the sphere of radius 1 centered on $\boldsymbol{x}_j$ such that $||\boldsymbol{x}_i - \boldsymbol{x}_k||_1 = n$ and $||\boldsymbol{x}_k - \boldsymbol{x}_j||_1 = 1$. Such $\boldsymbol{x}_k$ always exists with the characterization of $\mathcal{C}$ provided in Section 3.2. Then, by the triangle inequality, Assumption 3.1 and the recurrence assumption, we have

$$\begin{aligned} |f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)| &\leq |f(\boldsymbol{x}_i) - f(\boldsymbol{x}_k)| + |f(\boldsymbol{x}_k) - f(\boldsymbol{x}_j)| \\ &\leq Ln + L \\ &= L(n+1) \end{aligned}$$

□

Intuitively, Proposition 3.1 states that new promising configurations can be found near the best configurations discovered so far. Moreover, it states that a configuration $\boldsymbol{x}_i$ with an objective value $f(\boldsymbol{x}_i) = f(\boldsymbol{x}_j) + Ln$, if it exists, would be sampled at least $n$ units away from $\boldsymbol{x}_j$. To leverage this insight, we consider a Gaussian Mixture (GM) of $K$ distributions centered on the $K$ best configurations $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ discovered so far (*i.e.* until the current time $t$). To ensure the isotropy of the Gaussian distributions, we define their covariance matrices as scalar matrices $\boldsymbol{\Sigma}_i^t = \lambda_i^t \boldsymbol{I}_d$, with $d = \dim \mathcal{C}$. Recalling that $R$ is the reservoir, we consider $\mu_*^t = \max_{i \in [1, |R|]} \mu_i^t$ and we set

$$\lambda_i^t = \frac{(\mu_*^t + L - \mu_i^t)^2}{dL^2} \tag{3.10}$$

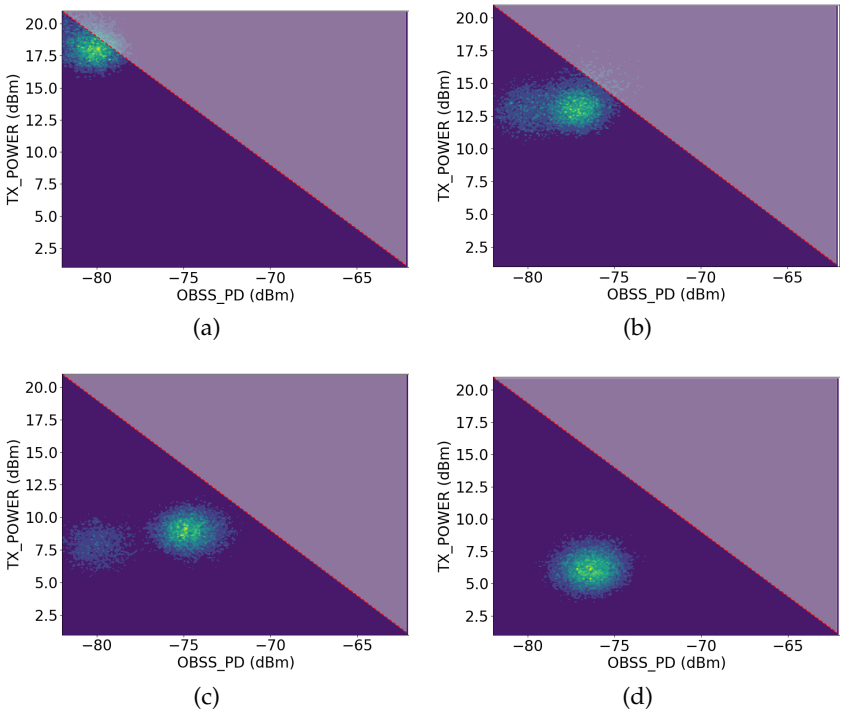



Figure 3.3: Illustration of the Gaussian mixture density (with $K = 3$) in the configuration space $\mathcal{C}$ of a single AP for four iterations of the sampler. The grayed upper triangle of the space $\mathcal{C}$ depicts the configurations that violate the constraint (1.1).

such that $\mathrm{tr}\left(\mathbf{\Sigma}_i^t\right) = \mathbb{E}\left[||\boldsymbol{x} - \boldsymbol{x}_i||_2^2\right] = \left(\frac{\mu_*^t + L - \mu_i^t}{L}\right)^2$, with $\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{x}_i, \Sigma_i^t\right)$.

With (3.10), a Gaussian from the mixture gets a low variance if it is centered on a configuration $\boldsymbol{x}_i$ that has an expected reward $\mu_i^t$ close to $\mu_*^t$. Conversely, it gets a larger variance if it is centered on a configuration that has a lower expected reward. Figure 3.3 illustrates this phenomenon by depicting four different iterations of the sampler with a mixture of $K = 3$ Gaussian distributions. For visualization purposes, we consider the parameters of a single AP, so that the configuration space $\mathcal{C}$ is only 2-dimensional. The GM is initiated (Figure 3.3a) near the default configuration of 802.11,[28] and navigates through the configuration space (Figures 3.3b and 3.3c), similarly to a gradient ascent algorithm. Eventually, the GM finds a basin of attraction towards which all the Gaussian distributions converge (Figure 3.3d).

[28] *That is,* $(-82, 20)$ *dBm for all APs.*

Algorithm 3.2 details the sampler approach described in this section. At each call by the optimizer, the sampler updates its state with the new provided observation (lines 1-3). Then, the Gaussian mixture is built, based on the $K$ best observed configurations in the history $H$ (lines 4-10) according to (3.10) as illustrated by line 8. Eventually, a new promising configuration $\boldsymbol{x}$ is sampled (line 11) and returned to the optimizer (line 12). Algorithm 3.2 has a computational complexity that directly depends on the size of the history $H$ (that grows with $t$)

**Algorithm 3.2** Gaussian Mixture-based Sampler

**Input**: Lipschitz constant $L$, number of Gaussians $K$, observation $(\boldsymbol{x}_i, y_i)$

1: Retrieve history $H$ and time step $t$
2: Add the new observation $(\boldsymbol{x}_i, y_i)$ to $H$
3: $t \leftarrow t + 1$
4: Find the $K$ $(\boldsymbol{x}_j, \mu_j^t)$ pairs in $H$ of largest expected rewards $\mu_j^t$
5: $\mu_*^t \leftarrow \max_j \mu_j^t$
6: $(G, W) \leftarrow (\emptyset, \emptyset)$
7: **for** $j \leftarrow 1$ **to** $K$ **do**
8: Add $\left\{\left(\boldsymbol{x}_j, \frac{\left(\mu_*^t + L - \mu_j^t\right)^2}{dL^2}\boldsymbol{I}\right)\right\}$ to $G$
9: Add $\mu_j^t$ to $W$
10: **end for**
11: Sample a new configuration $\boldsymbol{x}$ from mixture $(G, W)$
12: Transmit $\boldsymbol{x}$ to the optimizer

and on the number of Gaussian distributions $K$. Therefore, it has a computational complexity in $\mathcal{O}(K + t)$.

### 3.4.2 *Numerical Results*

#### 3.4.2.1 *Experimental Settings*

We evaluate the efficiency of this MAB solution at improving the spatial reuse of a WLAN by comparing it to other existing state-of-the-art strategies in the well-recognized, realistic, discrete-event simulator ns-3 [31]. The ns-3 code implementing our solution, the other strategies, as well as the considered WLAN topologies described below are available for download.[29] We compare five different strategies:

[29] https://github.com/abardou/IMAB-SR-STA-802.11ax

- `DEFAULT`: the legacy configuration of 802.11, with the configuration $(-82, 20)$ dBm for `TX_PWR` and `OBSS_PD`, enforced on all APs of the WLAN,
- $\epsilon$-`GREEDY`: the $\epsilon$-greedy strategie described in Section 2.2.3,
- `UNIF+GTS`: the state-of-the-art strategy [30], it is based on TS but with Gaussian priors and discovers new configurations through a uniform sampling,
- `GM+GTS`: it is similar to `UNIF + GTS` but discovers new configurations with our sampler (Gaussian mixture), described in Section 3.4.1,
- `GM+NGTS`: our proposition, it is based on TS with Normal-Gamma priors (see Section 3.3) and discovers new configurations with our sampler (Gaussian mixture), described in Section 3.4.1.

Note that considering GM+GTS helps us separate the benefits brought by the optimizer algorithm and the sampler algorithm. As a matter of fact, GM+GTS only differs from UNIF+GTS because it uses our sampler, and GM+NGTS only differs from GM+GTS because it uses our optimizer.

In each of our experiments, the simulation runs last for a total of 120 seconds of simulated time. For the sake of accuracy, each simulation was replicated 25 independent times. The duration of a test, corresponding to the time during which performance metrics are measured before being sent to the WLAN controller, was set to 50 ms (milliseconds). Therefore, 2,400 optimization steps can be performed before the simulation ends. Because we replicate 25 times each simulation, we obtain a matrix of $25 \times 2,400$ measures for each network performance metric. In order to provide a clear visualization of this large set of data, we chose to plot the median of the metric at each optimization step, framed by its first and third quartiles. Finally, we applied an Exponential Moving Average (EMA) to the three considered quartiles, to smooth the signals and focus on the trends caused by the optimization. This kind of visualization gives us an insight not only into the final performance of each strategy but also on its performance during the whole optimization process. The other parameters of the ns-3 simulation are given by Table 3.1. For the sake of comparison, all strategies are evaluated using the same simulation parameters as well as the same reward function, described in Section 3.2.1.

Table 3.1: ns-3 parameters.

| **Parameter** | **Value** |
|---|---|
| ns-3 version | 3.31 |
| Number of repetitions | 25 |
| Simulation duration | 120 s |
| Test duration | 50 ms |
| Packet size | 1,464 Bytes |
| Frequency band | 5 GHz |
| A-MDPU Aggregation | 4 |
| Path loss | `LogDistancePropagationLossModel` |
| MCS Control | `VhtMcs0` |
| MCS Data | `VhtMcs4` |

We consider here three examples out of the many we investigated corresponding to the WLAN topologies **T1**, **T2** and **T3** depicted in Figure 3.4. Each of them may correspond to a typical dense WLAN deployment. Topologies **T1** and **T2** are both composed of 6 APs, each being associated with two or three STAs. As for the topology **T3**, it is composed of 10 APs and 25 STAs. **T3** is particularly dense with an average of 5.6 conflicts per AP (when configured with the default setting of TX_PWR and OBSS_PD), and will allow us to test our solution on a larger, denser WLAN deployment. Note that the number of APs

here refers to APs belonging to the same WLAN and set on the same radio channel. Given the number of independent channels for a given frequency band,[30] topologies like **T1**, **T2** and **T3** could actually correspond to WLANs comprising dozens of APs.

[30] *In most countries, there are 3 independent channels in the 2.4 GHz frequency band and around 23 in the 5 GHz frequency band.*

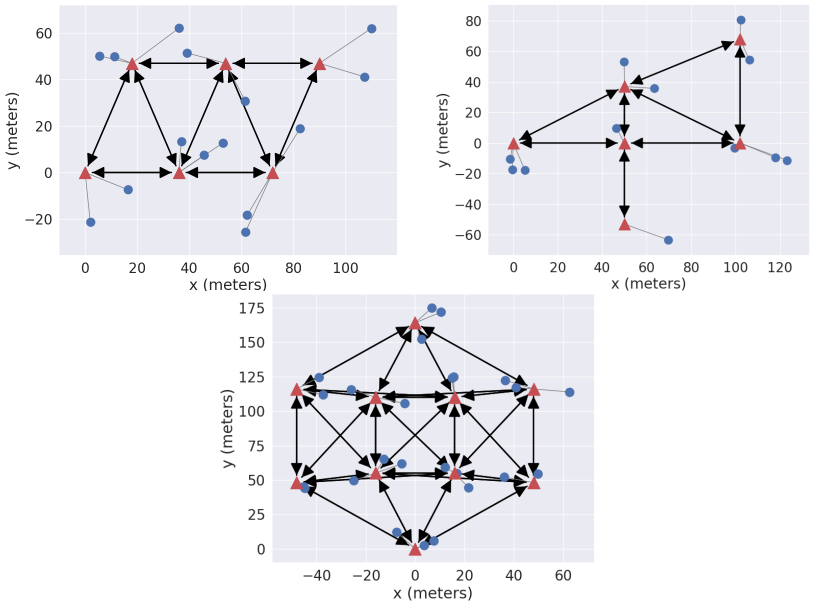



Figure 3.4: Conflict graphs of the 3 WLANs test topologies **T1**, **T2** and **T3**.

### 3.4.2.2 *Simulation Results*

The average regret[31] is the most important metric in our benchmark, as it quantifies the quality of the optimization of the objective function $f$. Table 3.2 provides the average regrets at the end of the experiment for each evaluated solution on the WLAN topologies **T1**, **T2** and **T3**. The results show that our approach (`GM+NGTS`) consistently outperforms the control strategies `DEFAULT` and $\epsilon$-`GREEDY` as well as the state-of-the-art strategy `UNIF+GTS`. In fact, the latter systematically achieves the poorest performance in our complex simulations with an objective function focused on starvations. This can be explained by a naive exploration of the configuration space $\mathcal{C}$, through a uniform sampling of new configurations, and by the unrealistic priors assumed for the distribution of each WLAN configuration. Conversely, `GM+GTS`, which only differs from `UNIF+GTS` because it uses our Gaussian mixture sampling strategy, is consistently getting good results. This highlights the benefits of our sampler. Finally, relaxing the assumption that is made by TS with Gaussian priors used by `UNIF+GTS` (which enforces each configuration to have the same fixed variance), can bring additional benefits. In fact, in two out of three experiments, using TS with our proposed Normal-Gamma priors (`GM+NGTS`) allows for an even lower regret.

[31] *That is, the ratio of the cumulative regret $R_t$ and the time step $t$.*

Table 3.2: Average regret $R_t/t \pm$ its standard error at the end of the simulation of the three considered WLAN topologies, with the five evaluated solutions. For each topology (column), the best metric is **in bold text**.

| Solution | T1 | T2 | T3 |
|---|---|---|---|
| DEFAULT | 0.492 ± 0.006 | 0.376 ± 0.008 | 0.530 ± 0.001 |
| $\epsilon$-GREEDY | 0.593 ± 0.009 | 0.458 ± 0.011 | 0.469 ± 0.006 |
| UNIF+GTS | 0.910 ± 0.002 | 0.773 ± 0.002 | 0.631 ± 0.002 |
| GM+GTS | **0.462 ± 0.006** | 0.354 ± 0.007 | 0.383 ± 0.003 |
| GM+NGTS | 0.476 ± 0.008 | **0.313 ± 0.007** | **0.351 ± 0.004** |

We now focus on network oriented metrics. We start our performance analysis by studying the evolution of the number of starving STAs with each strategy. Recall that starving STAs represent a major issue for WLANs and an efficient WLAN configuration should be able to remove as many starving STAs as possible. Figure 3.5 shows the corresponding results delivered by the simulator ns-3. We notice that with DEFAULT (*i.e.* with the default setting of TX_PWR and OBSS_PD), the number of STAs in starvation is in average at 8 for **T1**, 6 for **T2** and 15 for **T3**. All the other strategies manage to rapidly reduce the number of starving STAs across the three examples except for UNIF+GTS that consistently obtains worse values than DEFAULT. The results also show that GM-GTS significantly outperforms UNIF+GTS suggesting the importance of the sampling process in the overall optimization of $f$. Finally, Figure 3.5 indicates that our solution, denoted by GM+NGTS, leads to the removal of a proportion of starving STAs, which goes up to 40% when compared to DEFAULT on **T3**.

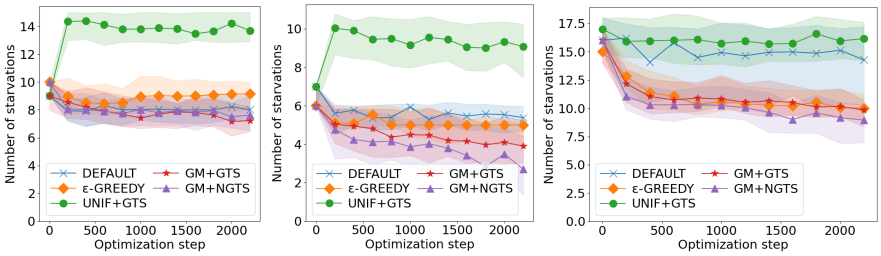



Figure 3.5: Evolution of the number of STAs starving of throughput for the five considered strategies on **T1**, **T2** and **T3**.

To further illustrate the gain that a better setting of the TX_PWR and OBSS_PD parameter values can have on the WLAN, we represent in Figure 3.6 the throughput of each STA for both the default configuration of 802.11ax and the one found by our solution on **T2**. Figure 3.6 shows that all STAs achieve higher throughputs when using the configuration found by our solution. More importantly, our solution enables most STAs to operate above the starvation threshold and only 3 of them (STA 4, STA 8 and STA 9) are occasionally experiencing

starvation of throughput. Conversely, in the case of the default configuration, most STAs are at least periodically experiencing starvation.

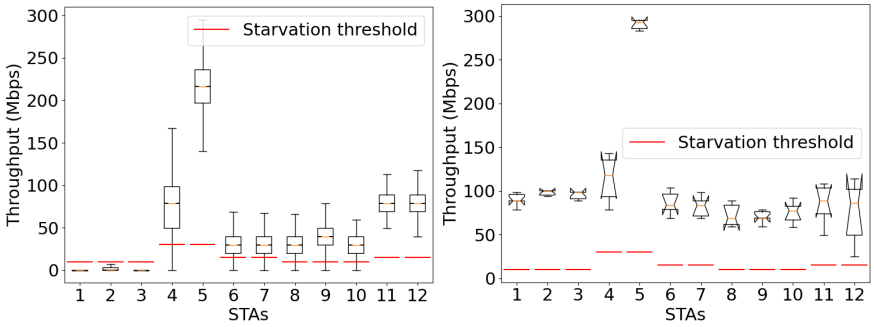



Figure 3.6: Throughputs obtained by STAs under the default 802.11 configuration (left) and the configuration found by our solution (right) on **T2**. Each STA throughput distribution is shown as a boxplot, with a red horizontal bar designating the starvation threshold: if the throughput is below, the STA is considered as starving.

We now explore the influence of our solution on the fairness that reflects how uniformly the throughputs are distributed among the STAs. For that purpose, we use Jain's index [60]. Figure 3.7 represents the corresponding results for each topology. Observe that our solution leads to an increase of the fairness when compared to `DEFAULT` and brings a substantial gain from the fairness associated with $\epsilon$-`GREEDY` or `UNIF-GTS`.

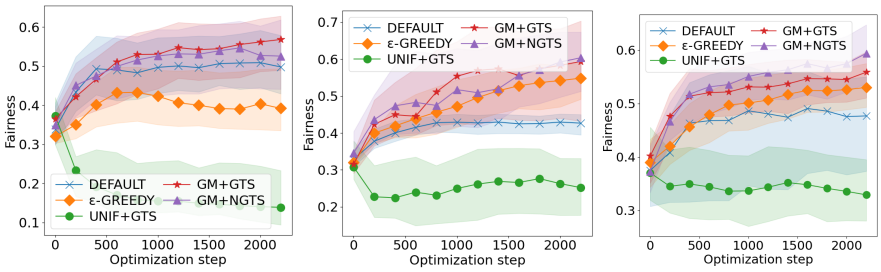



Figure 3.7: Evolution of the fairness between the STAs throughputs for the five considered strategies on **T1**, **T2** and **T3**.

Eventually, for the sake of completeness, we study the influence of all solutions over the aggregate throughput of the WLAN. Figure 3.8 reports the corresponding results. We observe that out of the 5 considered strategies, $\epsilon$-`GREEDY` is the one that leads to the largest improvement in terms of aggregate throughput. $\epsilon$-`GREEDY` performs respectively around 14% and 16% better than our method on the topologies **T2** and **T3** while attaining similar values for **T1**. However, keep in mind that maximizing the aggregate throughput is only a secondary objective in a WLAN and that it is often done at the expense of fairness and the number of starving STAs (see Figures 3.5 and 3.7). Note that for topology **T1**, our proposed solution maintains the aggregate throughput near its value obtained with the default setting of `TX_PWR`

and OBSS_PD. For topologies **T2** and **T3**, our solution is able to significantly increase the aggregate throughput when compared to DEFAULT. Overall, these results indicate that there is no downside for the aggregate throughput to the significant benefits brought by our solution.

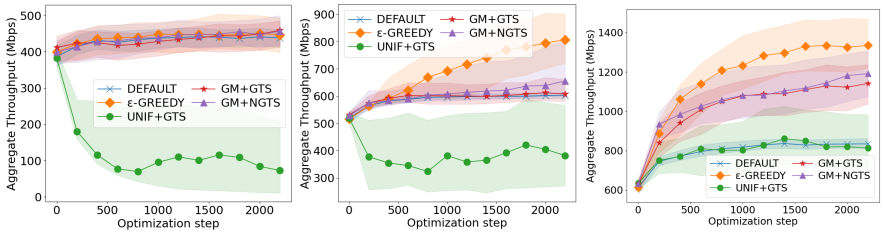



Figure 3.8: Evolution of the aggregate throughput of the WLAN for the five considered strategies on **T1**, **T2** and **T3**.

Overall, within 2,400 iterations, representing 120 seconds of simulated time and a very limited exploration of the high-dimensional space $\mathcal{C}$, our solution was always able to significantly reduce the number of starving STAs and to increase fairness between the throughputs of STAs without decreasing the aggregate throughput of the WLAN. Note that better results may be achieved with longer simulations. Overall, our solution consistently brings a significant improvement on every performance metrics when compared to the legacy default configuration of 802.11. It is able to remove 14%, 63% and 73% of the conflicts occurring with the default configuration in topologies **T1**, **T2** and **T3** respectively. We believe that these results demonstrate the capacity of a tailored MAB solution at improving the spatial reuse of radio channels in WLANs.

### 3.4.3 *Robustness Study*

#### 3.4.3.1 *Variations of the Workload*

In the previous section, we showed that our solution was able to increase the spatial reuse of WLANs by finding an appropriate configuration of TX_PWR and OBSS_PD when WLANs are facing a high but constant workload (see Section 3.1 for an enumeration of the assumptions regarding the WLAN). However, the workload within a WLAN may undergo variations with some STAs demanding more or less traffic to be exchanged over time. Thus, a good configuration of TX_PWR and OBSS_PD must be robust to these potential workload variations, by consistently bringing positive improvements to key performance metrics, when compared to the default configuration of 802.11.

To study the robustness of our solution, we consider **T3**, our densest topology, and we let its STAs run two different applications with different throughputs:

(i) an application requiring as much throughput as the STA can get,

(ii) an application requiring 1% of the attainable throughput of the STA.

In this section, we designate STAs running application of type (i) as "active" STAs. By controlling the proportion $p$ of active STAs, we let the workload of the WLAN vary. In order to evaluate the robustness of a strategy, we proceed as follows. First, we deploy the considered strategy on the WLAN with $p = 2/3$ for a first run of 90 seconds. At the end of this run, we obtain the configuration $\boldsymbol{x}^*$ recommended by the strategy. Recall that $\boldsymbol{x}^*$ refers to a set of values for `TX_PWR` and `OBSS_PD` at each AP. Then, we study the performance of $\boldsymbol{x}^*$ under different levels of workload. We configure the APs according to $\boldsymbol{x}^*$ and we run simulations on four different scenarios: $p = 0$, $1/3$, $2/3$ and $1$. Each of these simulations lasts 30 seconds so that the WLAN configured with $\boldsymbol{x}^*$ can converge to a stable state. At the end of each 30 seconds simulation, the reward, the fairness, and the aggregate throughput of the WLAN are collected.

We study the robustness of three different strategies:

(i) `DEFAULT`, the legacy default configuration of 802.11: (`TX_PWR`, `OBSS_PD`) = (20 dBm, -82 dBm) for each AP,

(ii) `UNIF+GTS`, the solution proposed by [30] and described in Section 2.1,

(iii) `GM+NGTS`, our proposed strategy using the optimizer described in Section 3.3 and the sampler described in Section 3.4.1.

Table 3.3: Confidence intervals for the reward, the fairness and the aggregate throughput (in Mbps) obtained by `DEFAULT`, `UNIF+GTS` and `GM+NGTS` under different proportions of active STAs.

| **Active STAs** | **Metric** | `DEFAULT` | `UNIF+GTS` | `GM+NGTS` |
|---|---|---|---|---|
| $p = 0$ | Reward | **0.98 ± 0.01** | 0.82 ± 0.01 | 0.97 ± 0.01 |
| | Fairness | **0.95 ± 0.01** | 0.79 ± 0.01 | 0.94 ± 0.01 |
| | Through. | **29.04 ± 0.03** | 25.22 ± 0.15 | 28.91 ± 0.17 |
| $p = \frac{1}{3}$ | Reward | 0.81 ± 0.04 | 0.71 ± 0.03 | **0.93 ± 0.02** |
| | Fairness | 0.26 ± 0.03 | 0.23 ± 0.02 | **0.27 ± 0.04** |
| | Through. | 562.0 ± 38 | 678.6 ± 70 | **716.1 ± 80** |
| $p = \frac{2}{3}$ | Reward | 0.61 ± 0.04 | 0.61 ± 0.02 | **0.84 ± 0.03** |
| | Fairness | 0.37 ± 0.03 | 0.38 ± 0.02 | **0.5 ± 0.04** |
| | Through. | 725.1 ± 27 | 1079 ± 65 | **1138 ± 72** |
| $p = 1$ | Reward | 0.5 ± 0.05 | 0.59 ± 0.02 | **0.86 ± 0.02** |
| | Fairness | 0.48 ± 0.04 | 0.54 ± 0.01 | **0.75 ± 0.01** |
| | Through. | 842.5 ± 10 | **1510 ± 43** | 1468 ± 26 |

Table 3.3 reports how a recommended WLAN configuration behaves under different workloads according to some performance metrics,

along with their standard errors, provided by the ns-3 simulator. For the lowest workload considered ($p = 0$), the best configuration is the default configuration of 802.11, with a reward near its maximal value. The configuration recommended by GM+NGTS is barely inferior with relative differences of -1.0%, -1.1% and -0.4% for the reward, the fairness and the aggregate throughput, respectively. The configuration recommended by UNIF+GTS also behaves properly, although the relative differences with the DEFAULT are higher. With $p = \frac{1}{3}$, our solution provides better values for each performance metrics considered, although the workload is lower than the one used to learn the recommended configuration. When compared to DEFAULT, the reward, the fairness and the aggregate throughput are greater by 14.8%, 3.8% and 27.4%, respectively. Although the gap between DEFAULT and UNIF+GTS is reduced, the former still performs better than the latter for this workload, except on the aggregate throughput. For $p = 2/3$, we observe the same dynamics: the relative difference between GM+NGTS and DEFAULT keeps increasing and UNIF+GTS seems equivalent to DEFAULT in regards to the reward and the fairness, except on the aggregate throughput where UNIF+GTS is better. Eventually, even with a higher workload than the one used for the learning of the recommended configuration ($p = 1$), our proposed solution consistently performs better than DEFAULT with relative differences of +72.0%, +56.3% and +74.3% for the reward, the fairness and the aggregate throughput, respectively. The configuration recommended by UNIF+GTS outperforms DEFAULT and provides the best aggregate throughput for this scenario. However, its reward is still significantly lower than the one obtained by GM+NGTS.

Based on these numerical results, we observe that the configuration recommended by our proposed solution ensures efficient performance metrics even when the WLAN workload is moved to higher or lower levels than the one used during the learning phase. Overall, GM+NGTS is found to be more robust than the state-of-the-art solution UNIF+GTS and significantly better than the default configuration DEFAULT. Once again, we can see that, except for the case $p = 1$, having the best reward means having the best values regarding the other performance metrics. This suggests that our reward function is a good quality criterion. As a side note, and not surprisingly, we observe through Table 3.3 that, in general, the higher the workload, the lower the reward. Therefore, we recommend to perform the learning phase under levels of workload as high as possible for the studied WLAN.

#### 3.4.3.2 *Variations of the Starvation Threshold*

We now study the impact of $\alpha$, the starvation threshold parameter, on the considered strategies. Recall that the value of this parameter reflects the level of requirements on STAs' performance and is decided

by the network administrator. In Figure 3.9, we show the confidence intervals for the cumulative regret at the end of a 120 seconds lasting simulation on the three considered topologies **T1**, **T2** and **T3** for multiple credible values of $\alpha$.

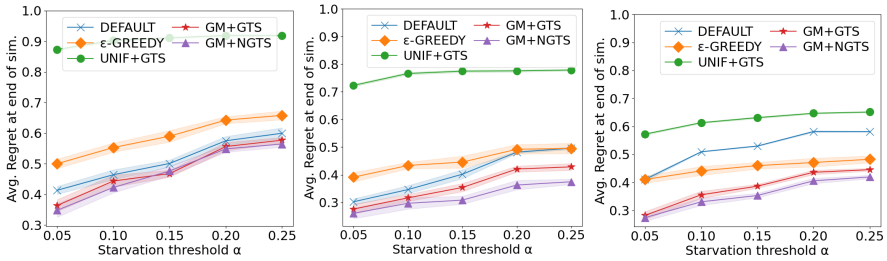



Figure 3.9: Cumulative regret at the end of the simulation w.r.t. $\alpha$ for the five considered strategies on **T1**, **T2** and **T3**.

Since the starvation threshold of any STA is proportional to $\alpha$, the larger $\alpha$, the more likely STAs are considered in a starvation situation, and ultimately, the more difficult it is for the optimizer to collect large rewards. This explains why the cumulative regret tends to increase for larger values of $\alpha$ as shown by Figure 3.9. Overall, we observe that for all the considered topologies and values of $\alpha$, our proposed solution is the most successful at minimizing the cumulative regret. This suggests that the superiority of our proposed solution is robust to changes in the starvation threshold parameter.

### 3.4.4 *Summary*

In this Section 3.4, we presented a strategy based on a Gaussian mixture to subsample the high-dimensional configuration space $\mathcal{C}$. This subsampling strategy allows the optimization of an ad-hoc objective function $f$ through the use of the MAB framework and TS. The resulting solution demonstrated better empirical performance than control and state-of-the-art strategies for the spatial reuse problem in WLANs, on non-trivial scenarios with a well-recognized network simulator. It also demonstrated robustness against the variation of two important WLAN parameters: the workload and the starvation threshold.

This approach also has limitations of different types:

(i) **Design**: the ad-hoc objective function $f$ experiences a threshold effect anytime the starvation status of a STA changes.

(ii) **Theoretical**: the problem breakdown described in Section 3.2.2 lacks some theoretical motivation.

(iii) **Theoretical**: the subsampling strategy also lacks theoretical analysis with respect to its convergence and optimality properties.

(iv) **Design**: the proposed approach, although demonstrating good empirical performance, has many hyperparameters that must

be carefully set, namely the Lipschitz constant $L$, the number of Gaussian distributions in the mixture $K$ and the exploration rate $\epsilon$.

(v) **Evaluation**: a static Modulation Coding Scheme (MCS) has been considered in the simulations, the performance with a dynamic MCS remains unknown.

(vi) **Evaluation**: the considered WLAN topologies are ad-hoc and the traffic is assumed one-directional. Adding some real-world deployments would strengthen the empirical results.

(vii) **Evaluation**: there is a trade-off between the quality of the estimation of the objective function by ns-3 and the cost of making an objective function call. In fact, the larger the test duration, the less noisy and the more accurate the estimation of the objective function. Unfortunately, a large test duration also means that less function calls can be made in a given time interval. This trade-off could be interesting to investigate.

## 3.5 EXPLORATION WITH A MIXTURE OF HYPERSPHERES

We now address some of the limitations listed in Section 3.4.4, namely (v) and (vi), by introducing more complex WLAN topologies closer to real-world deployments. We also propose another subsampling strategy better fitted to the resulting more complex evaluation context.

### 3.5.1 *Subsampling with Hyperspheres*

Unlike the subsampling strategy using a Gaussian mixture, described in Section 3.4, we propose to explore the configuration space $\mathcal{C}$ by subsampling on the surface of multiple hyperspheres. As a matter of fact, the use of multivariate Gaussian distributions is hampered by their relative sensitivity to the number of dimensions. When the number of dimensions is small, most of their density is centered around the mean value (i.e., within the standard deviation-ellipsoid) and, in our case, this translates to exploring configurations that are similar to the centers of the distributions. On the contrary, in high dimensions, a large part of the probability density is shifted to areas that are far away from the centers [68]. More importantly, even in lower-dimensional spaces, Assumption 3.1 could be better exploited. In fact, a significant fraction of the samples made by the mixture lie too close to the centers of the Gaussians. To overcome these inconveniences, we let our sampler use a mixture of hyperspheres. Sampling uniformly $Z_{\boldsymbol{x}_i,\lambda_i}$ on a surface of an hypersphere of radius $\lambda_i$, centered on a configuration $\boldsymbol{x}_i$ can be done with

$$Z_{\boldsymbol{x}_i,\lambda_i} = \boldsymbol{x}_i + \frac{\boldsymbol{v}}{||\boldsymbol{v}||_2}\lambda_i, \tag{3.11}$$

[68] *Wang et al. (2015), "Confidence analysis of standard deviational ellipse and its extension into higher dimensional Euclidean space"*

with $\boldsymbol{v}$ a $d$-dimensional standard normal random vector.

Similarly to our first subsampling approach, in order to concurrently explore multiple promising areas within $\mathcal{C}$, we consider a mixture of $K$ hypersphere distributions, wherein each hypersphere is centered on one of the $K$ best configurations explored so far by the sampler. At first, the mixture is initialized with two hyperspheres, each of them representing a starting point for our algorithm. The first hypersphere is initially centered on the default configuration of 802.11, namely (`TX_PWR`, `OBSS_PD`) = (20, -82) dBm for all APs. For the second starting point, its location is based on the conflict graph between APs: we simply decrease the `TX_PWR` of APs in a round-robin fashion until the conflict graph of APs reaches an average degree that is less than 1. By doing so, we ensure spatial diversity between the two starting points with the aim of speeding up the search when adequate parameter settings are mostly far from the default configuration.

**Algorithm 3.3** Hypersphere Mixture-Based Sampler

**Input**: $L$, number of hyperspheres $K$, observation $(\boldsymbol{x}_i, y_i)$

1: Retrieve history $H$ and time step $t$
2: Add the new observation $(\boldsymbol{x}_i, y_i)$ to $H$
3: $t \leftarrow t + 1$
4: Find the $K$ $(\boldsymbol{x}_j, \mu_j^t)$ pairs in $H$ of largest expected rewards $\mu_j^t$
5: $\mu_*^t \leftarrow \max_j \mu_j^t$
6: $(M, W) \leftarrow (\emptyset, \emptyset)$
7: **for** $j \leftarrow 1$ **to** $K$ **do**
8: Add $\left\{\left(\boldsymbol{x}_i, \frac{\mu_*^t + L - \mu_j^t}{L}\right)\right\}$ to $M$
9: Add $\mu_j^t$ to $W$
10: **end for**
11: Sample a new configuration $\boldsymbol{x}$ from mixture $(M, W)$
12: Transmit $\boldsymbol{x}$ to the optimizer

Algorithm 3.3 summarizes the behavior of the sampler based on a mixture of hyperspheres. As one may guess, it mainly differs from Algorithm 3.2 by line 8, where an hypersphere is initialized instead of a Gaussian distribution.

