
Mobility and Resource Management in O-RAN with Online Meta-Learning
Michail Kalntis

Mobility and Resource Management in O-RAN with Online Meta-Learning

Mobility and Resource Management in O-RAN with Online Meta-Learning

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus
Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Thursday, 19 March 2026, 10:00

by

Michail KALNTIS

This dissertation has been approved by the (co)promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir. F.A. Kuipers,	Delft University of Technology, promotor
Dr. G. Iosifidis,	Delft University of Technology, copromotor
Independent members:	
Prof.dr.ir. G.J.T. Leus,	Delft University of Technology
Prof.dr. A. Banchs,	University Charles III of Madrid, Spain
Prof.dr. N. Meratnia,	Eindhoven University of Technology
Dr. G. Neglia,	National Institute for Research in Digital Science and Technology, France
Prof.dr. K.G. Langendoen,	Delft University of Technology, reserve member

Keywords: Network Optimization, Datasets, Handovers, Resource Allocation

Printed by: Gildeprint B.V.

ISBN 000-00-0000-000-0

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Contents

Summary

Modern cellular networks are tasked to deliver guaranteed performance for a wide array of users that increasingly demand higher throughput and reliability, lower latency, seamless connectivity, ubiquitous coverage, energy efficiency, fairness, and security, to name a few. To meet these demands, networks are becoming increasingly complex, combining diverse deployments and multiple radio access technologies that are envisioned to extend beyond 5G. At the same time, the resources (e.g., spectrum, energy, capacity) needed to serve all users are limited and expensive; and control decisions, such as mobility and resource allocation, often require trading throughput with other user-perceived performance metrics such as lower delays, signaling/communication costs, and failure risks.

In these environments where traffic patterns change rapidly, signal qualities fluctuate unpredictably and cost/availability of resources is uncertain, it becomes apparent that static control rules and legacy mechanisms built on heuristics are poorly suited. In this context, the evolution of mobile network architectures, particularly the emergence of open Radio Access Network (RAN), represents a necessary and enabling change. The O-RAN Alliance, for example, is a global initiative aimed at softwarizing and standardizing RANs to improve their performance, reduce costs, and lower the entry barrier for a broader vendor ecosystem. It enables scalable, data-driven control loops that can be implemented centrally by intelligent *controllers* and enforced at different time scales, namely, near-real-time (near-RT) and non-real-time (non-RT). In this way, it becomes possible to embed online learning solutions in the RAN itself, where data are collected and used for *effective* and *robust* learning.

This dissertation responds to these challenges by developing *online (meta-) learning* algorithms for two coupled control layers in O-RAN: (*i*) mobility management (via user-cell association and traditional/conditional handovers) and (*ii*) resource allocation (via threshold, non-RT policies) for virtualized base stations. Online learning provides a principled way to make sequential decisions under uncertainty, and online meta-learning enables the system to combine various (online) learners, each tailored for different environments, achieving both effectiveness, which translates to high performance under all conditions, as well as robustness, which ensures this high performance without knowing precisely the conditions. All proposed methods deliver operation guarantees under all conditions (from stationary to even adversarial dynamics), as well as practical gains on country-scale operator data and O-RAN-compatible testbeds.

Samenvatting

Moderne mobiele netwerken hebben de taak om gegarandeerde prestaties te leveren voor een brede groep gebruikers die in toenemende mate hogere doorvoersnelheid en betrouwbaarheid, lagere vertraging (latency), naadloze connectiviteit, alomtegenwoordige dekking, energie-efficiëntie, eerlijkheid en veiligheid verlangen, om er maar enkele te noemen. Om aan deze eisen te voldoen, worden netwerken steeds complexer, doordat ze diverse uitrolscenario's en meerdere radio-technologieën combineren. Tegelijkertijd zijn de middelen ("resources", zoals spectrum, energie en capaciteit) die nodig zijn om alle gebruikers te bedienen beperkt en kostbaar; en vereisen controlebeslissingen, zoals mobiliteitsbeheer en toewijzing van middelen, vaak een afweging tussen doorvoersnelheid en andere prestatie-indicatoren, zoals lagere vertragingen, signalerings- en communicatiekosten en risico op uitval.

In deze omgevingen, waar verkeerspatronen snel veranderen, signaalkwaliteiten onvoorspelbaar fluctueren en de kosten/beschikbaarheid van middelen onzeker zijn, wordt duidelijk dat statische controle regels en traditionele mechanismen gebaseerd op heuristieken slecht geschikt zijn. In deze context vormt de evolutie van mobiele netwerkarchitecturen, in het bijzonder de opkomst van open Radio Access Network (RAN), een noodzakelijke en faciliterende verandering. De O-RAN Alliance is bijvoorbeeld een wereldwijde samenwerking die zich richt op het softwarematig maken en standaardiseren van RANs om hun prestaties te verbeteren, kosten te verlagen en de toetredingsdrempel voor een breder ecosysteem van leveranciers te verlagen. Het maakt schaalbare, datagedreven controlemechanismen mogelijk die centraal kunnen worden geïmplementeerd door intelligente *controllers* en op verschillende tijdschalen kunnen worden toegepast, namelijk near-real-time en non-real-time (near/non-RT). Op deze manier wordt het mogelijk om online leermodules in de RAN zelf te integreren, waar data worden verzameld en gebruikt voor *effectief* en *robuust* leren.

Dit proefschrift speelt in op deze uitdagingen door *online (meta-)leer* algoritmen te ontwikkelen voor twee gekoppelde controlelagen in O-RAN: (*i*) mobiliteitsbeheer (via gebruiker-celassociatie en traditionele/conditionele handovers) en (*ii*) toewijzing van middelen (via drempelgebaseerde non-RT beleidsregels) voor gevirtualiseerde basisstations. Online leren biedt een principiële manier om opeenvolgende beslissingen te nemen onder onzekerheid, en online meta-leren stelt het systeem in staat om verschillende (online) leeralgoritmen te combineren, elk afgestemd op verschillende omgevingen. Dit leidt tot zowel effectiviteit, wat zich vertaalt in hoge prestaties onder alle omstandigheden, als robuustheid, wat ervoor zorgt dat deze hoge prestaties behaald worden zonder de omstandigheden precies te hoeven kennen. Alle voorgestelde methoden bieden operationele garanties onder alle omstandigheden (van stationaire tot zelfs vijandige dynamieken), evenals praktische voordelen op gegevens op landelijk schaalniveau van operators en op O-RAN-compatibele testomgevingen.

List of Abbreviations

2G	Second Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5G-NR	5G New Radio
5G-NSA	5G-Non-Standalone
5G-SA	5G-Standalone
6G	Sixth Generation
AI	Artificial Intelligence
APN	Access Point Name
BBU	Baseband Unit
BS	Base Station
CHO	Conditional Handover
CN	Core Network
CP	Control Plane
CQI	Channel Quality Indicator
C-RAN	Cloud Radio Access Network
CS	Circuit Switched
DL	Downlink
ECDF	Empirical Cumulative Distribution Function
eNB	Evolved Node B
EPC	Evolved Packet Core
FR2	Frequency Range 2
gNB	Next Generation Node B
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSMA	Global System for Mobile Communications Association
HO	Handover
HOF	Handover Failure
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet-of-Things
KPI	Key Performance Indicator
LTE	Long Term Evolution
M2M	Machine-to-Machine
MANO	Management and Orchestration

MCS	Modulation and Coding Scheme
MEC	Mobile Edge Computing
ML	Machine Learning
MME	Mobile Management Entity
MNO	Mobile Network Operator
MR	Measurement Report
MSC	Mobile Switching Center
Near-RT	Near-Real-Time
NextG	Next Generation
NFV	Network Functions Virtualization
Non-RT	Non-Real-Time
NR	New Radio
OAI	OpenAirInterface
OCO	Online Convex Optimization
O-CU	O-RAN Centralized Unit
O-DU	O-RAN Distributed Unit
O-RU	O-RAN Radio Unit
PRB	Physical Resource Block
PS	Packet Switched
QoE	Quality of Experience
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAT	Radio Access Technology
RIC	Radio Access Network Intelligent Controller
RL	Reinforcement Learning
RRC	Radio Resource Control
RRM	Radio Resource Management
RRU	Remote Radio Unit
RSRQ	Reference Signal Received Quality
RT	Real-Time
SCF	Small Cell Forum
SDN	Software-Defined Networks
SGSN	Serving GPRS Support Node
SGW	Serving Gateway
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service-Level Agreement
SNR	Signal-to-Noise Ratio
SOL	Smoothed Online Learning
TAC	Type Allocation Code
TAU	Tracking Area Update
THO	Traditional Handover
TIP	Telecom Infra Project
TTT	Time-To-Trigger
UE	User Equipment

UL	Uplink
UP	User Plane
URLLC	Ultra Reliable Low Latency Communications
vBS	Virtual Base Station
VNF	Virtualized Network Function
vRAN	Virtualized Radio Access Network

1

Introduction

Cellular communication networks have become a critical part of modern society. Mobile Network Operators (MNOs) install cell sites / base stations (consisting of cells) in various locations to provide wireless service coverage through radio frequency (RF) signals to billions of heterogeneous users and devices, such as smartphones, tablets, vehicles, IoT devices, and other connected equipment, as well as verticals with distinct requirements. From streaming high-definition video to sending messages and supporting industrial automation, these diverse requirements range from high throughput, ultra-low latency, excellent reliability, and scalability; or any combination of them.

Behind the scenes, this *network system* must *continuously decide*, even in a matter of a few milliseconds, how to serve all requests in this complex arena. Naturally, if the resources of the network system were unlimited, such decisions would be simple and would fulfill every user's and vertical's demand without constraint. However, in practice, resources such as computational capacity and bandwidth are inherently limited, and trade-offs inevitably arise.

These multifaceted conflicts emerge across the heterogeneous users, where one's service must be prioritized over another's; across applications and verticals, where the requirements of, for example, mobile broadband and mission-critical IoT diverge; and across objectives, where maximizing performance often comes at the cost of higher energy consumption or signaling overhead. With multiple stakeholders (users, operators, vendors, and regulators) involved, the trade-offs are not only technical but also economic and operational, making mobility and resource management one of the most intricate challenges in modern and future networks.

Taking these considerations into account, it becomes apparent that the long-standing reliance on static rules or heuristics for network control, often based solely on expected usage patterns, cannot adequately address the complexity of today's networks. Modern cellular networks operate in *environments* (i.e., *conditions*) that can be highly unpredictable: signal conditions fluctuate rapidly due to fast-moving users and high radio frequencies, together with traffic loads from a plethora of users

and verticals. Crucially, in the context of this thesis, environments/conditions refer not only to the traditional radio parameters (e.g., interference, signal condition, load) but also to factors such as resource availability, cost of resources, signaling overheads, and, in general, any source of perturbation that introduces uncertainty in the network's operation. Nevertheless, the network is expected to deliver *guaranteed* performance under all these conditions.

To keep pace with this complexity, the mobile infrastructure is undergoing a major transformation. A new architecture known as the *open Radio Access Network* (RAN), and its embodiment through *O-RAN* Alliance, is revolutionizing how networks are built and managed. In O-RAN, control functions are disaggregated, software-driven, and programmable. This shift makes it possible to embed learning algorithms directly into the network, so that decisions about users and cell sites can be made centrally, ensuring more unified and informed decisions, based on data (and not just pre-defined rules) and in different *time scales*, ranging from a few milliseconds in (*near*) *real-time* to a couple of seconds or hours in *non-real-time*.

This dissertation explores how we can design such learning algorithms to tackle two interrelated problems that arise in the control of O-RAN systems:

1. **Mobility management via user-cell associations and handovers:** When users move and have active data connections, the network must decide which base station (or cell) should serve them at each point in time. Associating or keeping a user to a cell with a low signal strength would lead to poor throughput and potentially lack of connectivity, while frequent changes in association, causing traditional handovers (THOs, or HOs in general) may lead to excessive signaling and unneeded service interruptions. Selecting the "right" cell for each user, when network conditions change (e.g., as users move), is of paramount importance and requires balancing the effective throughput of the users with the (sometimes prolonged) delays associated with the HOs.