### 3.5.2 *Numerical Results*

#### 3.5.2.1 *Experimental Settings*

To address some of the experimental limitations raised at the end of Section 3.4, let us describe the new experimental settings. To evaluate the performance of our solutions, we consider three WLAN topologies denoted **T1**, **T2** and **T3** and depicted in Figure 3.10. **T1** is a simple example composed of six APs and a dozen of STAs. **T2** mimics the

[69] *Meraki (2022)*, High Density Wi-Fi Deployments

topology of the highly-dense WLAN deployed by Cisco on the 3rd floor of its office in San Francisco [69]. To account for APs from lower and upper floors, we replicate the Cisco deployment on three floors and we run a simple channel allocation algorithm before selecting the most crowded channel to obtain **T2**. The resulting WLAN has a total of 10 APs with an average of 5 STAs per AP, which are uniformly distributed within its vicinity (*i.e.* the intersection between its radio range at 20 dBm and its Voronoï cell in the WLAN). **T3** is very similar to **T2** as it only differs by the locations of STAs. On average, STAs from **T3** are much further away from their AP than in **T2**. As a matter of fact, STAs in **T3** are about 3 times closer to their AP than to the closest competing AP. In the case of **T2**, this ratio is exceeding 9. This significantly compounds the complexity of **T3**. In a sense, **T3** can be seen as a case in which the association between APs and STAs is far from optimal, or alternately, a case in which the number of available radio channels is too limited. In practice, this may turn to be the case thus making **T3** an interesting example but harder than **T2** to study.

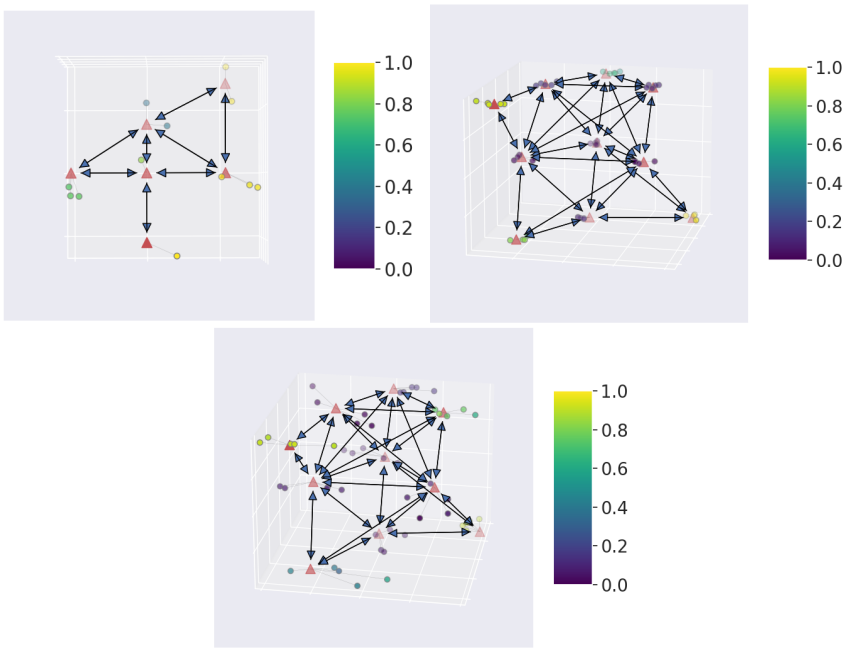



Figure 3.10: Topologies **T1**, **T2** and **T3**. APs are shown as red triangles, conflicts between APs as double-headed arrows, and STAs as colored dots. Their colors show how frequently a given STA has a reasonable throughput (*i.e.* is not in starvation) with the 802.11ax default configuration. A cool color means that the STA is often in starvation, while a warm color denotes a STA that is never in such a situation.

We set the rate of data streams between every AP and each of their STA at 50 Mbps downstream and 3.33 Mbps upstream. This asymmetry reflects that STAs are typically much more downloading than uploading. Additionally, with this level of workload, **T1**, **T2**, and **T3**

are all guaranteed to be in saturation: their APs are unable to properly serve all the needs of their STAs. Hence, we resort to the configuration of APs through the setting of their `TX_PWR` and `OBSS_PD` parameters to increase the Quality of Service (QoS) of the WLANs.

We compare the performance of five strategies:

(i) `DEFAULT`: the default configuration, namely (`TX_PWR`, `OBSS_PD`) = (20, -82) dBm for all APs,

(ii) `WCNC'15`: a dynamic sensitivity threshold solution [25] described in Section 2.1,

(iii) `UNIF+GTS`: a MAB solution [30] also described in Section 2.1,

(iv) `GM+NGTS`: the solution using the Gaussian mixture sampler (described in Section 3.4),

(v) `HM+NGTS`: the solution using the sampler based on a mixture of hyperspheres.

Note that all these strategies were amended to use the objective function defined in Equation 3.3 and to apply their optimization only for the settings of the APs.

We implemented all the aforementioned strategies in the open-source network simulator ns-3 [31]. Table 3.4 reports the settings we used. Note that, compared to Section 3.4, we have increased the realism of our simulations by incorporating the mechanism of rate adaptation that let APs and STAs dynamically vary the speed of their wireless links (through the use of different MCS) in response to the quality of the received signal.

Table 3.4: ns-3 parameters for the evaluation of the HM-based sampler.

| **Parameter** | **Value** |
|---|---|
| ns-3 version | 3.31 |
| Number of repetitions | 22 |
| Simulation duration | 120 s |
| Duration of an iteration | 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 | `LogDistancePropagationLossModel` |
| Wi-Fi Manager | `IdealWifiManager` |

All our simulations last 120 seconds. To account for the potential high variance of the studied performance parameters, we replicate

each simulation 22 times and we represent in the corresponding figures the first, second, and third quartiles. If a metric is subject to large variations, we extract and represent its trend using an exponential moving average in place of the raw data. Finally, to enable the reproducibility of all our results, we made available (in open-source) all the code we used for this section, including the three WLANs topologies and the implementations of the different strategies.[32]

[32] https://github.com/abardou/IMAB-RING-SR-AP-802.11AX

#### 3.5.2.2 *Simulation Results*

As in Section 3.4.2.2, we start by focusing on the most important metric, namely the average regret, along with its standard errors on Table 3.5. Although our first introduced solution GM+NGTS (see Section 3.4) still provides competitive performance, the use of more complex scenarios degraded its relative performance to state-of-the-art strategies. Our new proposed solution HM+NGTS, which uses the sampler based on a mixture of hyperspheres, consistently provides significant lower average regrets than the other strategies.

Table 3.5: Average regret $R_t/t \pm$ its standard error at the end of the simulation of the three considered WLAN topologies, with the five evaluated solutions. For each topology (column), the best metric is **in bold text**.

| Solution | T1 | T2 | T3 |
|---|---|---|---|
| DEFAULT | $0.248 \pm 0.001$ | $0.632 \pm 0.001$ | $0.659 \pm 0.003$ |
| WCNC'15 | $0.200 \pm 0.001$ | $0.438 \pm 0.001$ | $0.580 \pm 0.001$ |
| UNIF+GTS | $0.324 \pm 0.003$ | $0.399 \pm 0.001$ | $0.751 \pm 0.001$ |
| GM+NGTS | $0.243 \pm 0.004$ | $0.472 \pm 0.021$ | $0.642 \pm 0.005$ |
| HM+NGTS | $\mathbf{0.109 \pm 0.003}$ | $\mathbf{0.237 \pm 0.011}$ | $\mathbf{0.428 \pm 0.004}$ |

This better performance of HM-NGTS can also be observed on network-oriented metrics. Figure 3.11 reports the number of STAs in starvation for the three considered topologies. HM-NGTS manages to significantly reduce the number of STAs in starvation, to even address the problem completely in **T1**. As for **T2**, the number of STAs is reduced by 80% when compared to DEFAULT. Even on **T3** that constitutes a much more complex problem than **T2**, this reduction is still of 43%.

Finally, we focus on the measured throughput of the WLAN under the considered solutions. Figure 3.12 reports the results. On this performance metric also, HM-NGTS shows a significant improvement over DEFAULT, the state-of-the-art strategies and our first introduced solution GM-NGTS.

Overall, in all the examples we explored, our solution never failed to quickly discover a configuration of TX_PWR and OBSS_PD that significantly reduces the number of STAs in starvation, improves the fairness among STAs and increases the cumulated throughput. Interestingly,

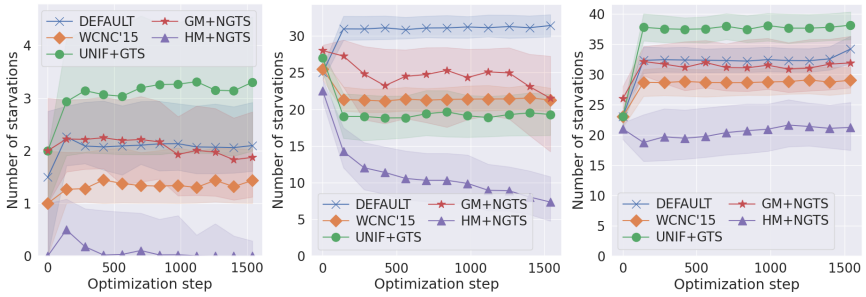



Figure 3.11: Evolution of the number of STAs in starvation for the five considered strategies on **T1**, **T2** and **T3**.

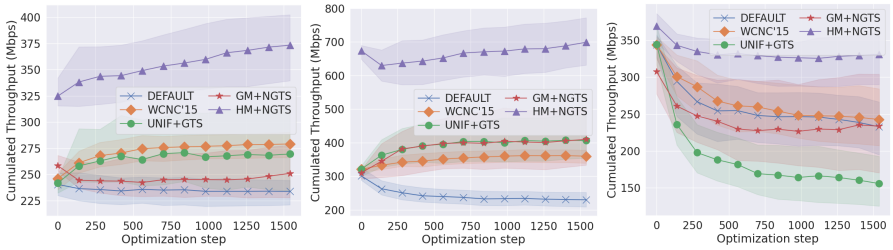



Figure 3.12: Evolution of the throughput of the WLAN for the five considered strategies on **T1**, **T2** and **T3**.

the configuration found by our solution consists, in general, in decreasing `TX_PWR` and increasing `OBSS_PD` at each AP. Intuitively this means that, at the end of the optimization, most APs tend to emit at a lower level but allow themselves to emit in more noised conditions. This dual change the favors spatial reuse of the radio channel with more simultaneous transmissions from nearby APs.

## 3.6 SUMMARY

In this chapter, we proposed two ad-hoc solutions to the spatial reuse problem in WLANs. They are both based on the MAB framework, and they address the very large cardinality of the configuration space $\mathcal{C}$ by subsampling it to build a reservoir of promising configurations. Concurrently, they use Thompson sampling to identify the best configuration in the reservoir. The first solution subsamples $\mathcal{C}$ with a Gaussian mixture, while the second uses a mixture of hyperspheres. Both of them were evaluated through simulation, using ns-3 on realistic scenarios involving complex mechanisms of the 802.11 ecosystem. They demonstrated competitive performance against state-of-the-art and control solutions on every performance metric considered. They were particularly efficient at improving the QoS of starving STAs. The first solution was presented at an international conference specialized in wireless communications [70] and extended in a special issue of a peer-reviewed journal [71], while the second solution was published in another peer-reviewed journal specialized in computer net-

[70] Bardou et al. (2021), *"Improving the Spatial Reuse in IEEE 802.11ax WLANs: A Multi-Armed Bandit Approach"*

[71] Bardou et al. (2022), *"Analysis of a Multi-Armed Bandit solution to improve the spatial reuse of next-generation WLANs"*

works [72]. Note that the hypersphere sampler was also presented at a national conference on operations research [73].

*[72] Bardou et al. (2023), "Mitigating starvation in dense WLANs: A multi-armed Bandit Solution"*

*[73] Bardou et al. (2022), "Multi-Armed Bandit Algorithm for Spatial Reuse in WLANs: Minimizing Stations in Starvation"*

Although we believe these solutions address the spatial reuse problem in WLANs in an innovative manner, some limitations raised in Section 3.4.4 remain, mainly regarding the design of the solution and the theoretical motivations underpinning it:

(i) **Design**: the ad-hoc objective function $f$ experiences a threshold effect anytime the starvation status of a STA changes,

(ii) **Theoretical**: the problem breakdown described in Section 3.2.2 lacks some theoretical motivation,

(iii) **Theoretical**: the subsampling strategies also lack theoretical analysis with respect to their convergence and optimality properties,

(iv) **Design**: the proposed approaches have many hyperparameters that must be carefully set

(v) **Evaluation**: the trade-off between the quality of the estimation of the objective function and the number of function calls in a given time interval remains to be discussed.

We address most of these limitations in the next chapter, by proposing a more theoretically grounded method to address the spatial reuse problem in WLANs.

# 4

# DECENTRALIZED BAYESIAN OPTIMIZATION FOR 802.11

**Outline.** In this chapter, we address the spatial reuse problem in a Wireless Local Area Network (WLAN) using Bayesian Optimization (BO) based on a Gaussian Process (GP) (see Section 2.2.4). We tackle the large dimensionality of the configuration space $\mathcal{C}$ by exploiting the natural additive decomposition of the objective function $f$. We end up with a decentralized Bayesian optimization algorithm, called INSPIRE, that offers minimax guarantee on the maximization of $f$ under a regularity assumption. We evaluate INSPIRE on the network simulator ns-3 with realistic WLAN scenarios. The obtained results show significant improvement over state-of-the-art methods for the spatial reuse problem, including the solutions developed in Chapter 3. This indicates that BO is a very promising approach for the decentralized performance optimization in computer networks.

## 4.1 MOTIVATIONS

Although the solutions proposed in Chapter 3 demonstrated good empirical performance, their many hyperparameters and our lack of understanding regarding their theoretical properties (convergence, optimality) can be considered as major drawbacks. Note that this is also the case for all the data-driven state-of-the-art strategies considered in Section 2.1.2. Thus, we consider that proposing a solution to the spatial reuse problem in WLANs[33] that has theoretical guarantee is of critical importance for the community.

Another motivation is to improve the domain of applicability of our proposed solution. In fact, the centralized approaches developed in Chapter 3 are systematically discarded in all the scenarios that cannot afford a centralized entity operating as a controller for the WLAN. A decentralized solution, on the other hand, can be deployed in every scenario. In addition, it is also more in line with to the vast majority of network algorithms that constitute the WLAN ecosystem, which are also decentralized.

[33] *Other than the analytical approaches described in Section 2.1.1 that do not scale and / or are too coarse-grained.*

## 4.2 PROBLEM FORMULATION

### 4.2.1 *System Modeling*

We consider a system very similar to the one described in Chapter 3. Let us quickly recall vocabulary and introduce some notations about

the studied WLAN. A wireless device, such as a phone or a computer, is called a Station (STA) and is necessarily associated with an Access Point (AP) that serves as a gateway to the Internet. In this chapter, we study a WLAN composed of $n$ APs. Each AP $i$ has a set of associated STAs in the set $\mathcal{S}_i$, and we denote $n_S$ the total number of STAs in the WLAN, that is $n_S = \sum_{i=1}^{n} |\mathcal{S}_i|$. Eventually, we denote $\mathcal{N}_i$ the APs that are within the communication range of AP $i$. Note that AP $i$ itself belongs to $\mathcal{N}_i$. We refer to $\mathcal{N}_i$ as the surroundings, or the neighborhood, of AP $i$.

We make no assumptions on the WLAN, including on the specific arrangement of its APs and STAs, other than the three detailed below.

**Assumption 4.1.** *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 assume 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.

**Assumption 4.2.** *The $n$ APs have their `TX_PWR` and `OBSS_PD` parameters configurable (as defined by the 802.11ax amendment).*

**Assumption 4.3.** *Each AP can periodically run performance tests under a configuration $\boldsymbol{x} \in \mathcal{C}$ and obtain, in return, the average throughput attained by each of its STAs over a short time interval $\Delta t$.*

We denote $T(\boldsymbol{x}) = (T_1(\boldsymbol{x}), \cdots, T_{n_S}(\boldsymbol{x}))$ the mean throughputs attained by the STAs and reported by the performance tests.

Our objective, as in the previous chapter, is to discover the optimal configuration $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$, according to the objective function $f$ that remains to be defined. Note that this configuration $\boldsymbol{x}^*$ must be discovered in a decentralized fashion.

### 4.2.2 *Objective Function and its Decomposition*

To quantify the quality of a given configuration $f$ we exploit $T(\boldsymbol{x})$, the performance tests that can periodically run on each AP as stated by Assumption 4.3. As in Chapter 3, we consider the Proportional Fairness (PF) $\mathrm{PF}(\boldsymbol{x}) = \prod_{i=1}^{n_S} T_i(\boldsymbol{x})$ since it provides a natural scalarization of the multi-objective problem of fairly optimizing the throughput within a WLAN. However, to avoid introducing threshold effects that may make the optimization problem harder, we do not seek to enforce the constraint (3.2) introduced in Chapter 3. To turn the PF into a much more pleasant closed-form to optimize, we consider its logarithm and we set

$$f(\boldsymbol{x}) = \log \mathrm{PF}(\boldsymbol{x}) = \sum_{i=1}^{n_S} \log T_i(\boldsymbol{x}). \tag{4.1}$$

Note that in the previous chapter, we defined an objective function over a discrete configuration space $\mathcal{C}$ (see Section 3.2) that provides a strong incentive to reduce starvations in the WLAN. In this chapter, we assume a compact configuration space $\mathcal{C} = \{[-82, -62] \times [1, 21]\}^n$ to avoid making restrictive assumptions about the discretization of the configuration space. Note that, even with a compact configuration space, the objective function defined in Section 3.2.1 induces discontinuities through thresholds effects (e.g. whenever the starvation status of a STA changes). This new definition (4.1) does not provide such a strong incentive to reduce starvations, but it is continuous and more likely to support regularity assumptions.

Observe that an AP cannot compute $f$ as defined in (4.1) as it should have access to every performance test realized in the WLAN in a centralized fashion. This goes against the development of a decentralized protocol. Therefore, we consider another function, denoted $f^{(i)}$ for each AP $i$, that must be computable with local information only.[34] In other words, we seek an additive decomposition of $f$ in $n$ factors, that is $f = \sum_{i=1}^{n} f^{(i)}$, so that each AP $i$ can focus on the optimization of a single factor $f^{(i)}$.

[34] *That is, information accessible at a single AP.*

A natural decomposition of (4.1) is to set $f^{(i)}(\boldsymbol{x}) = \sum_{j \in \mathcal{S}_i} \log T_j(\boldsymbol{x})$, that is, the local objective function of AP $i$ is built by aggregating the performance of its associated STAs in $\mathcal{S}_i$. However, this definition gives each AP a strong incentive to act selfishly. In fact, under such definition, maximizing $f^{(i)}$ is very likely to lead a set of APs to strongly increase their `TX_PWR` to favor their STAs at the expense of the Quality of Service (QoS) of neighboring devices. Although this may not be considered as a major drawback since some decentralized optimization techniques are designed to handle this case (e.g. see [74]), it is very likely to slow down the optimization process.

*[74] Mota et al. (2014), "Distributed optimization with local domains: Applications in MPC and network flows"*

To address this problem, recall that Assumption 4.1 allows an AP $i$ to communicate with APs in its neighborhood, that is APs in $\mathcal{N}_i$. Therefore, we define the factor function for AP $i$ as

$$f^{(i)}(\boldsymbol{x}) = \sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_j|} \sum_{k \in \mathcal{S}_j} \log T_k(\boldsymbol{x}). \tag{4.2}$$

Intuitively, (4.2) builds the local objective function of AP $i$ as a weighted sum of the performance of all the STAs associated with APs in $\mathcal{N}_i$. Since the performance of the STAs associated with an AP $j$ appear in exactly $\mathcal{N}_j$ factors, we introduce the weights $1/|\mathcal{N}_j|$ to ensure that $(f_1(\boldsymbol{x}), \cdots, f_n(\boldsymbol{x}))$ is indeed an additive decomposition of the global objective function $f$.

### 4.2.3 *Gaussian Processes as Surrogates*

We propose that each AP $i$ uses a GP as a surrogate model for its local objective function $f^{(i)}$, as described in Section 2.2.4. The surrogate

model of AP $i$ is based only on the settings of AP $i$ and its neighboring APs, denoted by $\boldsymbol{x}_{\mathcal{N}_i}$.

**Assumption 4.4.** *$\forall i \in [1, n]$, the factor function $f^{(i)}$ is a $\mathcal{GP}\left(0, k^{(i)}(\boldsymbol{x}_{\mathcal{N}_i}, \boldsymbol{x}'_{\mathcal{N}_i})\right)$.*

Choosing the covariance function is a critical step when designing a GP, as it determines some key features such as its isotropy and smoothness. Since little information is known about each factor function $f^{(i)}$, we choose $k^{(i)}$ to be a Matérn covariance function [75]

*[75] Genton (2001), "Classes of kernels for machine learning: a statistics perspective"*

$$M_\nu(\boldsymbol{x}, \boldsymbol{x}') = s^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{||\boldsymbol{x} - \boldsymbol{x}'||_2}{\rho}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{||\boldsymbol{x} - \boldsymbol{x}'||_2}{\rho}\right) \quad (4.3)$$

where $\nu$, $s$ and $\rho$ are hyperparameters, while $\Gamma$ and $K_\nu$ are the Gamma function and the modified Bessel function of the second type, respectively (see Chapter 30 of [76] for more details about the two functions).

*[76] Bronson et al. (2014), Schaum's outline of differential equations*

Note that (4.3) is a function of $||\boldsymbol{x} - \boldsymbol{x}'||_2$. This guarantees the isotropy of the resulting GP. Additionally, note that the $\nu$ hyperparameter offers a direct control onto the differentiability (in the mean-square sense) of the GP. As a matter of fact, with the Matérn covariance function $M_\nu$, the resulting GP is $\lceil \nu \rceil - 1$-times differentiable in the mean-square sense [77]. Since we do not have information about the smoothness of $f$, we set $\nu = 3/2$ so the resulting GP is only once differentiable. The two other hyperparameters $s$ and $\rho$ are set by Maximum Likelihood Estimation (MLE) throughout the optimization. We refer the interested reader to [78] for more details about the engineering of covariance functions. Eventually, note that with $\nu = 3/2$, (4.3) becomes

*[77] Santner et al. (2003), The design and analysis of computer experiments*

*[78] Palar et al. (2023), "On kernel functions for bi-fidelity Gaussian process regressions"*

$$k^{(i)}(\boldsymbol{x}, \boldsymbol{x}') = s_i^2 \left(1 + \frac{\sqrt{3}||\boldsymbol{x} - \boldsymbol{x}'||_2}{\rho_i}\right) e^{-\frac{\sqrt{3}||\boldsymbol{x} - \boldsymbol{x}'||_2}{\rho_i}}. \quad (4.4)$$

As discussed in Section 2.2.4, Bayesian optimization techniques deal with the exploration-exploitation dilemma through the use of an acquisition function, that exploits the posterior GP (given the previous queries $\boldsymbol{X}^{(i)}$ and the observed objective values $\boldsymbol{y}^{(i)}$) to find a promising query. Out of the many candidates, we select Expected Improvement (EI) because of its low computational overhead. Additionally, it returned the best results in our experiments (see Section 4.4). The EI acquisition function is

$$\varphi_t^{(i)}(\boldsymbol{x}) = \mathbb{E}\left[\left(f^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}\right)^+\right] \quad (4.5)$$

with $y_{*,t}^{(i)} = \max_{j \in [1,t]} y_j^{(i)}$ and $(a)^+ = \max(0, a)$.

The formulation (4.5) does not always have a closed form. However since $f^{(i)}(\boldsymbol{x})$ is a Gaussian variable, standard calculations (see [55]) show that (4.5) becomes

$$\varphi_t^{(i)}(\boldsymbol{x}) = \left(\mu_t^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}\right) P(z) + \sigma_t^{(i)}(\boldsymbol{x}) p(z) \quad (4.6)$$

with $z = \left(\mu_t^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}\right) / \sigma_t^{(i)}(\boldsymbol{x})$, $\mu_t^{(i)}$ computed with (2.10), $\sigma_t^{(i)}$ computed with (2.11), $p(z)$ and $P(z)$ the p.d.f and the c.d.f of $\mathcal{N}(0,1)$ respectively.

## 4.3 INSPIRE

### 4.3.1 *Optimization of the Acquisition Function*

As discussed in Section 2.2.4, a Bayesian optimization algorithm typically uses a global optimization algorithm such as DIRECT [34] to optimize its acquisition function at each iteration $t$. However, these global optimization algorithms struggle at optimizing functions of high dimension. To prevent this, as discussed in greater length in Chapter 6, we choose to exploit gradient descent (generally attributed to Cauchy [79]), a local optimization algorithm that only offers local optimum guarantees [80] but is able to deal with high-dimensional functions. Note that differentiating the closed form (4.6) is trivial.

*[79] Lemaréchal (2012), "Cauchy and the gradient method"*
*[80] Curry (1944), "The method of steepest descent for non-linear minimization problems"*

**Proposition 4.1.**

$$\nabla\varphi_t^{(i)}(\boldsymbol{x}) = \nabla\mu_t^{(i)}(\boldsymbol{x})P(z) + \nabla\sigma_t^{(i)}(\boldsymbol{x})p(z). \tag{4.7}$$

*Proof.*

$$\begin{aligned}
\nabla\varphi_t^{(i)}(\boldsymbol{x}) &= \nabla\left(\left(\mu_t^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}\right)P(z)\right) + \nabla\left(\sigma_t^{(i)}(\boldsymbol{x})p(z)\right) \\
&= \nabla\mu_t^{(i)}(\boldsymbol{x})P(z) + \left(\mu_t^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}\right)p(z)\nabla z \\
&\quad + \nabla\sigma_t^{(i)}(\boldsymbol{x})p(z) - \sigma_t^{(i)}(\boldsymbol{x})zp(z)\nabla z \qquad (4.8) \\
&= \nabla\mu_t^{(i)}(\boldsymbol{x})P(z) + \nabla\sigma_t^{(i)}(\boldsymbol{x})p(z). \qquad (4.9)
\end{aligned}$$

The line (4.8) follows from observing that $\nabla P(z) = p(z)\nabla z$ and that $\nabla p(z) = -zp(z)\nabla z$, while (4.9) follows from remarking that $\sigma_t^{(i)}(\boldsymbol{x})z = \mu_t^{(i)}(\boldsymbol{x}) - y_{*,t}^{(i)}$. □

### 4.3.2 *Consensus Function*

In the previous sections, we have described how each AP $i$ computes its local objective function $f^{(i)}$ and relies on a GP to explore promising configurations for the APs in $\mathcal{N}_i$. However, more coordination between APs is required. In fact, by construction, the collection $\mathcal{F} = (\mathcal{N}_j)_{1\leq j\leq n}$ is a cover of the set of APs but not a partition. In fact, if $\mathcal{F}$ had only null intersections,[35] then the spatial reuse of the radio channel would already be at its apex and there would be no need for improvement.

[35] *That is,* $\forall j,k \in [1,n], j \neq k, \mathcal{N}_j \cap \mathcal{N}_k = \emptyset$.

Figure 4.1 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

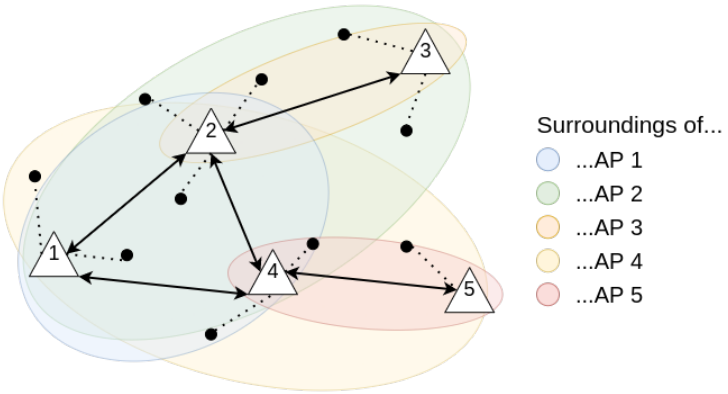



Figure 4.1: A WLAN represented by a graph with APs depicted as labelled triangles and STAs as black dots. An edge exists between two APs when they are in the communication range of each other. We use different colors to illustrate the surroundings of each AP in $\mathcal{F}$.

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, the setting $\boldsymbol{x}_i$ of an AP $i$ is involved in the local objective functions of its neighbors in $\mathcal{N}_i$. Therefore, independently maximizing each local objective 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 [81]). Without more information on the relationship between the settings of the APs and the measured throughputs of STAs, it seems difficult to provide an expression for the maximal argument of the global objective function $f$ given the maximal argument of the local objective functions $f^{(i)}$. However, recall that in our case we have $\sum_{i=1}^{n} f^{(i)}(\boldsymbol{x}_{\mathcal{N}_i}) = f(\boldsymbol{x})$. We propose to leverage this property to provide guarantees, under a regularity assumption.

*[81] Hardin (2009), "The Tragedy of the Commons"*

**Assumption 4.5.** *Each local objective function $f^{(i)}$ is Lipschitz continuous, with Lipschitz constant $L_i$.*

**Theorem 4.1.** *Let $\left\{f^{(i)}\right\}_{1\leq i\leq n}$ be a set of Lipschitz-continuous functions with Lipschitz constants $L_i$ and $\{x_j^{(i)}\}_{i\in\mathcal{N}_j}$ be the prescriptions received by the AP $j$. Let $\tilde{\boldsymbol{x}}$ be the marginal median of the received set of prescriptions weighted by the Lipschitz constants[36], so that an element $\tilde{x}_j$ is*

[36] *The weighted median of a sample corresponds to the 50% weighted percentile.*

$$\tilde{x}_j = \textit{median}\left(\underbrace{\left\{x_j^{(i)}\right\}_{i\in\mathcal{N}_j}}_{\textit{sample}}, \underbrace{\{L_i\}_{i\in\mathcal{N}_j}}_{\textit{weights}}\right). \tag{4.10}$$

*If* $\forall i \in [1, n]$, *the recommendation of AP* $i$ *is* $\boldsymbol{x}^{(i)} = \arg\max_{\boldsymbol{x}} f^{(i)}(\boldsymbol{x})$, *then* $\tilde{\boldsymbol{x}}$ *is a minimax optimum of* $\sum_{i=1}^{n} f^{(i)}(\boldsymbol{x}_{\mathcal{N}_i})$.

The proof of Theorem 4.1 is provided in Appendix A. Intuitively, Theorem 4.1 states that the distance $|f(\boldsymbol{x}^*) - f(\boldsymbol{x})|$ can be bounded from above, and that the upper bound is tight.[37] Furthermore, Theorem 4.1 states that $\tilde{\boldsymbol{x}}$ as defined in (4.10) minimizes this tight upper bound, making $\tilde{\boldsymbol{x}}$ a minimax optimal. Note that a component $\tilde{x}_j$ can be computed by AP $j$, as it is the median of the recommendations sent by the neighborhood $\mathcal{N}_j$ of AP $j$. This boils down to compute the marginal median (also called the component-wise median) of the received recommendations.

[37] *As far as we know, there is no lower upper bound for this quantity using only the assumed information about* $f$.

### 4.3.3 *Algorithm and Complexity*

Since $\sum_{i=1}^{n} f_i(x) = f(x)$, we leverage Theorem 4.1 to find a minimax optimum of the global objective function $f$.

**Algorithm 4.1** 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 $\boldsymbol{x}^{(i)} = \arg\max_{\boldsymbol{x}} \varphi_t^{(i)}(\boldsymbol{x})$ by gradient ascent
4: Broadcast $\boldsymbol{x}^{(i)}$ 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 $\tilde{\boldsymbol{x}}^{(i)}$ with (4.10)
7: Test $\tilde{\boldsymbol{x}}^{(i)}$ for $\Delta t$ seconds and compute $g^{(i)}$ by using (4.1) only on the STAs of AP $i$
8: Broadcast $g^{(i)}$, $|\mathcal{N}_i|$ and $\tilde{\boldsymbol{x}}^{(i)}$ to APs in $\mathcal{N}_i$
9: Receive $g^{(j)}$, $|\mathcal{N}_j|$ and $\tilde{\boldsymbol{x}}^{(j)}$ from AP $j, j \neq i, j \in \mathcal{N}_i$
10: Compute the local objective $y^{(i)}$ with (4.2) and the local configuration $\tilde{\boldsymbol{x}}_{\mathcal{N}_i}$
11: Add the pattern $\left(\tilde{\boldsymbol{x}}_{\mathcal{N}_i}, y^{(i)}\right)$ to $\mathcal{GP}_i$
12: **end while**

Algorithm 4.1 summarizes INSPIRE, our proposed solution. An AP finds a prescription using its own surrogate model (line 3) and, by interacting only with its neighbors[38] reaches a consensus on a promising configuration of its parameters thanks to Theorem 4.1 (lines 4-6). Then, it performs a test of these parameters on the WLAN (line 7), and broadcasts its observations to its neighbors (lines 8-9). Eventually, it receives enough information to compute its objective function (4.2) on

[38] *That is, APs in its communication range.*

line 10 before conditioning its surrogate model with the new observation on line 11. Overall, Algorithm 4.1 allows to iteratively learn and optimize a local objective function quantifying the quality of a configuration on the spatial reuse of its neighborhood. Note that, because we are addressing the spatial reuse problem, we seek to optimize the PF of the throughputs $(T_1(\boldsymbol{x}), \cdots, T_{n_S}(\boldsymbol{x}))$. However, it is worth noting that INSPIRE could be applied to any objective function satisfying Assumptions 4.4 and 4.5. This covers many applications, in wireless networks and beyond. As an example, it seems to be a promising algorithm to tackle the minimization of the energy consumption of a network, which is a current challenge in the networking community [82–84].

[82] Landaluce et al. (2020), "A review of IoT sensing applications and challenges using RFID and wireless sensor networks"

[83] Rajendra Prasad P (2019), "Efficient performance analysis of energy aware on demand routing protocol in mobile ad-hoc network"

[84] Giordani et al. (2020), "Non-terrestrial networks in the 6G era: Challenges and opportunities"

Contrary to what one might think, the most resource-intensive operation in Algorithm 4.1 is not the inversion, at time step $t$, of the matrix $\boldsymbol{K}_t$ in (2.10) and (2.11). In fact, if the Cholesky decomposition of the matrix $\boldsymbol{K}_{t-1} = LL^T$ is known, the Cholesky decomposition of $\boldsymbol{K}_t$ is easily obtained and so is $\boldsymbol{K}_t^{-1}$. The most resource-intensive operation is the maximization of the acquisition function through gradient ascent. Since the gradient ascent has a complexity in $\mathcal{O}\left(\xi^{-1}\right)$ for a desired accuracy $\xi$ [85], this requires computing $\mathcal{O}\left(\xi^{-1}\right)$ matrix-vector multiplications. The computational complexity of Algorithm 4.1 at time $t$ is therefore $O\left(\xi^{-1}t^2\right)$. This implies that the real burden to the execution time of INSPIRE is the time step $t$. This compels us to bound the size of the set of queries $\boldsymbol{X}$ and the set of observed values $\boldsymbol{y}$ in order 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, a windowing method (*e.g.* sliding window) can be applied to bound the size of the dataset and so the computational complexity of INSPIRE. We discuss such an alternative and we evaluate its impact on the empirical performance of the proposed solution in the following section.