   Apart from the traditional HO mechanism, 3GPP recently introduced Conditional HOs (CHOs) to mitigate the inherent limitation of THO that lies in its reactive nature. The novel CHO scheme enables proactive cell reservations and user-driven execution, thus increasing the probability of HO success and reducing delays in the procedure; especially in dense deployments and high-frequency bands. However, they introduce new challenges, such as the over-reservation of cells' resources and the signaling/communication overhead needed to materialize these reservations.

2. **Resource allocation for virtualized base stations in non-real-time:** Modern O-RAN systems rely on virtualized base stations (vBSs), which offer unprecedented flexibility due to their software-defined nature. However, this flexibility introduces new challenges: performance can become less predictable and energy consumption more volatile, particularly when vBSs run on general-purpose computing platforms.

   To meet these demands, it becomes essential to design intelligent resource allocation strategies that can guide how vBSs operate under different environments. Unlike traditional base stations, vBSs can be centrally configured

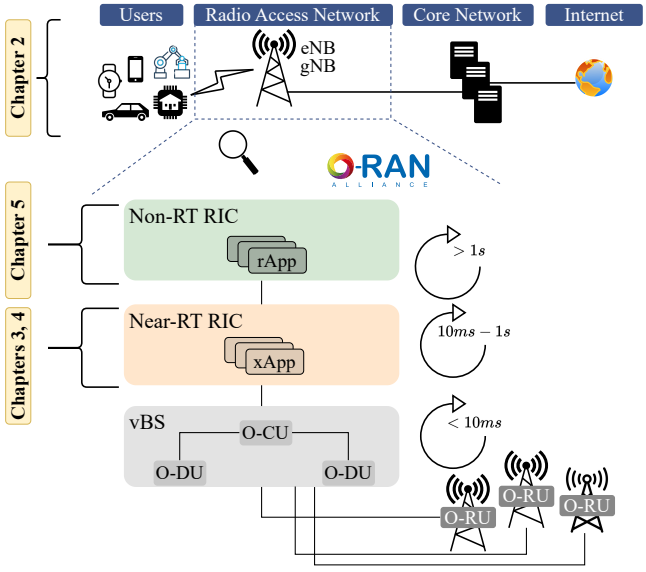


Figure 1.1: Top: heterogeneous users connect via RAN to the CN and the internet. Bottom: disaggregation of RAN to the non- and near-RT RIC, together with the main components of the vBS. Left: brief placement of each chapter's focus in (O-)RAN architecture.

via the O-RAN control architecture, which supports the use of non-real-time (non-RT) policies that influence how real-time schedulers allocate their scarce resources to individual users.

These two problems are tightly linked: associating each user with a cell (Problem 1) leads to allocating resources from this cell (Problem 2), affecting both users' and cells' performance (e.g., throughput and energy consumption). Extensive details on both problems are given in Section 1.4.

To solve them, this thesis proposes online (meta)-learning techniques that aim to be (*i*) *effective*, optimizing performance under a wide range of conditions, and (*ii*) *robust*, ensuring this performance without requiring access to accurate information about the conditions (e.g., mobility patterns), which might range from static to dynamic and potentially adversarial (i.e., picked by an adversary that tries to degrade the network operation). These two main features of our proposal fill a key gap in the literature and pave the way for the next generation of mobility and resource management solutions. All contributions in this thesis are extensively validated using a combination of real-world datasets and trace-driven simulations.

The following section traces the evolution of RAN, starting from rigid, hardwired modules to a virtualized, open, programmable, and versatile ecosystem. Then, we analyze why online (meta-)learning is essential in these systems and succinctly describe the main question that this thesis aspires to answer. Finally, we detail the problems addressed and the main contributions of this thesis.

1

1.1. Mobile Networks: From Legacy to Open RAN

Architecture. Given that RF signals propagate through the air, their quality is influenced by multiple factors, including distance, frequency band, interference, and physical obstacles. For that reason, cell sites are typically equipped with multiple antennas associated with *radio cells* (or *cells*) that serve users located in a bounded geographical area and support different radio access technologies (RATs), such as 2G, 3G, 4G, 5G, and 6G[1] [1].

The part of the infrastructure that operates these cells and manages the RF interface between users and the network is called RAN, illustrated in Figure 1.1(top). Apart from playing a central role in managing the air interface, the RAN acts as the first layer of control between the user and the core network (CN). The CN, in turn, handles centralized functions (e.g., authentication and mobility management) and provides the connection with external networks (e.g., the Internet or other operator networks), enabling end-to-end communications

Evolution of Radio Access Networks and Technologies. Early generations of RANs and their RATs were built as vertically integrated, single-vendor stacks. In second generation (2G) cellular networks, groups of Base Transceiver Stations (BTSs) reported over proprietary links to a central Base Station Controller (BSC), which performed tasks such as load balancing and handover control in a closed, vendor-specific environment [2]. The third generation (3G) upgrade preserved this tight coupling: NodeBs provided the air interface, while a Radio Network Controller (RNC) executed most radio resource management functions, again behind vendor-locked interfaces that limited interoperability [3]. The fourth generation (LTE, 4G) incorporated RNC tasks into the *eNodeB*, but still deployed the base station as a monolithic appliance whose hardware and firmware were tied to a single supplier, resulting in high costs and long innovation cycles [4].

With the advent of late-stage 4G, namely, LTE Advanced (Pro), Cloud-RAN (C-RAN) emerged [5]. In this model, the base station was split into a baseband unit (BBU) and a remote radio unit (RRU), where the former was deployed at a centralized location and the latter close to the cell sites. While C-RAN relied on proprietary hardware for BBU functionalities, the idea of virtualized RAN (vRAN) and virtualized base station (vBS) arose, where baseband functions are virtualized as Virtualizing Network Functions (VNFs) or containers, running on commercial off-the-shelf platforms. This architecture introduced the idea of partial virtualization, separating where processing occurs and enabling pooling gains.

Building on these efforts, LTE eNodeB evolved into gNodeB (gNB) in the 5G New Radio (NR), which itself was decomposed into three main components: the radio unit (RU, same as RRU), the distributed unit (DU), and the centralized unit (CU); the latter two composed the BBU in LTE.[2] Each of these splits implemented part of the protocol stack, allowing more finely-grained control, reduced latency,

[1] The Second, Third, Fourth, Fifth, and Sixth Generation networks, and their respective RATs, are henceforth referred to as 2G, 3G, 4G, 5G, and 6G, respectively.

[2] The CU is further divided into two logical components, the control and user plane (CP and UP), to enable the deployment of different functionalities in various parts of the networks. Another split considers alternatives for the physical layer functionalities in RU and DU [6].

and broader virtualization. Despite this architectural progress, implementations remained largely vendor-specific, constraining interoperability and innovation.

RAN Openness. In response to these limitations, a flurry of industrial and academic activities has focused on the development of *virtualized* and *open* RAN. Prominent initiatives include the Telecom Infra Project (TIP), which concentrates on real-world trials of open RAN systems and deployments in different operators' networks [7]; the O-RAN Software Community, a collaboration between the O-RAN Alliance and Linux Foundation, which develops open-source software for the RAN [8]; and the Small Cell Forum (SCF), which emphasizes open interfaces for an open RAN ecosystem with small cells [9].

Unlike these groups, the O-RAN Alliance, founded in 2018 by five major operators, has taken a leading role in formalizing open RAN specifications [6], [10]–[12]. The O-RAN Alliance has since grown rapidly to become a global community comprising more than 300 vendors, research and academic institutions, and operators, with the latter serving over 4.5 billion subscribers worldwide. The goal of this initiative is to define the next generation of open,[3] intelligent, and fully interoperable vRANs/vBSs by improving their performance, reducing their costs, and lowering the barrier for smaller vendors to enter the ecosystem, fostering innovation.

Importantly, O-RAN places intelligence at the center of its agenda, embedding AI/ML-driven automation into every layer of the architecture. This shift is reinforced by parallel initiatives around Mobile Edge Computing (MEC), which recognize the RAN not only as the point where wireless signals are processed, but also as a new opportunity to serve computation-intensive and latency-sensitive applications close to the resource-constrained mobile users [13].

O-RAN Intelligence. A key innovation for O-RAN is the introduction of flexible and programmable AI-native RAN Intelligent Controllers (RICs), enabling the implementation of custom control plane functions, regarding, for example, handovers or resource management decisions [14]. O-RAN envisions two RICs, one for the non real-time (RT), namely *non-RT RIC*, which involves large time scale operations with execution time more than 1 s, and one for the near RT (*near-RT RIC*), where operations range from 10 ms to 1 s.

This architecture enables exactly the capabilities needed by the algorithms developed in this thesis: hosting various intelligent mechanisms that continuously "learn" by acting, observing, and adapting; and all these in different time scales, depending on the concerned task. For instance, handover decisions are taken in millisecond-level granularity, and therefore, are implemented as *xApps* in near-RT RIC. On the other hand, longer-term policies, acting as thresholds for the actual, real-time resource allocation decisions (to avoid intervening with the proprietary RT schedulers), can be made in second-level granularity; and thus are deployed as *rApps* in the non-RT RIC. These actions, enabled through the RICs, provide an intelligent centralized control over multiple vBSs and users, making unified, and thus, more informed decisions, as shown in Figure 1.1.

[3] As shown in Figure 1.1, the gNB's RU, DU, and CU are termed in O-RAN terminology as O-RU, O-DU, and O-CU, respectively.

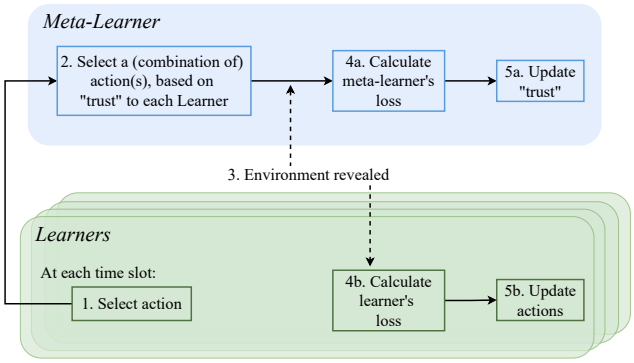


Figure 1.2: The protocol of online meta-learning at each time slot. In online learning (i.e., without a meta-learner), steps 2, 4a, and 5a are skipped.

1.2. Learning Frameworks

Online Learning. Online Learning is a Machine Learning (ML) paradigm that enables learning (i.e., updates predictions) while a stream of data becomes available sequentially. Allowing the model to dynamically adapt its decisions as new data arrive is necessary in situations where data patterns change over time, such as in volatile/adversarial network conditions where mobility patterns are unknown. This approach contrasts with the data-hungry batch (offline) learning, which requires the entire dataset (or a batch) for training to identify the best predictor(s); and changes in distribution or new data would necessitate retraining, a procedure that requires a significant amount of time. More specifically, this thesis leverages the theory and practicality of Online Convex Optimization (OCO), a powerful and mathematically structured subset of online learning. OCO was first introduced in [15] and has since been the workhorse of sequential decision-making in a wide range of problems [16].

In this setting, illustrated in the lower part of Figure 1.2, the following iterative (i.e., over *time slots*) process occurs: a *learner* (i.e., player, agent, or algorithm) repeatedly selects *actions* (i.e., *decisions*) under possibly different *conditions/environment* (e.g., signal qualities resource availability, signaling costs). At each time slot, the action must be chosen *before* the outcomes (i.e., how good or bad this action is) and conditions are revealed. Once the action (e.g., associating a user to a base station or allocating base station resources) is chosen, the actual conditions are observed and the learner incurs a *loss* from this outcome. The algorithm then uses the loss, conditions, and actions taken as feedback to improve its future decisions in a way that minimizes its average loss over time. These losses are not only unknown to the learner beforehand, but can even be chosen by an *adversary*, rendering the solution of this problem challenging.

While it may seem that the learner could observe the conditions before making a decision, in practice, there is a non-negligible delay between observing the conditions and processing them to devise and implement the decision. And in highly volatile conditions, this delay will yield outdated information, as the conditions may have

changed drastically. This means that an online learning approach is needed, where decisions are made based on historical and previous observations, without presuming knowledge of their future values.

Online Meta-Learning. This thesis also leverages the idea of *online meta-learning* (henceforth, sometimes referred to as simply "meta-learning"), which boosts performance whenever possible: a pool of specialized learners tuned for different conditions (e.g., mobility patterns) is used, and the meta-learner assesses in real-time and "trusts" the one(s) that perform the best; see Figure 1.2. For instance, algorithms that are designed to perform well in adversarial conditions might be too conservative when the conditions are static/stationary, or when the network has access to context information [17]. In these latter cases, different data-centric learners can leverage the available information to identify optimal solutions faster. Hence, the question that arises naturally is how to combine the required robustness of some algorithms that might work optimally under adversarial conditions, without compromising learning performance (in terms of convergence speed) whenever the conditions are known; therefore, obtaining the *best-of-both-worlds*.

To address this, we adopt ideas from the expert-learning paradigm [18] where meta-learners intelligently select among actions proposed by different algorithms, which, in turn, rely on and perform better under different assumptions. A key challenge is that learning occurs on two levels: the meta-learner must learn which algorithm(s) are the best-performing, and each algorithm must learn which action(s) are the best ones. In the *full-feedback* setting, the meta-learner can observe and potentially combine the actions chosen by all learners. In contrast, in the *bandit-feedback* setting, the meta-learner chooses a single algorithm; and the action of only this algorithm is observed, rendering the learning even more demanding.

Meta-learning is finding increasing applications in online learning [19] and communication systems [20] due to its robustness to distribution shifts and fast adaptation. The power of this framework lies in the generic nature of the deployed learners, as it is not limited to online learning algorithms, but can also encompass forecasters trained offline [21], [22]. As shown later in this thesis, if these forecasters are superior to the other learners because, for instance, there is an abundance of training data, then the overall performance improves; and when the forecaster is found to be inaccurate, our meta-learners maintain the robust performance of the online learners. This design enables robust adaptation to *all* environments without prior statistics and ensures that the achieved performance quickly approaches that of a powerful oracle with full knowledge of the future.

Performance Assessment. The goal of an online (meta-)learning algorithm is to minimize its cumulative loss. One of the primary metrics to assess how well the algorithm does so is through *regret*: the difference between the cumulative loss of the algorithm and that of an ideal (but unknown) benchmark (i.e., *oracle*), which has *full* information about the future. In other words, it measures how much the (meta-)learner "regrets", in hindsight, not having followed the oracle [23]. For an algorithm to "learn", the regret, on *average*, should diminish (i.e., approach zero) as the number of time slots increases; or similarly, the gap between the (meta-)learner and the oracle should decrease as time passes and the algorithm learns.

The notion of regret takes many forms in the literature, depending on the definition of the oracle. In the simplest but most studied scenario known as *static regret* [15], [16], [24], the oracle chooses the best *fixed* action in hindsight. Nevertheless, if the best action changes over time, choosing the single best action (e.g., single cell to serve a moving user) may not be a suitable-to-compare benchmark. For that, the notion of *dynamic regret* was introduced that measures the algorithm's actions w.r.t. *any sequence* of actions; that is, an oracle that can change its decision at each time slot. Clearly, it is impossible to compete with an arbitrarily changing oracle [16]. However, a diminishing dynamic regret can be achieved when the oracle's actions do not change too often [15], [25], a concept known as *path length*. The algorithms in this thesis provide robust theoretical guarantees using both the static and dynamic flavors of regret, and are assessed even in scenarios where the best actions are changing adversarially.

1.3. Problem Statement

Encompassing the key aspects and challenges mentioned above, this thesis addresses the following question:

> What are the key mobility and resource management challenges in the emerging O-RAN-enabled next-generation mobile networks, and how can these be addressed towards enabling efficient and robust operation?

To answer the question, this thesis collects multiple countrywide datasets from a top-tier MNO and develops principled online meta-learning algorithms that offer performance guarantees. The following sections describe the key mobility and resource management challenges addressed and summarize the main contributions.

1.4. Problem Description

To address the question posed in Section 1.3, this thesis focuses on two tightly connected problems at the core of mobile network operation: mobility management through effective handovers, and allocation of vBS resources.

1.4.1. Problem 1: Mobility Management via User-Cell Associations and Handovers

Traditional Handovers. Mobility management has been a primary consideration for every generation of mobile networks and occupies a prominent position in the agendas of both industry and academia [26], [27]. At its core lies the problem of *user-cell associations/assignments*, namely, which cell should serve each user at each point in time (i.e., time slot). Since each radio cell can only cover a limited geographic area, these associations inevitably change as users move; thus causing *handovers*, referred to also as traditional handovers (THOs).

The THO mechanism is illustrated in Figure 1.3a, where a user moves away from its serving cell C_1. At the intersection of the three cells, the user may either return to C_1 or continue toward C_2 (or C_3). In the latter case, a THO is triggered, but only

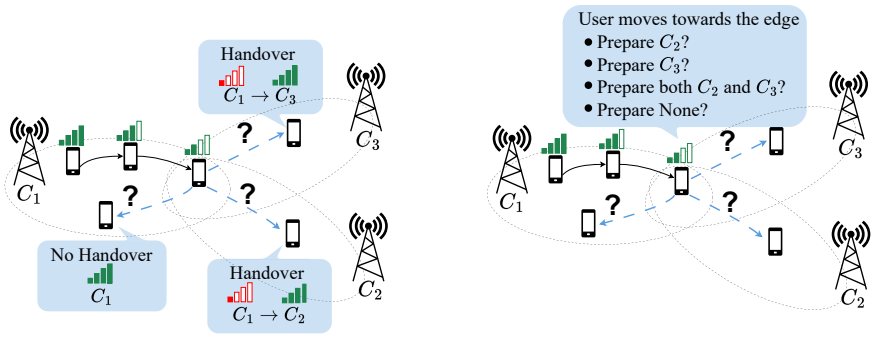


Figure 1.3: Decision-making in (a) traditional and (b) conditional handovers, after and before signal has degraded, respectively.

after the user's signal from C_1 has already degraded; meaning the user is already well within the coverage region of C_2 (or C_3).

As becomes apparent, HOs are the fundamental elements for enabling seamless connectivity for mobile users.[4] Optimizing this procedure is crucial, as inefficient or suboptimal HOs can have a dire twofold impact: on the network's side, they might lead to increased resource consumption and signaling overhead [29]; and on the user's side, they may cause service interruptions and battery depletion [30], [31]. With the advent of 5G and 6G, HO management has become more intricate due to the coexistence and integration of multiple generations of mobile technologies [32].

Thus, the latest proposals depart from conventional signal-to-noise ratio (SNR) or signal-to-interference-plus-noise ratio (SINR) based association rules, towards solutions where user-cell associations are based on network-wide criteria, e.g., aggregate throughput or fairness/load-balancing metrics, while also aiming to reduce HOs, and their (sometimes prolonged) delays whenever possible [33]–[36]. The recent O-RAN paradigm [12] facilitates such holistic approaches and enables central *controllers* to coordinate network operations dynamically [37], [38], including the implementation of user-cell associations. The contributions of the thesis to this problem are analyzed in Section 1.5.

Conditional Handovers. Even though THOs have long served as the backbone of mobile connectivity, they remain inherently reactive, triggered only after signal conditions have degraded significantly; see Figure 1.3a. This results in increased failures and delays in environments with dense deployments or high frequency bands, where signal conditions change more abruptly.

To mitigate these limitations, 3GPP introduced CHOs [39], [40]. The main idea is to *proactively* reserve resources in multiple *candidate* cells while signal conditions are favorable, and delegate the final HO decision to the user. This novel approach of *network-configured, user-decided HOs*, instead of the *network-configured, user-assisted* THOs, has been shown to reduce HO delays and failures [41], [42]. At

[4]The HO procedure is different from *cell (re-)selection*, which happens when users do not maintain an active data connection but still need to change cells to ensure the reception of signaling messages [28].

the same time, CHOs introduce new challenges: selecting the optimal set of cells to prepare is highly context-sensitive; preparing too many cells leads to resource overutilization, while preparing too few risks service interruption due to fallback to traditional HOs [43]–[46]. Nevertheless, for a user that moves slowly, it might still be beneficial to execute THOs, as signal conditions tend to remain more stable.

Figure 1.3b illustrates these challenges: if, at the intersection of the three cells, we prepare only C_2 (C_3) but the user moves towards C_3 (C_2), then C_2 (C_3) resources will be wasted; also, at the next time slot, extra signaling/communication will be needed to release these resources and a traditional HO will occur. Preparing both C_2 and C_3, if they have enough capacity, is beneficial to the user, and soon after the trajectory becomes clear and a CHO is executed, the resources allocated to the unneeded cell can be released, creating, however, some signaling cost. Ideally, if only C_2 (C_3) is prepared at the intersection of the three cells and the user indeed moves towards C_2 (C_3), the CHO can be executed efficiently, ensuring optimal resource allocation with minimal signaling overhead. In case no candidate cells are prepared, the user executes THO, as in Figure 1.3a.

Although the 5G architecture introduced flexibility to support such capabilities [47], [48], the fundamental approach to HO has remained largely unchanged. In contrast, the 6G vision, expressed in various 3GPP workshops and white papers [49]–[53], paints a different picture, where mobility should be *intelligent, proactive, and relying on native Artificial Intelligence (AI)*. To support highly demanding 6G use cases, the network must adapt in real-time to user requirements and the current signal conditions [54]. Specifically, seamless mobility across different service domains, ultra-reliable low-latency HO and resource efficiency are identified as critical enablers for 6G [51], [52]. Static or predefined HO strategies cannot meet these requirements. We thus call for a *paradigm shift* in mobility management that can jointly optimize CHO and THO strategies, underpinned by (near) real-time data-driven control. More details on the contributions of the thesis to this problem are given in Section 1.5.

1.4.2. Problem 2: Resource Allocation for Virtualized Base Stations in Non-Real-Time

At the core of virtualized and open RAN architectures [12] lie vBSs, such as srsLTE [55] and OpenAirInterface (OAI) [56], which offer OPEX/CAPEX savings and performance gains, since their operational parameters can be adjusted with high granularity at runtime [57]. Alas, these benefits come at a cost. Softwarized base stations are found to have less predictable performance and more volatile energy consumption [58]–[60], an issue that is amplified when instantiating them in general-purpose computing infrastructure. This induces operation and cost uncertainties at times when there is an increased need for robustness and performance guarantees in mobile networks. Therefore, it becomes imperative to understand how to configure or schedule these vBSs (i.e., how to allocate their resources) without relying on strong assumptions or compromising network performance, in order to unblock their deployment and maintain energy costs at sustainable levels.

As mentioned before, the O-RAN architecture offers new opportunities to achieve

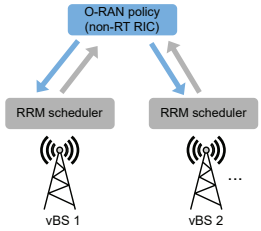


Figure 1.4: The proposed policy operates in the non-RT RIC and decides MCS, Power, and PRB thresholds that are sent to each vBS's scheduler.

this goal. Namely, the emerging O-RAN standards [10], [11] have provisions for multi-tier control solutions for resource management that can be implemented centrally, i.e., by the RIC, and enforced at different time scales. In particular, our focus here is on non-RT policies that determine the operation envelope (or resource allocation *thresholds*) of the vBSs over time intervals (rounds) of a few seconds. These policies are fed to, and enforced by, the real-time radio scheduler of each vBS, which devises their assignments subject to global rules about, e.g., the maximum transmission power and the eligible modulation and coding schemes (MCS); as can be seen in Figure 1.4. Such centralized threshold policies have been recently introduced, e.g., see [60]–[62], and have several practical advantages. First, O-RAN includes heterogeneous base stations that are challenging, if not impossible, to configure directly by intervening with their real-time schedulers. The global non-RT policies, on the other hand, offer an easy path to shape the operation of each vBS. Secondly, using such central policies, the O-RAN controllers can coordinate the operation of their vBSs in a unified fashion, managing jointly their resources, and also use AI/ML mechanisms that can benefit from this centralized view.