[85] Xie et al. (2020), "Linear convergence of adaptive stochastic gradient descent"

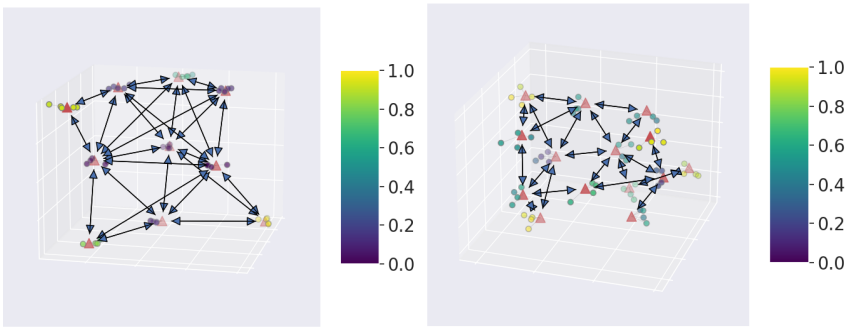
## 4.4 NUMERICAL RESULTS

### 4.4.1 *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, previously discussed in Section 3.5.2. We use **T1** to refer to this WLAN topology, which is illustrated in Figure 4.2a. **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 m² 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 4.2b depicts the topology **T2**.




(a) (b)

Figure 4.2: Topologies **T1** and **T2**. APs are shown as red triangles, conflicts between APs as double-headed arrows, and STAs as colored dots. Their colors show how frequently a given STA has a reasonable throughput (*i.e.* is not in starvation) with the 802.11ax default configuration. A cool color means that the STA is often in starvation, while a warm color denotes a STA that is never in such a situation.

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 are likely to be in saturation. 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 conditions.

To better appraise the quality of INSPIRE, we also consider a control strategy, several state-of-the-art solutions which are discussed in Section 2.1.2 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 [25];
- `JNCA'19`: each AP solves a Multi-Armed Bandit (MAB) problem using Thompson sampling to dynamically update their `TX_PWR` and `OBSS_PD` parameters [30];
- `GM+NGTS`: the solution described in Section 3.4;
- `HM+NGTS`: the solution described in Section 3.5;
- `INSPIRE_LIM`: similar to INSPIRE, except that only the last 50 observations are considered to make predictions.

*[86] Blum (2016),* LibGP

We implemented INSPIRE (based on the open-source GP C++ library LibGP [86]) as well as the seven strategies described above in the open-source network simulator ns-3 [31]. We report in Table 4.1 the simulation parameters used in the rest of this section. Unlike previous works (e.g., [8, 25, 28, 30, 70])) with the exception of [26], 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.

Table 4.1: ns-3 parameters for the evaluation of INSPIRE.

| **Parameter** | **Value** |
|---|---|
| ns-3 version | 3.31 |
| Number of repetitions | 22 |
| Simulation duration | 30 s |
| Test duration ($\Delta t$) | 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` |

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 (4.1), whereas the evaluated strategies internally use the local objective function defined in (4.2). Then, we compute the classical performance metrics already discussed in Chapter 3: the average regret (with a normalized version of the global objective function( 4.1), the number of starving STAs, which we define as STAs experiencing a very low throughput (namely, less than 10% of their attainable throughput) and the cumulated throughput, which simply sums all STAs' throughput. Additionally, we measure the execution time of the solutions.

Each simulation lasts 30 seconds and we replicate 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 simulation, we apply an exponential moving average 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.

### 4.4.2 *Performance Metrics*

Table 4.2: Average regret $R_t/t \pm$ its standard error at the end of the simulation of the two considered WLAN topologies, with the seven evaluated solutions. For each topology (column), the best metric is **in bold text**.

| Solution | T1 | T2 |
|---|---|---|
| DEFAULT | 0.652 ± 0.001 | 0.429 ± 0.004 |
| WCNC'15 | 0.470 ± 0.001 | 0.327 ± 0.005 |
| JNCA'19 | 0.437 ± 0.001 | 0.398 ± 0.006 |
| GM+NGTS | 0.527 ± 0.016 | 0.375 ± 0.006 |
| HM+NGTS | 0.305 ± 0.006 | 0.379 ± 0.005 |
| INSPIRE | **0.193 ± 0.005** | **0.294 ± 0.006** |
| INSPIRE_LIM | 0.233 ± 0.005 | 0.329 ± 0.005 |

As in Chapter 3, we start by providing in Table 4.2 the average regrets of all the considered strategies on the topologies **T1** and **T2**. The results show that INSPIRE and, at a lesser extent, its windowing version INSPIRE_LIM, offer a significant improvement over the control and state-of-the-art solutions. On **T1**, INSPIRE achieves an average regret that is 70% lower than the control DEFAULT configuration and 37% lower than the best state-of-the-art strategy. On **T2**, these quantities

are 31% and 10% respectively. Discussing in greater depth the complexity of the two scenarios in Section 4.5.1 will shed more light on these performance differences.

We now consider the performance metrics (number of starving STAs, cumulated throughput, computational overhead) on **T1**, depicted by Figure 4.3. Taking `DEFAULT` as a baseline, the results show that `INSPIRE` reduces the number of STAs in starvation by 80% and manages to increase the cumulated throughput by 600%. Considering the computational overhead of the strategies, we observe that the increase in the performance metrics comes at a computational cost. However, we argue that a computational overhead of around a tenth of a second remains acceptable when making decisions on WLANs. Comparing the computational overhead of `INSPIRE` and `INSPIRE_LIM`, we observe that capping the size of the dataset also caps the computational overhead, which is in line with the explanations detailed in Section 4.3.3. Overall, although its performance are slightly below those of `INSPIRE`, `INSPIRE_LIM` constitutes a significant improvement over the considered state-of-the-art strategies for all the considered performance metrics, including the average regret (see Table 4.2). Hence, it is a good alternative to `INSPIRE` if one accepts to trade off some empirical performance for a constant computational overhead.

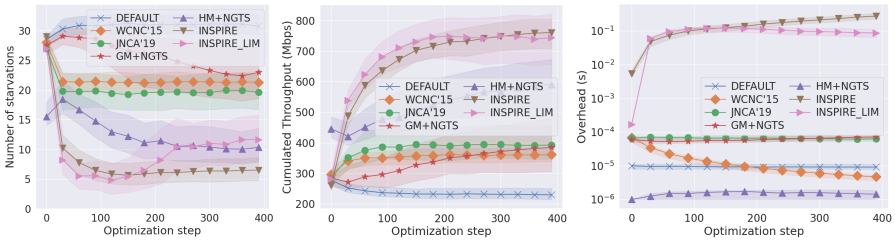



Figure 4.3: Performance metrics of `INSPIRE` on **T1**.

We now turn to the case of topology **T2**. The results depicted in Figure 4.4 show 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 best state-of-the-art strategy on this topology, by a margin of 14%. Looking at the performance of WLANs and of their STAs, observe that `INSPIRE` is able to limit the number of STAs starving from throughput by 36% when compared to the `DEFAULT` configuration. Similarly, the cumulated throughput of STAs have their value increased by 28% and nearly doubled with `INSPIRE`. Although the dimensionality of a configuration is larger for **T2** than for **T1** ($d = 28$ against $d = 20$), the computational overhead 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 4.3.3). Eventually, note that `INSPIRE_LIM` is slightly outperformed by `INSPIRE`, but remains roughly equivalent to the best considered state-of-the-art strategy.

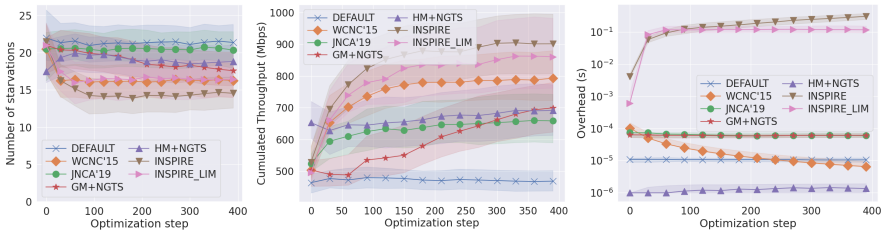



Figure 4.4: Performance metrics of `INSPIRE` on **T2**.

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 GPs which we further discuss in the next section.

## 4.5 DISCUSSION

### 4.5.1 *Two Scenarios with Different Complexity*

The topologies **T1** and **T2** may not seem so different from each other, but INSPIRE performed differently on each of them. By the end of the optimization process,[39] 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 the case of **T2** the progress was lower, 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 4.2, 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 inter-

[39] *That is, 400 steps.*

ference to the surrounding APs. Similarly, STAs that are further away from their AP are significantly more affected by the transmissions of concurrent APs.

To verify that **T2** constitutes a more complex example than **T1**, we examine the shape of the objective function in both cases. Because of the high dimensionality of the objective functions and the lack of a closed-form expression, we resort to a slicing technique to provide a visualization of the objective function (4.1). We postpone to Appendix B the details of this slicing. Figure 4.5 illustrates the obtained random slices for the objective function in **T1** and **T2**. It suggests a relatively smooth objective function in **T1**. On the other hand, the objective function in the case of **T2** is much more erratic, featuring a lot of local maxima.

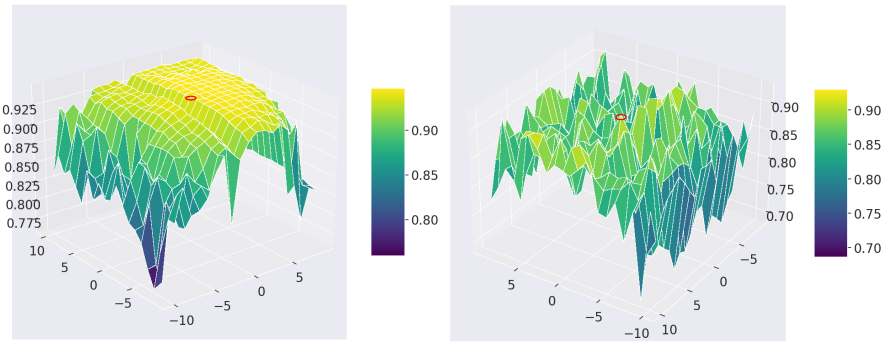



Figure 4.5: Random slices of the global objective function for the topologies **T1** and **T2**. The best solution found by `INSPIRE` is at $(0, 0)$. The maximum of the slice is shown with a red circle.

Interestingly, Figure 4.5 shows that `INSPIRE` succeeded to find a configuration that is maximal in this random slice of the objective function in **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 4.5 shows that, at least, a 6% better configuration exist. Nonetheless, we argue that `INSPIRE` was able to find an efficient configuration, which is already not easy given the erraticness of the objective function.

### 4.5.2 *The Benefits of Decentralization*

Eventually, to justify our choice of letting APs exploit only local information and prescribe network configurations to their surrounding APs, we consider two alternative versions of `INSPIRE`:

- `GPs w/o agg.`: each AP keeps using the local, altruistic objective function (4.2) but does not aggregate local prescriptions. In other words, each AP directly prescribes the value of its own parameters, without using the consensus (4.10);

- `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.

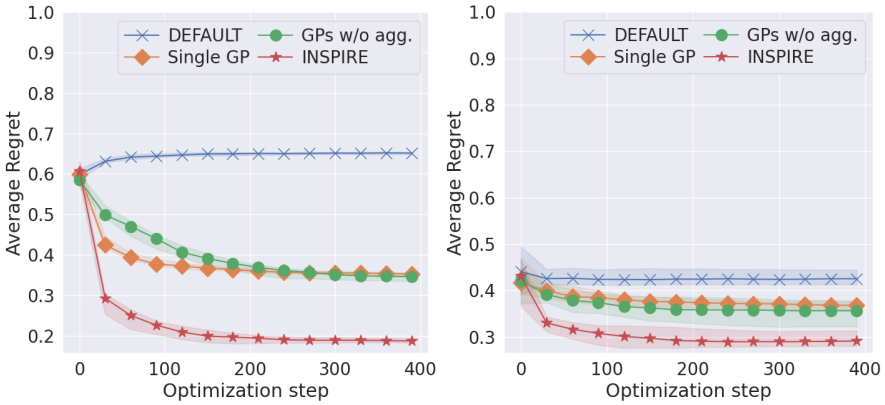



Figure 4.6: Average regret of alternative versions of `INSPIRE` on the topologies **T1** and **T2**.

We compare these alternative strategies with `INSPIRE` and the `DEFAULT` strategy by considering their average regret on topologies **T1** and **T2** in Figure 4.6. It shows that, on **T1**, the alternative strategies have an average regret 46% lower than `DEFAULT`. However, `INSPIRE` manages to reduce its average regret by an extra 25%. Despite the greater complexity of the function in **T2**, this extra reduction factor persists at a value of 13%. 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 WLAN. 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, recent works such as [87] bring evidence that having access to an additive decomposition of the objective function improves the performance of BO algorithms based on GPs, by lowering the variance of their predictions. Overall, `INSPIRE`, as a decentralized algorithm using a minimax optimal consensus (see Theorem 4.1), manages to improve the spatial reuse of the radio channel at the scale of the WLANs more than its simpler alternatives.

*[87] Wang et al. (2020), "Improving GP-UCB algorithm by harnessing decomposed feedback"*

## 4.6 SUMMARY AND LIMITATIONS

In this chapter, we proposed INSPIRE, a decentralized BO algorithm based on GPs to address the spatial reuse problem in WLANs. The algorithm iteratively discovers and simultaneously optimizes its objective function. It exploits a natural additive decomposition of this objective function to allow each AP to work only with other APs in its

communication range. Each AP agrees with its neighboring APs on a configuration to test thanks to a consensus that we proved optimal in the minimax sense. We compared INSPIRE to the state-of-the-art approaches for the spatial reuse problem in WLANs on a realistic simulator emulating real-world scenarios. INSPIRE demonstrated a significant improvement over the performance of state-of-the-art techniques. This efficiency comes at the cost of a larger computational overhead, that remains acceptable for this technological application. Nevertheless, we proposed an alternative strategy that uses a windowing method to cap the computational complexity of the algorithm without too much degrading its performance. Eventually, we empirically showed that naive and simpler alternative of INSPIRE (including a centralized version of the algorithm) are outperformed by our solution on the considered WLAN topologies. This suggests that the complex features of INSPIRE, namely its decentralized nature and its consensus function, are essential to quickly optimize the spatial reuse of the considered WLANs. INSPIRE was the subject of a publication in an international conference [88] (extended in a peer-reviewed journal [89]) and in a national conference [90]. Both of them awarded this work with their Best Paper Award.

*[88] Bardou et al. (2022), "INSPIRE: Distributed Bayesian Optimization for ImproviNg SPatIal REuse in Dense WLANs"*

*[89] Bardou et al. (2023), "Analysis of a decentralized Bayesian optimization algorithm for improving spatial reuse in dense WLANs"*

*[90] Bardou et al. (2022), "INSPIRE: Optimisation bayésienne distribuée pour l'amélioration de la réutilisation spatiale des WLANs denses"*

We believe that INSPIRE redefines the state-of-the-art of the spatial reuse optimization in WLANs. Moreover, we argue that its good empirical performance are not specific to this technological application, since the Assumptions 4.4 and 4.5 can be undoubtedly retrieved in other applications, as demonstrated in the next chapter.

Nevertheless, our approach keeps several limitations:

(i) **Design**: INSPIRE assumes that the configuration space $\mathcal{C}$ is a compact space. Although, in theory, the hardware of a 802.11 card could be configured with an arbitrary configuration, some manufacturers and standards may enforce a discretization of this space. An efficient method to adapt the recommendation of INSPIRE to a discrete space remains to be proposed.

(ii) **Design**: the computational overhead of INSPIRE may remain unacceptable for some technological applications.

(iii) **Theoretical**: an implicit assumption made by Theorem 4.1 is that each GP manages to find the maximal argument of its local objective function. This is not straightforward and needs to be properly verified.

(iv) **Theoretical**: INSPIRE does not provide the same asymptotic optimality guarantees as other BO algorithms. More work needs to be done to better understand the solution and modify it to provide an algorithm with stronger theoretical guarantees.

(v) **Evaluation**: the trade-off between the quality of the estimation of the objective function and the number of function calls in a given time interval remains to be discussed.

In the next chapter, we demonstrate that INSPIRE successfully thrives in another technological context: the cellular networks. Then, in Chapter 6, we address the two theoretical limitations (iii) and (iv) raised in the previous paragraph.

# 5

# ASSESSING THE BENEFITS OF NOMA FOR NEXT-GENERATION CELLULAR NETWORKS

**Outline.** In this chapter, we consider a technological context that differs from the Wireless Local Area Network (WLAN) introduced in Chapter 1 (Section 1.1.2) and optimized in Chapters 3 and 4. Instead, we focus on assessing the benefits of a new multiple access approach, called NOMA, which is envisioned for the future generations of cellular networks. We start by briefly introducing the cellular networks and their applications in Section 5.1. Then, we discuss the motivations of this contribution and we properly describe the problem we address in Sections 5.2 and 5.3 respectively. Finally, we conduct our experiments in Section 5.4 and we demonstrate that NOMA can bring significant gains providing it is properly configured. Through this chapter, we also show that INSPIRE, the algorithm developed in Chapter 4 (see Algorithm 4.1) can find applications in other technological contexts than WLANs.

## 5.1 ANOTHER TECHNOLOGICAL CONTEXT: CELLULAR NETWORKS

### 5.1.1 *Cellular Networks: Overview*

A cellular network is a high-speed, high-capacity, wireless telecommunication network designed for voice and data. As its name suggests, the network is distributed over land areas, called cells. Within each cell resides a base transceiver station, or Base Station (BS) for short, that ensures the network coverage of the cell. Joined together, the BS cover a large geographic area, providing connectivity to numerous end-users through their cellular devices (such as cell phones). Nowadays, the major part of the inhabited areas in the world are covered with a cellular network.

Nowadays, cellular networks still account for less traffic than WLANs, carrying between 2.5 [1] and 4.4 [91] less traffic data in 2020. Nevertheless, the increase in popularity of cellular devices gradually shifted the use cases of cellular networks from phone calls and entertainment to more critical applications. As an example, [92] identifies cellular networks as the lifeline of communication, since they have become the primary means of communication for finance-sensitive business transactions, lifesaving emergencies, and life/mission-critical services such as E-911 in the USA. Given these needs of efficiency and reliability, the local, analog wireless networks (deployed mainly in Japan and in the USA) constituting the first generation of cellular networks (1G)

*[91] Manson (2021), "Wireless network data traffic: worldwide trends and forecasts 2021–2026"*

*[92] Vacca (2014),* Network and system security

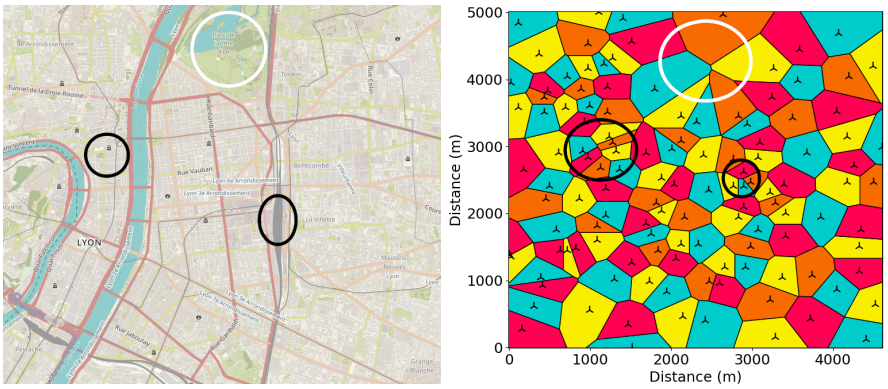
quickly gave way to the first commercial deployment of digital cellular networks (2G) in 1991, during the wireless revolution [93]. Since then, cellular networks have undergone three major updates, namely 3G, 4G and 5G. Each of these updates was released by the 3rd Generation Partnership Project (3GPP) and systematically introduced new frequency bands as well as higher data rates to face an increasing demand. The latest one (5G) was introduced in 2016. In this section, we describe some aspects of the cellular networks, relevant to the understanding and the study of the benefits of a new multiple access approach. For a thorough description of cellular networks, please refer to [94, 95].

*[93] Rappaport (1991), "The wireless revolution"*

*[94] Miao et al. (2016),* Fundamentals of mobile data networks

*[95] Ghosh et al. (2019), "5G evolution: A view on 5G cellular technology beyond 3GPP release 15"*

As mentioned before, the cell denotes the area of coverage of a BS. The division of the radio service into cells is based on terrain and reception characteristics. As an example, it can be based on the Signal to Noise Ratio (SNR) of the end-users' celullar devices, that is the ratio between the signal strength at reception and the ambient noise of the radio channel.




(a) (b)

Figure 5.1: (a) Map of the city of Lyon, France. (b) Visualization of a corresponding cellular network covering the city. A BS is depicted in black. Each cell of the cellular network is determined by a Voronoï cell. For the sake of identification, black circles indicate the most dense areas in the city and the white circle indicates the least dense area in the city.

Figure 5.1 depicts a potential cell partitioning covering the city of Lyon, France.[40] The locations of the BS are those of the main telecommunication operator in France, Orange,[41] extracted from the Autorité de Régulation des Communications Électroniques, des Postes et de la distribution de la presse (ARCEP) database.[42] The division into cells is based on the SNR of end-users cellular devices, assuming the same propagation model for all areas of the city, and an equal transmission power for all the BS.

To ensure a minimal Quality of Service (QoS) to the end-users, operators tend to correlate the density of deployed BS with the density

[40] *Map data © OpenStreetMap contributors. Tiles style by Humanitarian OpenStreetMap Team hosted by OpenStreetMap France.*

[41] https://www.orange.fr/portail

[42] https://files.data.gouv.fr/arcep_donnees/mobile/sites/2022_T3/

of cellular devices. As a few examples, observe that the most dense areas (namely the city center and the main train station, marked with black circles on Figure 5.1) require a denser deployment of BS and, consequently, smaller cells. Conversely, the least dense area (in terms of cellular devices) of Lyon, namely the Parc de la Tête d'Or (marked with a white circle on Figure 5.1), does not require a large number of BS.

### 5.1.2 *Intra-Cell Resource Allocation*

Within a single cell, multiple users are requesting service from the BS. To share the radio resource among them, a cell can use multiple techniques. Note that these techniques differ from the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, used by WLANs and discussed in Section 1.1.2. We start by discussing three of them, illustrated by Figure 5.2.

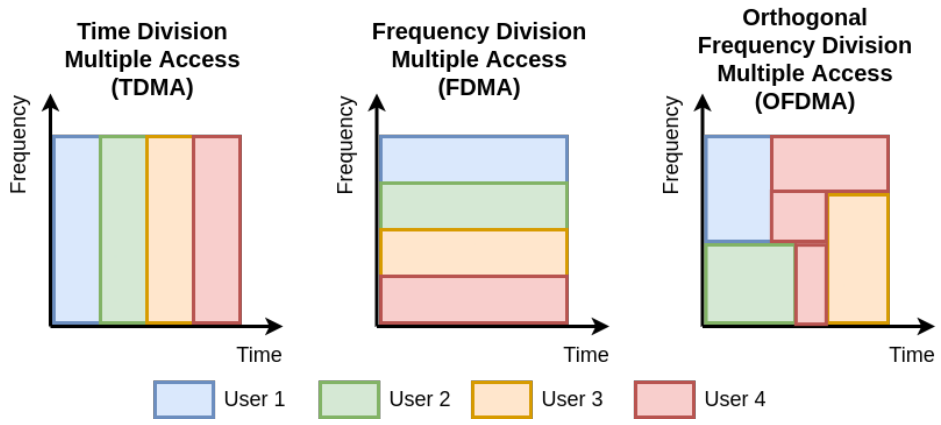



Figure 5.2: Resource allocation techniques employed in a cellular network cell.

**Time Division Multiple Access (TDMA).** TDMA consists in attributing non-overlapping time slots to end-users. Each time slot is used to transmit a segment of the data to a specific user, using the whole radio channel. The Global System of Mobile Communications (GSM) standard, used for the 2G cellular networks, is TDMA-based [96].

[96] *Mehrotra (1997), GSM system engineering*

**Frequency Division Multiple Access (FDMA).** FDMA, unlike TDMA, divides the radio channel into multiple non-overlapping individual frequency band. Each band can be used as long as necessary, to transmit data to a specific user. The cable television system is FDMA-based [97].

[97] *Ciochina et al. (2010), "A review of OFDMA and single-carrier FDMA"*

**Orthogonal Frequency Division Multiple Access (OFDMA).** OFDMA combines the advantages of TDMA and FDMA by allowing a division

within the frequency and the temporal domains. The radio channel is divided into multiple individual bands, and the time is divided into time slots. Resource blocks of different sizes can then be attributed to users according to their needs. Thanks to its flexibility, OFDMA is very spectrally efficient [97]. OFDMA is implemented in the last generations of cellular networks, namely the Long-Term Evolution (LTE) cellular system (belonging to 4G) and the 5G standard [98].

*[98] Rahnema et al. (2017),* From LTE to LTE-Advanced Pro and 5G

These three presented schemes belong to the class of Orthogonal Multiple Access (OMA) techniques. However, all OMA techniques (including OFDMA that brings more flexibility in the resource allocation) struggle to meet the high demands of next-generation cellular networks. Another recent resource allocation scheme, introduced by the 13th release of the 3GPP for 4G cellular networks, seems very promising since it uses overlapping resource blocks to increase the cell data rate. We introduce it below.

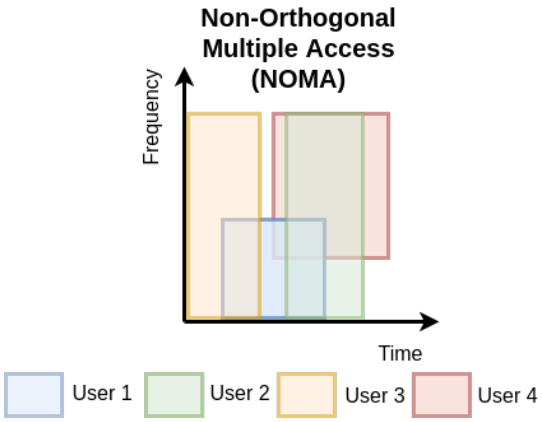



Figure 5.3: Illustration of NOMA using overlapping resource blocks.

**Non-Orthogonal Multiple Access (NOMA).** NOMA is able to use overlapping resource blocks for its end-users, as illustrated by Figure 5.3. To do so, NOMA exploits a technique called Successive Interference Cancellation (SIC) [99]. SIC allows a cellular device to decode multiple received signals, by decoding the strongest signal first[43] and subtracting it to the received data to gain access to the other signals, as illustrated by Figure 5.4. With NOMA, the same frequency bands can now be used to transmit to all the end-users of a cell. For more details about NOMA, one can refer to [100].

*[99] Sen et al. (2010), "Successive interference cancellation: A back-of-the-envelope perspective"*

[43] *The other, weakest signals are temporarily treated as interference.*

*[100] Kizilirmak et al. (2016), "Non-orthogonal multiple access (NOMA) for 5G networks"*

### 5.1.3 *Inter-Cell Coordination*

The resource allocation schemes described in Section 5.1.2 (namely TDMA, FDMA, OFDMA and NOMA) allow the BS to communicate with the intra-cell users without interference (or at least to deal with them in the case of NOMA). However, the users at the edge of a cell still

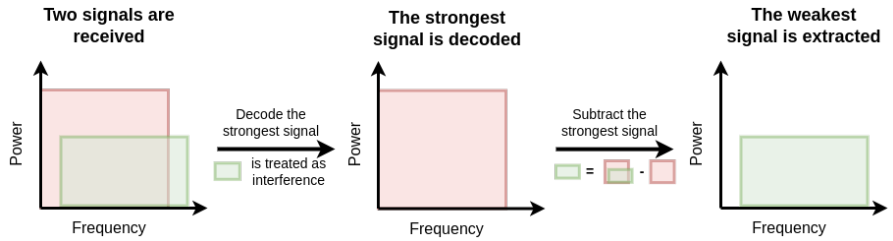



Figure 5.4: Illustration of SIC on the transmission of two overlapping resource blocks.

experience interference from the neighboring cells. This is one of the main problems affecting the QoS of cellular networks, called Inter-Cell Interference (ICI). To address ICI, the research community in cellular networks proposed many Inter-Cell Interference Coordination (ICIC) strategies [101]. In this section, we introduce the three strategies considered in our study.

*[101] Hamza et al. (2013), "A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks"*

#### 5.1.3.1 *Coloring*

One of the most naive strategy for mitigating ICI is to allocate different fractions of the frequency band to neighboring cells, so they cannot interfere. This is ensured by computing a coloring of the cells. That is, each cell is assigned a color in such a way that two neighboring cells cannot have the same color, as suggested by Figure 5.1b and illustrated with Figure 5.5. This forces neighboring cells to emit in different fractions of the frequency band.

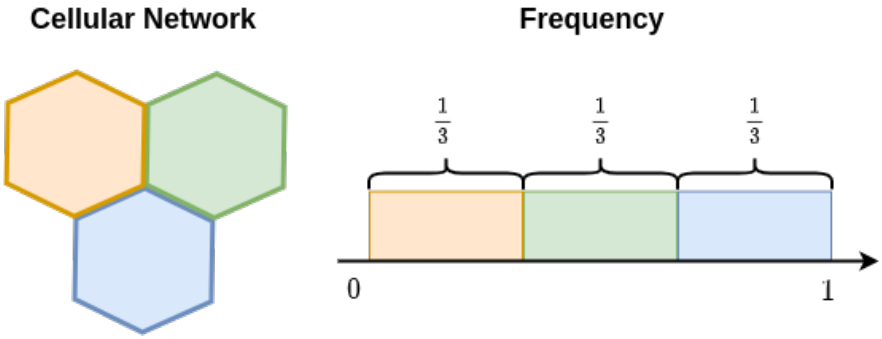



Figure 5.5: Illustration of the coloring strategy with three hexagonal cells and a frequency band normalized between 0 and 1.

Naturally, this strategy significantly reduces the spectral efficiency of the cellular network, as a cell uses only a fraction of the frequency band. Note that, by the four colors theorem [102], each cellular network is at most 4-colorable. Therefore, in the worst case, each cell would be allowed to transmit using 25% of the frequency band only. Naturally, this significantly hampers the spectral efficiency of the cellular network.

*[102] Appel et al. (1977), "The solution of the four-color-map problem"*

#### 5.1.3.2 *Fractional Frequency Reuse*

Fractional Frequency Reuse (FFR) proposes to divide the end-users of each cell into two subsets, based on their geographic positions. As a consequence, these two subsets define two areas, namely an inner and an outer area (see Figure 5.6). The inner areas of each cell, assumed too far from each others to cause and experience mutual interference, are allowed to use the same fraction of the frequency band. The outer areas apply the coloring strategy (see Section 5.1.3.1) on the remaining of the frequency band.

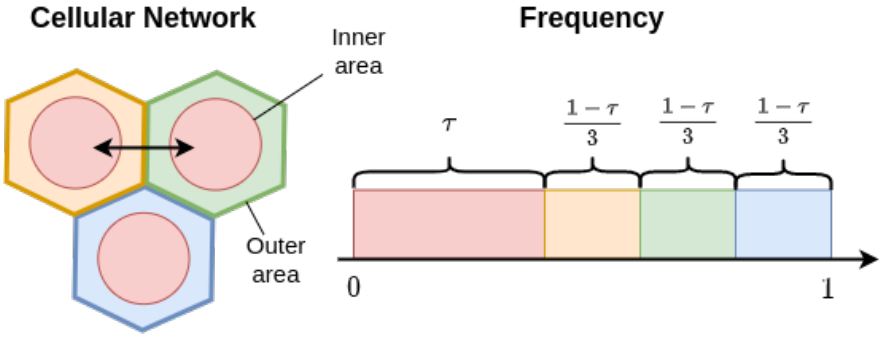



Figure 5.6: Illustration of FFR with three hexagonal cells and a frequency band normalized between 0 and 1. The interference between two neighboring cells is depicted with a black two-headed arrow.

Figure 5.6 illustrates the split of the frequency band proposed by FFR. It is more spectrally efficient than the simple coloring strategy since the inner areas within each cell can reuse a fraction of the frequency band. FFR can be controlled by two parameters:

(i) the radius of the inner area and,

(ii) the fraction $\tau$ of the frequency band allocated to the inner areas.

As far as we know, no derivation of the optimal values for these two parameters has reached a consensus in the cellular network community. However, numerous heuristics have been proposed, especially to derive a suitable value of $\tau$ [103–105].

*[103] Xu et al. (2012), "Throughput and optimal threshold for FFR schemes in OFDMA cellular networks"*

*[104] Elwekeil et al. (2019), "Performance evaluation of an adaptive self-organizing frequency reuse approach for OFDMA downlink"*

*[105] Ezhilarasi et al. (2021), "Maximization of sum throughput in LTE heterogeneous network using region splitting-based resource partitioning scheme"*

#### 5.1.3.3 *Combining FFR and NOMA?*

Although FFR can improve the spectral efficiency of the cellular network as compared to the coloring strategy, there is still room for improvement as the frequency band allocated to each outer area is significantly reduced. To tackle this problem, we consider a strategy that has not yet been implemented at large scale: a combination between FFR and NOMA. This combination would allow the cellular network to take advantage of the SIC features. In particular, each cell of the network could use overlapping resource blocks to serve its inner and outer users (as illustrated by Figures 5.3 and 5.4).

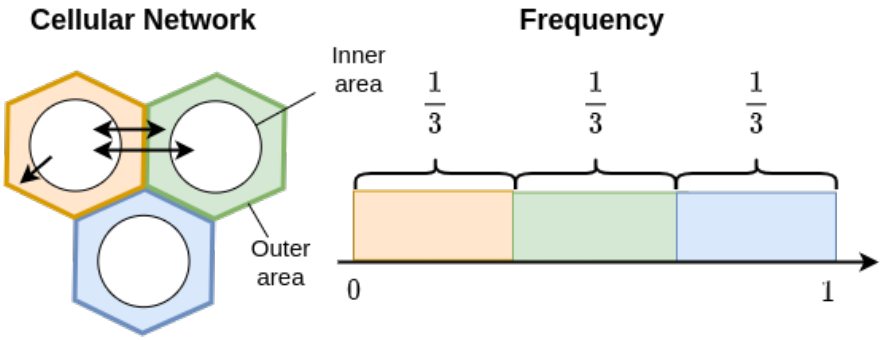



Figure 5.7: Illustration of NOMA used in conjunction with FFR, on three hexagonal cells and a frequency band normalized between 0 and 1. The white color depicts the use of the whole frequency band. The interference between two neighboring cells is depicted with black arrows.