Nevertheless, the effective design of such policies is a new and particularly intricate problem. Due to their coarse time scale (seconds) and unlike the typical Radio Resource Management (RRM) decisions (updated in ms), these policies do not have access to the network conditions and user traffic that will be realized during the interval they will be applied. And, further, these parameters can change arbitrarily during such large time windows, not necessarily following a stationary distribution. Moreover, due to the heterogeneity and volatile operation of the vBSs, the effect of such policies on the KPIs of interest is challenging, if not impossible, to predict or quantify with analytical expressions. Coupled with the typically large number of possible policies, this compounds finding the optimal policy for each vBS. In light of these observations, it is not surprising that the first works in this area focused on O-RAN operations under static network conditions and demands, [60], [62]. The following section details the contributions of the thesis to this problem.

1.5. Contributions and Outline

This section presents the contributions, problems addressed, and proposed solutions for each individual chapter, as illustrated in Figure 1.5. The focus of each chapter

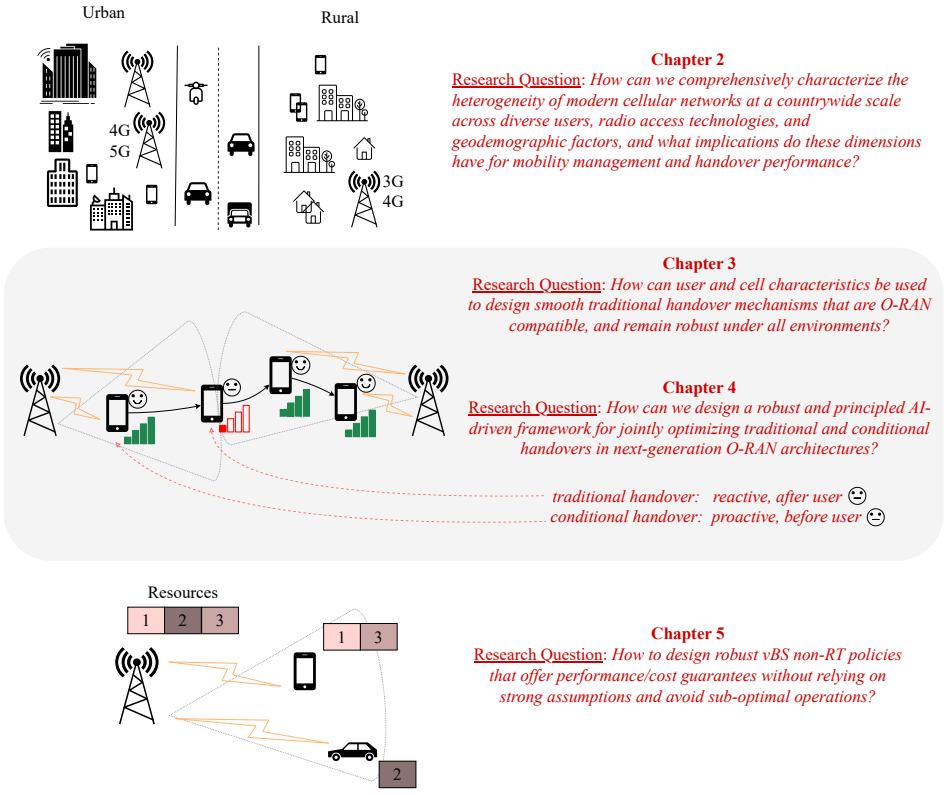


Figure 1.5: Schematic focus and research questions of each chapter of the thesis.

with respect to the O-RAN architecture is shown in Figure 1.1 (left).

Heterogeneity and Mobility Management of Cellular Networks – Chapter 2. This chapter examines the heterogeneity of modern cellular networks and their mobility management through a countrywide analysis of a top-tier MNO in Europe. Unlike the vast majority of studies that employ data from measurement campaigns within confined areas and with limited devices, thereby providing only a partial view of these aspects, we collect traffic for four weeks from approximately 40M users. By analyzing these datasets, we capture the heterogeneity of RATs, varying deployments based on population, and a broad spectrum of user types (e.g., smartphones, M2M/IoT devices), with different characteristics, such as manufacturers (e.g., Samsung, Apple, Google).

Subsequently, we delve into HOs, the fundamental element for seamless connectivity of mobile users. We characterize the geo-temporal dynamics of horizontal (intra-RAT) and vertical (inter-RAT) HOs at the district level and at millisecond granularity, leveraging open datasets from the country's official census office to as-

sociate our findings with the population. Additionally, we investigate the frequency, duration, and causes of HO failures, and model them using statistical tools.

These insights provide the empirical basis for the next chapter, where we shift from characterization to improvement and develop mechanisms that enhance HO performance in line with the ambitions of 5G and beyond.

Mobility Management through Smooth Handovers – Chapter 3. In this chapter, we shift our focus to HO optimization. To ensure that our proposed methodologies align with reality, we begin by building on the datasets analyzed in the previous chapter and enriching them with targeted measurements and crowdsourced signal data. Our findings quantify the impact of HOs on network and user key performance indicators (KPIs), such as packet loss and throughput, and highlight a correlation between HO failures/delays and the characteristics of cells and users. Leveraging these new mobility insights, we subsequently model for the first time, to the best of our knowledge, UE–cell associations as dynamic decisions under the framework of Smoothed Online Learning (SOL), which enriches the online convex optimization (OCO) toolbox, accounting for costs induced by decision changes (and thus, HOs).

We propose a realistic system model for smooth and accurate HOs that extends existing approaches by *(i)* incorporating device and cell features in HO optimization, and *(ii)* eliminating (prior) strong assumptions about requiring future signal measurements and knowledge of users' mobility. Our proposed online meta-learning algorithm, aligned with the O-RAN paradigm, offers robust dynamic regret guarantees even in challenging environments and demonstrates superior performance in multiple scenarios with real-world crowdsourced and synthetic data.

Yet, these traditional HOs, the backbone of mobility management, remain reactive by design, as they are triggered only after conditions have already deteriorated. This raises the natural question of whether networks and users can act proactively, when needed, by preparing cells before a disruption occurs. To answer this question, the next chapter turns to CHOs and investigates how to design a robust and AI-driven framework for jointly optimizing THOs and CHOs in 5G, 6G, and next generation (NextG) O-RAN architectures.

From Reactive to Proactive Handovers – Chapter 4. This chapter advances our study of mobility management by moving beyond THOs alone and incorporating CHOs into a unified optimization framework. To capture these dynamics, we extend the MNO datasets with both source and target cell features (not only the former, as in the chapters before), allowing us to capture the unique dynamics of CHOs, where multiple target cells could be allocated in advance. Building on these insights, we introduce online meta-learning algorithms that adapt to runtime observations and guarantee performance comparable to a theoretical oracle with perfect future information, without access to system conditions. Additionally, they are designed for near-real-time deployment as xApps within the O-RAN architecture, aligning with 6G and NextG goals of flexible and intelligent control. Extensive evaluations leveraging our real-world countrywide dataset demonstrate that they improve user throughput and reduce signaling overhead, outperforming 3GPP-compliant baselines.

Once traditional and conditional handovers are optimized, the focus must shift to the other side of the link, namely, how (v)BS and their cells adapt their available resources to serve connected users efficiently. Apart from proportional-fair / round-robin schedulers used up to this section, can resources be assigned in a more effective and robust way?

Resource Allocation for Virtualized Base Stations in Non-Real-Time – Chapter 5. This chapter focuses on optimizing the allocation of vBS's radio resources in the O-RAN ecosystem. Although such systems offer increased flexibility, reduced costs, vendor diversity, and improved interoperability, optimizing the allocation of their radio resources raises new challenges due to the volatile operation of vBSs and the dynamic network conditions and user demands they must support. Using the novel multi-tier control architecture of O-RAN, we propose a new set of resource allocation non-real-time (i.e., threshold) policies, designed to balance the performance and energy consumption of vBS in a robust and provably optimal way.

To that end, we introduce an online learning algorithm that operates under minimal assumptions and without requiring knowledge of the environment, hence being suitable even for "challenging" environments with non-stationary or adversarial demands and conditions. We also develop an online meta-learning solution that leverages other available algorithmic schemes, e.g., tailored for more "easy" environments, by choosing dynamically the best-performing algorithm; thus enhancing the system's effectiveness. We prove that the proposed solutions achieve sub-linear regret (zero optimality gap) and characterize their dependence on the main system parameters. The performance of the algorithms is evaluated using real-world data from a testbed, under both stationary and adversarial conditions, yielding energy savings of up to 64.5% compared to several state-of-the-art benchmarks.

2

Heterogeneity and Mobility Management of Cellular Networks

In this chapter, we take the first step in understanding the heterogeneity of modern cellular networks and their mobility management from the perspective of a top-tier mobile network operator (MNO), offering a realistic view of what deployed networks look like today. Unlike the vast majority of studies, which employ data from measurement campaigns within confined areas and with limited devices [30], [32], [63]–[73], we collect traffic from approximately 40M users for four weeks and study the heterogeneity of users (e.g., different manufacturers and types) and the coexistence of multiple radio access technologies (RATs) from different generations (2G–5G), as well as the geodemographic segmentation. Special attention is paid to mobility and handover (HO), the primary mechanism through which seamless connectivity is achieved as users move across the network. We quantify the geo-temporal dynamics of both horizontal and vertical HOs, and associate them with data sourced from the country's national census. Finally, we analyze the patterns, causes, and durations of HO failures (HOFs) and model them using statistical tools.

The content of this chapter has been published in:

M. Kalntis, J. Suárez-Varela, J. O. Iglesias, A. K. Bhattacharjee, G. Iosifidis, F. A. Kuipers, and A. Lutu, "Through the Telco Lens: A Countrywide Empirical Study of Cellular Handovers," in *Proc. of ACM Internet Measurement Conference (IMC)*, 2024.

2.1. Challenges and Contributions

As highlighted in Chapter 1, the advent of 5G, and soon, its successors, 6G and next generation (NextG), marks a shift in the telecommunications landscape, offering unprecedented speed, ultra-low latency, exceptional reliability, and, importantly, ubiquitous connectivity to a wide array of devices [74]. However, like any emerging technology, the pace of real-world deployments does not match instantly the pace of innovation [75], [76], resulting in multiple generations of technology operating simultaneously. This coexistence of multiple RATs induces economic and deployment trade-offs, highlighting the importance of understanding the ensuing *heterogeneity* under real-world conditions.

Motivated by this unique transitional phase, we begin the chapter by capturing novel, large-scale datasets from a top-tier MNO in Europe[1]; enabling, in this way, the analysis of network dynamics at a crucial moment when –at the time of capturing the datasets– *all* digital RATs developed during the last three decades are concurrently operational within the same network. Specifically, we analyze three main aspects that affect the complexity of modern cellular networks: the heterogeneity of (*i*) RATs and (*ii*) UEs, as well as the (*iii*) geodemographic diversity.

A central element that cuts across these three axes is *mobility management*, which has become increasingly intricate, yet crucial for maintaining seamless connectivity. Thus, we continue our analysis by presenting the *first, to our knowledge, countrywide analysis of mobility management from the perspective of a top-tier MNO in Europe.* As can be seen in Table 2.1, we have recorded *all mobility events*, and more precisely, the induced HOs and HOFs for four weeks at millisecond granularity.

We merge this data with: (*i*) information from the MNO's deployment, to study HO performance across its topology and supported RATs, (*ii*) device-specific information, to associate HOs and HOFs with specific UE types and manufacturers, and (*iii*) data from the country's official census office, to account for the geodemographic distribution of HOs across 300+ districts with various population densities. At the time of capturing the datasets, the MNO was initiating its commercial 5G-Standalone (SA) deployment; thus, we measured only the 5G-Non-Standalone (NSA) deployment to avoid any early-stage issues with SA [77].

We analyze the spatio-temporal dynamics of horizontal (intra-RAT), and vertical (inter-RAT) HOs, at the district-level and with ms granularity, and characterize their pattern across the country to identify regional trends. Furthermore, we dissect the impact of UE types (smartphones, M2M/IoT devices, low-tier feature phones) and manufacturers on HOs, HOFs, and mobility/performance metrics. We also analyze the causes behind HOFs, using 3GPP-based and vendor-specific failure descriptions. Lastly, we leverage statistical methods to model how the coexistence of multiple RATs affects HO performance, especially when UE connections are downgraded to older technologies (e.g., 2G, 3G).

Below, we present the key findings and contributions of this chapter.

• **Heterogeneity & Complexity of Networks / Mobility (Section 2.3)**. In the MNO's deployment, 5G cells make up 8.4% while 4G accounts for 55%, with 2G

[1]To maintain the confidentiality of the operator, we are only able to disclose general location.

Table 2.1: Dataset statistics.

Feature	Value
Area covered	Country in Europe (300+ districts)
# of cell sites	24k+
# of radio cells	350k+
# of UEs measured	≈ 40M
# of handovers (daily)	1.7B+
Measurement period	29-Jan-2024 to 25-Feb-2024 (4 weeks, 28 days)
Trace size (daily)	≈ 8 TB

and 3G cells covering the remaining ≈36% and handling 18% of user connectivity time. Despite this, older RATs carry only 5.23% of the uplink (UL) and 2.07% of the downlink (DL) data flowing through the network. Among all UEs, 59.1% are smartphones, primarily from Apple (54.8%) and Samsung (30.2%), from which 51.5% do not support 5G, relying instead on 4G. Additionally, over 32% of the UEs, mainly M2M/IoT devices and feature phones, support only up to 3G. This blend of technologies highlights the challenges of phasing out older RATs, particularly in an environment where IoT manufacturers still rely on 3G/2G for devices with limited connectivity needs. Our geodemographic analysis points to a large disparity between the density of HOs in urban centers with larger population density (2.1M HOs per sq. km), and less populated rural areas (60 HOs per sq. km); in a network deployment that registers on average 13.1k HOs per sq. km.

• **HO Analysis (Section 2.4)**. Taking as a reference the HOs registered in the 4G EPC, approximately 94% of HOs are horizontal (between 4G/5G-NSA radio cells), complete within 90 ms (median of 43 ms), and correspond to smartphone activity. M2M/IoT UEs and low-tier feature phones –accounting for >40% of the device population– share the remaining 6% of HOs. By investigating the top-5 smartphone manufacturers (Apple, Samsung, Motorola, Google, Huawei), we discover similar patterns in terms of HOs (±10% of variation between them) and low HOF rates (Google exhibits -27% of HOFs w.r.t. other UEs, but with higher variability). Moreover, we find some smartphone manufacturers outside the top-5 (e.g., KVD) that exhibit higher HOF rates (up to +600% w.r.t. other UEs) and HO signaling (up to +293%).

• **HOF Analysis & Modeling (Section 2.5)**. Rural areas (with sparser deployments) experience 32.4% more HOFs during peak hours [7:00–8:00) than urban areas. Moreover, HOF rate is close to zero for the majority of the UEs; for the ones with high mobility metrics (>100 visited cells, >100km radius of gyration), which are mostly smartphones (85%), HOF rate rises up to 0.4% (pct-75).

Furthermore, we dissect the reasons why HOs fail by using 1k+ 3GPP and vendor-specific descriptions that explain the causes. Interestingly, we find that 92% of the HOs in the entire country fail with solely 8 causes; and from the studied failures, 75% (0.03%) occur in HOs to 3G (2G), and 25% of them are due to an excessive load in the target cell (Cause #4). Moreover, we measure the duration of

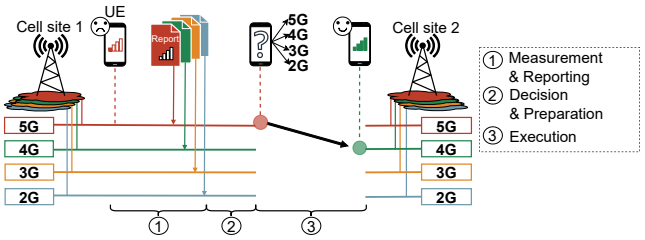


Figure 2.1: High-level description of HO procedure [78].

these 8 causes and highlight that the ones related to specific cancellations (Cause #1) and timeouts (Cause #8) require on average >2 s to complete, reaching up to 10 s in the latter case.

In addition to the previous analysis, we aim to identify which network-related features correlate with HOFs. Specifically, we test whether the HO type (intra/inter-RAT) is a good predictor for the HOF rate. Statistical analysis verifies that, although they occur infrequently (only 6% of all HOs are to 2G/3G), HOs to 3G (2G) increase the HOF rate by 166% (915%) compared to HOs between the newer RATs (intra 4G/5G-NSA).

2.2. Methodology and Datasets

In this section, we briefly present the HO mechanism and demonstrate our measurement infrastructure in a large countrywide MNO. We describe the three datasets built for this study and introduce the official census dataset we used to complement our analysis. Finally, we detail the performance and mobility metrics employed.

2.2.1. Handover Mechanism

Every UE relies on its *primary cell* (i.e., the cell it is connected to), serving as the pivotal link for control-plane signaling and HO management. Figure 2.1 depicts the HO process from a source (i.e., primary) to a target radio cell [79]. When a UE attaches to a new cell, it receives a set of mobility management configurations, including parameters for the triggering of HO events (e.g., hysteresis, offsets, etc.). Based on these configurations, the UE performs signal strength/quality measurements – e.g., Reference Signal Received Quality (RSRQ) – of the source and neighboring cells, and sends a Measurement Report (MR) to the source periodically, or if any of the mobility management criteria is met. For instance, in 4G and 5G NR, a HO triggering event typically occurs when either the serving cell's signal falls below a threshold (A2 event) or when the signal of a neighboring cell becomes offset better than the serving cell (A3 event) [80], [81].

Based on the MR, the source identifies the best target cell and initiates the HO. After the target cell accepts the request, the source transmits a HO command to the UE. For example, in 4G/5G NR, the source sends a Radio Resource Control (RRC) Connection Reconfiguration message to the UE to begin its cell synchronization with

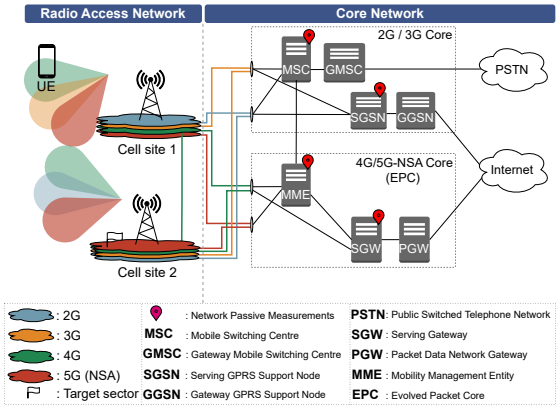


Figure 2.2: Network architecture & measurement points.

the target cell and the Random Access Channel (RACH) procedure. After the UE reports successful access to the target cell, the source releases its resources. More details are available in [78], [82].

2.2.2. Network Data Collection

Measurement Infrastructure. We collect passive measurements using commercial tools integrated into the MNO's infrastructure, see Figure 2.2. In a nutshell, the cellular network architecture can be divided into three primary components: (*i*) the devices accessing the network, (*ii*) the Radio Access Network (RAN), responsible for managing wireless communication, and (*iii*) the Core Network (CN), which provides the required services and functions for the network operation (e.g., user authentication and mobility management). This is consistent for all the different radio technology generations that coexist in the network. Our monitoring locations, which we depict with red pins in Figure 2.2, focus on key components of the core network, including the Mobile Management Entity (MME tracks and manages the mobility of devices in 4G and 5G-NSA), the Mobile Switching Center (MSC is responsible for communication switching functions), the Serving GPRS Support Node (SGSN manages data routing for 2G/3G), the Serving Gateway (SGW routes packages between RAN and the CN), and the cell sites in the RAN.

Data is collected in a private cloud environment for a given retention period, and is already anonymized before processing. Particularly, we organize the collected data into three datasets, providing various information at the radio cell and UE-level.

Mobility Management Signaling Dataset. The captured data spans from 29-Jan-2024 to 25-Feb-2024 for the entire country (see Table 2.1). We analyzed the activity of users in the control plane for all RATs supported by the MNO. For each RAT, the dataset includes the (control plane) signaling messages related to events such as service requests, HOs, attach/detach, paging, and Tracking Area Update (TAU). We direct our attention to HOs, for which we capture six main variables that enable an in-depth analysis: (*i*) *timestamp*, with millisecond granularity, (*ii*) *HO re-*

sult (i.e., success/failure), (*iii*) *HO duration* (msec granularity), (*iv*) *cause codes for HO failures based on 3GPP* [79], [83], which are enriched with *sub-cause descriptions* specified by the antenna vendors, (*v*) *anonymized user ID*, based on the International Mobile Subscriber/Equipment Identity (IMSI/IMEI),[2] and (*vi*) source and target radio cells along with their RATs. As mentioned before, due to the early stages of 5G-SA deployment in the studied MNO, we base our analysis on 5G-NSA, which relies on the 4G EPC for mobility management.

Radio Network Topology. We utilize this dataset to integrate in our analysis the upgrades in the MNO's network deployment footprint (e.g., newly deployed sites). We capture this dataset daily during the period of analysis; it contains information on each deployed radio cell, such as geographic location (longitude and latitude), the postcode of the area, and the supported technologies (i.e., 2G, 3G, 4G, 5G).

Devices Catalog. We leverage a daily commercial database, provided by the Global System for Mobile Communications (GSM) Association (GSMA) to examine correlations of device-specific characteristics with HOs. This catalog associates the TAC of each device with attributes such as the supported radio bands and RATs, the manufacturer, and the device type. We apply a heuristic to classify the devices into three types: smartphones, M2M/IoT devices, and low-tier feature phones [84]. For this, we rely on the observation that the Access Point Name (APN) configured for the UEs may contain keywords associated with IoT verticals (e.g., "m2m", "smartmeter"), and combine the information from the APN with the device characteristics of our daily commercial GSMA database.

2.2.3. Census Datasets

We leverage open datasets published by the official census office in the studied European country to enrich our mobility study with the geodemographic characteristics of different areas. Specifically, we take as a reference the 300+ districts defined by the census office, and collect the population density and the postcodes within each of them. Then, based on census information we classify postcode areas into two main categories (*urban* and *rural*), which correlate with population density (more than 10k and less than 10k residents) and also serve as a proxy for areas with denser and sparser RAN deployments, respectively.

2.2.4. Performance and Mobility Metrics

Performance Metrics. In line with prior works [28], [85], we focus on:

- *HO count*, which represents the number of HOs over a time interval. We usually set it to either 30 minutes, 60 minutes, or one day. We use this metric to show how users' mobility fluctuates over time and space, and how it differs per RAT, device type, and manufacturer.
- *HO duration*, which represents the time interval (in ms) to complete the HO, see [72], [86]–[88]. Minimizing this interval is crucial for seamless connectivity and improves the users' Quality of Experience (QoE).