Figure 5.7 illustrates how the combination of NOMA and FFR differs from pure FFR (Figure 5.6). Observe that the inner areas are now able to exploit the whole frequency band, while the outer areas apply the simple coloring strategy on the whole frequency band to protect its end-users from interference. Thanks to SIC, users in the inner area do not experience interference with the users in the outer area of the same cell. However, observe that the inner area of the orange cell causes three types of interference (depicted with the areas of the green cell only for the sake of clarity):

(i) with its own cell's outer area, because of SIC,

(ii) with the other cells' outer areas,

(iii) with the other cells' inner areas.

## 5.2 MOTIVATIONS

As mentioned in Section 5.1.2, NOMA is a very promising resource allocation scheme recently introduced by 3GPP but has not yet been implemented at a large scale. This is mainly due to the complexity of the SIC algorithms that require a significant amount of computing power at the end users. Some doubts regarding the efficiency of NOMA were also raised in the research community. As an example, [106] argue that FFR and NOMA are incompatible. However, this claim is contradicted by [107, 108]. Overall, the performance of the combination of NOMA and FFR in a multi-cells context remains largely unknown. In this chapter, we consider the three ICIC strategies described in Section 5.1.3, namely the coloring strategy, FFR and the combination of FFR and NOMA. Each strategy has its advantages and drawbacks:

[106] *Wang et al. (2019), "User association and power allocation for multi-cell non-orthogonal multiple access networks"*

[107] *Vaezi et al. (2019), "Non-orthogonal multiple access: Common myths and critical questions"*

[108] *Ramesh et al. (2022), "Non orthogonal multiple access requirements for 5G and its myths"*

- The coloring strategy causes little interference but is also less spectrally efficient. This could reduce the overall QoS of the cellular network.
- FFR is more spectrally efficient as each cell is allowed to reuse a fraction of the frequency band in its inner area. However, this causes additional interference between the inner areas of neighboring cells, as illustrated by Figure 5.6.
- The combination of NOMA and FFR is the most spectrally efficient strategy as it is able to exploit overlapping resource blocks. However, this causes additional interference within the cell and between neighboring cells, as illustrated by Figure 5.7.

Using INSPIRE, the solution developed in Chapter 4 (see Algorithm 4.1), we propose to optimize different cellular networks under these three ICIC strategies. Doing so, we are able to evaluate and compare the ICIC strategies in a multi-cells context by comparing the performance achieved by the optimized cellular networks. In particular, our study provides material that clearly show the benefits of the combination of FFR and NOMA and could be of some interests for the design choices of the next-generation of cellular networks.

## 5.3 PROBLEM FORMULATION

*[109] Banerjee et al. (2022), "Joint Power and Subcarrier Allocation in Multi-Cell Multi-Carrier NOMA"*

Regarding the optimization of cellular networks, most contributions, such as [109], propose to optimize the frequency band and the transmission power of the BS at the scale of each end-user. In this chapter, we tackle a different optimization problem. We consider only two transmission powers for each cell, one for the inner area and one for the outer area (see Section 5.3.1). By tuning these two transmission powers, we control more easily the interference and the association of the end-users to one BS and one of the cell areas. Through an efficient configuration of the transmission powers at each BS, we seek to optimize the QoS of the cellular network (see Section 5.3.2).

### 5.3.1 *Problem Parameters*

As mentioned before, each cell $i$ is required to tune its transmission powers, denoted $p_1^{(i)}$ and $p_2^{(i)}$ for its inner and outer area, respectively. They are to be tuned within a given domain $[a, b]^2$, under the two following constraints:

(i) by construction, the transmission power for the inner area must be lower than the transmission power for the outer area, that is $p_1^{(i)} \leq p_2^{(i)}$,

(ii) the sum of the transmission powers must not exceed the maximal power capacity of a BS, denoted $P^+$, that is $p_1^{(i)} + p_2^{(i)} \leq P^+$.

Because of the constraints (i) and (ii), the domains of $p_1^{(i)}$ and $p_2^{(i)}$ are not rectangular. This is a problem for some optimization algorithms. To design parameters with rectangular domains that naturally incorporate these constraints, we define

(i) $\theta_1^{(i)} = p_1^{(i)} + p_2^{(i)}$ the total transmission power of the BS, to be chosen within the interval $[2a, P^+]$, and

(ii) $\theta_2^{(i)} = \frac{p_2^{(i)}}{\theta_1^{(i)}}$ the fraction of the transmission power allocated to the outer area, to be chosen within the interval $\left[\frac{1}{2}, 1\right[$.

Naturally, the two transmission powers $p_1^{(i)}$ and $p_2^{(i)}$ can be retrieved since $p_1^{(i)} = \theta_1^{(i)}\left(1 - \theta_2^{(i)}\right)$ and $p_2^{(i)} = \theta_1^{(i)}\theta_2^{(i)}$. Note that a cellular network comprising $n$ cells has $2n$ parameters to be tuned.

### 5.3.2 *Objective Function*

In this chapter, as in Chapters 3 and 4, we would like to rely on the end-users achieved throughputs to build our objective function. In the previous chapters, we instrumented a well-known simulator to collect these throughputs. However, as far as we know, there is no well-established open-source simulator in the cellular network community. Therefore, for the sake of simplicity and of the generality of our results, we rather rely on an upper bound on the achieved throughput of each end-user, that is the Shannon capacity [110].

*[110] Kemperman (1974), "On the Shannon capacity of an arbitrary channel"*

The Shannon capacity $C_j$ for an end-user $j$ associated with a BS $i$ is an upper bound on the throughput that the end-user is able to achieve. It is expressed in bits per second and is defined as

$$C_j = W_j \log_2 (1 + S_{i,j}) \tag{5.1}$$

with $W_j$ the bandwidth allocated to the end-user $j$ and $S_{i,j}$ the Signal to Interference plus Noise Ratio (SINR) of the transmission between the BS $i$ and the end-user $j$. To ease exposition, more details about the computation of SINR are provided in Section 5.3.3.

As with a WLAN, a cellular network must achieve a satisfying QoS through a good trade-off between a large system throughput (*i.e.* the cumulative sum of the individual throughput of each end-user) and a high level of fairness. In Chapters 3 and 4, we considered that the Proportional Fairness (PF) was a satisfying trade-off (see Sections 3.2.1 and 4.2.2 for more details). In this chapter, we consider a metric that generalizes multiple fairness measures (including PF), called the $\alpha$-fairness [111]. It is defined, for $\alpha \in [0, +\infty[$ and a $d$-dimensional

*[111] Mo et al. (2000), "Fair end-to-end window-based congestion control"*

vector $\boldsymbol{x}$ of the Shannon capacity for the $d$ end-users of the cellular network, as

$$f_\alpha(\boldsymbol{x}) = \sum_{i=1}^{d} \frac{x_i^{1-\alpha}}{1-\alpha}. \tag{5.2}$$

For $\alpha = 1$, (5.2) is not defined. However, observe that

$$\nabla f_\alpha(\boldsymbol{x}) = \left(x_1^{-\alpha}, \cdots, x_d^{-\alpha}\right). \tag{5.3}$$

Therefore, $\lim_{\alpha\to 1} \nabla f_\alpha(\boldsymbol{x}) = \left(x_1^{-1}, \cdots, x_d^{-1}\right)$. In addition, note that $\lim_{\alpha\to 1} f_\alpha(\boldsymbol{x}) = \lim_{\alpha\to 1} \left(\int \nabla f_\alpha\right)(\boldsymbol{x}) = \left(\int \nabla f_1\right)(\boldsymbol{x})$. Computing $\int \nabla f_1$ is immediate and yields $\left(\int \nabla f_1\right)(\boldsymbol{x}) = C + \sum_{i=1}^{d} \log x_i$, with $C$ an integration constant. $C$ does not have any influence on the maximization of $f_1$. Therefore, in the following, we set $C = 0$ and we have

$$\lim_{\alpha\to 1} f_\alpha(\boldsymbol{x}) = \sum_{i=1}^{d} \log x_i. \tag{5.4}$$

Therefore, (5.2) can be extended to

$$f_\alpha(\boldsymbol{x}) = \begin{cases} \sum_{i=1}^{d} \log x_i & \text{if } \alpha = 1 \\ \sum_{i=1}^{d} \frac{x_i^{1-\alpha}}{1-\alpha} & \text{otherwise.} \end{cases} \tag{5.5}$$

As mentioned before, $\alpha \in [0, +\infty[$. Its value controls the importance of the fairness in the objective function. In fact, setting $\alpha = 0$ enforces virtually no fairness in $\boldsymbol{x}$ as this boils down to optimizing $f_0 = \sum_{i=1}^{d} x_i$. Recall that $\boldsymbol{x}$ denotes the Shannon capacity of the $d$ end-users of the cellular network, therefore $f_0(\boldsymbol{x})$ is the capacity of the network. As $\alpha$ increases, the fairness constraint becomes stronger. As can be seen with (5.5), $\alpha = 1$ corresponds to the PF. Asymptotically, considering $f_\infty(\boldsymbol{x}) = \lim_{\alpha\to+\infty} f_\alpha(\boldsymbol{x})$, we have

$$f_\infty(\boldsymbol{x}) = \min_{i\in[1,d]} x_i. \tag{5.6}$$

Note that (5.6) is one of the most stringent definitions of fairness, as the objective value of the entire cellular network is defined as the lowest capacity experienced by a single user.

Finally, observe that (5.5) is a sum of $d$ terms, each term being associated to the capacity of a user. Therefore, we can build an additive decomposition with any partition of the set of users. As an example, a partition of users could be $\{\mathcal{U}(i)\}_{i\in[1,n]}$, with $\mathcal{U}(i)$ denoting the users associated with cell $i$. More generally, for a partition of users $\mathcal{P} = \{P_i\}_{i\in[1,k]}$, $k > 0$, we have

$$f_\alpha(\boldsymbol{x}) = \sum_{\mathcal{P}_i\in\mathcal{P}} \sum_{j\in\mathcal{P}_i} \frac{x_i^{1-\alpha}}{1-\alpha} \tag{5.7}$$

$$= \sum_{\mathcal{P}_i\in\mathcal{P}} f_\alpha^{(i)}(\boldsymbol{x}_{\mathcal{P}_i}) \tag{5.8}$$

with $\boldsymbol{x}_{\mathcal{P}_i}$ the projection of $\boldsymbol{x}$ onto the dimensions whose indices are in $\mathcal{P}_i$.

### 5.3.3 *Users Association to BS and Cell Areas*

In the cellular network, each user must be associated with the BS of a cell before being associated to either the inner or the outer area of the cell (that is, to a SIC level). We discuss the two association methods chosen in this section.

#### 5.3.3.1 *Association with a BS*

For a given user, the association with a BS is based on the SINR at reception. For a BS $i$ and a user $j$, the SINR is denoted by $S_{i,j}$ and defined by

$$S_{i,j} = \frac{p^{(i)} L_{i,j}}{N + \sum_{k \in \mathcal{I}(j)} p^{(k)} L_{k,j}} \tag{5.9}$$

with $p^{(i)}$ the transmission power of the signal at the BS (either $p_1^{(i)}$ or $p_2^{(i)}$ depending on the location of the end-user in the cell), $L_{i,j}$ the loss of the signal between BS $i$ and user $j$, $N$ a background noise and $\mathcal{I}(j)$ the set of BS interfering with user $j$.

To determine which BS to associate with, a user collects the SINR for all its surrounding BS (by considering $p_1^{(i)}$ for their inner areas and considering $p_2^{(i)}$ for their outer areas), and associates with the BS that provides the largest SINR. Note that the set of interfering BS $\mathcal{I}(j)$ also depends on which area (inner or outer) the user $j$ belongs to (see Section 5.1.3 for more details).

By tuning the cellular network configuration (that is, the transmission powers of each BS), the SINRs will change and end-users will naturally change their associations to BS. All other things being equal, a BS increasing its transmission power will naturally serve more users, and hence increase the area of its cell.

#### 5.3.3.2 *Assignment to a Cell Area*

Once the users are associated with a cell, they need to be associated with either the inner or the outer area of the cell. Finding the association that maximizes the objective function $f_\alpha$ is a difficult combinatorial problem, as it boils down to finding the optimal 2-partition of the set of users among the $2^{d_i - 1} - 1$ that exist.[44]

To circumvent this combinatorial problem, we reduce the number of candidate associations by sorting the users according to their SNR at reception.[45] Then, we split the sorted list of users in two: those with the lowest SNR are affected to the outer area, the others are affected to the inner area. This drastically reduces the number of associations to

[44] *Here, $d_i$ denotes the number of users in the cell.*

[45] *They are necessarily known from the BS as its users are associated with a BS based on their SINR.*

$d_i - 1$ candidates. We argue that the number of candidate associations is now sufficiently reduced to discover the optimal users association with a greedy algorithm. Therefore, within each cell, the $\alpha$-fairness (see (5.5)) corresponding to each candidate association is computed, and the association achieving the largest $\alpha$-fairness value is chosen.

### 5.3.4 *Optimal Scheduler*

Thanks to NOMA, the BS can use overlapping resource blocks to communicate with inner and outer users concurrently. However, two users sharing the same SIC level[46] must still use orthogonal resource blocks and, hence, share the resource in the frequency and/or time domain. In this section, we derive the optimal fraction of the resource to allow to each user $j$, so that the objective function (5.5) applied to the users of a given area in a cell is maximized. We opt for the optimal scheduler, as it is the only one that guarantees no negative impact on the objective function values. In fact, any other scheduler resulting from arbitrary design choices is likely to reduce the objective function values and, hence, to prevent us from studying the impact of the considered ICIC strategies on the objective function.

[46] *That is, two users in the same area (inner or outer).*

Given the capacities $\boldsymbol{c} = (c_1, \cdots, c_k)$ of $k$ users, we seek to solve

$$\begin{aligned} &\underset{\boldsymbol{t}\in[0,1]^k}{\text{maximize}}\ f_\alpha\left(\boldsymbol{t} \odot \boldsymbol{c}\right) \\ &\text{such that } \boldsymbol{t}^\top \mathbf{1} = 1 \end{aligned} \tag{5.10}$$

with $\boldsymbol{t}$ the scheduling, $\mathbf{1}$ the conformable vector of ones and $\odot$ the Hadamard product.

**Proposition 5.1.** *Let $c_1, \cdots, c_k$ the Shannon capacities of $k$ users that need to share the resource, the optimal scheduling $\boldsymbol{t}^* = (t_1^*, \cdots, t_k^*)$ is*

$$\boldsymbol{t}^* = \begin{cases} \left(\mathbb{1}_{i=i^*}\right)_{i\in[1,k]} & \text{if } \alpha = 0, \\ \left(\frac{c_i^{(1-\alpha)/\alpha}}{\sum_{i=1}^k c_i^{(1-\alpha)/\alpha}}\right)_{i\in[1,k]} & \text{otherwise} \end{cases} \tag{5.11}$$

*with $i^* = \arg\max_{i\in[1,k]} c_i$ and $\mathbb{1}_{i=i^*}$ the indicator function indicating if $i$ is equal to $i^*$.*

*Proof.* Since the problem (5.10) boils down to finding the optimal convex combination of $c_1, \cdots, c_k$, let us consider the Lagrangian relaxation when $\alpha \neq 1$, that is

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{t}, \lambda) &= f_\alpha(\boldsymbol{t}^\top \boldsymbol{c}) - \lambda(\boldsymbol{t}^\top \mathbf{1} - 1) \\ &= \frac{1}{1-\alpha} \sum_{i=1}^k t_i^{1-\alpha} c_i^{1-\alpha} - \lambda \left( \sum_{i=1}^k t_i - 1 \right) \end{aligned} \tag{5.12}$$

It is known that finding the critical points of $\mathcal{L}$ provides the maximal argument of $f_\alpha(\boldsymbol{t}^\top \boldsymbol{c})$. Let us consider the system that a critical point of (5.12) yields, that is

$$\nabla \mathcal{L}_\alpha = \mathbf{0} \implies \begin{cases} c_1^{1-\alpha} t_1^{-\alpha} - \lambda = 0, \\ \vdots \\ c_k^{1-\alpha} t_k^{-\alpha} - \lambda = 0, \\ 1 - \sum_{i=1}^{k} t_i = 0 \end{cases} . \tag{5.13}$$

Setting $\alpha \neq 0$, solving the system (5.13) is trivial and yields a single critical point, with $t_i = \frac{c_i^{(1-\alpha)/\alpha}}{\sum_{i=1}^{k} c_i^{(1-\alpha)/\alpha}}$, $\forall i \in [1, k]$. This is necessarily the optimal scheduling we are looking for. Note that solving the case $\alpha = 0$ is trivial as it boils down to allocating all the resource to the user with maximal capacity, at the expense of all the others.

Finally, considering the relaxation for $\alpha = 1$ (see (5.5)), the Lagrangian is

$$\mathcal{L}_1(\boldsymbol{t}, \lambda) = \sum_{i=1}^{k} \log t_i c_i - \lambda \left( \sum_{i=1}^{k} t_i - 1 \right)$$

and looking for the critical points, we have

$$\nabla \mathcal{L}_1 = \mathbf{0} \implies \begin{cases} \frac{1}{t_1} - \lambda = 0, \\ \vdots \\ \frac{1}{t_k} - \lambda = 0, \\ 1 - \sum_{i=1}^{k} t_i = 0 \end{cases} . \tag{5.14}$$

The system (5.14) can be trivially solved and yields a single critical point, with $t_i = \frac{1}{k}$. Consequently, this is also the maximal argument of our objective function. This concludes our proof. □

### 5.3.5 *Optimization of the Objective Function*

We now have all the information to characterize the addressed optimization problem. This is a distributed optimization problem (of high dimension when the number of BS is large) whose parameters are described in Section 5.3.1 and objective function is (5.5) (see Section 5.3.2). Unfortunately, the relation between a change in the transmission powers of the BS and the distribution of the end-users on the different cells and cell areas (see Section 5.3.3) is hard, if not impossible, to express in a closed-form. The relation between the cellular network configuration and the scheduling of each cell (see Section 5.3.4) is also hard to write down precisely. Because the different associations

and the scheduling have a direct impact on the objective function, the objective function (5.5) can be seen as a black-box.

Therefore, we are facing a distributed black-box optimization problem, and we use INSPIRE, the solution developed in Chapter 4 (see Algorithm 4.1) to explore and simultaneously optimize (5.5).

## 5.4 NUMERICAL RESULTS

In this section, we describe in Section 5.4.1 the setting implemented to assess the performance of a cellular network using the three ICIC strategies described in Section 5.1.3, before providing and comparing the obtained performance metrics in Section 5.4.2.

### 5.4.1 *Experimental Setting*

In these experiments, we compare the three ICIC strategies described in Section 5.1.3. We briefly recall them below:

(i) `COLORING`: the control ICIC strategy, consisting into simply coloring the cellular network as described in Section 5.1.3.1,

(ii) `FFR`: the FFR solution, as described in Section 5.1.3.2,

(iii) `NOMA + FFR`: the combination of NOMA and FFR, as described in Section 5.1.3.3.

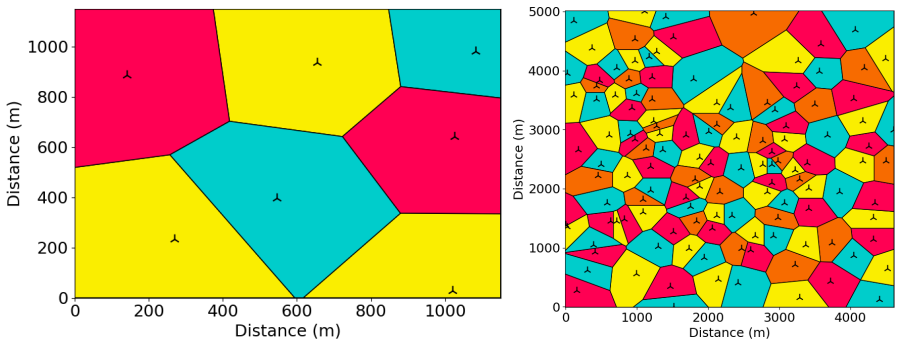
We evaluate the performance of these solutions with an homemade simulator, implementing the users association to the cells and to the SIC levels, as well as the optimal scheduling, as descibed in Sections 5.3.3 and 5.3.4. The simulator is instrumented to collect the Shannon capacity of each user and the Jain's fairness index [60] at the network level. Table 5.1 gathers the parameters of the simulator and their values.

Table 5.1: Homemade simulator parameters for the evaluation of the ICIC strategies.

| **Parameter** | **Value** |
|---|---|
| Number of iterations | 110 |
| Background noise | -100 dBm/Hz |
| Path loss | `LogDistance(`$d_{\texttt{ref}} = 1$ `km,` $L_{\texttt{ref}} = 128.1$ `dBm,` $n = 3.76$`)` |
| Bandwidth | 20 MHz |

At each optimization step, given the previous configurations of the cellular network and their objective values, we use INSPIRE (Chapter 4, Algorithm 4.1) to find a promising configuration of the cellular network and maximize the objective function $f_\alpha$. For `FFR` and `NOMA +`

FFR, the configuration of each BS is a set of two parameters describing respectively the total transmission power of the BS and the fraction of the power used for the inner area (as described in Section 5.3.1). However, in the case of FULL REUSE and COLORING, cells do not have an inner area and each BS is configured with a single parameter only, describing its transmission power.




(a) **T1** (b) **T2**

Figure 5.8: Cellular networks for the evaluation of the ICIC strategies.

Finally, we consider two different cellular networks for the benchmarking of the ICIC strategies, depicted by Figure 5.8. Figure 5.8a illustrates a small cellular network, denoted **T1**. It is composed of 7 BS and 1400 users (not shown here), distributed according to a uniform grid. Figure 5.8b depicts a more realistic cellular network covering the city of Lyon (France) with the 125 BS of the main french telecommunication operator (Orange). Please refer to Section 5.1.1 for more details about the position of the BS. As for the users (not shown here), they are generated by a Log-Gaussian Cox process [112], to have a realistic users distribution with spatial heterogeneity. More concretely, we used the method described in [113] to generate the users. To have a realistic number of users, we reproduced the population density of Lyon (around 11,000 inhabitants per squared kilometer) and we selected a fraction of these users, based on the market share of Orange.[47] Then, we assumed that only 10% of these users were actually active users on the cellular network, leading us to randomly remove 90% of them. Overall, **T2** ends up with a total of around 9,000 users.

*[112] Møller et al. (1998), "Log gaussian cox processes"*

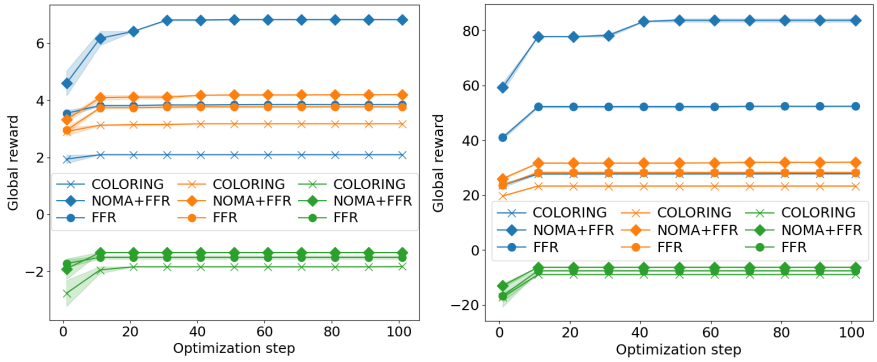
*[113] Wang et al. (2014), "The impact of user spatial heterogeneity in heterogeneous cellular networks"*

*[47] 38.5% according to Statista (https://fr.statista.com/themes/3851/orange/#topicOverview).*

### 5.4.2 *Performance Results*

First of all, let us consider the performance of INSPIRE on the objective function $f_\alpha$. Recall that $\alpha$ determines the importance of fairness in the objective function. Figure 5.9 gathers the results for three values of $\alpha$. Note that each set of colored curves evolves on a different range of values, as it corresponds to a different value of $\alpha$. On both cellular networks **T1** and **T2** and regardless of the value of $\alpha$, INSPIRE is able

to consistently optimize $f_\alpha$ by discovering promising configurations of the networks. Figure 5.9 also shows that the combination `NOMA + FFR` consistently obtains better objective function values for the considered values of $\alpha$, followed by `FFR` and `COLORING`. This strongly suggests that among the considered ICIC strategies, `NOMA+FFR` is able to achieve the best trade-off between the Shannon capacity and the fairness on both cellular networks.




(a) **T1**, 3 replications (b) **T2**, 3 replications

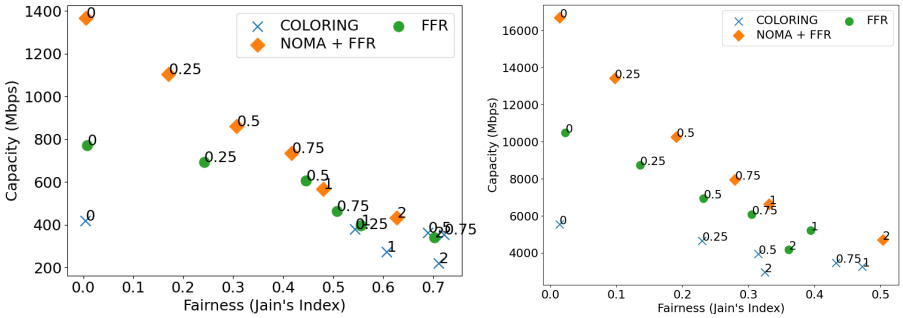
Figure 5.9: Graph of maximal reward (scaled down for visualization purposes) w.r.t. the optimization steps for the considered ICIC strategies on (left) **T1** and (right) **T2**. Multiple values of $\alpha$ are considered: (blue) $\alpha = 0$, (orange) $\alpha = 0.5$ and (green) $\alpha = 2$. The average performance is shown with a solid line, while the standard errors are depicted with shaded areas.

To confirm this, let us consider the Shannon capacity and Jain's fairness index achieved by each strategy. Note that, from the perspective of multi-objective optimization, $f_\alpha$ is a scalarization as it is able to turn the multi-objective problem into a scalar optimization problem. To better compare the different ICIC technologies, let us consider for each experiment the best configuration found (according to $f_\alpha$) and visualize it in the output space (that is, the space where the capacity and the fairness have their own axis) [114]. If a configuration $\boldsymbol{x}_1$ achieves a higher throughput *and* a higher fairness than a configuration $\boldsymbol{x}_2$, then $\boldsymbol{x}_2$ is said to be Pareto-dominated [115] by $\boldsymbol{x}_1$.

*[114] Deb et al. (2013), "Multi-objective optimization"*

*[115] Voorneveld (2003), "Characterization of pareto dominance"*

Figure 5.10 depicts, in the output space, the best configurations found by INSPIRE for different values of $\alpha$ (ranging from 0 to 2) and the considered ICIC strategies. The combination `NOMA + FFR` appears clearly as the ICIC strategy able to achieve the highest capacity, without trading off on the network fairness. In fact, most of the configurations found for `FFR` or `COLORING` in the output space of **T1** (Figure 5.10a) and **T2** (Figure 5.10b) are Pareto-dominated by a configuration of `NOMA + FFR`. This confirms that the combination of NOMA and FFR is very promising, as it is very likely to improve the throughput of cellular network without leading to unfair situations.




(a) **T1** (b) **T2**

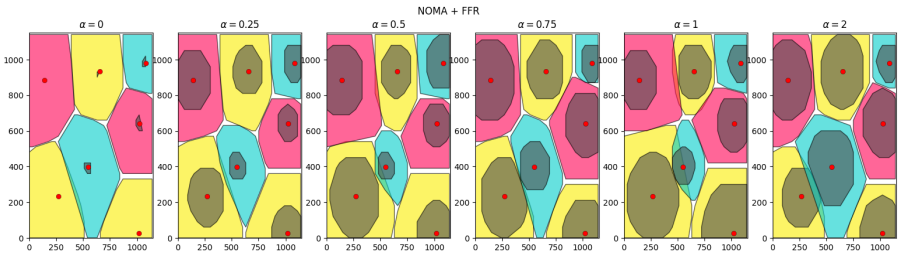
Figure 5.10: Performance of the considered ICIC strategies in the output space. The value of $\alpha$ used to find this configuration is written next to each point.

Finally, to understand how the cellular network answers to an increased demand of fairness among the end-users, let us visualize **T1** and **T2** under the best configurations found according to $f_\alpha$, for different values of $\alpha$.

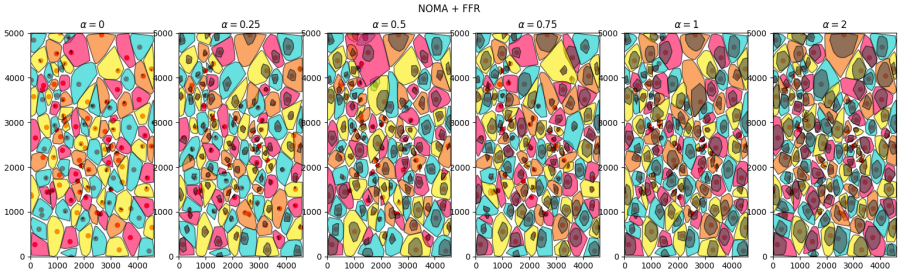
Figure 5.11 depicts the cellular networks **T1** (Figure 5.11a) and **T2** (Figure 5.11b) under such configurations. Observe that, for $\alpha = 0$, the inner areas are small and comprise a few users only. Doing so, these lucky users get an excellent QoS, at the expense of all the other users who get a very poor QoS. This configuration actually maximizes the objective function value $f_0$ by achieving a very high Shannon capacity at the network scale. However, as soon as some fairness is required (with $\alpha > 0$), the inner areas start inflating. More generally, for the tested values of $\alpha$, the larger $\alpha$, the bigger the inner areas. The effect of $\alpha$ on the throughput distribution of each cell is studied in Appendix C.

## 5.5 SUMMARY AND LIMITATIONS

In this chapter, we have exploited INSPIRE, the decentralized Bayesian optimization solution developed in Chapter 4 to propose a power control algorithm in cellular networks. This algorithm tries to achieve a compromise between the Shannon capacity and Jain's fairness index by optimizing a scalarization: the $\alpha$-fairness. By tuning $\alpha$, the amount of fairness in the objective function can be controlled. Thanks to INSPIRE, we were able to compare different strategies to prevent interference at the edge of the cells in cellular networks. More precisely, we studied the combination of a new resource allocation scheme (NOMA) and an ICIC technique (FFR). We demonstrated, through simulation, that this combination seems very promising as it manages to Pareto-dominate other classical ICIC strategies such as the pure FFR and the




(a) **T1**




(b) **T2**

Figure 5.11: Visualization of the cellular networks **T1** and **T2** under the best configurations found by INSPIRE for the combination `NOMA + FFR` and different values of $\alpha$. Each cell is shown as a colored polygon, with another shaded polygon inside it to depict the inner area. Each BS is shown as a red dot.

coloring of the cellular network, in terms of Shannon capacity and Jain's fairness index.

Although the experimental results are very promising, this contribution is still a work in progress that is not yet published. As such, it suffers from several limitations:

(i) **Design**: more features must be implemented in the homemade simulator, such as the fading or the beamforming (which alterate the propagation model), to increase the realism of the simulator and the accuracy of the presented results.

(ii) **Evaluation**: cellular networks are almost exclusively used in a dynamic context, where end-users are mobile. The performance of the proposed combination (NOMA + FFR) and the optimizing algorithm (INSPIRE) need to evaluated in this context.

Since the solution proposed in this chapter relies on INSPIRE, note that the limitations raised in Chapter 4 (see Section 4.6) also apply to this work. In the next chapter, we propose to address these limitations by designing a high-dimensional Bayesian optimization algorithm that provides more theoretical guarantees.

# 6

# CONTRIBUTION TO THE OPTIMAL BAYESIAN OPTIMIZATION OF HIGH-DIMENSIONAL FUNCTIONS

**Outline.** In this chapter, we directly address the Bayesian optimization of an unknown high-dimensional function $f$ that is noisy and costly to evaluate. This is still an open research problem, as described briefly in Section 2.2.4 and discussed in greater length in this chapter. We address the over-exploration issue in decentralized Bayesian Optimization (BO) algorithms by proposing a provably better approximation of the famous acquisition function GP-UCB [57] in a decentralized context. We also describe a decentralized algorithm, called DuMBO, to maximize this new acquisition function. We tackle the large dimensionality of the input space $\mathcal{C}$ by assuming an additive decomposition of the objective function $f$, and we prove that DuMBO can optimize it with no-regret guarantees. Finally, we demonstrate its competitive empirical performance against other state-of-the-art high-dimensional BO algorithms on a benchmark comprising synthetic functions and real-world problems.

## 6.1 MOTIVATIONS

In the previous chapters, we addressed the spatial reuse problems in a Wireless Local Area Network (WLAN) (Chapters 3 and 4) and we demonstrated that one of our algorithms, devised for this purpose, can successfully apply to different technological contexts (Chapter 5). However, theoretical limitations remain as more work needs to be put in the theoretical modeling of our strategies to tackle the high-dimensionality of the objective function $f$. This chapter provides DuMBO, an algorithm that addresses the theoretical limitations of the previously proposed solutions. Through a thorough theoretical modeling, we assess its optimality and its superiority over state-of-the-art, provably optimal algorithms. In doing so, we aim at giving a broader impact to our contributions. As a matter of fact, this chapter demonstrates the ability of DuMBO to optimize objective functions from a variety of technological applications, ranging from telecommunication networks to astrophysics.

Furthermore, recall that one of the current major drawbacks of BO is its inability to optimize high-dimensional functions. In fact, although BO has become a highly effective framework for black-box optimization in input spaces of low effective dimension [116], classical BO algorithms struggle with high-dimensional functions as their acquisition

*[116] Wang et al. (2013), "Bayesian Optimization in High Dimensions via Random Embeddings."*

functions become hard to maximize. However, many real-world applications, such as computer vision, robotics and of course computer networks[48] often involve a high-dimensional objective function. This chapter aims to contribute to the field of BO by providing a way to extend it to the optimization of high-dimensional functions.

[48] *As illustrated throughout this thesis.*

## 6.2 RELATED WORKS

In this section, we extend Section 2.2.4 by describing how state-of-the-art BO algorithms address the high dimensionality of input spaces. Before discussing this, recall that the quality of the optimization is quantified through the notion of regret, with the instantaneous regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)$ (see Chapter 1, Definition 1.1) and the cumulative regret $R_T = \sum_{t=1}^{T} r_t$ (see Chapter 1, Definition 1.2). Also, remember that a Black-Box Optimization (BBO) algorithm is said to be asymptotically optimal (or equivalently, no-regret) if $\lim_{T \to +\infty} R_T / T = 0$ (see Chapter 1, Definition 1.3).

### 6.2.1 *High-Dimensional Bayesian Optimization*

When facing high-dimensional problems, global optimization algorithms, such as DIRECT (see Section 2.2.1), struggle at maximizing the acquisition function $\varphi_t$ of a BO algorithm in an acceptable amount of time. This is mainly due to their computational complexity, which is exponential in the number of dimensions $d$ of the input space $\mathcal{C}$. To tackle this issue, BO algorithms use a variety of techniques that leads to mitigating the complexity of the optimization of the acquisition function.

As an example, TuRBO [117] maintains a trust region in the input space $\mathcal{C}$ to perform local Bayesian optimization. Although its Gaussian Process (GP) has a poor fit globally, it serves as an excellent surrogate model within the trust region. Naturally, all the subtlety of this algorithm lies in how the trust region is built and updated at each iteration.

[117] *Eriksson et al. (2019), "Scalable global optimization via local bayesian optimization"*

Apart from TuRBO, high-dimensional BO algorithms generally fall into one of the two following general categories.

#### 6.2.1.1 *Embedding BO Algorithms*

Embedding BO algorithms assume that only a few dimensions are significantly impacting $f$ and project its high-dimensional input space into a low-dimensional one where the optimization is actually performed. REMBO [118] and ALEBO [119] make the hypothesis that the effective dimensionality of the objective function is lower than $d$. They optimize $f$ after projecting its high-dimensional input into a linear subspace using random Gaussian matrices. Other approaches ex-

[118] *Binois et al. (2015), "A warped kernel improving robustness in Bayesian optimization via random embeddings"*

[119] *Letham et al. (2020), "Re-examining linear embeddings for high-dimensional Bayesian optimization"*

ist, such as [120] that uses variational auto-encoders to learn a continuous embedding of molecules or [121] that discovers the embedding function using a manifold GP. [122] proposes the SAASBO algorithm that assumes a hierarchy of relevance in the input dimensions and introduces a structured prior over the kernel hyperparameters of the GP. These hyperparameters are designed in such a way that they induce a sparse GP, able to learn an embedding of the high-dimensional inputs and to handle large amount of data. Finally, there exist algorithms that propose to simply select some dimensions of the input space to project onto. Such recent methods include Dropout [123], inspired by the dropout algorithm in neural networks, that proposes the dimension dropout for the high-dimensional BO. At each optimization, it samples a small number of dimensions and performs the optimization on these dimensions only. Another approach is MCTS-VS [124], based on Monte-Carlo Tree Search to sample the subset of dimensions to project onto. Note that these methods demonstrate excellent empirical performance but offer no theoretical guarantees. In particular, they do not offer no-regret guarantees through an asymptotic optimality property.

[120] *Gómez-Bombarelli et al. (2018), "Automatic chemical design using a data-driven continuous representation of molecules"*

[121] *Moriconi et al. (2020), "High-dimensional Bayesian optimization using low-dimensional feature spaces"*

[122] *Eriksson et al. (2021), "High-dimensional Bayesian optimization with sparse axis-aligned subspaces"*

[123] *Li et al. (2018), "High dimensional Bayesian optimization using dropout"*

[124] *Song et al. (2022), "Monte Carlo Tree Search based Variable Selection for High Dimensional Bayesian Optimization"*

#### 6.2.1.2 *Decomposing BO Algorithms*

Decomposing BO algorithms assume an additive structure for $f$ and optimize the factors of the induced decomposition with classical BO. This assumption leads to exponential gains in the complexity, as the critical parameter is now $\bar{d}$, the Maximum Factor Size (MFS) of the decomposition.[49] Classical approaches such as ADD-GPUCB [125], MES [126] or QFF [127] assume a decomposition with a MFS equal to 1 and orthogonal domains. More recent approaches like DEC-HBO [128] are able to optimize decompositions with larger MFS and shared input components. Still, the MFS of the decomposition must be low to avoid a prohibitive computational complexity.

Note that, like any BO algorithm, decomposing BO algorithms approximate the true objective function $f$ as the existence of its additive decomposition is not guaranteed. However, unlike the others approaches, the vast majority of decomposing BO algorithms offer no-regret guarantees, as they are provably asymptotically optimal provided that their introduced assumptions on $f$ hold. Also note that some of them, such as ADD-GPUCB [125] and DEC-HBO [128], can be used in a decentralized context.

[49] *That is, the dimensionality of the highest-dimensional factor of the decomposition.*

[125] *Kandasamy et al. (2015), "High dimensional Bayesian optimisation and bandits via additive models"*

[126] *Wang et al. (2017), "Max-value entropy search for efficient Bayesian optimization"*

[127] *Mutny et al. (2018), "Efficient high dimensional bayesian optimization with additivity and quadrature fourier features"*

[128] *Hoang et al. (2018), "Decentralized high-dimensional Bayesian optimization with factor graphs"*

### 6.2.2 *The DuMBO (Decentralized Message-passing Bayesian Optimization) Algorithm*

In this chapter, we propose DuMBO, a decomposing BO algorithm. To better appreciate how DuMBO differentiates from other decomposing

BO algorithms, let us discuss about one particular challenge within the decomposing BO community.

The decomposing algorithms [125–128] are all based on the assumption that the inferred additive decomposition of $f$ has a low MFS. As an example, ADD-GPUCB [125] and QFF [127] require the simplest form of additive decompositions ($\bar{d} = 1$). As stated before, this may lead to the optimization of a coarse approximation of $f$. In return, they are systematically able, at each time step $t$, to query $\arg\max \varphi_t$ in a reasonable amount of time thanks to a global optimization algorithm. From this, a methodological question arises: is it better

(i) to systematically query $\boldsymbol{x}_t = \arg\max \varphi_t$, with $\varphi_t$ an acquisition function built from a simple (and often inexact) decomposition of $f$, or

(ii) to find the query $\boldsymbol{x}_t$ using an acquisition function $\varphi_t$ built from an exact, more complex decomposition of $f$ by giving up on the guarantee of finding the global maximum of $\varphi_t$?

Case (i) has been extensively studied [125–127], but it seems that only [128] has taken a few steps in the direction of case (ii). In fact, DEC-HBO tolerates more complex additive decompositions ($\bar{d} = 3$), but is no longer guaranteed to query $\arg\max \varphi_t$ (because it uses a variant of the max-sum algorithm [129] that requires $f$ to have a sparse additive decomposition to converge). Table 6.1 gathers the main difference between the state-of-the-art of decomposing algorithms, and the proposed DuMBO algorithm. Observe that, overall, DuMBO is the only algorithm that exploits weaker guarantees on the maximization of $\varphi_t$ to lower its computational complexity. This allows DuMBO to relax the restrictive MFS assumption and hence, to handle decompositions with an arbitrary MFS without the need to approximate them with a simpler one.

*[129] Rogers et al. (2011), "Bounded approximate decentralised coordination via the max-sum algorithm"*

Table 6.1: Comparison of decomposing state-of-the-art BO algorithms with DuMBO on relevant criteria. Here, $n$ is the number of factors in the decomposition, $d$ the number of dimensions of $f$, $\bar{d}$ the MFS of the decomposition, $t$ the optimization step, $\xi$ the desired accuracy when maximizing $\varphi_t$ and $N_A$ a constant defined in Section 6.4.3. $N_m$ is a constant defined in [128].

| Solution | Complexity | MFS Asm. | Find $\arg\max \varphi_t$ |
|---|---|---|---|
| ADD-GPUCB [125] | $\mathcal{O}\left(t^3 + nt^2 + n^2\xi^{-1}\right)$ | $\bar{d} = 1$ | Yes |
| QFF [127] | $\mathcal{O}\left((\xi^{-1}t^{3/2}(\log t)^{\bar{d}})^{\bar{d}}\right)$ | $\bar{d} = 1$ | Yes |
| DEC-HBO [128] | $\mathcal{O}\left(N_m\xi^{-\bar{d}}n(t^3 + n)\right)$ | $\bar{d} \leq 3$ | If the dec. is sparse |
| DuMBO (Ours) | $\mathcal{O}\left(\bar{d}N_A n t^3 \xi^{-1}\right)$ | None | See Section 6.4.2 |

## 6.3 PROBLEM FORMULATION

We now introduce the core assumptions about the black-box objective function $f : \mathcal{C} \to \mathbb{R}$ to obtain an additive decomposition (Section 6.3.1), along with an introductory example to ease exposition. Next, we exploit these assumptions to derive inference formulas (Section 6.3.2) and to adapt GP-UCB to a decentralized context (Section 6.3.3).

### 6.3.1 *Core Assumptions*

In order to optimize $f$ in a decentralized fashion, we make several assumptions.

**Assumption 6.1.** *The unknown objective function $f$ can be decomposed into a sum of factor functions $\left(f^{(i)}\right)_{i\in[1,n]}$, with domains $\left(\mathcal{C}^{(i)}\right)_{i\in[1,n]}$, such that $\mathcal{C} = \cup_{i=1}^{n}\mathcal{C}^{(i)}$ and*

$$f = \sum_{i=1}^{n} f^{(i)}. \tag{6.1}$$

Any decomposition, including (6.1), can be represented by a factor graph where each factor and variable node denote, respectively, one of the $n$ factors of the decomposition and one of the $d$ input components of $f$. There is an edge between a factor node $i$ and a variable node $j$ if and only if $f^{(i)}$ uses $x_j$ as an input component. We use $\mathcal{V}_i, 1 \leq i \leq n$, and $\mathcal{F}_j, 1 \leq j \leq d$, to denote respectively the set of variable nodes connected to factor node $i$ and the set of factor nodes connected to variable node $j$.

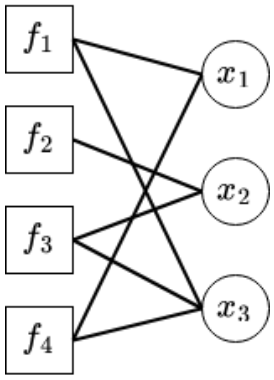



Figure 6.1: Factor graph of an additive decomposition, with $n = 4$ factors and $d = 3$ variables.

**Example.** An introductory example is shown in Figure 6.1, where $\mathcal{V}$-sets and $\mathcal{F}$-sets can be easily inferred. For instance, $\mathcal{V}_1 = \{1, 3\}$ since $f^{(1)}$ uses $x_1$ and $x_3$ whereas $\mathcal{F}_1 = \{1, 4\}$ since $x_1$ is used by $f^{(1)}$

and $f^{(4)}$. As a sanity check, you can verify that this decomposition has an MFS $\bar{d} = 2$.

To make predictions about the factor functions without any prior knowledge, we need a model that maps the previously collected inputs with their noisy outputs. Denoting $\boldsymbol{x}_{\mathcal{V}_i} = (x_j)_{j \in \mathcal{V}_i}$, let us introduce the following assumption.

**Assumption 6.2.** *Factor functions $f^{(i)}$ are independent $\mathcal{GP}\left(\mu_0^{(i)}, k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)\right)$, with prior mean $\mu_0^{(i)} = 0$ and covariance function $k^{(i)}$.*

Since $f$ is a sum of independent GPs, Assumption 6.2 implies that $f$ is also $\mathcal{GP}\left(\mu_0, k(\boldsymbol{x}, \boldsymbol{x}')\right)$ with prior mean $\mu_0 = 0$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)$.

### 6.3.2 *Inference Formulas*

For any $\boldsymbol{x} \in \mathcal{C}$ and given the previous $t$ input queries $\boldsymbol{X} = (\boldsymbol{x}^1, \cdots, \boldsymbol{x}^t)$, the vector $(f(\boldsymbol{x}), f(\boldsymbol{x}^1), \cdots, f(\boldsymbol{x}^t))$ is Gaussian. Given the $t$-dimensional vector of noisy outputs $\boldsymbol{y} = (y_1, \cdots, y_t)^\top$, with $y_i = f(\boldsymbol{x}^i) + \epsilon$ and $\epsilon$ a centered Gaussian variable, the posterior distribution of the factor $f^{(i)}(\boldsymbol{x})$ is also Gaussian. Since $f$ can be decomposed, the posterior mean $\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and variance $(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2$ of the factor $f^{(i)}$ at time $t+1$ can be expressed with the posterior means and covariance functions of the factor functions involved in decomposition (6.1).

**Proposition 6.1.** *Let $\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and $(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2$ be the posterior mean and variance of $f^{(i)}$ at input $\boldsymbol{x}_{\mathcal{V}_i}$. Then, for the decomposition (6.1),*

$$\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) = \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)\top} \boldsymbol{K}^{-1} \boldsymbol{y} \tag{6.2}$$

$$(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2 = k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}) - \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)\top} \boldsymbol{K}^{-1} \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)} \tag{6.3}$$

*with $t \times 1$ vectors $\boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)} = (k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}^j))_{j \in [1,t]}$ and $t \times t$ matrices $\boldsymbol{K} = (k(\boldsymbol{x}_{\mathcal{V}_i}^j, \boldsymbol{x}_{\mathcal{V}_i}^k))_{j,k \in [1,t]}$.*

For the sake of generality, Proposition 6.1 only requires an additive decomposition of $f$. Note that Proposition 6.1 does *not* assume that a corresponding additive decomposition of the observed outputs in $\boldsymbol{y}$ is available. However, note that, in a significant portion of real-world applications (e.g. network throughput maximization [88], energy consumption minimization [130] or UAVs-related applications [131]), a natural output decomposition is observable. As demonstrated by [87], having access to a decomposed output can only improve the predictive performance of the GP surrogate model. Therefore, we now derive the inference formulas when the output decomposition is known.

Observing the output decomposition of $f$ means that $f$ is now a function $\mathbb{R}^d \to \mathbb{R}^n$, with $n$ the number of factors in its additive decomposition. At a time $t$, a BO algorithm no longer has access to a

*[130] Bourdeau et al. (2019), "Modeling and forecasting building energy consumption: A review of data-driven techniques"*

*[131] Xie et al. (2018), "Throughput maximization for UAV-enabled wireless powered communication networks"*

$t$-dimensional output vector $\boldsymbol{y}$ but at a $t \times n$ matrix $\boldsymbol{Y}$. Note that $\boldsymbol{y}$ and $\boldsymbol{Y}$ are connected through the relation $\boldsymbol{Y}\mathbf{1} = \boldsymbol{y}$, with $\mathbf{1}$ the $n$-dimensional vector of ones.

Having access to the matrix $\boldsymbol{Y}$ allows to train $n$ different GPs instead of a single one with an additive kernel [132], so that the $i$th $\mathcal{GP}\left(0, k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)\right)$ serves as a surrogate model only for the $i$th factor of the decomposition of $f$. To condition the $i$th GP, we consider the data set $\mathcal{S}_i = \left\{\left(\boldsymbol{x}^j_{\mathcal{V}_i}, Y_{j,i}\right)\right\}_{j\in[1,t]}$. Given $\mathcal{S}_i$, the posterior mean $\mu^{(i)}_{t+1}$ and the posterior variance $\left(\sigma^{(i)}_{t+1}\right)^2$ formulas are simple instances of the conditioned Gaussian distribution formulas, with

*[132] Duvenaud et al. (2011), "Additive gaussian processes"*

$$\mu^{(i)}_{t+1}(\boldsymbol{x}) = \boldsymbol{k}^{(i)\top}_{\boldsymbol{x}_{\mathcal{V}_i}} \boldsymbol{K}^{-1}_{(i)} \boldsymbol{Y}_{:i}, \tag{6.4}$$

$$\left(\sigma^{(i)}_{t+1}(\boldsymbol{x})\right)^2 = k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}\right) - \boldsymbol{k}^{(i)\top}_{\boldsymbol{x}_{\mathcal{V}_i}} \boldsymbol{K}^{-1}_{(i)} \boldsymbol{k}^{(i)}_{\boldsymbol{x}_{\mathcal{V}_i}}, \tag{6.5}$$

with $\boldsymbol{Y}_{:i}$ the $i$th column of $\boldsymbol{Y}$, $t \times 1$ vectors $\boldsymbol{k}^{(i)}_{\boldsymbol{x}_{\mathcal{V}_i}} = \left(k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}^j_{\mathcal{V}_i})\right)_{j\in[1,t]}$ and $t \times t$ matrices $\boldsymbol{K}_{(i)} = \left(k^{(i)}(\boldsymbol{x}^j_{\mathcal{V}_i}, \boldsymbol{x}^k_{\mathcal{V}_i})\right)_{j,k\in[1,t]}$.

Note that (6.4) and (6.5) mainly differ from (6.2) and (6.3) by their ability to exploit the inverse of the Gram matrix $\boldsymbol{K}^{-1}_{(i)}$ built only from the $i$th covariance function $k^{(i)}$, and of course, the outputs of the $i$th factor of the decomposition $\boldsymbol{Y}_{:i}$. The impact of having access to the decomposed output of $f$ is explored in Section 6.5.

### 6.3.3 *Improved Acquisition Function*

Having defined a surrogate model for $f$, we can now turn to finding an optimal policy for querying the objective function. In this section, we exploit the decomposition of $f$ to build an acquisition function for our BO algorithm that approximates GP-UCB in a decentralized context. Proofs for all the presented results can be found in Appendix D.

Recall that GP-UCB is defined by (2.12) as the sum of an exploitation term $\mu_t(\boldsymbol{x})$ and an exploration term $\sigma_t(\boldsymbol{x})$ weighted by some scalar $\beta_t^{1/2}$. Finding an additive decomposition for GP-UCB is hard because $\sigma_t(\boldsymbol{x})$ cannot be expressed as a sum. To circumvent this caveat, [125] proposed to apply GP-UCB to each factor of the additive decomposition of $f$, with $\varphi^{(i)}_t = \mu^{(i)}_t + \beta_t^{1/2}\sigma^{(i)}_t$. Then, they proved that their algorithm ADD-GPUCB offers no-regret performance by considering $\sum_{i=1}^n \varphi^{(i)}_t = \mu_t + \beta_t^{1/2}\sum_{i=1}^n \sigma^{(i)}_t$ as an acquisition function. Although the exploitation term $\mu_t$ is preserved, the exploration term is now overweighted since $\sum_{i=1}^n \sigma^{(i)}_t \geq \sqrt{\sum_{i=1}^n \left(\sigma^{(i)}_t\right)^2} = \sigma_t$. To reach better empirical performance, one could look for a tighter additive upper bound of $\sigma^2_t$. This is the purpose of this section. We start by decomposing the variance of the $i$th factor function $(\sigma^{(i)}_t(\boldsymbol{x}))^2$ into two terms.

**Epistemic vs. aleatoric uncertainty.** The variance $(\sigma_t^{(i)}(\boldsymbol{x}))^2$ of the random variable $f^{(i)}(\boldsymbol{x})$ is composed of two fundamentally different terms. The *epistemic* uncertainty refers to the uncertainty caused by having an undersampled dataset, with not enough data points to accurately estimate the value of $f^{(i)}(\boldsymbol{x})$. In contrast, the *aleatoric* uncertainty, also called the observational noise, can intuitively be seen as the intrinsic noise of $f^{(i)}(\boldsymbol{x})$ due, for instance, to a poor measurement quality. For more details, refer to [133].

*[133] Hüllermeier et al. (2021), "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods"*

Let $v_-^{(i)} \geq 0$ be the aleatoric uncertainty of the factor function $f^{(i)}$, which can be seen as a lower bound of the posterior variance of $f^{(i)}$, that is $\forall \boldsymbol{x} \in \mathcal{C}^{(i)}, \forall t \in \mathbb{N}$, $v_-^{(i)} \leq (\sigma_t^{(i)}(\boldsymbol{x}))^2$. We argue that a better approximation of the exploration term can be proposed if the posterior variance of the GP is assumed to be bounded.

**Assumption 6.3.** *$\forall t \in \mathbb{N}$, the posterior variance of the objective function $f$, $\sigma_t^2(\boldsymbol{x}) = \sum_{i=1}^n \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2$ satisfies $\sigma_t^2(\boldsymbol{x}) \leq v_+$, with $v_+ = \left(\sqrt{v_-} + 2\delta_-\right)^2$, $v_- = \sum_{i=1}^n v_-^{(i)}$ and $\delta_-^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j\neq i}}^n \sqrt{v_-^{(i)} v_-^{(j)}}$.*

Note that the restrictiveness of Assumption 6.3 fades as the number $n$ of factors grows.

**Example.** Consider the case where all the $n$ factor functions $f^{(i)}$ have the same aleatoric uncertainty $v_0$. We have $v_- = nv_0$, $\delta_-^2 = n(n-1)v_0$ and $v_+ = nv_0(1+2\sqrt{n-1})^2$. Thus, the ratio $v_+/v_- = (1+2\sqrt{n-1})^2$ increases as $n$ grows, which suggests that Assumption 6.3 is more easily verified when $n$ is large. Note that $v_+/v_-$ can be quite large for reasonable values of $n$. Considering the decomposition in Figure 6.1 with $n = 4$, $v_+/v_- \approx 20$. In this particular context where we consider an additive decomposition of $f$ composed of 4 factors with the same aleatoric uncertainty $v_0$, Assumption 6.3 holds for any objective function whose posterior variance is less than *20 times* its aleatoric uncertainty.

Under Assumption 6.3, we propose to bound from above the exploration term with the following proposition.

**Proposition 6.2.** *Under Assumption 6.3,*

$$a \sum_{i=1}^{n} {\sigma_t^{(i)}}^2 + \frac{1}{4a} \tag{6.6}$$

*is the the tightest linear upper bound of the exploration term $\sigma_t$ in the least squares sense, where $a$ is the single positive real root of the quartic polynomial*

$$P(a) = \frac{2\left[u^3\right]_{v_-}^{v_+}}{3} a^4 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^3 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3} a - \frac{[u]_{v_-}^{v_+}}{8}, \tag{6.7}$$

*and $[h(u)]_{v_-}^{v_+} = h(v_+) - h(v_-)$.*

We show that (6.6) is a tighter upper bound of $\sigma_t(\boldsymbol{x})$ than the one proposed in [125, 128].

**Theorem 6.1.** *Let Assumptions 6.1, 6.2, 6.3 hold. Then the following inequality holds for all $\boldsymbol{x} \in \mathcal{C}$*

$$a \sum_{i=1}^{n} \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2 + \frac{1}{4a} \leq \sum_{i=1}^{n} \sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}). \tag{6.8}$$

Therefore, we propose an acquisition function $\varphi_t = \sum_{i=1}^{n} \varphi_t^{(i)}$ corresponding to the described additive decomposition (6.1) with

$$\varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) = \mu_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) + a\beta_t^{\frac{1}{2}} \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2 \tag{6.9}$$

## 6.4 DUMBO

In this section, we describe DuMBO, a BO algorithm that exploits the results from Section 6.3 to find $\arg\max_{\boldsymbol{x}\in\mathcal{C}} \sum_{i=1}^{n} \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$, which is equal to $\arg\max_{\boldsymbol{x}\in\mathcal{C}} \mu_t(\boldsymbol{x}) + \beta_t^{1/2}\left(a\sigma_t^2(\boldsymbol{x}) + 1/4a\right)$ since both expressions differ only by a constant term. We start by explaining how DuMBO infers an additive decomposition of the objective function $f$ from observed data (Section 6.4.1). Then, we formulate how we maximize $\varphi_t$ in a decentralized fashion (Section 6.4.2). After that, we describe the DuMBO algorithm and we discuss its computational complexity (Section 6.4.3). Next, we provide an early-stopped version of DuMBO and discuss the weaker guarantees achieved in this case (Section 6.4.4). Finally, we prove the asymptotic optimality of DuMBO (Section 6.4.5).

### 6.4.1 *Additive Decomposition Inference*

Assumption 6.1 requires an additive decomposition of the objective function $f$. However, if the decomposition is not provided by the user, it must be inferred from observed data. In this section, we describe how such a decomposition can be inferred from data [134, 135]. We adopt the method proposed by [134], as in Appendix B of [128].

As in [128], let us associate each candidate additive decomposition $\mathcal{A}$ (represented by its factor graph, see Figure 6.1 for an illustration) with the kernel of an additive GP [132]. Given $k$ candidates $\mathcal{A}_1, \cdots, \mathcal{A}_k$, we reformulate the acquisition function $\varphi_t$ as a weighted average with respect to the posterior of each candidate given the dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[1,t]}$ composed of the selected input queries and their observed noisy outputs, that is

*[134] Gardner et al. (2017), "Discovering and exploiting additive structure for Bayesian optimization"*

*[135] Wang et al. (2017), "Batched high-dimensional Bayesian optimization via structural kernel learning"*

$$\varphi_t(\boldsymbol{x}) = \sum_{i=1}^{k} p(\mathcal{A}_i|\mathcal{D})\varphi_t^{\mathcal{A}_i}(\boldsymbol{x}) \tag{6.10}$$

$$= \sum_{i=1}^{k} p(\mathcal{A}_i|\mathcal{D}) \sum_{j=1}^{|\mathcal{A}_i|} \varphi_t^{(j)}(\boldsymbol{x}_{\mathcal{V}_j}) \tag{6.11}$$

$$\approx \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{|\mathcal{A}_i|} \varphi_t^{(j)}(\boldsymbol{x}_{\mathcal{V}_j}), \tag{6.12}$$

with $\varphi_t^{\mathcal{A}_i}$ our proposed acquisition function given the additive decomposition $\mathcal{A}_i$, $\varphi_t^{(j)}$ given by (6.9), (6.11) following from (6.10) since the additive decomposition $\mathcal{A}_i$ also provides an additive decomposition of our proposed acquisition function, and (6.12) an approximation of (6.11) as proposed by [134].

*[136] Robert et al. (2010), "Metropolis–hastings algorithms"*

As for the candidates $\mathcal{A}_1, \cdots, \mathcal{A}_k$, they are sampled by Monte-Carlo Markov Chain (MCMC) with the Metropolis-Hastings algorithm [136]. Starting from the fully dependent decomposition $\mathcal{A}_0 = \{\{1, \cdots, d\}\}$ at $t = 0$. When the decomposition is unknown, at each time step $t$, $k$ promising decompositions are sampled by MCMC starting from the last sampled decomposition at time step $t-1$. We briefly detail below how a new additive decomposition $\mathcal{A}_{i+1}$ is sampled from an additive decomposition $\mathcal{A}_i$. A thorough description of this process can be found in [134].

To generate $\mathcal{A}_{i+1}$, we sample an alternative additive decomposition $\mathcal{A}'$ from a proposal distribution $g(\mathcal{A}'|\mathcal{A}_i)$ (see Figure 1 in [134]). To do so, one of two operations on $\mathcal{A}_i$ is chosen randomly. The `Merge` operation chooses randomly two factors in the decomposition $\mathcal{A}_i$ and merge them together, while the `Split` operation chooses a factor and split it in two. Either `Merge` or `Split` is applied to generate another configuration $\mathcal{A}'$. Then, the model evidence $p(\boldsymbol{y}|\boldsymbol{X}, \mathcal{A}')$ is computed. Finally, we consider the acceptance probability $\mathcal{P}(\mathcal{A}'|\mathcal{A}_i)$ of the Metropolis-Hastings algorithm, that is

$$\mathcal{P}(\mathcal{A}'|\mathcal{A}_i) = \min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{X}, \mathcal{A}')g(\mathcal{A}'|\mathcal{A}_i)}{p(\boldsymbol{y}|\boldsymbol{X}, \mathcal{A}_i)g(\mathcal{A}_i|\mathcal{A}')}\right). \tag{6.13}$$

Then, $\mathcal{A}_{i+1} = \mathcal{A}'$ with probability $\mathcal{P}(\mathcal{A}'|\mathcal{A}_i)$ and $\mathcal{A}_{i+1} = \mathcal{A}_i$ with probability $1 - \mathcal{P}(\mathcal{A}'|\mathcal{A}_i)$.

Note that the computational overhead of sampling $k$ alternative additive decomposition of $f$ is in $\mathcal{O}\left(kt^3\right)$ with a dataset $\mathcal{D}$ of $t$ data points, since each sampled additive decomposition requires the Maximum Likelihood Estimation (MLE) of the likelihood $p(\boldsymbol{y}|\boldsymbol{X}, \mathcal{A})$, which involves inverting a $t \times t$ matrix (see [53]). Once $\mathcal{A}_1, \cdots, \mathcal{A}_k$ are sampled, (6.12) is maximized by our decentralized algorithm to find the input $\boldsymbol{x}$ to query.

### 6.4.2 *ADMM Formulation*

Optimizing $\varphi_t(\boldsymbol{x}) = \sum_{i=1}^{n} \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ while ensuring the compatibility between shared input components amounts to solving the following constrained optimization problem:

$$\begin{aligned} &\underset{\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}}{\text{maximize}} \sum_{i=1}^{n} \varphi_t^{(i)}\left(\boldsymbol{x}^{(i)}\right) \\ &\text{such that } \boldsymbol{x}^{(i)}_{\mathcal{V}_i \cap \mathcal{V}_j} = \boldsymbol{x}^{(j)}_{\mathcal{V}_i \cap \mathcal{V}_j}, \forall i, j \in [1, n] \end{aligned} \tag{6.14}$$

with $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}$ inputs with dimension indices respectively listed in $\mathcal{V}_1, \cdots, \mathcal{V}_n$ of the factor functions $\varphi_t^{(1)}, \cdots, \varphi_t^{(n)}$.

To simplify the constraints (6.14), we introduce a global consensus variable $\bar{\boldsymbol{x}} \in \mathcal{C}$ and we reformulate the optimization problem as

$$\begin{aligned} &\underset{\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}}{\text{maximize}} \sum_{i=1}^{n} \varphi_t^{(i)}\left(\boldsymbol{x}^{(i)}\right) \\ &\text{such that } \boldsymbol{x}^{(i)} = \bar{\boldsymbol{x}}_{\mathcal{V}_i}, \forall i \in [1, n]\,. \end{aligned} \tag{6.15}$$

We now turn the problem (6.15) into an unconstrained optimization problem by considering its augmented Lagrangian $\mathcal{L}_\eta(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}, \bar{\boldsymbol{x}}, \boldsymbol{\lambda})$:

$$\mathcal{L}_\eta = \sum_{i=1}^{n} \varphi_t^{(i)}(\boldsymbol{x}^{(i)}) - \boldsymbol{\lambda}^{(i)\top}(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}) - \frac{\eta}{2}||\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}||_2^2 \tag{6.16}$$

with $\boldsymbol{\lambda}_k^{(i)}$ a column vector of dual variables with $|\mathcal{V}_i|$ components and a hyperparameter $\eta > 0$ that controls the influence of the second-order penalty $||\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}||_2^2$ on the augmented Lagrangian. To properly choose a value for $\eta$, we refer the interested reader to [137].

To maximize (6.16), we consider the Alternating Directions Method of Multipliers (ADMM), proposed by [138]. We now describe how we apply ADMM to our problem and present some relevant well-known results. For further details, please refer to [137].

ADMM is an iterative method that proposes, at iteration $k$, to solve sequentially the problems

$$\begin{aligned} \boldsymbol{x}_{k+1}^{(1)} &= \arg\max_{\boldsymbol{x}^{(1)}} \mathcal{L}_\eta(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}_k^{(n)}, \bar{\boldsymbol{x}}_k, \boldsymbol{\lambda}_k) \\ &\vdots \\ \boldsymbol{x}_{k+1}^{(n)} &= \arg\max_{\boldsymbol{x}^{(n)}} \mathcal{L}_\eta(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n-1)}, \boldsymbol{x}^{(n)}, \bar{\boldsymbol{x}}_k, \boldsymbol{\lambda}_k) \\ \bar{\boldsymbol{x}}_{k+1} &= \arg\max_{\bar{\boldsymbol{x}}} \mathcal{L}_\eta(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}, \bar{\boldsymbol{x}}, \boldsymbol{\lambda}_k) \qquad (6.17) \\ \boldsymbol{\lambda}_{k+1} &= \arg\max_{\boldsymbol{\lambda}} \mathcal{L}_\eta(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}, \bar{\boldsymbol{x}}_{k+1}, \boldsymbol{\lambda}). \qquad (6.18) \end{aligned}$$

*[137] Boyd et al. (2011), "Distributed optimization and statistical learning via the alternating direction method of multipliers"*

*[138] Gabay et al. (1976), "A dual algorithm for the solution of nonlinear variational problems via finite element approximation"*

Note that $\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}$ can be found concurrently, by gradient ascent (e.g. with ADAM [139]) of

[139] Kingma et al. (2014), "Adam: A method for stochastic optimization"

$$\mathcal{L}_{\eta}^{(i)} = \varphi_t^{(i)}(\boldsymbol{x}^{(i)}) - \boldsymbol{\lambda}^{(i)\top}(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}) - \frac{\eta}{2}||\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}||_2^2. \tag{6.19}$$

If the choice of $\boldsymbol{\lambda}_0$ verifies $\forall i \in [1,n], \sum_{j\in\mathcal{F}_i} \lambda_{0,i}^{(j)} = 0$, it is known (see [137]) that the closed-forms for (6.17) and (6.18) are

$$\bar{\boldsymbol{x}}_{k+1} = \left( \frac{1}{|\mathcal{F}_i|} \sum_{j\in\mathcal{F}_i} x_{k+1,i}^{(j)} \right)_{i\in[1,d]} \tag{6.20}$$

$$\boldsymbol{\lambda}_{k+1} = \left( \boldsymbol{\lambda}_k^{(i)} + \eta \left( \boldsymbol{x}_{k+1}^{(i)} - \bar{\boldsymbol{x}}_{k+1,\mathcal{V}_i} \right) \right)_{i\in[1,n]}. \tag{6.21}$$

Since the maximization of $\varphi_t = \sum_{i=1}^n \varphi_t^{(i)}$ relies on ADMM, let us discuss its maximization guarantees. It is well known that ADMM converges towards the global maximum of a convex $\varphi_t$. Furthermore, ADMM has also demonstrated very good performance at optimizing non-convex functions. As examples, [140, 141] exploited ADMM to perform parallel constrained tensor factorization and as a splitting method for orthogonality constrained problems, respectively. It has also been used as a way to compute sparse representations in [142]. This success in solving nonconvex optimization tasks has been explained by recent works such as [143], which extended the maximization guarantees of ADMM to the class of *restricted prox-regular* functions.

[140] Liavas et al. (2015), "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers"

[141] Lai et al. (2014), "A splitting method for orthogonality constrained problems"

[142] Chartrand et al. (2013), "A nonconvex ADMM algorithm for group sparsity with sparse groups"

[143] Wang et al. (2019), "Global convergence of ADMM in nonconvex nonsmooth optimization"

**Definition 6.1** (restricted prox-regularity, see [143])**.** *For a lower semi-continuous function $f$, let $M \in \mathbb{R}^+$, $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $\partial f$ the set of general subgradients of $f$. Define the exclusion set*

$$S_M = \{\boldsymbol{x} \in dom(f) : ||\boldsymbol{d}|| > M \text{ for all } \boldsymbol{d} \in \partial f(\boldsymbol{x})\}.$$

*$f$ is called restricted prox-regular if, for any $M > 0$ and bounded set $T \subseteq dom(f)$, there exists $\gamma > 0$ such that*

$$f(\boldsymbol{y}) + \frac{\gamma}{2}||\boldsymbol{x} - \boldsymbol{y}||^2 \geq f(\boldsymbol{x}) + \boldsymbol{d}(\boldsymbol{y} - \boldsymbol{x}), \tag{6.22}$$

$\forall \boldsymbol{x} \in T \setminus S_M$, $\boldsymbol{y} \in T$, $\boldsymbol{d} \in \partial f(\boldsymbol{x})$, $||\boldsymbol{d}|| \leq M$.