[2]The first 8 digits of the IMEI represent the Type Allocation Code (TAC), which we use later to classify devices.

• *HOF rate*, which refers to the number of HOFs divided by the total number of triggered HOs.[3] HOFs dramatically affect the users' experience and typically happen due to poor signal strength, configuration and synchronization errors, or capacity issues in the network. In Section 2.5, we uncover the reasons why these failures occur and emphasize that a comprehensive understanding of HOFs can only be achieved by incorporating the perspective of MNOs.

Mobility Metrics. To showcase the mobility characteristics of users, we focus on two metrics from the MNO's perspective, as follows.

• *Number of cells* quantifies the number of distinct radio cells that a user successfully communicates with, per day. We highlight that this metric does not necessarily translate to the distance traveled by users in a given area, as the density of radio cells in the area also plays a role. For instance, urban areas typically have denser deployments and, as a result, users connect to a larger number of cells even if they travel the same distance as in less populated areas (e.g., rural) with sparser deployments.

• *Radius of gyration* complements the previous metric by capturing the traveled distance for the UEs [89]. It is defined as the root mean squared distance between each visited cell (weighted by the time spent there) and the center of mass. The radius of gyration is defined as follows:

$$g = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(t_j\mathbf{l}_j - \mathbf{l}_{\text{cm}})^2},$$

where $\mathbf{l}_j$ represents the location of the j^{th} visited cell site, t_j represents the time spent in the j^{th} visited cell site and $\mathbf{l}_{\text{cm}}$ represents the location of the user's center of mass, calculated as $\mathbf{l}_{\text{cm}} = \frac{1}{N}\sum_{j=1}^{N}\{t_j\mathbf{l}_j\}$, where N is the total number of cell sites visited by the user. A high radius of gyration indicates that the user travels far and wide (i.e., their moves span a large geographical area). Conversely, a lower radius of gyration points to more localized movements, relatively close to a central location.

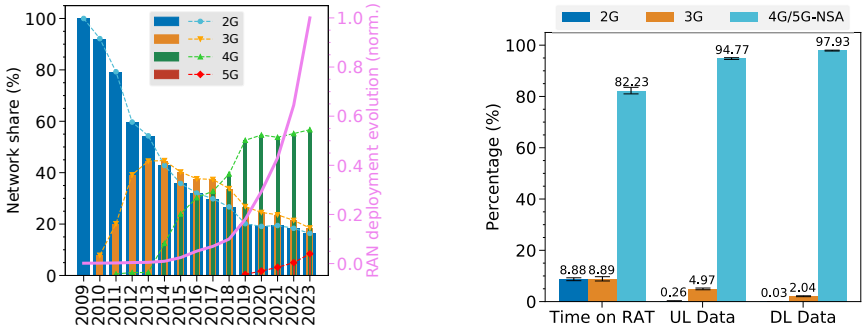
2.3. A First Look at the Network

Our datasets capture the heterogeneity and complexity of HOs across the entire MNO's deployment in the studied country, which includes diverse deployment densities and RATs, as well as a broad spectrum of UEs (e.g., smartphones, M2M/IoT, etc). In this section, we explore the heterogeneity of these datasets along three particularly interesting axes, from the network's perspective: (*i*) heterogeneity of RATs, (*ii*) heterogeneity of UEs, and (*iii*) geodemographic complexity.

2.3.1. Radio Access Technologies

Figure 2.3a shows the deployment evolution in the network from 2009 to 2023. The number of cells (solid pink line) has increased at an exponential pace in the last 15 years, with an average growth of 59% during the last 5 years (2018-2023).

[3] We primarily focus on HO failures rather than explicitly detailing HO successes; however, successes and failures are complementary to each other.



(a) (b)

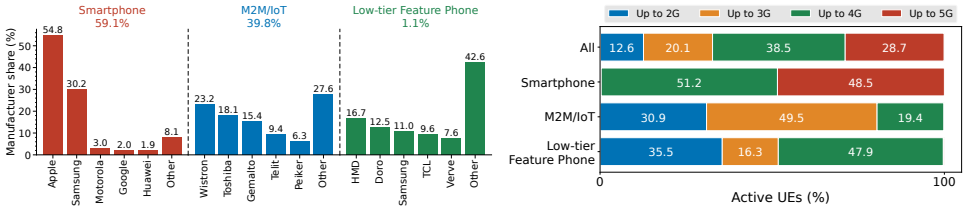
Figure 2.3: (a) Evolution of network deployment in a commercial MNO. The left y-axis corresponds to all bars and lines, except the pink line (right y-axis). (b) Average daily RAT use. Error bars show the min/max daily values over 4 weeks.

Throughout these 15 years, different RATs have been coexisting with a varying mix. The latest major network upgrade occurred in 2019 with the deployment of 5G-NR, which accounted for 8.4% of the cells by the end of 2023. At the same time, we observe the gradual decommissioning of 2G and 3G cells (≈18% each in 2023), while 4G is still the dominant RAT (≈55%) in terms of infrastructure. This heterogeneity does not come as a surprise, since decommissioning legacy RATs is a challenging process that needs to account for various techno-economic factors, such as the turnover rate of customers or the radio coverage [90]. Nonetheless, it compounds network management and affects both the number and the success of HOs as we present in Section 2.5, and as prior studies have also identified [73], [91].

To further understand the use of the RATs, we compute the overall time that UEs spend on each of them by using the timestamps of mobility events in the dataset. With the current 5G-NSA deployment, we do not distinguish from the events captured in the core network (i.e., MME) when devices are served by a 4G or a 5G-NR cell (see Section 2.2); thus, we use the term "4G/5G-NSA". In Figure 2.3b, we notice that UEs rely mostly on 4G/5G-NSA (≈82% of the time on average), while 2G and 3G serve users during a non-negligible 8.9% of the time each. In terms of aggregated data volumes, the share for 4G/5G-NSA rises up to 94.77% and 97.93%, respectively, for UL and DL traffic, leaving marginal values for 2G and 3G. Yet, these legacy RATs still serve a noteworthy number of UEs that support only these technologies (see Section 2.3.2).

The heterogeneity of the network appears also in terms of the antenna vendor. Four principal vendors (V1, V2, V3, V4) employ antennas (and thus, RATs) for this network, with their deployment distributed asymmetrically across different regions. All vendors support 4G/5G-NSA and 3G RATs, and accommodate nearly the full spectrum of devices. Details are provided in Appendix 2A.

Key takeaways: *The cellular network we measure includes all RATs (2G-5G), where 2G and 3G radio cells account for 36% of the total deployment. These RATs (2G & 3G) connect users on average for 18% of their up-time, while UEs generate only 5.23% (2.07%) of the UL (DL) data over them.*



(a) (b)

Figure 2.4: Percentages of (a) manufacturers and device types, and (b) the RATs they support (excluding <0.5%).

2.3.2. User Equipment

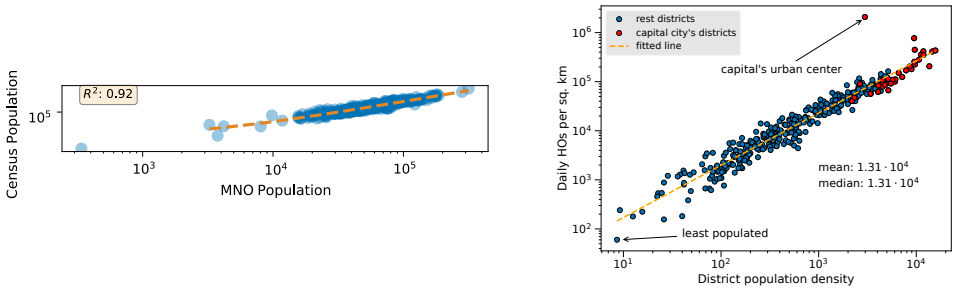
The number of devices accessing the network over the 4-week period is ≈40M. We classify these devices into three types based on their capabilities, namely, **smartphones**, **M2M/IoT devices**, and **low-tier feature phones**, accounting for 59.1%, 39.8%, and 1.1% of UEs, respectively. Figure 2.4a shows the top-5 manufacturers in the three types of devices. In the larger category – smartphones – we observe that most devices are manufactured by Apple (54.8%) or Samsung (30.2%). For M2M/IoT UEs, we find a diversified set of manufacturers; namely, over 27% of these UEs are from manufacturers outside the top-5.

We infer the connectivity capabilities of mobile devices from the GSMA device catalog (see Section 2.2). We find that 12.6% of all UEs support only 2G and 20.1% up to 3G (see Figure 2.4b), which partially explains the slow pace of decommissioning legacy RATs. These legacy devices are mostly M2M/IoT devices or feature phones, where $> 80\%$ and $> 50\%$, respectively, support at most 3G. The overall number of devices that support 4G or 5G adds up to 67.2%. The majority of these devices are smartphones: 51.4% of smartphones support up to 4G, and 48.5% are 5G-capable.

Key takeaways: *Over 32% of all devices support only up to 3G – predominantly M2M/IoT UEs and feature phones – and 51.5% of smartphones do not support 5G yet (the majority relies on 4G). These factors contribute to the presence of a mixture of old and new RATs in current deployments, stressing the challenges associated with decommissioning the older ones.*

2.3.3. Geodemographic Segmentation

Population Sampling. This section demonstrates that the dataset we collect through the commercial MNO is representative of the country's overall population. Figure 2.5a shows the population according to census (y-axis) and the population we inferred from the MNO (x-axis), where each data point refers to the districts in the country (see Section 2.2). We derive the end-user's home location at postcode granularity from their connectivity patterns during nighttime [92]. To achieve this, we consider the main cell site the user connects to between 00:00 and 08:00 (i.e., night hours) for at least 14 days (not necessarily consecutive) during February 2024. We then aggregate their mapped home postcode at the district level. These results show a clear linear relationship ($R^2 = 0.92$) between the census data and the MNO



(a) (b)

Figure 2.5: In district-level (a) comparison between the inferred population from the MNO data and the actual population from the census data, and (b) daily HOs per sq. km.

user base, which reinforces that our dataset accurately captures the country's population distribution [93]. This renders our dataset especially interesting for analyzing mobility in the entire country, including regions with diverse population dynamics.

Mobility & Geodemographics. We investigate the distribution of mobility events by examining the number of HOs across districts. Figure 2.5b shows the number of daily HOs per sq. km in each district, together with the population density there (residents per sq. km). This analysis facilitates the characterization of mobility patterns across distinct geodemographic segments (e.g., densely populated urban areas or less populated rural areas). Overall, our findings indicate a strong positive correlation (Pearson correlation of 0.97), between the number of HOs per day and the residential population density in the corresponding district.

As anticipated, dense urban areas exhibit a high number of HOs per square km. For instance, in the district that covers the urban center of the capital, we observe approximately 2.1M HOs per square km each day. In this city the studied network's infrastructure itself comprises more than 500 radio cells per square km. Conversely, in less populated areas the intensity of HOs is significantly lower (60 HOs per sq. km in the least densely populated district). This value is more than 200× lower compared to the district-level mean in the country (13.1k HOs per sq. km daily), reflecting the stark contrast in mobile network activity between highly urbanized and more remote areas.