Note that the class of restricted prox-regular functions includes some non-convex, non-smooth functions.

### 6.4.3 *Algorithm and Complexity*

The formulation in the previous section is the foundation of a fully decentralized message-passing algorithm, called DuMBO, that can

**Algorithm 6.1** DuMBO

**Input**: factor graph of the acquisition function $\varphi_t$, $\eta > 0$.

1: $t = 0$
2: **while true do**
3: $\boldsymbol{\lambda}_0 = \mathbf{0}$
4: $k = 0$
5: **while** ADMM stopping criterion not met **do**
6: **for all** factor node $i$ [concurrently] **do**
7: **if** $k = 0$ **then**
8: Compute $\boldsymbol{x}_{k+1}^{(i)}$ by maximizing $\varphi_t^{(i)}$ with gradient ascent
9: **else**
10: Update $\boldsymbol{\lambda}_k^{(i)}$ with (6.21)
11: Compute $\boldsymbol{x}_{k+1}^{(i)}$ by maximizing (6.19) with gradient ascent starting from $\boldsymbol{x}_k^{(i)}$
12: **end if**
13: Send $x_{k+1,j}^{(i)}$ to the variable node $j$, $\forall j \in \mathcal{V}_i$
14: **end for**
15: **for all** variable node $j$ [concurrently] **do**
16: Compute $\bar{x}_{k+1,j}$ with (6.20)
17: Send $\bar{x}_{k+1,j}$ to the factor nodes in $\mathcal{F}_j$
18: **end for**
19: $k = k + 1$
20: **end while**
21: Observe $y_{t+1} = f(\bar{\boldsymbol{x}}_k)$
22: Update the surrogate model $\mathcal{GP}$ with $(\bar{\boldsymbol{x}}_k, y_{t+1})$
23: $t = t + 1$
24: **end while**

run on the factor graph of $f$. It is described in details in this section, followed by a short discussion about its computational complexity.

Algorithm 6.1 describes the DuMBO algorithm. At each step of the external optimization loop (lines 2-24), ADMM maximizes the acquisition function $\varphi_t$ with the internal optimization loop (lines 5-20). Each factor node and each associated variable node communicate back and forth at lines 13 and 17, until some stopping criterion for the convergence of ADMM is satisfied. Note that DuMBO is decentralized, since loops at lines 6 and 15 are run concurrently.

We now provide a time complexity analysis for DuMBO. The analysis assumes that a gradient ascent performs $\mathcal{O}\left(\xi^{-1}\right)$ steps for a desired accuracy $\xi$ [85] and ADMM converges in at most $N_A$ steps. We also denote $d^{(i)}$ the factor size of the $i$th factor in the decomposition, used by the local acquisition function $\varphi_t^{(i)}$. Note that, within the factor graph of $\varphi_t$, $n$ factor nodes and $d$ variable nodes are working concurrently to run ADMM in a decentralized fashion. We discuss the time complexities for the two types of node (note that the costs of the com-

munication between factor and variable nodes have been neglected for the clarity of the analysis).

**Factor node.** For a factor node $i$, it is known that, at iteration $t$, the time complexity of the inference with a GP is $\mathcal{O}\left(t^3 d^{(i)}\right)$, since the number of previous observations is $t$. Thus, the time complexity of evaluating (6.19) is $\mathcal{O}\left(t^3 d^{(i)}\right)$. Since it is required $\mathcal{O}\left(\xi^{-1}\right)$ times by the gradient ascent, the time complexity of finding $x_{k+1}^{(i)}$ is $\mathcal{O}\left(\xi^{-1} t^3 d_m^{(i)}\right)$. A factor node also needs to compute $\boldsymbol{\lambda}_{k+1}^{(i)}$, which is $\mathcal{O}\left(d^{(i)}\right)$. Since the factor node is called at least once and at most $N_A$ times for ADMM to converge, its time complexity is $\mathcal{O}\left(d^{(i)} \xi^{-1} t^3 N_A\right)$.

**Variable node.** A variable node $j$ is simply in charge of collecting messages from $|\mathcal{F}_j|$ factor nodes, and to aggregate them into $\bar{x}_{k+1,j}$ by averaging. Its time complexity is therefore $\mathcal{O}\left(|\mathcal{F}_j|\right)$.

### 6.4.4 *Early-stopped Version*

Although DuMBO has a competitive time complexity (see Table 6.1), we now discuss how to properly early-stop it and still get (weaker) guarantees on the maximization of $\varphi_t$ *before* ADMM converges. This can be of critical importance for some real-world applications. Note that the proofs for the results in this section can be found in Appendix E. We start with the following assumption.

**Assumption 6.4.** *For all $i \in [1, n]$, the covariance function $k^{(i)}$ from Assumption 6.2 is Lipschitz continuous, with Lipschitz constant $L\left(k^{(i)}\right)$.*

Assumption 6.4 holds for a large class of covariance functions, such as Matérn (see 4.3) or the squared exponential (equivalent to a Matérn kernel when $\nu \to +\infty$). For such covariance functions, we have the following result.

**Proposition 6.3.** *Let Assumptions 6.1, 6.2, 6.3 and 6.4 hold. Then, $\varphi_t^{(i)}$ is Lipschitz continuous, with Lipschitz constant*

$$L\left(\varphi_t^{(i)}\right) = tL\left(k^{(i)}\right) \rho\left(\boldsymbol{K}^{-1}\right) M_t^{(i)} \tag{6.23}$$

*with $M_t^{(i)} = \max\left(|y_t^+ - 2a\beta_t^{1/2} v_-^{(i)}|, |y_t^- - 2a\beta_t^{1/2} v_+^{(i)}|\right)$, $y_t^+ = \max_{k\in[1,t]} y_k$, $y_t^- = \min_{k\in[1,t]} y_k$ and $\rho\left(\boldsymbol{K}^{-1}\right)$ the spectral radius of $\boldsymbol{K}^{-1}$.*

Thanks to the Lipschitz continuity of the acquisition function, we have the following result.

**Theorem 6.2.** *$\forall i \in [1, n]$, let $\boldsymbol{x}_{k+1}^{(i)} = \arg\max_{\boldsymbol{x}} \mathcal{L}_\eta^{(i)}(\boldsymbol{x})$, with $\mathcal{L}_\eta^{(i)}$ defined in (6.19). Then*

$$\tilde{\boldsymbol{x}} = \left( \frac{1}{\sum_{i \in \mathcal{F}_j} L(\varphi_t^{(i)})} \sum_{i \in \mathcal{F}_j} L(\varphi_t^{(i)}) x_{k+1,i}^{(j)} \right)_{j \in [1,d]} \tag{6.24}$$

*is optimal in the minimax sense.*

Intuitively, Theorem 6.2 proves that a tight upper bound for $|\sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}^*) - \sum_{i=1}^n \varphi_t^{(i)}(\tilde{\boldsymbol{x}})|$ (with $\boldsymbol{x}^*$ the optimal solution) exists, and that (6.24) minimizes this upper bound. In addition, if not enough information or computing capacity is available to compute $L(\varphi_t^{(j)})$, (6.20) can be used instead of (6.24) to early-stop DuMBO (see Appendix E for a proof of this result).

Observe that Theorem 6.2 provides a way to aggregate the set of recommended values for the $j$th input (that is, $\left\{x_{k+1,i}^{(j)}\right\}_{i \in \mathcal{F}_j}$) without waiting for ADMM to converge. Although $\tilde{\boldsymbol{x}}$ is not guaranteed to be the maximizer of $\varphi_t(\boldsymbol{x})$, it is nonetheless a minimax optimal of $\varphi_t(\boldsymbol{x})$. Computing $\tilde{\boldsymbol{x}}$ after a few ADMM iterations instead of waiting for ADMM to converge could be a great trade-off between maximizing $\varphi_t(\boldsymbol{x})$ and the computational overhead brought by ADMM.

### 6.4.5 *Asymptotic Optimality*

In this section, we provide a regret bound for DuMBO and the theoretical results needed to assess its asymptotic optimality, assuming the convergence of ADMM.

**Assumption 6.5.** *The acquisition function $\varphi_t$ is restricted prox-regular (see Definition 6.1).*

We start by providing an upper bound on its immediate regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$, with $\boldsymbol{x}^t = \arg\max_{\boldsymbol{x} \in \mathcal{C}} \varphi_t(\boldsymbol{x})$ for a finite, discrete domain $\mathcal{C}$. Its proof can be found in Appendix F.

**Theorem 6.3.** *Let $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$ denote the immediate regret of DuMBO. Let $\delta \in (0, 1)$ and $\beta_t = 2 \log\left(\frac{|\mathcal{C}|\pi^2 t^2}{6\delta}\right)$. Then $\forall t \in \mathbb{N}$ we have*

$$r_t \leq 2\beta_t^{\frac{1}{2}} \left( a \sum_{i=1}^n \left(\sigma_t^{(i)}(\boldsymbol{x}^t)\right)^2 + \frac{1}{4a} \right) \tag{6.25}$$

*with probability at least $1 - \delta$.*

We demonstrate the asymptotic optimality of DuMBO by piggybacking on the asymptotic optimality of DEC-HBO [128]. This is a decomposing BO algorithm with an immediate regret bound of $2\beta_t^{1/2} \sum_{i=1}^n \sigma_t^{(i)}(\boldsymbol{x}^t)$ (see Theorem 1 in [128]). Interestingly, Theorem 6.1

directly implies that the immediate regret bound (6.25) is lower than the immediate regret bound of DEC-HBO. As a consequence, the immediate regret of DuMBO is bounded with the regret bound of DEC-HBO. This allows us to rely on proofs in [128] to demonstrate some properties of DuMBO. In particular, DEC-HBO is provably asymptotically optimal whether the domain $\mathcal{C}$ is discrete or continuous (see Theorems 2 and 3 in [128]). These results directly apply to DuMBO and imply the following corollary.

**Corollary 6.1.** *Let $\delta \in (0,1)$ and $R_t = \sum_{k=1}^{t} r_k$ denote the cumulative regret of DuMBO. Then there exists a monotonically increasing sequence of $\{\beta_t\}_t$ such that $\beta_t \in \mathcal{O}(\log t)$ and $\lim_{t\to+\infty} R_t/t = 0$ with probability at least $1-\delta$.*

Corollary 6.1 states that DuMBO is asymptotically optimal, as defined by Definition 1.3.

## 6.5 NUMERICAL RESULTS

In this section, we detail the experiments carried out to evaluate the empirical performance of DuMBO. Our benchmark comprises four synthetic functions (Section 6.5.1) and three real-world applications (Section 6.5.2). We consider two state-of-the-art decomposing BO algorithms: ADD-GPUCB [125] that assumes that $\bar{d} = 1$ and DEC-HBO [128] that assumes that $\bar{d} \leq 3$. We also consider two recent state-of-the-art BO algorithms that do not assume an additive decomposition of the objective function: TuRBO [117] and SAASBO [122]. We compare these solutions with two versions of the proposed algorithm: DuMBO that must systematically infer the additive decomposition of $f$ (see Section 6.4.1) and ADD-DuMBO that, conversely, can observe the decomposition of $f$ if it exists (see (6.4) and (6.5) in Section 6.3.2). Note that, in Section 6.5.3, we also evaluate ES-DuMBO and ES-ADD-DuMBO, the early-stopped versions (see Section 6.4.4) of DuMBO and ADD-DuMBO respectively. Finally, the wall-clock times of all the described BO algorithms are provided in Section 6.5.4.

Since BO is often used in the optimization of expensive black-box functions, we are interested in the ability to obtain good performance in a small number of iterations. Therefore, in every experiment, all the BO algorithms are given 110 iterations to optimize $f$. Each experiment is repeated 5 independent times. Table 6.2 gathers the averaged results that were obtained. Note that, for each experiment, a BO algorithm is considered as one of the best performing algorithms if its average performance lies within the confidence interval of the best performing solution.

Table 6.2: Comparison of four state-of-the-art solutions against four different versions of DuMBO on synthetic functions and real-world problems. The reported metrics correspond to the minimal regret attained for the synthetic functions, and to the average negative reward for the real-world problems. The best performance metric among all the strategies is written in **bold text**, and the best among the strategies that do not have access to the additive decomposition is underlined. Decomposing BO algorithms can be identified with the prefix "(+)".

| Algorithm | Synthetic Functions ($d$-$\bar{d}$) | | | | Real-World Problems ($d$-$\bar{d}$) | | |
|---|---|---|---|---|---|---|---|
| | SHC | Hartmann | Powell | Rastrigin | Cosmo | WLAN | Rover |
| *Unknown Add. Dec.* | (2-2) | (6-6) | (24-4) | (100-5) | (9-) | (12-6) | (60-) |
| ADD-GPUCB | 0.102 | 1.29 | 10,258 | N/A | 7.46 | -119.05 | 26.57 |
| DEC-HBO | **0.005** | 1.47 | 9,025 | N/A | 14.90 | -116.58 | 10.07 |
| TuRBO | 0.322 | 1.89 | 711 | 1,109 | **5.82** | -118.39 | **6.06** |
| SAASBO | 0.013 | 0.89 | 2,544 | 1,073 | 16.55 | -116.40 | 10.82 |
| (+) ES-DuMBO | **0.006** | 0.87 | 1,745 | 1,045 | 7.10 | -118.91 | 7.55 |
| (+) DuMBO | 0.029 | 0.76 | **542** | 1,010 | **5.86** | -118.57 | **6.38** |
| *Known Add. Dec.* | | | | | | | |
| (+) ES-ADD-DuMBO | 0.047 | 1.02 | 621 | 932 | N/A | -119.40 | N/A |
| (+) ADD-DuMBO | 0.102 | **0.72** | **542** | **822** | N/A | **-121.06** | N/A |

### 6.5.1 *Synthetic Functions*

In this section, we compare the BO algorithms mentioned above (excluding the early-stopped versions for the sake of clarity) using four synthetic functions: the 2d Six-Hump Camel (SHC), the 6d Hartmann, the 24d Powell and the 100d Rastrigin. These functions are described in details in Appendix G.

Figure 6.2 reports the performance of the solutions on the synthetic functions with their minimal regrets. Observe that, in the specific example of the SHC function (Figure 6.2a), DEC-HBO obtains the best performance. This is due to the relative simplicity of SHC. In fact, the SHC function satisfies all the assumptions made by DEC-HBO: a MFS lower than 3 and a sparse factor graph. In this case, the variant of the max-sum algorithm used by DEC-HBO is guaranteed to query $\arg\max \varphi_t$ at each time step $t$. Since DuMBO does not offer global maximization guarantees in that case, it is outperformed by DEC-HBO. Still, note that it exhibits competitive performance when compared to the other BO algorithms.

As for the Hartmann, Powell and Rastrigin functions (Figures 6.2b, 6.2c and 6.2d respectively) they exhibit similar dynamics. Observe that the two decomposing algorithms, ADD-GPUCB and DEC-HBO,

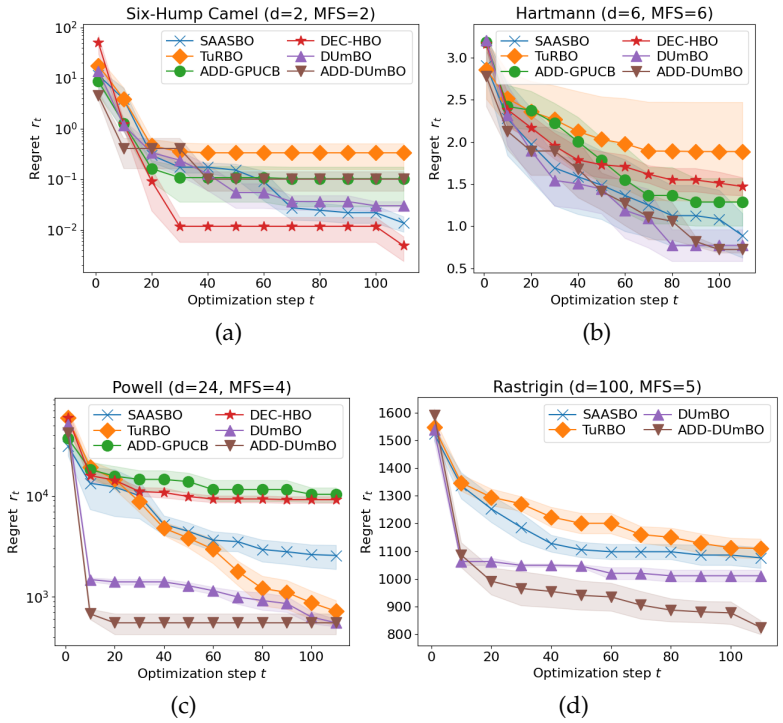


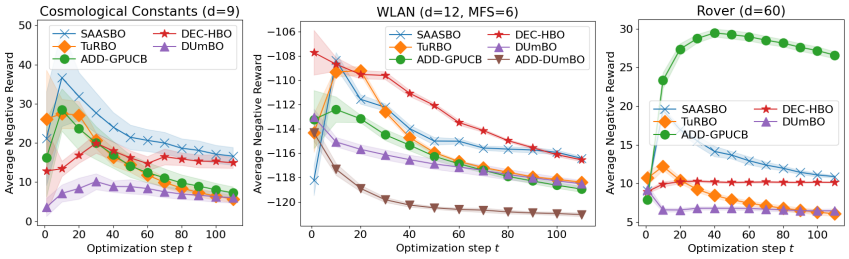

Figure 6.2: Graphs of regret achieved by the considered BO algorithms for (a) the SHC function, (b) the Hartmann function, (c) the Powell function and (d) the Rastrigin function. Note that the results in (a) and (c) are shown with a logarithmic scale. The shaded areas indicate the standard error intervals.

obtain poor minimal regrets (note that they were not evaluated on the Rastrigin function since they did not finish a single execution in 24 hours). This is because they infer an additive decomposition of $f$ based on the assumption that $\bar{d} \leq 3$ when actually $\bar{d} > 3$. Conversely, DuMBO, which does not make any restrictive assumption on $\bar{d}$, manages to quickly achieve a low regret by inferring an efficient additive decomposition of $f$. Observe that DuMBO also outperforms SAASBO and TuRBO on all three synthetic functions. Finally, the three Figures 6.2b, 6.2c and 6.2d show that, when given access to the true additive decomposition of $f$, ADD-DuMBO achieves its lowest regret in a lower number of iterations. Overall, note that, among all the BO algorithms tested in the experiments, the two versions of DuMBO are the only ones able to properly infer and/or exploit the additive decomposition of $f$ when it has a large MFS.

### 6.5.2 *Real-World Experiments*

We now evaluate DuMBO on three real-world problems, described in details in Appendix G: (a) fine-tuning some cosmological constants to maximize the likelihood of observed astronomical data, (b) con-

trolling the power of devices in a WLAN to maximize its Shannon capacity [110] and (c) planning the least costly trajectory to a provided destination for a rover on a rough terrain. Note that the problem (b) is similar to the problem addressed in Chapters 3 and 4, except that only the transmission power of each Access Point (AP) is tuned.




(a) (b) (c)

Figure 6.3: Negative average reward achieved by the considered BO algorithms for (a) the fine-tuning of cosmological constants, (b) the optimization of the Shannon capacity of a WLAN and (c) the planning of the rover trajectory. The shaded areas indicate the standard error intervals.

Figure 6.3 depicts the performance of the considered BO algorithms. Figure 6.3b shows that DuMBO obtains competitive performance against other state-of-the-art BO algorithms. On this problem as well, the performance of ADD-DuMBO demonstrate that having access and being able to handle additive decompositions with large MFS is a significant advantage. As a matter of fact, it allows to outperform other BO algorithms unable to exploit this additional information. Finally, Figures 6.3a and 6.3c exhibit patterns similar to those already identified in Figure 6.2: ADD-GPUCB and DEC-HBO fail to infer an adequate additive decomposition because of the restrictive MFS assumption. Conversely, DuMBO, which does not make such an assumption on the size of the MFS, demonstrates its competitiveness by achieving the best performance along with TuRBO. Note that ADD-DuMBO is not evaluated on problems (a) and (c) since the objective functions cannot be decomposed.

### 6.5.3 *Performance of the Early-Stopped Versions*

In this section, we evaluate the performance of ES-DuMBO and ES-ADD-DuMBO, the early-stopped versions of DuMBO and ADD-DuMBO, respectively. Recall that the early-stopping procedure and guarantees are described in Section 6.4.4. The solutions are early-stopped at the end of the very first ADMM iteration. For the sake of readability, we only depict the performance of the additive algorithms (ADD-GPUCB, DEC-HBO, DuMBO and ADD-DuMBO) to compare with the early-stopped versions.

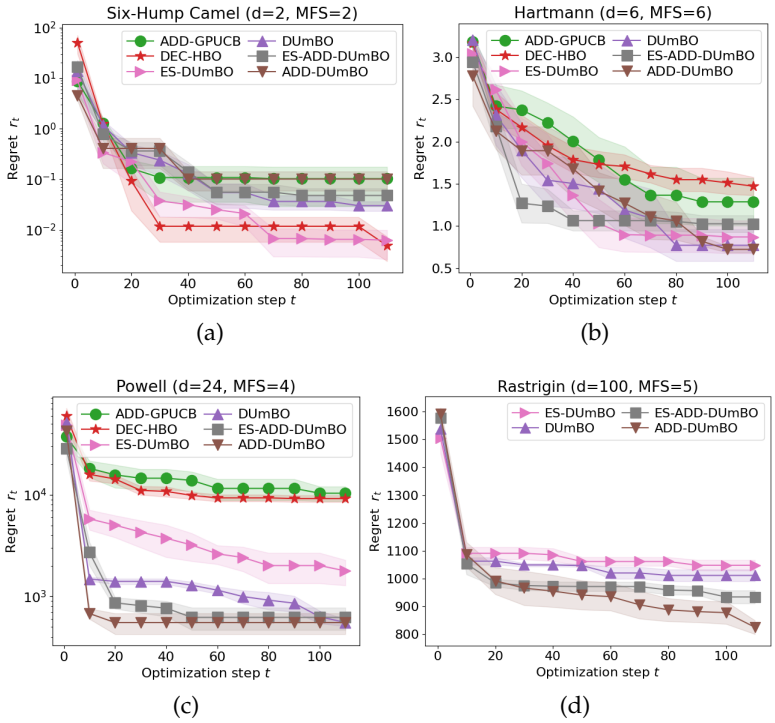
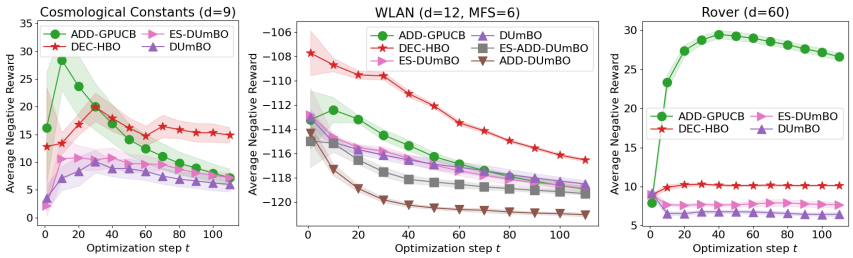



Figure 6.4: Performance achieved by the decomposing BO algorithms and the early-stopped versions of DuMBO for (a) the 2d Six-Hump Camel function, (b) the 6d Hartmann function, (c) the 24d Powell function and (d) the 100d Rastrigin function. The shaded areas indicate the standard error intervals.

Figure 6.4 depicts the performance of the early-stopped versions on the synthetic functions described in Appendix G. Except for the SHC function (Figure 6.4a), the same dynamic can be observed. The early-stopped versions ES-DuMBO and ES-ADD-DuMBO obtain slightly worse performance than their counterparts DuMBO and ADD-DuMBO. However, they remain very competitive, as they outperform the state-of-the-art decomposing BO algorithms in 3 out of 4 synthetic experiments. Regarding the SHC function, the early-stopped versions achieve better performance than their counterparts. ES-DuMBO even achieves similar performance than DEC-HBO. As a future work, we plan to conceptually better understand this observation.

Figure 6.5 depicts the performance of the early-stopped versions on the real-world problems considered in Appendix G. The three experiments are in agreement with the results observed for the synthetic functions in Figure 6.4. ES-DuMBO and ES-ADD-DuMBO perform slightly worse than their counterparts, but they remain very competitive as they outperform DEC-HBO in all three experiments, and ADD-GPUCB on the rover trajectory planning problem (Figure 6.5c). In the two remaining problems (Figures 6.5a, 6.5b), the early-stopped versions achieve better (or equivalent) performance than ADD-GPUCB.




(a) (b) (c)

Figure 6.5: Performance achieved by the decomposing BO algorithms and the early-stopped versions of DuMBO for (a) the cosmological constants fine-tuning, (b) the maximization of the Shannon capacity in a WLAN and (c) the trajectory planning of a rover. The shaded areas indicate the standard error intervals.

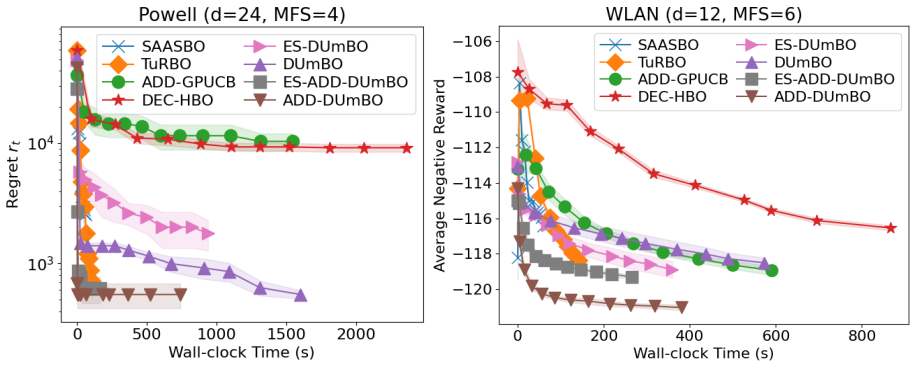
These result demonstrate that, although the early-stopped version of DuMBO provides only minimax guarantees on the maximization of $\varphi_t$, its excellent empirical performance make it a very interesting solution. This is especially true for technological contexts that cannot afford high computing capabilities, as we discuss in the next section.

### 6.5.4 *Wall-Clock Time*

In this section, we provide wall-clock time measurements (excluding the evaluation time of the objective function) of the described BO algorithms on a synthetic function (24d Powell) and a real-world problem (WLAN) described in Appendix G. The measurements were taken using a server equipped with two Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, with 14 cores (28 threads) each.

Figure 6.6 gathers the wall-clock time measurements. Observe that DuMBO does not only offer very competitive performance, it also exhibits a lower overhead when compared to the other decomposing algorithms (DEC-HBO and ADD-GPUCB). However, SAASBO and TuRBO manage to get lower runtimes than DuMBO. This is not surprising since, by design, these methods have minimal overheads, at the expense of any theoretical guarantees.

Nevertheless, observe that the early-stopped version of DuMBO, ES-DuMBO, also reaches very good performance, with a significantly reduced response time. With ADD-DuMBO, observe that having access to the true additive decomposition of the function also reduces the overhead of the solution, since the decomposition does not need to be inferred anymore. Finally, observe that ES-ADD-DuMBO, the early-stopped version of DuMBO when the additive decomposition is provided, obtains similar results as TuRBO and SAASBO, with only a slightly larger runtime, especially on the Powell synthetic function (Figure 6.6a). Therefore, we argue that DuMBO, due to its excellent




(a) (b)

Figure 6.6: Performance achieved by all the described BO algorithms (including the two versions of DuMBO and their early-stopped alternatives) w.r.t. their wall-clock time for (a) the 24d Powell synthetic function and (b) the maximization of the Shannon capacity in a WLAN. The shaded areas indicate the standard error intervals.

empirical performance and its satisfying runtime (thanks to its capacity to be early-stopped) is a competitive solution even in critical applications where the response time needs to be low.

## 6.6 SUMMARY AND LIMITATIONS

In this chapter, we investigated the benefits of relaxing the restrictive assumptions of low-MFS additive decomposition that limit the applicability domain of state-of-the-art decomposing BO algorithms. As illustrated by Table 6.1, we chose to optimize our acquisition function with algorithms that scale well with the number of dimensions. This enables us to exploit, when available, the true (and potentially complex) additive decomposition of the objective function $f$ instead of approximating it with a simple decomposition. To illustrate the effectiveness of such design choices, we proposed DuMBO, an asymptotically optimal BO algorithm that optimizes $f$ in a decentralized fashion, thanks to a tighter decentralized approximation of GP-UCB that requires less exploration than the previously proposed approximations. As demonstrated by Sections 6.4.5 and 6.5, DuMBO is a competitive alternative to state-of-the-art BO algorithms, able to optimize complex objective functions in a small number of iterations. Compared to other decomposing algorithms, such as ADD-GPUCB and DEC-HBO, DuMBO constitutes a significant improvement, particularly when the decomposition of $f$ has numerous factors with large MFS (Figures 6.2b, 6.2c, 6.2d and 6.3b), or simply does not exist (Figures 6.3a and 6.3c). Overall, DuMBO makes a step further in the direction of using provably optimal BO algorithms to optimize black-box functions in decentralized systems. It has been recently submit-

ted to a flagship international conference. Note that the preprint of the submission is accessible on arXiv [144]. Follow-up to this work are plenty and include the extension of DuMBO to batch mode [145, 146] or its application to other suitable technological contexts such as telecommunication networks [88], UAVs [131] or within a multi-robots team [147].

Although DuMBO has demonstrated good performance on hard problems, it still has some limitations:

(i) **Design**: although it has a lower complexity than the other additive algorithms, its execution time is still significantly larger than those of algorithms that trade-off theoretical guarantees for low execution times, such as TuRBO [117] or SAASBO [122].

(ii) **Theoretical**: although we showed that DuMBO has a lower immediate regret bound than DEC-HBO (Theorem 6.3), a lower cumulative regret bound remains to be found to theoretically assess its superiority over other algorithms.

(iii) **Theoretical**: since the convergence of ADMM is ensured when the acquisition function is restricted prox-regular, proving that our proposed acquisition function belongs to this class would increase the guarantees provided by DuMBO.

*[144] Bardou et al. (2023), "Relaxing the Additivity Constraints in Decentralized No-Regret High-Dimensional Bayesian Optimization"*

*[145] Li et al. (2016), "Multiple queries as bandit arms"*

*[146] Daxberger et al. (2017), "Distributed batch Gaussian process optimization"*

*[147] Chen et al. (2013), "Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system"*

# 7 CONCLUSION

## 7.1 THESIS SUMMARY

Throughout this thesis, we investigated algorithms that allow the optimization of a black-box function, driven by the need of efficiently optimizing wireless networks. In particular, we put a strong focus on the optimization of the spatial reuse of the radio channel in a Wireless Local Area Network (WLAN).

In Chapter 2, we started by describing in details the existing approaches addressing the spatial reuse problem in WLANs through the control of the `TX_PWR` and `OBSS_PD` parameters (Section 2.1). We showed that trying to analytically model the throughput within a WLAN is a challenging approach that appears less suitable than data-driven solutions. Then, we discussed the black-box optimization algorithms that seem to represent a promising option to conceive novel efficient solutions to improve the spatial reuse of WLANs (Section 2.2). Multi-Armed Bandit (MAB) and Bayesian Optimization (BO) stand out as two well-suited frameworks able to provide optimality guarantees along with their generally excellent empirical performance.

Based on these observations, we proposed in Chapter 3 two centralized ad-hoc solutions to the spatial reuse problem in WLANs. Based on the MAB framework, they address the high-dimensionality of the input space $\mathcal{C}$ by subsampling it to build a reservoir of promising configurations. Concurrently, they use Thompson sampling to identify the best configuration in the reservoir. The two approaches mainly differ by their subsampling strategies. They are based on a Gaussian mixture and on a mixture of hyperspheres, respectively. Both rely on a regularity assumption of the objective function $f$. Although they show good empirical performance on complex scenarios, they do not provide theoretical guarantees regarding their convergence nor their optimality.

In Chapter 4, the BO framework was considered for the spatial reuse problem in WLANs. We proposed a decentralized solution, called INSPIRE. It exploits a natural additive decomposition of the objective function $f$ to allow each Access Point (AP) to work and interact only with other APs in its communication range. To come to an agreement on which configuration to test in its neighborhood, each AP relies on a consensus function that we demonstrated optimal in the minimax sense. INSPIRE was tested on complex, realistic WLANs deployments and exhibited excellent empirical performance, significantly better than other state-of-the-art solutions.

With Chapter 5, we demonstrated that INSPIRE can also be successfully applied to other technological contexts. In fact, we used it to propose a decentralized solution to the power control of next-generation cellular networks. We demonstrated that the introduction of a new feature, called Non-Orthogonal Multiple Access (NOMA), can, if optimized properly, significantly increase the throughput of cellular networks and preserve their fairness towards end-users.

Finally, in Chapter 6, we directly addressed the no-regret BO of a noisy, black-box, high-dimensional function that is costly to evaluate. We proposed DuMBO, a decentralized no-regret BO algorithm that enables us to relax a restrictive assumption that limited the applicability domain of state-of-the-art, no-regret BO algorithms. Advantageously, DuMBO can be early-stopped whenever deemed necessary and still provide minimax optimal guarantee on the maximization of its acquisition function. Besides, we proposed an alternative approximation of the acquisition function GP-UCB in a decentralized context. It results in a lower immediate regret bound than the one obtained by using the current state-of-the-art decentralized approximation of GP-UCB. DuMBO was evaluated on difficult synthetic functions and real-world problems (including the throughput optimization in WLANs) and systematically demonstrated competitive performance against state-of-the-art BO algorithms.

## 7.2 FUTURE WORKS

In this section, we describe four research directions that could be natural extensions of this work.

### 7.2.1 *Technological Applications*

Although the performance of the solutions proposed in this thesis were illustrated through the spatial reuse optimization of WLANs, our solutions could be beneficial to many technological contexts. More precisely, INSPIRE, DuMBO or its early-stopped version could be very efficient at optimizing a black-box objective function in complex systems where little assumptions can be made. Furthermore, if the complex system provides access to a natural additive decomposition of the objective function, the performance gain is even larger, as demonstrated in Chapter 6.

As illustrated throughout this thesis, wireless networks constitute such complex systems wherein our solutions thrive, but other examples include multi-robots teams [147], UAVs fleets [131], or energy consumption minimization in sustainable cities [130].

### 7.2.2 *Dynamic Setting*

In this thesis, the objective function $f$ was considered static. However, many interesting problems, including the spatial reuse optimization in WLANs may be time-varying. Extending our solutions (and their theoretical guarantees) to the dynamic setting could be an impactful extension to this work. Note that these extensions could be inspired by the existing work on dynamic BO [148] and spatio-temporal Gaussian Process (GP) [149].

Optimizing a time-varying objective function $f(\boldsymbol{x}, t)$ is hard, as each query is now a point in space-time $(\boldsymbol{x}, t)$, with $\boldsymbol{x} \in \mathcal{C}$ and $t \in \mathbb{R}^+$. This means that the BO algorithm would have to take into account its own response time, and to decide by itself the moment when a relevant input needs to be queried. Some approaches, such as [148], have already started to address this problem. Some questions also arise from the fact that time only goes forward. In fact, any observation becomes less and less relevant, as time goes by, for predicting the future values of $f$. How does this affect the cumulative regret of the BO algorithm? What is a relevant criterion to declare that an observation is outdated and should be removed from the dataset? This is a subset of questions that remain to be answered.