***Key takeaways**: We infer the home locations of approximately 40M UEs across the studied country to ensure that our data accurately represents the entire population ($R^2 = 0.92$ with census data). By analyzing HOs per square km at the district level, we observe significant disparities – from 2.1M daily HOs per sq. km in the in the center of the capital city to 60 HOs per sq. km in remote areas – highlighting the complexity of managing HOs across different regions.*

2.4. Characteristics of Handovers

Analyzing mobility patterns is crucial for various purposes, including urban planning, social policy design, and optimizing network infrastructure [68], [94]. In this

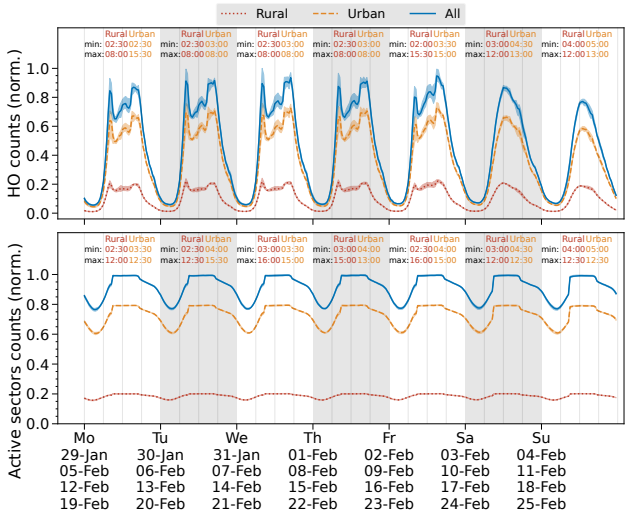


Figure 2.6: Temporal evolution of HOs (top) and active cells (bottom) in urban and rural areas in 2024. Curves show the average HO volume in 30-minute intervals over the four weeks; shadows show the min/max values. All values are normalized by the max HO and cell count observed.

section, we take as reference the three axes of heterogeneity from Section 2.3, and characterize geo-temporal cellular mobility patterns through HOs. We examine the horizontal and vertical HOs across UE types and districts, and investigate how mobility and UE manufacturers relate to HO performance.

2.4.1. Geo-temporal Analysis

First, we analyze HO patterns as a function of geodemographic factors, focusing on the difference between urban and rural areas, classified at the postcode level.[4] This broader urban/rural classification enables us to robustly capture variations in HO dynamics across areas with different demographic characteristics.

HO Patterns. Figure 2.6 (top) shows the weekly temporal evolution (with *30-min granularity*) of HO counts over the 4 weeks (shadows show the min/max values). To adhere to privacy and security guidelines of the MNO, we normalize the HO counts by the max value (over 30-min intervals) of the studied period. The total number of HOs in urban areas represents, on average, 78% of all HOs, which is consistent with findings from other studies [31]. Namely, we discover that 80% of the cells are installed in urban areas, which cover only 49.6% of the total country's territory.

From the daily HO patterns, we observe that weekdays (Mo-Fr) experience higher number of HOs compared to weekends (Sa-Su). Concretely, we find a 33% reduction on average in the peak of HOs during Sundays compared to Fridays. Moreover, we identify the peak HO times during weekdays at 08:00–08:30 and 15:00–15:30 for both rural and urban areas. Also, weekday HO patterns exhibit notable fluctua-

[4] We drop from this analysis 3.1% of the postcodes due to the lack of reliable census information in these areas.

Table 2.2: Statistics per handover and device type.

	Horizontal	Vertical		All HOs (%)
	Intra 4G/ 5G-NSA (%)	4G/5G-NSA to 3G (%)	4G/5G-NSA to 2G (%)	
Smartphones	88.28 ± 0.77	5.84 ± 0.77	< 0.001	94.12 ± 0.77
M2M/IoT	5.73 ± 0.52	0.02 ± 0.01	< 0.001	5.75 ± 0.53
Feature phones	0.13 ± 0.05	< 0.001	< 0.001	0.13 ± 0.05
All devices	94.14 ± 1.29	5.86 ± 0.78	< 0.001	-

tions, with a sharp $\times 3$ increase in the HOs observed from 06:00 to 08:00; this is in contrast to weekends, which have a single peak of mobility between 12:00 and 13:00. During weekdays, after the second peak at 15:00–15:30, the number of HOs gradually decreases (on average 11% per 30 minutes), leading to the minimum at 02:00–03:30 (or 03:00–05:00 over the weekends).

Likewise, Figure 2.6 (bottom) shows the number of active cells – handling at least one HO – over 30-min intervals. As underlined in the sequel, MNOs apply dynamic energy-saving policies to switch off cells when they are not needed to satisfy capacity demand. Comparing Figures 2.6 (top) and 2.6 (bottom), we see that the portion of active cells highly correlates with the HO counts (Pearson correlation of 0.9). Weekdays and weekends present no significant differences in terms of active cells. More precisely, after 08:00 (first peak hour) $\approx$99% of cells remain active until 17:00, when a decrease of $\approx$1% per 30-min is observed, until midnight. As mentioned earlier, we conjecture that this decrease correlates not only with the reduced mobility of the UEs (notice the HO drop at the same hours), but also with the reduced capacity demand (i.e., less user activity) in densely deployed areas, which triggers energy-saving mechanisms to switch off cells that act as capacity boosters [95], [96].

Key takeaways: *HO patterns vary significantly across: (i) urban and rural areas (urban cells account for 78% of HOs, while covering only 49.6% of the territory), and (ii) during weekdays and weekends (33% of difference during peak hours).*

2.4.2. Horizontal vs Vertical Handovers

To understand how devices interact with the different RATs in the network (see Section 2.3), we take as a reference the behavior of devices connected to 4G/5G-NSA (i.e., 4G and 5G-enabled devices). We differentiate three main *HO types*, namely, intra 4G/5G-NSA (horizontal), 4G/5G-NSA $\rightarrow$ 3G (vertical), and 4G/5G-NSA $\rightarrow$ 2G (vertical). Our intent is to characterize how frequently these devices still rely on older RATs, and in which circumstances.

HO Frequencies. Table 2.2 depicts the percentage of the different HO types we registered across UE types. The vast majority of HOs are intra 4G/5G-NSA HOs (94.14%), while vertical HOs – from 4G/5G-NSA to 3G or 2G – correspond to 5.86% and 0.001%, respectively.

Furthermore, smartphones primarily initiate intra-4G/5G-NSA HOs, contributing to 88.28% of the total, with a fallback to 3G occurring in 5.84% of the cases.

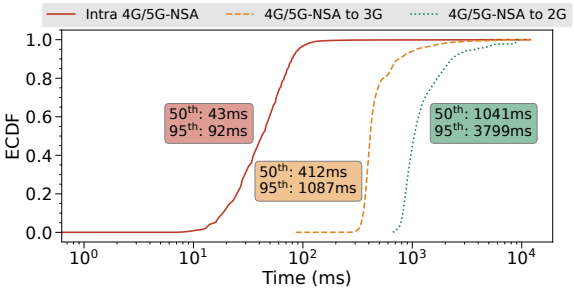


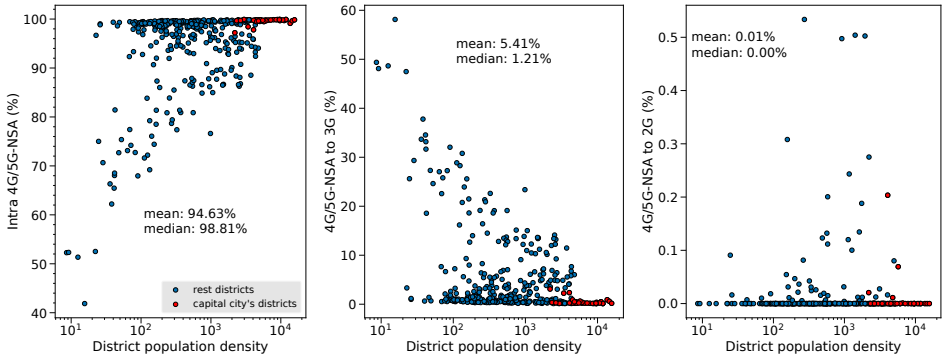
Figure 2.7: HO duration (horizontal vs vertical).

M2M/IoT devices engage mostly in intra 4G/5G-NSA HOs, with a minority transitioning to 3G, a pattern echoed by feature phones on a smaller scale (i.e., 0.13% intra 4G/5G-NSA HOs). This is particularly important given that ≈80% of M2M/IoT devices support only up to 3G (see Figure 2.4b). It is an artifact of the IoT vertical applications employing massive M2M deployments (e.g., smart meter applications), which often require only stationary devices with limited connectivity demands [84].

HO Duration. Figure 2.7 illustrates the signaling times of HOs (see definition in Section 2.2.4), revealing that 95% of intra 4G/5G-NSA HOs complete within ≈90 ms (median of 43 ms). These results align with previous studies [72], [86], [88]. In contrast, HOs from 4G/5G-NSA to 3G are one order of magnitude longer, with a median of 412 ms and their 95th percentile exceeding 1s. The latency further increases for vertical HOs to 2G, where the median time matches the 95th percentile for HOs to 3G (≈1s), and the 95th percentile stretches beyond 3.8s. Even if these HO types rarely occur (see Table 2.2) their large duration reveals a clear negative impact of vertical HOs. We delve into the duration of HOFs in Section 2.5.

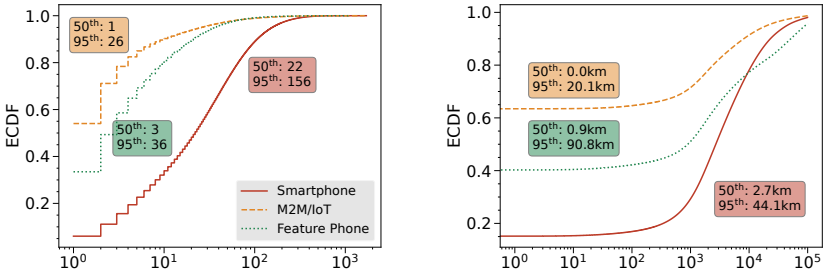
HOs per District. Figure 2.8 provides a comprehensive view of HO dynamics across districts in the studied country. In this way, we are able to pinpoint the areas that are more dependent on newer/older RATs. Notably, densely populated urban districts – which include the districts of the capital city – exhibit a high penetration of 4G/5G-NSA (up to 99.92% of all HOs, see Figure 2.8a), while some less populated rural areas show more transitions to legacy RATs. For example, in the 6% least densely populated districts, HOs to 3G account for 26.5% on average of all HOs, and reach up to 58.1% for a specific remote district (Figure 2.8b). Likewise, the percentage of transitions to 2G remains marginal for most of the districts, with a maximum of ≈0.5% for 4 specific districts. (Figure 2.8c).

Key takeaways: *(i) 94% of HOs are intra 4G/5G-NSA, and are triggered by smartphones. (ii) HOs to 3G/2G take up to 3.8 seconds (pct-95) to execute and still represent 6% of all HOs. (iii) The most densely populated urban areas rely almost exclusively on 4G/5G-NSA for HOs (>99%); less densely populated rural areas still use older RATs (HOs to 3G are up to 58.1% in a remote area and on average 26.5% in the least densely populated districts). This analysis helps the MNO to identify areas where a great volume of 4G and 5G-capable devices are frequently using legacy RATs, thus building a realistic strategy towards their decommissioning.*



(a) (b) (c)

Figure 2.8: Distribution of (a) intra 4G/5G-NSA, (b) 4G/5G-NSA to 3G, (c) 4G/5G-NSA to 2G, HOs across districts. Y-axes have different scales.



(a) (b)

Figure 2.9: Mobility metrics (a) number of cells and (b) gyration (km), across device types.

2.4.3. Mobility across Device Types

This section examines the relation between UEs' mobility and their HO performance. We first characterize mobility metrics across different device types. Then, we analyze the relation between the mobility metrics of the UEs and the HOF rate that they experience, serving as an indicator of how these UEs suffer from disruptions.

We take as a reference the two mobility metrics described in Section 2.2.4: radius of gyration and number of cells. Figure 2.9 shows the empirical cumulative distribution function (ECDF) of both mobility metrics across device types. Overall, we observe that smartphones are considerably more mobile than the two other types, exhibiting a median of 22 distinct visited cells per day, and a median radius of gyration of 2.7km. Conversely, the majority of M2M/IoT devices and low-tier feature phones are more static, with median values of 1 and 3 visited cells per day, respectively, and a median gyration of 0.0km and 0.9km. This reflects that these UEs are mostly static, and the few HOs that these devices experience are typically between cells in the same sites.