*[148] Nyikosa et al. (2018), "Bayesian optimization for dynamic problems"*

*[149] Hamelijnck et al. (2021), "Spatio-temporal variational Gaussian processes"*

### 7.2.3 *Multi-Objective Optimization*

In this thesis, we systematically optimized scalar objective functions. For the most part of our contributions, these objective functions were scalarizations of an underlying multi-objective problem, consisting in jointly optimizing the throughput and the fairness of the wireless network. Although the empirical performance were satisfying, scalarizing a multi-objective problem often means to make enough arbitrary choices to simplify the optimization task.

Developing Black-Box Optimization (BBO) solutions able to address the true multi-objective optimization task in a constrained environment (e.g. distributed computing resources or low computing power...) would make a natural extension to this work, and an interesting solution for technological contexts such as wireless networks. Some approaches, such as [150–152], already proposed to use BO for solving multi-objective problems.

*[150] Laumanns et al. (2002), "Bayesian optimization algorithms for multi-objective optimization"*

*[151] Hernández-Lobato et al. (2016), "Predictive entropy search for multi-objective bayesian optimization"*

*[152] Paria et al. (2020), "A flexible framework for multi-objective bayesian optimization using random scalarizations"*

### 7.2.4 *More General Processes*

Although GPs demonstrate excellent empirical performance and theoretical guarantees when used as surrogate models, they require some prior knowledge about the function to optimize, embedded in the covariance function $k$. Another extension to this work would be to adapt our solutions to more general processes, that place a nonpara-

metric prior on the GP covariance kernel function. To do so, existing works on Student-t processes [153, 154] and on the matrix-t distribution [155] could be used.

In a more theoretical point of view, contributing to the understanding of these processes and deriving lower cumulative regret bounds for BO algorithms that use them as surrogate models would constitute a significant contribution to the BBO community.

[153] Heyde et al. (2005), "Student processes"

[154] Shah et al. (2014), "Student-t processes as alternatives to Gaussian processes"

[155] Kibria (2006), "The matrix-t distribution and its applications in predictive inference"

## 7.3 PERSONAL STATEMENT

To conclude this thesis, I provide in this section a personal, subjective opinion about the motivations behind this work and the research directions that I deem beneficial for the computer networks community.

Over the course of my doctoral studies, I was increasingly interested in producing contributions that are supported with theoretical analysis. This was mainly due to a simple observation: the vast majority of the scientific contributions (that I encountered) addressing the spatial reuse optimization in 802.11 networks was almost exclusively supported by the good empirical performance of ad-hoc optimization algorithms that do not offer guarantees of any kind. Although some contributions consider optimization algorithms that initially offer interesting properties, their rough adaptations to the technological context resulted in exotic versions with uncharacterized behavior. These works often justify the absence of theoretical guarantees by the fact that they address hard optimization problems in complex systems, where very few assumptions on the objective function are allowed.

Throughout this thesis, I wanted to reconsider the notion of best-effort in telecommunication technologies to include, whenever possible, optimality guarantees. In my view, black-box optimization is a great tool to address hard optimization problems in complex systems without jeopardizing such guarantees. I hope that this thesis will help bridging the gap between the computer networks community and the black-box optimization community. I believe that the interaction between these two communities will produce exciting results allowing major performance gains for telecommunication technologies.

# BIBLIOGRAPHY


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

[2] Mingzhe Chen, Deniz Gündüz, Kaibin Huang, Walid Saad, Mehdi Bennis, Aneta Vulgarakis Feljan, and H Vincent Poor. "Distributed learning in wireless networks: Recent progress and future challenges." In: *IEEE Journal on Selected Areas in Communications* 39.12 (2021), pp. 3579–3605.

[3] M. Gast. *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2005.

[4] Ovidiu Iacoboaiea, Jonatan Krolikowski, Zied Ben Houidi, and Dario Rossi. "Real-time channel management in WLANs: Deep reinforcement learning versus heuristics." In: *2021 IFIP Networking Conference (IFIP Networking)*. IEEE. 2021, pp. 1–9.

[5] M. Gast. *802.11n: the survival guide*. O'Reilly Media, Inc., 2012.

[6] Oluwaseun J Adeyemi, Segun I Popoola, Aderemi A Atayero, David G Afolayan, Mobolaji Ariyo, and Emmanuel Adetiba. "Exploration of daily Internet data traffic generated in a smart university campus." In: *Data in brief* 20 (2018), pp. 30–52.

[7] 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." In: *IEEE Std 802.11ax-2021* (2021), pp. 1–767. DOI: 10.1109/IEEESTD.2021.9442429.

[8] Francesc Wilhelmi, Sergio Barrachina-Munoz, Boris Bellalta, Cristina Cano, Anders Jonsson, and Gergely Neu. "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs." In: *Journal of Network and Computer Applications* 127 (2019), pp. 26–42.

[9] 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. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, 944–949.

[10] 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.

[11] Gregory Hornby, Al Globus, Derek Linden, and Jason Lohn. "Automated Antenna Design with Evolutionary Algorithms." In: *American Institute of Aeronautics and Astronautics*. 2006. DOI: 10.2514/6.2006-7242.

[12] Oded Berger-Tal, Jonathan Nathan, Ehud Meron, and David Saltz. "The exploration-exploitation dilemma: a multidisciplinary framework." In: *PloS one* 9.4 (2014), e95693.

[13] Peter Auer. "Using confidence bounds for exploitation-exploration trade-offs." In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 397–422.

[14] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. "A Client-Driven Approach for Channel Management in Wireless LANs." In: *IEEE International Conference on Computer Communications*. 2006.

[15] Julien Herzen, Ruben Merz, and Patrick Thiran. "Distributed spectrum assignment for home WLANs." In: *2013 Proceedings Ieee Infocom*. IEEE. 2013, pp. 1573–1581.

[16] Sergio Barrachina-Muñoz, Francesc Wilhelmi, and Boris Bellalta. "Online primary channel selection for dynamic channel bonding in high-density WLANs." In: *IEEE Wireless Communications Letters* 9.2 (2019), pp. 258–262.

[17] Álvaro López-Raventós and Boris Bellalta. "Concurrent decentralized channel allocation and access point selection using multi-armed bandits in multi BSS WLANs." In: *Computer Networks* 180 (2020), p. 107381.

[18] Francesc Wilhelmi, Sergio Barrachina-Muñoz, Cristina Cano, Ioannis Selinis, and Boris Bellalta. "Spatial reuse in IEEE 802.11 ax WLANs." In: *Computer Communications* 170 (2021), pp. 65–83.

[19] 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." In: *Wireless Communications and Mobile Computing* 4.8 (2004), pp. 933–946.

[20] Youngsoo Kim, Jeonggyun Yu, and Sunghyun Choi. "SP-TPC: a self-protective energy efficient communication strategy for IEEE 802.11 WLANs." In: *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*. Vol. 3. IEEE. 2004, pp. 2078–2082.

[21] Shuwei Qiu, Xiaowen Chu, Yiu-Wing Leung, and Joseph Kee Yin Ng. "Joint access point placement and power-channel-resource-unit assignment for 802.11 ax-based dense WiFi with QoS requirements." In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 2569–2578.

[22] Giuseppe Bianchi. "Performance analysis of the IEEE 802.11 distributed coordination function." In: *IEEE Journal on selected areas in communications* 18.3 (2000), pp. 535–547.

[23] Rafael Laufer and Leonard Kleinrock. "The capacity of wireless CSMA/CA networks." In: *IEEE/ACM Transactions on Networking* 24.3 (2015), pp. 1518–1532.

[24] Marija Stojanova, Thomas Begin, and Anthony Busson. "Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach." In: *Performance Evaluation* (2019).

[25] M Shahwaiz Afaqui, Eduard Garcia-Villegas, Elena Lopez-Aguilera, Graham Smith, and Daniel Camps. "Evaluation of dynamic sensitivity control algorithm for IEEE 802.11 ax." In: *2015 IEEE wireless communications and networking conference (WCNC)*. IEEE. 2015, pp. 1060–1065.

[26] Hyunjoong Lee, Hyung-Sin Kim, and Saewoong Bahk. "LSR: link-aware spatial reuse in IEEE 802.11 ax WLANs." In: *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2021, pp. 1–6.

[27] Szymon Szott, Katarzyna Kosek-Szott, Piotr Gawłowicz, Jorge Torres Gómez, Boris Bellalta, Anatolij Zubow, and Falko Dressler. "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning." In: *IEEE Communications Surveys & Tutorials* 24.3 (2022), pp. 1843–1893.

[28] Elif Ak and Berk Canberk. "FSC: Two-scale AI-driven fair sensitivity control for 802.11 ax networks." In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[29] Francesc Wilhelmi, Jernej Hribar, Selim F Yilmaz, Emre Ozfatura, Kerem Ozfatura, Ozlem Yildiz, Deniz Gündüz, Hao Chen, Xiaoying Ye, Lizhao You, et al. "Federated spatial reuse optimization in next-generation decentralized IEEE 802.11 WLANs." In: *arXiv preprint arXiv:2203.10472* (2022).

[30] 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." In: *Ad Hoc Networks* 88 (2019), pp. 129–141.

[31] The ns3 Project. *The Network Simulator ns-3*. https://www.nsnam.org/. Accessed: 2021-09-30. 2020.

[32] Robert Michael Lewis, Virginia Torczon, and Michael W Trosset. "Direct search methods: then and now." In: *Journal of computational and Applied Mathematics* 124.1-2 (2000), pp. 191–207.

[33] Stéphane Alarie, Charles Audet, Aïmen E Gheribi, Michael Kokkolaras, and Sébastien Le Digabel. "Two decades of blackbox optimization applications." In: *EURO Journal on Computational Optimization* 9 (2021), p. 100011.

[34] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. "Lipschitzian optimization without the Lipschitz constant." In: *Journal of optimization Theory and Applications* 79 (1993), pp. 157–181.

[35] Donald R Jones and Joaquim RRA Martins. "The DIRECT algorithm: 25 years Later." In: *Journal of global optimization* 79.3 (2021), pp. 521–566.

[36] Michel Gendreau, Jean-Yves Potvin, et al. *Handbook of metaheuristics*. Vol. 2. Springer, 2010.

[37] Nikolaus Hansen and Andreas Ostermeier. "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation." In: *Proceedings of IEEE international conference on evolutionary computation*. IEEE. 1996, pp. 312–317.

[38] Anne Auger and Nikolaus Hansen. "Performance evaluation of an advanced local search evolutionary algorithm." In: *2005 IEEE congress on evolutionary computation*. Vol. 2. IEEE. 2005, pp. 1777–1784.

[39] Nikolaus Hansen and Anne Auger. "CMA-ES: evolution strategies and covariance matrix adaptation." In: *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. 2011, pp. 991–1010.

[40] Nikolaus Hansen. "The CMA evolution strategy: A tutorial." In: *arXiv preprint arXiv:1604.00772* (2016).

[41] Mohammad Nabi Omidvar and Xiaodong Li. "A comparative study of CMA-ES on large scale global optimisation." In: *AI 2010: Advances in Artificial Intelligence: 23rd Australasian Joint Conference, Adelaide, Australia, December 7-10, 2010. Proceedings 23*. Springer. 2011, pp. 303–312.

[42] Yizao Wang, Jean-Yves Audibert, and Rémi Munos. "Algorithms for infinitely many-armed bandits." In: *Advances in Neural Information Processing Systems* 21 (2008).

[43] Rajeev Agrawal. "The continuum-armed bandit problem." In: *SIAM journal on control and optimization* 33.6 (1995), pp. 1926–1951.

[44] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[45] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." In: *Machine learning* 47 (2002), pp. 235–256.

[46] Tze Leung Lai, Herbert Robbins, et al. "Asymptotically efficient adaptive allocation rules." In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[47] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples." In: *Biometrika* 25.3-4 (1933), pp. 285–294.

[48] Shipra Agrawal and Navin Goyal. "Analysis of thompson sampling for the multi-armed bandit problem." In: *Conference on learning theory*. JMLR Workshop and Conference Proceedings. 2012, pp. 39–1.

[49] Shipra Agrawal and Navin Goyal. "Further Optimal Regret Bounds For Thompson Sampling." In: *Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2013.

[50] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. "Thompson sampling: An asymptotically optimal finite-time analysis." In: *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings 23*. Springer. 2012, pp. 199–213.

[51] Aurélien Garivier and Olivier Cappé. "The KL-UCB algorithm for bounded stochastic bandits and beyond." In: *Proceedings of the 24th annual conference on learning theory*. JMLR Workshop and Conference Proceedings. 2011, pp. 359–376.

[52] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. "Contextual bandits with linear payoff functions." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 208–214.

[53] Christopher K. I. Williams and Carl Edward Rasmussen. "Gaussian Processes for Regression." In: *Conference on Neural Information Processing Systems (NeurIPS'95)*. 1995.

[54] Shanti S Gupta and Klaus J Miescke. "Bayesian look ahead one-stage sampling allocations for selection of the best population." In: *Journal of statistical planning and inference* 54.2 (1996), pp. 229–244.

[55] Donald R. Jones, Matthias Schonlau, and William J. Welch. "Efficient global optimization of expensive black-box functions." In: *Journal of Global optimization* 13.4 (1998), 455–492.

[56] Jonas Mockus. "Application of Bayesian approach to numerical methods of global and stochastic optimization." In: *Journal of Global Optimization* 4 (1994), pp. 347–365.

[57] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting." In: *IEEE Transactions on Information Theory* 58.5 (2012), 3250–3265. DOI: doi:10.1109/tit.2011.2182033.

[58] Chao Qin, Diego Klabjan, and Daniel Russo. "Improving the expected improvement algorithm." In: *Advances in Neural Information Processing Systems* 30 (2017).

[59] 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*. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, 2013, pp. 115–123.

[60] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. "A quantitative measure of fairness and discrimination." In: *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA* 21 (1984).

[61] Li Bin Jiang and Soung Chang Liew. "Proportional fairness in wireless LANs and ad hoc networks." In: *IEEE Wireless Communications and Networking Conference, 2005*. Vol. 3. IEEE. 2005, pp. 1551–1556.

[62] Li Li, Martin Pal, and Yang Richard Yang. "Proportional fairness in multi-rate wireless LANs." In: *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE. 2008, pp. 1004–1012.

[63] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. "A Client-Driven Approach for Channel Management in Wireless LANs." In: *IEEE International Conference on Computer Communications*. 2006.

[64] M. Garetto, T. Salonidis, and E.W. Knightly. "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks." In: *IEEE/ACM Transactions on Networking* (2008).

[65] Yahel David and Nahum Shimkin. "Infinitely many-armed bandits with unknown value distribution." In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*. Springer. 2014, pp. 307–322.

[66] Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. "Pure exploration in infinitely-armed bandit models with fixed-confidence." In: *Algorithmic Learning Theory*. PMLR. 2018, pp. 3–24.

[67] William M Bolstad and James M Curran. *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.

[68] Bin Wang, Wenzhong Shi, and Zelang Miao. "Confidence analysis of standard deviational ellipse and its extension into higher dimensional Euclidean space." In: *PloS one* 10.3 (2015), e0118537.

[69] Cisco Meraki. *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. 2022.

[70] Anthony Bardou, Thomas Begin, and Anthony Busson. "Improving the Spatial Reuse in IEEE 802.11ax WLANs: A Multi-Armed Bandit Approach." In: *MSWiM '21: Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. Ed. by ACM. 2021.

[71] Anthony Bardou, Thomas Begin, and Anthony Busson. "Analysis of a Multi-Armed Bandit solution to improve the spatial reuse of next-generation WLANs." In: *Computer Communications* 193 (2022), pp. 279–292.

[72] Anthony Bardou, Thomas Begin, and Anthony Busson. "Mitigating starvation in dense WLANs: A multi-armed Bandit Solution." In: *Ad Hoc Networks* 138 (2023).

[73] Anthony Bardou, Thomas Begin, and Anthony Busson. "Multi-Armed Bandit Algorithm for Spatial Reuse in WLANs: Minimizing Stations in Starvation." In: *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*. 2022.

[74] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Püschel. "Distributed optimization with local domains: Applications in MPC and network flows." In: *IEEE Transactions on Automatic Control* 60.7 (2014), pp. 2004–2009.

[75] Marc G Genton. "Classes of kernels for machine learning: a statistics perspective." In: *Journal of machine learning research* 2.Dec (2001), pp. 299–312.

[76] Richard Bronson and Gabriel B Costa. *Schaum's outline of differential equations*. McGraw-Hill Education, 2014.

[77] Thomas J Santner, Brian J Williams, William I Notz, and Brain J Williams. *The design and analysis of computer experiments*. Vol. 1. Springer, 2003.

[78] Pramudita Satria Palar, Lucia Parussini, Luigi Bregant, Koji Shimoyama, and Lavi Rizki Zuhal. "On kernel functions for bi-fidelity Gaussian process regressions." In: *Structural and Multidisciplinary Optimization* 66.2 (2023), p. 37.

[79] Claude Lemaréchal. "Cauchy and the gradient method." In: *Doc Math Extra* 251.254 (2012), p. 10.

[80] Haskell B Curry. "The method of steepest descent for non-linear minimization problems." In: *Quarterly of Applied Mathematics* 2.3 (1944), pp. 258–261.

[81] Garrett Hardin. "The Tragedy of the Commons." In: *Journal of Natural Resources Policy Research* 1.3 (2009), pp. 243–253. DOI: 10.1080/19390450903037302.

[82] Hugo Landaluce, Laura Arjona, Asier Perallos, Francisco Falcone, Ignacio Angulo, and Florian Muralter. "A review of IoT sensing applications and challenges using RFID and wireless sensor networks." In: *Sensors* 20.9 (2020), p. 2495.

[83] Shivashankar Rajendra Prasad P. "Efficient performance analysis of energy aware on demand routing protocol in mobile ad-hoc network." In: (2019).

[84] Marco Giordani and Michele Zorzi. "Non-terrestrial networks in the 6G era: Challenges and opportunities." In: *IEEE Network* 35.2 (2020), pp. 244–251.

[85] Yuege Xie, Xiaoxia Wu, and Rachel Ward. "Linear convergence of adaptive stochastic gradient descent." In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 1475–1485.

[86] Manuel Blum. *LibGP*. GitHub, https://github.com/mblum/libgp. 2016.

[87] Kai Wang, Bryan Wilder, Sze-chuan Suen, Bistra Dilkina, and Milind Tambe. "Improving GP-UCB algorithm by harnessing decomposed feedback." In: *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*. Springer. 2020, pp. 555–569.

[88] Anthony Bardou and Thomas Begin. "INSPIRE: Distributed Bayesian Optimization for ImproviNg SPatIal REuse in Dense WLANs." In: *MSWiM '22: Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. Ed. by ACM. 2022.

[89] Anthony Bardou and Thomas Begin. "Analysis of a decentralized Bayesian optimization algorithm for improving spatial reuse in dense WLANs." In: *Computer Communications* (2023).

[90] Anthony Bardou and Thomas Begin. "INSPIRE: Optimisation bayésienne distribuée pour l'amélioration de la réutilisation spatiale des WLANs denses." In: *AlgoTel 2022-24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*. 2022.

[91] Analysys Manson. "Wireless network data traffic: worldwide trends and forecasts 2021–2026." In: *Forecast Report* (2021).

[92] John Vacca. *Network and system security*. Elsevier, 2014.

[93] Theodore S Rappaport. "The wireless revolution." In: *IEEE Communications Magazine* 29.11 (1991), pp. 52–71.

[94] Guowang Miao, Jens Zander, Ki Won Sung, and Slimane Ben Slimane. *Fundamentals of mobile data networks*. Cambridge University Press, 2016.

[95] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. "5G evolution: A view on 5G cellular technology beyond 3GPP release 15." In: *IEEE access* 7 (2019), pp. 127639–127651.

[96] Asha Mehrotra. *GSM system engineering*. Artech House, Inc., 1997.

[97] Cristina Ciochina and Hikmet Sari. "A review of OFDMA and single-carrier FDMA." In: *2010 European Wireless Conference (EW)*. IEEE. 2010, pp. 706–710.

[98] Moe Rahnema and Marcin Dryjanski. *From LTE to LTE-Advanced Pro and 5G*. Artech House, 2017.

[99] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. "Successive interference cancellation: A back-of-the-envelope perspective." In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. 2010, pp. 1–6.

[100] Refik Caglar Kizilirmak and Hossein Khaleghi Bizaki. "Non-orthogonal multiple access (NOMA) for 5G networks." In: *Towards 5G Wireless Networks-A Physical Layer Perspective* 83 (2016), pp. 83–98.

[101] Abdelbaset S Hamza, Shady S Khalifa, Haitham S Hamza, and Khaled Elsayed. "A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks." In: *IEEE Communications Surveys & Tutorials* 15.4 (2013), pp. 1642–1670.

[102] Kenneth Appel and Wolfgang Haken. "The solution of the four-color-map problem." In: *Scientific American* 237.4 (1977), pp. 108–121.

[103] Zhikun Xu, Geoffrey Ye Li, Chenyang Yang, and Xiaolong Zhu. "Throughput and optimal threshold for FFR schemes in OFDMA cellular networks." In: *IEEE Transactions on Wireless Communications* 11.8 (2012), pp. 2776–2785.

[104] Mohamed Elwekeil, Masoud Alghoniemy, Osamu Muta, Adel B Abdel-Rahman, Haris Gacanin, and Hiroshi Furukawa. "Performance evaluation of an adaptive self-organizing frequency reuse approach for OFDMA downlink." In: *Wireless networks* 25 (2019), pp. 507–519.

[105] S Ezhilarasi and PTV Bhuvaneswari. "Maximization of sum throughput in LTE heterogeneous network using region splitting-based resource partitioning scheme." In: *Wireless Personal Communications* 121 (2021), pp. 905–938.

[106] Kaidi Wang, Yuanwei Liu, Zhiguo Ding, Arumugam Nallanathan, and Mugen Peng. "User association and power allocation for multi-cell non-orthogonal multiple access networks." In: *IEEE Transactions on Wireless Communications* 18.11 (2019), pp. 5284–5298.

[107] Mojtaba Vaezi, Robert Schober, Zhiguo Ding, and H Vincent Poor. "Non-orthogonal multiple access: Common myths and critical questions." In: *IEEE Wireless Communications* 26.5 (2019), pp. 174–180.

[108] Parameswaran Ramesh and PTV Bhuvaneswari. "Non orthogonal multiple access requirements for 5G and its myths." In: *2022 IEEE Delhi Section Conference (DELCON)*. IEEE. 2022, pp. 1–6.

[109] Subhankar Banerjee, Chung Shue Chen, Marceau Coupechoux, and Abhishek Sinha. "Joint Power and Subcarrier Allocation in Multi-Cell Multi-Carrier NOMA." In: *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE. 2022, pp. 180–185.

[110] JHB Kemperman. "On the Shannon capacity of an arbitrary channel." In: *Indagationes Mathematicae (Proceedings)*. Vol. 77. 2. North-Holland. 1974, pp. 101–115.

[111] Jeonghoon Mo and Jean Walrand. "Fair end-to-end window-based congestion control." In: *IEEE/ACM Transactions on networking* 8.5 (2000), pp. 556–567.

[112] Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. "Log gaussian cox processes." In: *Scandinavian journal of statistics* 25.3 (1998), pp. 451–482.

[113] Ziyang Wang, Rainer Schoenen, Halim Yanikomeroglu, and Marc St-Hilaire. "The impact of user spatial heterogeneity in heterogeneous cellular networks." In: *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2014, pp. 1278–1283.

[114] Kalyanmoy Deb and Kalyanmoy Deb. "Multi-objective optimization." In: *Search methodologies: Introductory tutorials in optimization and decision support techniques*. Springer, 2013, pp. 403–449.

[115] Mark Voorneveld. "Characterization of pareto dominance." In: *Operations Research Letters* 31.1 (2003), pp. 7–11.

[116] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. "Bayesian Optimization in High Dimensions via Random Embeddings." In: *IJCAI*. Vol. 13. 2013, pp. 1778–1784.

[117] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. "Scalable global optimization via local bayesian optimization." In: *Advances in neural information processing systems* 32 (2019).

[118] Mickaël Binois, David Ginsbourger, and Olivier Roustant. "A warped kernel improving robustness in Bayesian optimization via random embeddings." In: *Learning and Intelligent Optimization: 9th International Conference, LION 9, Lille, France, January 12-15, 2015. Revised Selected Papers 9*. Springer. 2015, pp. 281–286.

[119] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. "Re-examining linear embeddings for high-dimensional Bayesian optimization." In: *Advances in neural information processing systems* 33 (2020), pp. 1546–1558.

[120] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. "Automatic chemical design using a data-driven continuous representation of molecules." In: *ACS central science* 4.2 (2018), pp. 268–276.

[121] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. "High-dimensional Bayesian optimization using low-dimensional feature spaces." In: *Machine Learning* 109.9 (2020), pp. 1925–1943.

[122] David Eriksson and Martin Jankowiak. "High-dimensional Bayesian optimization with sparse axis-aligned subspaces." In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 493–503.

[123] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. "High dimensional Bayesian optimization using dropout." In: *arXiv preprint arXiv:1802.05400* (2018).

[124] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. "Monte Carlo Tree Search based Variable Selection for High Dimensional Bayesian Optimization." In: *Advances in Neural Information Processing Systems*. 2022.

[125] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. "High dimensional Bayesian optimisation and bandits via additive models." In: *International conference on machine learning*. PMLR. 2015, pp. 295–304.

[126] Zi Wang and Stefanie Jegelka. "Max-value entropy search for efficient Bayesian optimization." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3627–3635.

[127] Mojmir Mutny and Andreas Krause. "Efficient high dimensional bayesian optimization with additivity and quadrature fourier features." In: *Advances in Neural Information Processing Systems* 31 (2018).

[128] Trong Nghia Hoang, Quang Minh Hoang, Ruofei Ouyang, and Kian Hsiang Low. "Decentralized high-dimensional Bayesian optimization with factor graphs." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[129] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nicholas R Jennings. "Bounded approximate decentralised coordination via the max-sum algorithm." In: *Artificial Intelligence* 175.2 (2011), pp. 730–759.

[130] Mathieu Bourdeau, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. "Modeling and forecasting building energy consumption: A review of data-driven techniques." In: *Sustainable Cities and Society* 48 (2019), p. 101533.

[131] Lifeng Xie, Jie Xu, and Rui Zhang. "Throughput maximization for UAV-enabled wireless powered communication networks." In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 1690–1703.

[132] David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. "Additive gaussian processes." In: *Advances in neural information processing systems* 24 (2011).

[133] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods." In: *Machine Learning* 110.3 (2021), pp. 457–506.

[134] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. "Discovering and exploiting additive structure for Bayesian optimization." In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1311–1319.

[135] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. "Batched high-dimensional Bayesian optimization via structural kernel learning." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3656–3664.

[136] Christian Robert, George Casella, Christian P Robert, and George Casella. "Metropolis–hastings algorithms." In: *Introducing Monte Carlo Methods with R* (2010), pp. 167–197.

[137] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[138] Daniel Gabay and Bertrand Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation." In: *Computers & mathematics with applications* 2.1 (1976), pp. 17–40.

[139] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[140] Athanasios P Liavas and Nicholas D Sidiropoulos. "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers." In: *IEEE Transactions on Signal Processing* 63.20 (2015), pp. 5450–5463.

[141] Rongjie Lai and Stanley Osher. "A splitting method for orthogonality constrained problems." In: *Journal of Scientific Computing* 58.2 (2014), pp. 431–449.

[142] Rick Chartrand and Brendt Wohlberg. "A nonconvex ADMM algorithm for group sparsity with sparse groups." In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 6009–6013.

[143] Yu Wang, Wotao Yin, and Jinshan Zeng. "Global convergence of ADMM in nonconvex nonsmooth optimization." In: *Journal of Scientific Computing* 78 (2019), pp. 29–63.

[144] Anthony Bardou, Patrick Thiran, and Thomas Begin. "Relaxing the Additivity Constraints in Decentralized No-Regret High-Dimensional Bayesian Optimization." In: *arXiv preprint arXiv:2305.19838* (2023).

[145] Cheng Li, Paul Resnick, and Qiaozhu Mei. "Multiple queries as bandit arms." In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2016, pp. 1089–1098.

[146] Erik A Daxberger and Bryan Kian Hsiang Low. "Distributed batch Gaussian process optimization." In: *International conference on machine learning*. PMLR. 2017, pp. 951–960.

[147] Jie Chen, Kian Hsiang Low, and Colin Keng-Yan Tan. "Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system." In: *arXiv preprint arXiv:1306.1491* (2013).

[148] Favour M Nyikosa, Michael A Osborne, and Stephen J Roberts. "Bayesian optimization for dynamic problems." In: *arXiv preprint arXiv:1803.03432* (2018).

[149] Oliver Hamelijnck, William Wilkinson, Niki Loppi, Arno Solin, and Theodoros Damoulas. "Spatio-temporal variational Gaussian processes." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23621–23633.

[150] Marco Laumanns and Jiri Ocenasek. "Bayesian optimization algorithms for multi-objective optimization." In: *Parallel Problem Solving from Nature—PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings 7*. Springer. 2002, pp. 298–307.

[151] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. "Predictive entropy search for multi-objective bayesian optimization." In: *International conference on machine learning*. PMLR. 2016, pp. 1492–1501.

[152] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. "A flexible framework for multi-objective bayesian optimization using random scalarizations." In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 766–776.

[153] Chris C Heyde and Nikolai N Leonenko. "Student processes." In: *Advances in Applied Probability* 37.2 (2005), pp. 342–365.

[154] Amar Shah, Andrew Wilson, and Zoubin Ghahramani. "Student-t processes as alternatives to Gaussian processes." In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 877–885.

[155] BM Golam Kibria. "The matrix-t distribution and its applications in predictive inference." In: *Journal of Multivariate Analysis* 97.3 (2006), pp. 785–795.

[156] C.-H. Chuang et al. "The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: single-probe measurements and the strong power of f(z) 8(z) on constraining dark energy." In: *Monthly Notices of the Royal Astronomical Society* 433.4 (2013), pp. 3559–3571. DOI: 10.1093/mnras/stt988. URL: https://doi.org/10.1093%2Fmnras%2Fstt988.

[157] J. Zuntz, M. Paterno, E. Jennings, D. Rudd, A. Manzotti, S. Dodelson, S. Bridle, S. Sehrish, and J. Kowalkowski. "CosmoSIS: Modular cosmological parameter estimation." In: *Astronomy and Computing* 12 (2015), pp. 45–59. DOI: 10.1016/j.ascom.2015.05.005. URL: https://doi.org/10.1016%2Fj.ascom.2015.05.005.

[158] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. "Batched large-scale Bayesian optimization in high-dimensional spaces." In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 745–754.

# Appendix

# A

# PROOF FOR THE OPTIMALITY OF THE CONSENSUS FUNCTION OF INSPIRE

In this appendix, we provide the proof for Theorem 4.1, which states that INSPIRE is minimax optimal.

*Proof.* Let $\boldsymbol{x}^{(i)}$ be the prescription of Access Point (AP) $i$, with $x_j^{(i)}$ the prescription of AP $i$ for AP $j$ and $\boldsymbol{w} \in [0,1]^n$ the weights of a convex combination ($||\boldsymbol{w}||_1 = 1$). Let $f^* = \sum_{i=1}^n f^{(i)}(\boldsymbol{x}^{(i)})$. If $\boldsymbol{x}^{(i)} = \arg\max_{\boldsymbol{x}} f^{(i)}(\boldsymbol{x})$, then it follows that $\forall \tilde{\boldsymbol{x}} \in \mathcal{C}, f^* - f(\tilde{\boldsymbol{x}}) \geq 0$. To find an optimal consensus $\tilde{\boldsymbol{x}}$, this difference must be minimized.

If all the functions $f^{(i)}$ are Lipschitz-continuous with Lipschitz constants $L_i$, we have

$$\begin{aligned}
f^* - f(\tilde{\boldsymbol{x}}) &= \sum_{i=1}^n f^{(i)}(\boldsymbol{x}^{(i)}) - \sum_{i=1}^n f^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{N}_i}) \\
&= \sum_{i=1}^n f^{(i)}(\boldsymbol{x}^{(i)}) - f^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{N}_i}) \\
&\leq \sum_{i=1}^n L_i ||\boldsymbol{x}^{(i)} - \tilde{\boldsymbol{x}}_{\mathcal{N}_i}||_1 \qquad \text{(A.1)} \\
&= \sum_{i=1}^n L_i \sum_{j \in \mathcal{N}_i} \sum_{d=1}^2 |x_{j,d}^{(i)} - \tilde{x}_{j,d}| \qquad \text{(A.2)} \\
&= \Psi(\tilde{\boldsymbol{x}})
\end{aligned}$$

with (A.1) following from each $f^{(i)}$ being Lipschitz continuous with Lipschitz constant $L_i$ and (A.2) following from developing the norm.

There is no lower upper bound of $f^* - f(\boldsymbol{x})$ than $\Psi(\boldsymbol{x})$, because we assume no information about $f$ other than its Lipschitz-continuity (see Assumption 4.5). Therefore, observe that if the minimizer of $\Psi(\boldsymbol{x})$ has a closed-form, it is the one of a minimax optimal. By rearranging the indices and splitting the absolute values we have

$$\begin{aligned}
\Psi(\tilde{\boldsymbol{x}}) &= \sum_{d=1}^2 \sum_{j=1}^n \sum_{i \in \mathcal{N}_j} L_i |x_{j,d}^{(i)} - \tilde{x}_{j,d}| \\
&= \sum_{d=1}^2 \sum_{j=1}^n \left( \sum_{\substack{i \in \mathcal{N}_j \\ x_{j,d}^{(i)} < \tilde{x}_{j,d}}} L_i (\tilde{x}_{j,d} - x_{j,d}^{(i)}) + \sum_{\substack{i \in \mathcal{N}_j \\ x_{j,d}^{(i)} \geq \tilde{x}_{j,d}}} L_i (x_{j,d}^{(i)} - \tilde{x}_{j,d}) \right). \\
&= \sum_{d=1}^2 \sum_{j=1}^n \psi_{j,d}(\tilde{x}_{j,d}) \qquad \text{(A.3)}
\end{aligned}$$

Observe that $\Psi(\tilde{\boldsymbol{x}})$ is be minimal if $\tilde{x}_{j,d}$ is built to minimize $\psi_{j,d}(\tilde{x}_{j,d})$. Therefore, let us focus on building $\tilde{x}_{j,d}$ so that it minimizes

$$\psi_{j,d}(\tilde{x}_{j,d}) = \sum_{\substack{i \in \mathcal{N}_j \\ x_{j,d}^{(i)} < \tilde{x}_{j,d}}} L_i(\tilde{x}_{j,d} - x_{j,d}^{(i)}) + \sum_{\substack{i \in \mathcal{N}_j \\ x_{j,d}^{(i)} \geq \tilde{x}_{j,d}}} L_i(x_{j,d}^{(i)} - \tilde{x}_{j,d}). \quad \text{(A.4)}$$

It is immediate to see that (A.4) is piecewise linear and convex since it is a sum of absolute values (*i.e.* piecewise linear and convex functions). Furthermore, the set of nonlinearity points is necessarily $\mathcal{P}_{j,d} = \left\{x_{j,d}^{(i)}\right\}_{i \in \mathcal{N}_j}$. This directly implies that $\min_{x \in \mathbb{R}^+} \psi_{j,d}(x) \in \mathcal{P}_{j,d}$. Therefore, we need to find $k^* = \arg\min_{k \in \mathcal{N}_j} \psi_{j,d}(x_{j,d}^{(k)})$. Without loss of generality, let us assume that $\mathcal{P}_{j,d}$ is sorted, so that $\forall k \in \mathcal{N}_j$,

$$\psi_{j,d}(x_{j,d}^{(k)}) = \sum_{\substack{i \in \mathcal{N}_j \\ i < k}} L_i(x_{j,d}^{(k)} - x_{j,d}^{(i)}) + \sum_{\substack{i \in \mathcal{N}_j \\ i > k}} L_i(x_{j,d}^{(i)} - \tilde{x}_{j,d}). \quad \text{(A.5)}$$

Let $d_k = \psi_{j,d}(x_{j,d}^{(k)}) - \psi_{j,d}(x_{j,d}^{(k+1)})$. It is trivial to see that

$$d_k = \left(x_{j,d}^{(k+1)} - x_{j,d}^{(k)}\right) \left( \sum_{\substack{i \in \mathcal{N}_j \\ i > k}} L_i - \sum_{\substack{i \in \mathcal{N}_j \\ i < k+1}} L_i \right). \quad \text{(A.6)}$$

Since $\psi_{j,d}$ is convex, $k^*$ is necessarily the smallest $k$ for which $d_k$ is negative. Since $\left(x_{j,d}^{(k+1)} - x_{j,d}^{(k)}\right) \geq 0$ because $\mathcal{P}_{j,d}$ is sorted, $k^*$ is necessarily the smallest $k$ for which $\sum_{\substack{i \in \mathcal{N}_j \\ i > k}} L_i - \sum_{\substack{i \in \mathcal{N}_j \\ i < k+1}} L_i$ is negative. Therefore, we want $\tilde{x}_{j,d} = \min_{x \in \mathbb{R}^+} \psi_{j,d}(x) = x_{j,d}^{(k^*)}$, which corresponds to the median of the values in $\mathcal{P}_{j,d}$, weighted by the Lipschitz constants $\{L_i\}_{i \in \mathcal{N}_j}$.

Because each element in the vector $\tilde{\boldsymbol{x}} = (\tilde{x}_{j,d})_{d \in \{1,2\}, j \in \{1,n\}}$ is built to be the weighted median of the prescriptions for the value of $x_{j,d}$, $\tilde{\boldsymbol{x}}$ is the weighted marginal median of the prescriptions. Eventually, since the weighted marginal median $\tilde{\boldsymbol{x}}$ of the prescriptions (see (4.10)) is indeed a minimum of $\Psi(\boldsymbol{x})$, it is a minimax optimum for the objective function $f$. □

# B

# RANDOM SLICING OF A HIGH-DIMENSIONAL FUNCTION

We detail in this appendix how we build the random slices of the objective function depicted in Figure 4.5 of Chapter 4. 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 $\boldsymbol{x}^*$ found by `INSPIRE` on a given topology, we uniformly draw two random configurations $\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}} \in \mathcal{C}$ and use $(\boldsymbol{x}^*, \hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ as a vector basis. By plotting the reward of each configuration attainable with this basis,[50] we obtain a 3d plot representing a random slice of the considered objective function, with the best configuration found $\boldsymbol{x}^*$ at the origin of the plot.

[50] *That is, any valid configuration $\boldsymbol{x}$ that can be expressed as $\boldsymbol{x} = \boldsymbol{x}^* + a\hat{\boldsymbol{u}} + b\hat{\boldsymbol{v}}, (a, b) \in \mathbb{R}^2$.*

Details of the vector basis for the Wireless Local Area Network (WLAN) topology **T1** (see Figure 4.2a) are as follows:

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

Corresponding values for **T2** (see Figure 4.2b) are:

- $\boldsymbol{x}^*$ = (-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)
- $\hat{\boldsymbol{u}}$ = (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)
- $\hat{\boldsymbol{v}}$ = (-.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)

# C

# THE IMPACT OF FAIRNESS ON THE THROUGHPUT DISTRIBUTION OF A CELLULAR NETWORK

In this appendix, we study the effect of increasing $\alpha$ (see (5.5)) on the throughput distribution of **T1** (Figure 5.8a), one of the cellular networks considered for the benchmarking of the combination of Non-Orthogonal Multiple Access (NOMA) and Fractional Frequency Reuse (FFR).

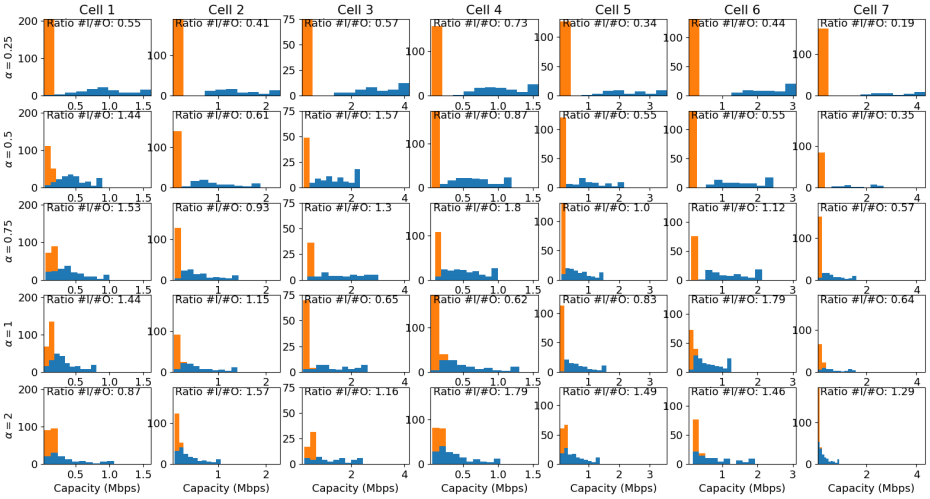



Figure C.1: Throughput distribution of the cells in **T1** and different values of $\alpha$. The throughput distribution of the users in the inner areas is depicted with a blue histogram, the throughput distribution of the users in the outer area is depicted with an orange histogram. For each cell, the ratio between the number of inner users and outer users is given.

Figure C.1 depicts the throughput distribution within the cells in **T1**. Observe that since the inner users are closer to their Base Station (BS) and use the whole frequency band, they achieve a better Shannon capacity than the outer users. However, observe that as $\alpha$ increases, the throughput distribution shrinks to the left. This is due to the inner areas getting bigger and incorporating more users. In the meantime, the outer areas get smaller and the outer users can experience slightly better Shannon capacities. This results in lower Shannon capacity at the network scale, but this also causes an increase in Jain's fairness index, as depicted by Figure 5.10.

# D

# PROOFS FOR THE IMPROVED ACQUISITION FUNCTION OF DUMBO

In this appendix, we provide the proofs for our proposed approximation of the exploration term in the GP-UCB acquisition function as well as our proposed acquisition function $\varphi_t$. We start by proving the following lemma.

**Lemma D.1.** *The best linear overestimation of $\sqrt{x}$ (in the least-squares sense) is (6.6), with $a$ one of the positive roots of the quartic polynomial (6.7).*

*Proof.* We want a linear approximation $ax + b$ that consistently overestimates $\sqrt{x}$ over the interval $[v_-, v_+]$. Since $\sqrt{x}$ is concave, the overestimation is ensured if $ax + b - \sqrt{x} = 0$ has at most a single solution in $\mathbb{R}^+$. This can be achieved by adjusting the $b$ parameter so that the polynomial $Q(x) = ax^2 - x + b$ has a single root. The discriminant of $Q$ is $1 - 4ab$, so $\forall a > 0, b = \frac{1}{4a}$ ensures the overestimation of $\sqrt{x}$.

The linear approximation $ax + \frac{1}{4a}$ must also be optimal in the least squares sense. Therefore, we must find

$$\begin{aligned} a^* &= \arg\min_{a \in \mathbb{R}^+} \int_{v_-}^{v_+} \left(\sqrt{u} - \left(au + \frac{1}{4a}\right)\right)^2 du \\ &= \arg\min_{a \in \mathbb{R}^+} \frac{\left[u^3\right]_{v_-}^{v_+}}{3} a^2 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a + \frac{3\left[u^2\right]_{v_-}^{v_+}}{4} - \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3a} + \frac{[u]_{v_-}^{v_+}}{16a^2}. \end{aligned} \tag{D.1}$$

Differentiating (D.1) by $a$ and multiplying by $a^3$ to turn the expression into a polynomial, we get the desired quartic

$$P(a) = \frac{2\left[u^3\right]_{v_-}^{v_+}}{3} a^4 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^3 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3} a - \frac{[u]_{v_-}^{v_+}}{8}.$$

Therefore, $a$ must be one of the positive roots of $P$. Since $ax + \frac{1}{4a}$ consistently overestimates $\sqrt{x}$, $P$ has at least one positive root by construction. □

With Lemma D.1, we can prove Proposition 6.2.

*Proof.* We need to prove that the quartic (6.7) has a single positive root, which is also the solution of (D.1). Let us consider the derivative $P'$ of $P$, which, by construction, has the same sign as the second derivative of the expression in (D.1).

$$P'(a) = \frac{8\left[u^3\right]_{v_-}^{v_+}}{3} a^3 - \frac{12\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^2 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3} \tag{D.2}$$

Studying the discriminant of (D.2) reveals that it is always non-positive, meaning that $P'$ has a single real root. Furthermore, it is easy to show that

(i) $\lim_{a\to-\infty} P'(a) = -\infty$ and,

(ii) $\lim_{a\to 0} P'(a) > 0$.

Hence, with the intermediate value theorem we know that the single real root of $P'$ belongs to $\mathbb{R}^-$. As a consequence, $\forall a \in \mathbb{R}^+, P'(a) > 0$. Therefore, we can conclude that the positive root of $P$ is a minimum.

Going further, since $\forall a \in \mathbb{R}^+, P'(a) > 0$, $P$ is increasing and cannot have more than one root in $\mathbb{R}^+$. Remark that $P(0) < 0$ and $\lim_{a\to+\infty} P(a) = +\infty$, hence $P$ has a unique positive root in $\mathbb{R}^+$ by the intermediate value theorem. As shown by Lemma D.1, this root is solution of (D.1) and gives us the optimal value for $a$ in the approximation (6.6). □

We can now prove Theorem 6.1.

*Proof.* We consider

$$S = \{\boldsymbol{x} : \boldsymbol{x} \in \left[\sqrt{v_-^{(1)}}, \sqrt{v_+^{(1)}}\right] \times \cdots \times \left[\sqrt{v_-^{(n)}}, \sqrt{v_+^{(n)}}\right], \\ v_- \leq ||\boldsymbol{x}||_2^2 \leq v_+\}$$

and we need to prove that $\forall \boldsymbol{x} \in S, a||\boldsymbol{x}||_2^2 + \frac{1}{4a} \leq ||\boldsymbol{x}||_1$. This is equivalent to finding $a$ so that $-a^2||\boldsymbol{x}||_2^2 + a||\boldsymbol{x}||_1 - \frac{1}{4} \geq 0$. As a matter of fact, if it exists a fixed $a$ that satisfies the inequality $\forall \boldsymbol{x} \in S$, it is the one computed by the Proposition 6.2, since the approximation is the optimal linear overestimation of $||\boldsymbol{x}||_2$.

We know that $-a^2||\boldsymbol{x}||_2^2 + a||\boldsymbol{x}||_1 - \frac{1}{4}$ is positive between its roots, which are

$$a_1(\boldsymbol{x}) = \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2}}{2||\boldsymbol{x}||_2^2}$$
$$a_2(\boldsymbol{x}) = \frac{||\boldsymbol{x}||_1 + \sqrt{||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2}}{2||\boldsymbol{x}||_2^2}.$$

In order to ensure the existence of $a$ satisfying the equation for all the elements of $S$, we need to make sure that

$$\max_{\boldsymbol{x}\in S} a_1(\boldsymbol{x}) \leq \min_{\boldsymbol{x}\in S} a_2(\boldsymbol{x}) \tag{D.3}$$

To ease the maximization of $a_1(\boldsymbol{x})$, let us break down the maximization by considering $\max_{\boldsymbol{x}\in S} \max_{y\in[v_-, ||\boldsymbol{x}||_1^2 - \delta_-^2]} \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y}$, with $\delta_-^2 =$

$||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j\neq i}}^n x_i x_j$. A trivial study of the variations of the expression shows that

$$\begin{aligned}\max_{y\in[v_-,||\boldsymbol{x}||_1^2-\delta_-^2]} \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y} &= \frac{||\boldsymbol{x}||_1 - \delta_-}{2(||\boldsymbol{x}||_1^2 - \delta_-^2)} \\ &= \frac{1}{2(||\boldsymbol{x}||_1 + \delta_-)}\end{aligned}$$

Therefore, $\max_{\boldsymbol{x}\in S} \frac{1}{2(||\boldsymbol{x}||_1+\delta_-)} \leq \frac{1}{2(\sqrt{v_-}+\delta_-)}$ and $\delta_-^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j\neq i}}^n \sqrt{v_-^{(i)} v_-^{(j)}}$.

Similarly, we study the variation of $\frac{||\boldsymbol{x}||_1+\sqrt{||\boldsymbol{x}||_1^2-y}}{2y}$ for $y \in [v_-, ||\boldsymbol{x}||_1^2 - \delta_+^2]$. It is trivial to show that

$$\begin{aligned}\min_{y\in[v_-,||\boldsymbol{x}||_1^2-\delta_+^2]} \frac{||\boldsymbol{x}||_1 + \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y} &= \frac{||\boldsymbol{x}||_1 + \delta_+}{2(||\boldsymbol{x}||_1^2 - \delta_+^2)} \\ &= \frac{1}{2(||\boldsymbol{x}||_1 - \delta_+)}\end{aligned}$$

Therefore, $\min_{\boldsymbol{x}\in S} \frac{1}{2(||\boldsymbol{x}||_1-\delta_+)} \geq \frac{1}{2(\sqrt{v_+}-\delta_+)}$, with $\delta_+ \geq \delta_-$. We can now rewrite our criterion (D.3) as $\sqrt{v_-} + \delta_- \geq \sqrt{v_+} - \delta_+$, and we replace $\delta_+$ by $\delta_-$ for the sake of simplicity. This leads to the desired criterion expressed only with the variance bounds

$$\sqrt{v_+} \leq \sqrt{v_-} + 2\delta_- \tag{D.4}$$

Therefore, whenever (D.4) holds, $a||\boldsymbol{x}||_2^2 + \frac{1}{4a} \leq ||\boldsymbol{x}||_1$, which is the desired result. □

# E

# PROOFS FOR THE EARLY-STOPPING GUARANTEES OF DUMBO

This appendix contains the proofs of Proposition 6.3 and Theorem 6.2. Let us start by proving Proposition 6.3.

*Proof.* We want to show that $\varphi_t^{(i)} = \mu_t^{(i)} + a\beta_t^{\frac{1}{2}} {\sigma_t^{(i)}}^2$ is Lipschitz continuous. This is true if $||\nabla\varphi_t^{(i)}||_2$ is bounded. Replacing $\mu_t^{(i)}$ and ${\sigma_t^{(i)}}^2$ by their expressions and differentiating shows that we need to bound

$$\begin{aligned} ||\nabla\varphi_t^{(i)}||_2 &= ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top} \boldsymbol{K}^{-1} \left( \boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)} \right) ||_2 \\ &\leq ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2 ||\boldsymbol{K}^{-1}||_2 ||\boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)}||_2 \end{aligned} \tag{E.1}$$

Let us bound properly all the terms in (E.1). By Assumption 6.3, we know that $\forall j \in [1,t], v_-^{(i)} \leq k^{(i)}(\boldsymbol{x}, \boldsymbol{x}^j) \leq v_+^{(i)}$. Similarly, $y_t^- \leq y_j \leq y_t^+$, with $y_t^- = \min_{j\in[1,t]} y_j$ and $y_t^+ = \max_{j\in[1,t]} y_j$. Therefore, denoting $M_t^{(i)} = \max\left( |y_t^+ - 2a\beta_t^{\frac{1}{2}} v_-^{(i)}|, |y_t^- - 2a\beta_t^{\frac{1}{2}} v_+^{(i)}| \right)$, we have $||\boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)}||_2 \leq \sqrt{t} M_t^{(i)}$. Moreover, by the Raleigh–Ritz theorem, it is known that the spectral norm of a symmetric positive semi-definite matrix coincide with its spectral radius (*i.e.* its largest eigenvalue). Therefore, $||\boldsymbol{K}^{-1}||_2 = \rho\left(\boldsymbol{K}^{-1}\right)$. Eventually, we bound $||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2$ using the definition of the spectral norm

$$\begin{aligned} ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2 &= \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top} \boldsymbol{z}||_2}{||\boldsymbol{z}||_2} \\ &\leq \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{\sum_{j=1}^t ||\nabla \boldsymbol{k}^{(i)}(\boldsymbol{x}, \boldsymbol{x}^j)||_2 |z_j|}{\frac{1}{\sqrt{t}} \sum_{j=1}^t |z_j|} \\ &\leq \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{\sqrt{t} L\left(k^{(i)}\right) \sum_{j=1}^t |z_j|}{\sum_{j=1}^t |z_j|} \\ &= \sqrt{t} L\left(k^{(i)}\right) \end{aligned} \tag{E.2}$$

where (E.2) follows from $k^{(i)}$ being Lipschitz continuous with Lipschitz constant $L\left(k^{(i)}\right)$ according to Assumption 6.4.

Combining all these upper bounds, we have an upper bound for the gradient of $\varphi_t^{(i)}$

$$||\nabla\varphi_t^{(i)}||_2 \leq tL\left(k^{(i)}\right) \rho\left(\boldsymbol{K}^{-1}\right) M_t^{(i)} \tag{E.3}$$

with $M_t^{(i)} = \max\left(|y_{i,t}^+ - 2a\beta_t^{\frac{1}{2}} v_-^{(i)}|, |y_{i,t}^- - 2a\beta_t^{\frac{1}{2}} v_+^{(i)}|\right)$, which is the desired result. □

We now prove Theoreom 6.2, which claims that a solution from an early-stopped version of DuMBO is still optimal in a weaker sense. Note that this proof uses ideas from the proof of Theorem 4.1 in Appendix A.

*Proof.* Denoting $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and having $\left\{\boldsymbol{x}_{k+1}^{(i)}\right\}_{i\in[1,n]}$, let us try to find a closed form for the upper bound of $G = |\sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^*) - \sum_{i=1}^n \varphi_t^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{V}_i})|$, before building $\tilde{\boldsymbol{x}}$ that minimizes this upper bound. Since Proposition 6.3 holds, each $\varphi_t^{(i)}$ is Lipschitz and we have

$$\begin{aligned} G &\leq \sum_{i=1}^n |\varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^*) - \varphi_t^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{V}_i})| \\ &\leq \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}_{\mathcal{V}_i}^* - \tilde{\boldsymbol{x}}_{\mathcal{V}_i}||_2 \\ &\leq \sum_{i=1}^n L(\varphi_t^{(i)}) \left(||\boldsymbol{x}_{\mathcal{V}_i}^* - \boldsymbol{x}_{k+1}^{(i)}||_2 + ||\boldsymbol{x}_{k+1}^{(i)} - \tilde{\boldsymbol{x}}_{\mathcal{V}_i}||_2\right). \end{aligned} \tag{E.4}$$

(E.4) is ensured by the triangle inequality. Note that, as far as we know, there is no tighter upper bound than (E.4) for $G$. From (E.4), we see that $\tilde{\boldsymbol{x}}$ must satisfy $\tilde{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}_{k+1}^{(i)} - \boldsymbol{x}_{\mathcal{V}_i}||_2$ to minimize the upper bound. This is equivalent to finding $\arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}) = \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}_{k+1}^{(i)} - \boldsymbol{x}_{\mathcal{V}_i}||_2^2$. Developing this expression, we have

$$\Psi(\tilde{\boldsymbol{x}}) = \sum_{i=1}^{n_\xi} \sum_{j\in\mathcal{V}_i} L(\varphi_t^{(i)}) \left(x_{k+1,j}^{(i)} - \tilde{x}_j\right)^2. \tag{E.5}$$

Differentiating (E.5) by $\tilde{x}_j$, we get

$$\frac{\partial\Psi}{\partial\tilde{x}_j} = 2\sum_{i\in\mathcal{F}_j} L(\varphi_t^{(i)}) \left(\tilde{x}_j - x_{k+1,j}^{(i)}\right)$$

Solving $\frac{\partial\Psi}{\partial\tilde{x}_j} = 0$ with the Hessian $H(\Psi)(\tilde{\boldsymbol{x}})$ positive definite is trivial and leads to the minimum

$$\tilde{\boldsymbol{x}} = \left(\frac{1}{\sum_{j\in\mathcal{F}_i} L(\varphi_t^{(j)})} \sum_{j\in\mathcal{F}_i} L(\varphi_t^{(j)}) x_{k+1,i}^{(j)}\right)_{i\in[1,d]} \tag{E.6}$$

which is the desired result. Note that if $L(\varphi_t^{(j)})$ cannot be computed explicitly in (E.6), we can upper bound $L(\varphi_t^{(j)})$ by $\max_{j\in[1,n]} L(\varphi_t^{(j)})$

which then cancels out in the numerator and denominator of (E.6) to become

$$\tilde{\boldsymbol{x}} = \left( \frac{1}{|\mathcal{F}_i|} \sum_{j \in \mathcal{F}_i} x_{k+1,i}^{(j)} \right)_{i \in [1,d]} . \tag{E.7}$$

Therefore, (E.7) can be used to early-stop DuMBO. Note that, as stated in the main text, the aggregation E.7 is exactly $\bar{\boldsymbol{x}}_{k+1}$ as defined in (6.20). □

# F

# PROOF OF THE ASYMPTOTICAL OPTIMALITY OF DUMBO

In this appendix, we discuss the asymptotic optimality of DuMBO and provide the proof for Theorem 6.3. We start by proving the following inequality connecting $f(\boldsymbol{x})$ with the posterior mean and variance of $f$.

**Lemma F.1.** *Pick $\delta \in (0,1)$ and let $\beta_t = 2\log\left(\frac{|\mathcal{C}|\pi^2 t^2}{6\delta}\right)$. Then, with probability at least $1-\delta$,*

$$|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right) \tag{F.1}$$

*for all $\boldsymbol{x} \in \mathcal{C}$ and $t \in \mathbb{N}$, $\mu_t(\boldsymbol{x})$ and $\sigma_t^2(\boldsymbol{x})$ defined through our proposed decomposition in Proposition 6.1.*

*Proof.* For all $\boldsymbol{x} \in \mathcal{C}$ and $t \in \mathbb{N}$, we have $f(\boldsymbol{x}) \sim \mathcal{N}\left(\mu_t(\boldsymbol{x}), \sigma_t^2(\boldsymbol{x})\right)$. Defining $s_t(\boldsymbol{x}) = \frac{f(\boldsymbol{x})-\mu_t(\boldsymbol{x})}{\sigma_t(\boldsymbol{x})}$, we know that $s_t(\boldsymbol{x}) \sim \mathcal{N}(0,1)$. Therefore,

$$\begin{aligned}
\Pr\left(|s_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}}\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \\
\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}}\sigma_t(\boldsymbol{x})\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \\
\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \qquad \text{(F.2)}
\end{aligned}$$

where (F.2) follows from Proposition 6.2. The inequality (F.2) is true for a single pair $(t, \boldsymbol{x})$. Applying the union bound for all pairs in $\mathbb{N} \times \mathcal{C}$, we have $\forall t \in \mathbb{N}, \forall \boldsymbol{x} \in \mathcal{C}$

$$\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) \geq 1 - |\mathcal{C}| \sum_{t=1}^{+\infty} e^{-\frac{\beta_t}{2}}. \tag{F.3}$$

Pick $\delta \in (0,1)$ and let $\beta_t = 2\log\left(\frac{|\mathcal{C}|\pi^2 t^2}{6\delta}\right)$. Then,

$$\begin{aligned}
|\mathcal{C}| \sum_{t=1}^{+\infty} e^{-\frac{\beta_t}{2}} &= |\mathcal{C}| \sum_{t=1}^{+\infty} e^{-\log\left(\frac{|\mathcal{C}|\pi^2 t^2}{6\delta}\right)} \\
&= \frac{6\delta}{\pi^2} \sum_{t=1}^{+\infty} \frac{1}{t^2} \\
&= \delta
\end{aligned}$$

Therefore, (F.3) becomes

$$\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}}\left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) \geq 1 - \delta \tag{F.4}$$

which is the desired result. □

We are now ready to bound the immediate regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$ and prove Theorem 6.3.

*Proof.* By definition, $\boldsymbol{x}^t = \arg\max_{\boldsymbol{x}} \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$. As a consequence, $\sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t) \geq \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^*)$. Developing the left term of the inequality with the expression of $\varphi_t^{(i)}$ and adding $\frac{\beta_t^{\frac{1}{2}}}{4a}$ on both sides, we have

$$\begin{aligned}\frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t) + a\beta_t^{\frac{1}{2}}\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t)^2 &\geq \frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i'}^*) \\ &\geq f(\boldsymbol{x}^*)\end{aligned} \tag{F.5}$$

with (F.5) following from (F.4) with high probability. We can now bound the immediate regret $r_t$

$$\begin{aligned} r_t &= f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t) \\ &\leq \frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t) + a\beta_t^{\frac{1}{2}}\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t)^2 - f(\boldsymbol{x}^t) \\ &= \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t) - f(\boldsymbol{x}^t) + \beta_t^{\frac{1}{2}}\left(a\sum_{i=1}^n \sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}^t)^2 + \frac{1}{4a}\right) \\ &= \mu_t(\boldsymbol{x}^t) - f(\boldsymbol{x}^t) + \beta_t^{\frac{1}{2}}\left(a\sigma_t^2(\boldsymbol{x}^t) + \frac{1}{4a}\right) \end{aligned} \tag{F.6}$$

Combining the conclusion of Lemma F.1 with (F.6), we get

$$\Pr\left(r_t \leq 2\beta_t^{\frac{1}{2}}\left(a\sigma_t^2(\boldsymbol{x}^t) + \frac{1}{4a}\right)\right) \geq 1 - \delta \tag{F.7}$$

which is the desired result. □

# G

# SYNTHETIC FUNCTIONS AND REAL-WORLD EXPERIMENTS FOR THE DUMBO EVALUATION

In this appendix, we describe the synthetic functions constituting our benchmark in Section 6.5.1, before discussing the real-world experiments conducted in Section 6.5.2.

**Six-Hump Camel.** The Six-Hump Camel function is a 2-dimensional function defined by

$$f(x_1, x_2) = \left(-4 + 2.1x_1^2 - \frac{x_1^4}{3}\right) x_1^2 - x_1 x_2 + \left(4 - 4x_2^2\right) x_2^2. \quad \text{(G.1)}$$

It is composed of $n = 3$ factors, with a Maximum Factor Size (MFS) $\bar{d} = 2$. In our experiment, we optimize it on the rectangle $\mathcal{C} = [-3, 3] \times [-2, 2]$. It has 6 local maxima, two of which are global with $f(\boldsymbol{x}^*) = 1.0316$.

**Hartmann.** The Hartmann function is a 6-dimensional function defined by

$$f(\boldsymbol{x}) = \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij} \left(x_j - P_{ij}\right)^2\right), \quad \text{(G.2)}$$

with $\boldsymbol{\alpha} = (\alpha_i)_{i \in [1,4]}$, $\boldsymbol{A} = (A_{ij})_{(i,j) \in [1,4] \times [1,6]}$ and $\boldsymbol{P} = (P_{ij})_{(i,j) \in [1,4] \times [1,6]}$ given as constants.

It is composed of $n = 4$ factors, with a MFS $\bar{d} = 6$. In our experiment, we optimize it on the hypercube $\mathcal{C} = [0, 1]^6$. It has 6 local maxima and a global maximum with $f(\boldsymbol{x}^*) = 10.5364$.

**Powell.** The Powell function is a function of an arbitrary number $d = 4k$ of dimensions, defined by

$$\begin{aligned} f(\boldsymbol{x}) = -\sum_{i=1}^{d/4} &(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 \\ &+ (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4. \end{aligned} \quad \text{(G.3)}$$

We chose to set $k = 6$, so that the resulting Powell function lives in a $d = 24$ dimensional space. It is composed of $n = 6$ factors, with a MFS $\bar{d} = 4$. In our experiment, we optimize it on the hypercube $\mathcal{C} = [-4, 5]^{24}$. It has a global maximum at $\boldsymbol{x}^* = \boldsymbol{0}$, with $f(\boldsymbol{x}^*) = 0$.

**Rastrigin.** The Rastrigin function is a function of an arbitrary number $d$ of dimensions, defined by

$$f(\boldsymbol{x}) = -10d - \sum_{i=1}^{d} x_i^2 - 10\cos(2\pi x_i)\,. \tag{G.4}$$

We chose to set $d = 100$. We also chose to aggregate some factors to make the optimization problem harder. The resulting Rastrigin function is composed of $n = 20$ factors, with a MFS $\bar{d} = 5$. In our experiment, we optimize it on the hypercube $\mathcal{C} = [-5.12, 5.12]^{100}$. It has multiple, regularly distributed local maxima, with a global maximum at $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$.

We now describe in details the real-world experiments conducted in Section 6.5.2.

**Cosmological Constants.** The cosmological constants problem consists in fine-tuning an astrophysics tool to optimize the likelihood of some observed data. We chose to compute the likelihood of the galaxy clustering [156] from the Data Release 9 (DR9) CMASS sample of the SDSS-III Baryon Oscillation Spectroscopic Survey (BOSS). To compute the likelihood, we instrumented the cosmological parameter estimation code CosmoSIS [157].[51]

We considered nine cosmological constants in our optimization task, going from the Hubble's constant to the mass of the neutrinos. If a Bayesian Optimization (BO) algorithm provided a set of inconsistent cosmological constants, a likelihood of $y = -60$ was returned.

Note that similar experiments were considered in other works, such as [117, 125]. However, they were conducted on another, older dataset, with a deprecated NASA simulator.[52] This makes the conducted experiments painful to reproduce on a modern computer. Hopefully, CosmoSIS is well documented and easier to install and instrument, so we conducted our experiment with CosmoSIS to make it easier to replicate.

**Shannon Capacity of a Wireless Local Area Network (WLAN).** The Shannon capacity [110] sets a theoretical upper bound on the throughput of a wireless communication, depending on the Signal to Interference plus Noise Ratio (SINR) of the communication. Denoting by $S_{i,j}$ the SINR between two wireless devices $i$ and $j$ communicating on a radio channel of bandwidth $W$ (in Hz), the Shannon capacity $C(S_{i,j})$ (in bits) is defined by

$$C(S_{i,j}) = W \log_2(1 + S_{i,j}) \tag{G.5}$$

In this problem, we consider a Wireless Local Area Network (WLAN) with end-users associated to nodes streaming a continuous, large amount of data. The WLAN topology is depicted in Figure G.1. It is

[156] Chuang et al. (2013), "The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: single-probe measurements and the strong power of f(z) 8(z) on constraining dark energy"

[157] Zuntz et al. (2015), "CosmoSIS: Modular cosmological parameter estimation"

[51] https://cosmosis.readthedocs.io/en/latest/reference/standard_library/BOSS.html

[52] https://lambda.gsfc.nasa.gov/toolbox/lrgdr/

populated with 36 end-users, each one associated to one of the 12 depicted nodes. Note that each node is within the radio range of at least two other nodes. This creates interference and, consequently, reduces the SINRs between nodes and end-users.

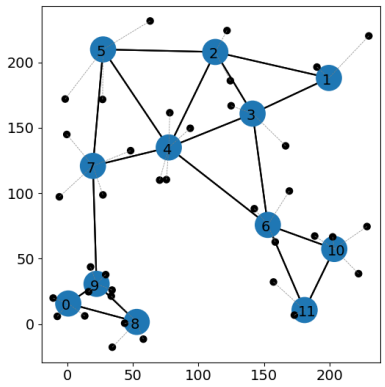



Figure G.1: The WLAN topology considered in the Shannon capacity optimization experiment. The end-users are depicted as black dots, the nodes as numbered blue circles and the associations between end-users and nodes as thin gray lines. Two nodes are connected with a black line if they are within the radio range of each other.

Each node has an adjustable transmission power $x_i \in [10^{0.1}, 10^{2.5}]$ in mW (milliwatts). This task consists in jointly optimizing the Shannon capacity (G.5) of each pair of (node, associated end-user) by tuning the transmission power of the nodes. That is, the objective function $f$ is a 12-dimensional function defined by

$$f(\boldsymbol{x}) = \sum_{i=1}^{12} \sum_{j \in \mathcal{N}_i} C(S_{i,j}), \tag{G.6}$$

with $\mathcal{N}_i$ the set of end-users associated to node $i$.

A difficult trade-off needs to be found because a node cannot simply use the maximum transmission power as this would cause a lot of interference for the neighboring nodes. Given a configuration $\boldsymbol{x} \in \mathcal{C} = [10^{0.1}, 10^{2.5}]^{12}$, the SINRs are provided by the well-recognized network simulator ns-3 [31] that reliably reproduces the WLAN internal dynamics. The additive decomposition comprises $n = 12$ factors, with a MFS of $\bar{d} = 5$, obtained by making the reasonable assumption that only the neighboring nodes of node $i$ (*i.e.* those within the radio range of node $i$) are creating interference for the communications of node $i$.

**Rover Trajectory Planning.** This problem was also considered by [117, 158]. The goal is to optimize the trajectory of a rover from a starting point $\boldsymbol{s} \in [0,1]^2$ to a target $\boldsymbol{t} \in [0,1]^2$, over a rough terrain.

*[158] Wang et al. (2018), "Batched large-scale Bayesian optimization in high-dimensional spaces"*

The trajectory is defined by a vector of $d = 60$ dimensions, reshaped into 30 2-d points in $[0,1]$. A B-spline is fitted to these 30

points, determining the trajectory of the rover. The objective function to optimize is

$$f(\boldsymbol{x}) = -c(\boldsymbol{x}) - 10(||\boldsymbol{x}_{0,1} - \boldsymbol{s}||_1 + ||\boldsymbol{x}_{59,60} - \boldsymbol{t}||_1), \tag{G.7}$$

with $c(\boldsymbol{x})$ the cost of the trajectory, obtained by integrating the terrain roughness function over the B-spline, and the two $L_1$-norms serving as incentives to start the trajectory near $\boldsymbol{s}$, and to end it near $\boldsymbol{t}$.

## COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

*Final Version* as of August 17, 2023 (`classicthesis` version 4.2).