Given the heterogeneity of M2M/IoT vertical applications, there are devices in the 95$^{\text{th}}$ percentile that show high mobility, with gyrations of 20.1km for M2M/IoT

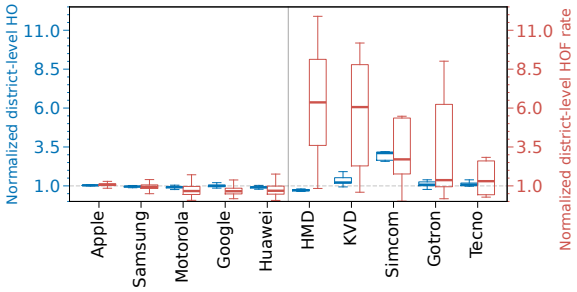


Figure 2.10: Normalized district-level HOs (left) and HOFs (right) per UE manufacturer. Boxplots include top-5 UE manufacturers and the top-5 UE manufacturers with greater median HOF values.

devices (see Figure 2.9b). These UEs mainly correspond to modems and routers that are deployed in fast-moving vehicles (e.g., trains), integrated into modern cars, embedded in industrial equipment, or wearable IoT devices carried by users who typically travel long distances. While feature phones (green line) surpass smartphones (red line) at around the 80$^{\text{th}}$ percentile, the former comprise only about 1% of the total UEs, while the latter makes up approximately 60% (see Figure 2.4a).

Manufacturer Impact. We assess whether higher HO counts and HOF rate correlate with specific UE manufacturers (e.g., due to a suboptimal mobility management implementation). We observe that the distribution of UEs is remarkably unbalanced across the studied country, e.g., Samsung and Apple smartphones are considerably more common in densely populated areas. To make a fair comparison and account for potential deviations due to the area itself (e.g., population, deployment density) – see Figure 2.5b – we create a metric that makes a unified comparison of UE manufacturers at the district level. That is, in each district we get the average HOs per UE for a specific manufacturer and divide it by the average HOs per UE including all manufacturers within that district (i.e., *normalized district-level HO*);[5] and similarly for the HOF rate (i.e., *Normalized district-level HOF rate*). A value greater than 1 indicates that UEs of a specific manufacturer generate more HOs (or HOF rate) on average than the total population of UEs in the same district.

Figure 2.10 shows the results for the top-5 manufacturers in the studied country (see Figure 2.4a), as well as the 5 manufacturers exhibiting the highest *Normalized district-level HOF rate*, based on the median behavior across all districts (see boxplots). For the top-5 manufacturers ratios are close to 1, which means that devices behave similarly to their peers in the same district, both in terms of HOs and HOF rate. Specifically, we observe that Apple smartphones, the most popular ones (≈32% of all UEs), generate slightly more HOs per UE and HOF rates than other devices (respectively +4% HOs and +8% HOF w.r.t. their peers). Likewise, Google smartphones are the ones that experience the smallest HOF rates (-27% w.r.t. their peers). Moreover, we find that some manufacturers show high HOF rates, such as KVD smartphones or HMD feature phones (+600% HOF rate), as well as others that generate higher HOs per UE, such as Simcom M2M/IoT (+293% HOs per UE).

[5]Some manufacturers have few devices in specific districts. We exclude district-manufacturer pairs that account for <1k devices.

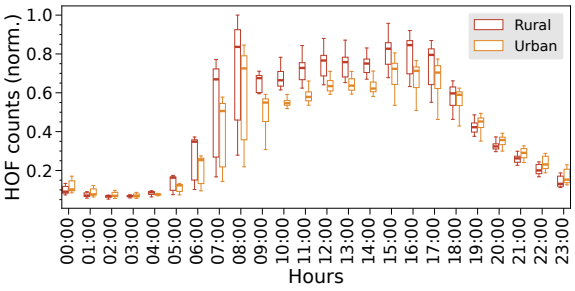


Figure 2.11: HOF counts per hour in urban and rural areas, normalized separately with the number of active cells in each class (i.e., urban/rural).

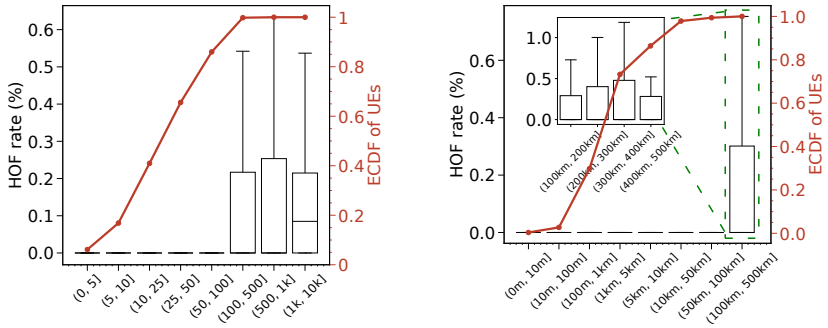
Key takeaways: *(i) Different UE types exhibit different mobility patterns; smartphones are, on median, connecting to more distinct cells (22 cells per day), with a daily median radius of gyration of 2.7km. (ii) The most popular device manufacturers exhibit a consistent behavior in terms of HOs (±10% of variation between them). While HOF rates are considerably small, some manufacturers (e.g., Google) exhibit lower HOF rates (-27%) than other manufacturers. For some niche manufacturers, we find high HOF rates (up to +600%) and HO counts (up to +293%). Based on these results, we conjecture that manufacturer-specific mobility management implementations and application-specific usage correlate with HO performance.*

2.5. Handover Failure Analysis

This section provides an in-depth analysis of HOFs. Initially, we examine the daily patterns of HOFs and their correlation with key mobility metrics. Next, we explore the causes of HOFs from the network's perspective and present modeling techniques that assess how network features at the radio cell level influence the HOF rate. Our analysis puts the spotlight on the need to reduce the network's complexity by decommissioning legacy RATs.

2.5.1. Patterns and Impact

HOF Patterns. We analyze the temporal evolution of HOF rate (see Section 2.2.4) along the day, aggregating data over the 4-week period. Figure 2.11 presents the hourly evolution of HOFs, where boxplots aggregate data from all active radio cells at a specific hour. To comply with the privacy policies of the MNO and account for the different distribution of cells in rural and urban areas, we have separately normalized the hourly HOFs for rural (urban) areas with the number of active cells observed in rural (urban) settings (see Figure 2.6, bottom). Overall, we observe that HOFs reach a local peak during the morning commuting time [7:00–9:00), and a lower local peak can be observed during the afternoon commuting time [15:00–18:00). Moreover, urban areas experience fewer HOFs compared to rural ones, especially during peak hours; e.g., the median HOF count is 32.4% higher in rural areas than in urban ones during [7:00–8:00). We conjecture that this pattern is likely due to the more limited 4G/5G coverage in these areas, which makes 4G and



(a) (b)

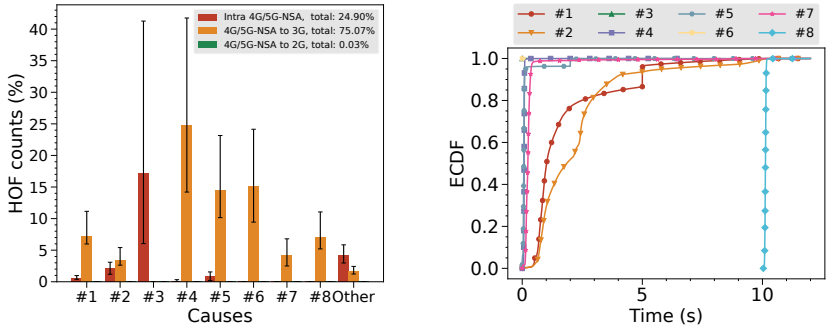
Figure 2.12: HOF rate (left y-axis) and ECDF for the number of UEs (right y-axis) w.r.t. binned device-level mobility metrics (log scale) of (a) number of cells and (b) gyration.

5G-capable devices fall back more frequently on older RATs (i.e., 2G, 3G) to keep connectivity. We further delve into this aspect in Section 2.5.3 by modeling the negative impact of vertical HOs on HOFs and inspecting the causes of such failures.

HOFs & Mobility. We explore the association of radius of gyration and number of cells with the HOF rate. In Figure 2.12, the left y-axis shows the daily average HOF rate for the UEs according to the number of cells (Figure 2.12a), or radius of gyration (Figure 2.12b). Meanwhile, the right y-axis displays the ECDF for the number of UEs along the bins in the x-axis (in log scale).

Concretely, Figure 2.12a shows that the HOF rate is close to zero for 87% of the UEs, which connect to 100 or less cells per day. For the remaining 13% of the UEs (traveling >100 cells), the HOF rate slightly increases (up to 0.4% for pct-75), but the median is still close to zero; except for <0.0001% of the UEs that connect to >1k cells and have a median HOF rate of 0.1%. Similarly, from Figure 2.12b, HOFs mainly occur in devices that move within a radius higher than 100km (which is the case for 0.007% of the devices, see the right y-axis), with the HOF rate reaching up to 0.4% (pct-75). Yet, the median HOF rates remain close to zero for all bins. We observe that the devices with increased mobility (>100 visited cells, >100km radius of gyration) are mostly smart/feature-phones (90%) and M2M devices (10%) – such as modems, routers and IoT wearables – attached or carried in fast-moving vehicles, like trains. It is interesting to note that UEs with <10km radius of gyration and <50 visited cells, which show almost zero HOF rate, include a very similar share of UE types (85% smart/feature-phones and 15% M2M); which confirms that the increase in HOFs in UEs with higher mobility metrics cannot be explained by an unequal distribution of UE types in this group.

Key takeaways: *(i) Rural areas suffer from 32.4% more HOFs during peak hours than urban. (ii) A small number of UEs with high mobility metrics daily (>100 visited cells, >100km radius of gyration) experience a non-negligible HOF rate (0.4% for pct-75); the number of visited cells and the radius of gyration are good predictors to flag UEs that can potentially experience high HOF rates.*



(a) (b)

Figure 2.13: (a) Percentage of HOF causes w.r.t. the total HOFs; (b) Distribution of HO signaling time per cause.

2.5.2. Causes

We study the HO failures using cause codes delineated by the 3GPP standards [79], [83] and the antenna vendors. In total, we collect 1k+ different causes for the failures. Our causes analysis complements prior studies that had exclusively focused on the user side, being mostly coarser, and solely for specific devices and failure types [64], [73]. In Figure 2.13a, we present the HOF counts in percentage, by calculating the HOF for each cause and dividing it by the total HOFs per day. We also plot alongside the minimum and maximum values observed in this period (i.e., 4 weeks). Our analysis reveals that (*i*) 92% of all HOFs occur because of 8 causes from the 1k+ that exist, and that (*ii*) 75% of all HOFs occur in transitions from 4G/5G-NSA to 3G, with the remainder (i.e., ≈25%) associated with intra 4G/5G-NSA HOs. HOFs attributable to transitions to 2G represent 0.03% of all. This distribution highlights the real-world implications of managing a layered cellular deployment, where $<6\%$ of handovers are vertical handovers to 3G, and the remaining 95% are intra 4G/5G-NSA handovers. We present next the 8 most common handover failure causes. Additional insights for the reasons for HOFs in rural/urban areas, different smartphone manufacturers, and UE types can be seen in Figure 2.14.

• **Cause #1**: "The source cell canceled the HO" relates to the cancellation of an ongoing or prepared handover. HO Cancellation procedure [79] can occur for several reasons, such as timeouts on the MSC, cell site, or issues with the size of the Forward Relocation Request [83]. This cause is predominantly observed in HOs to 3G, affecting 7.3% to 11.2% of cases daily, which is significantly higher compared to intra 4G/5G-NSA and 4G/5G-NSA to 2G HOs ($<1\%$ per day). We observe that this failure cause affects evenly all UE types, but is 50% more prevalent in rural than in urban areas (see Figure 2.14).

• **Cause #2**: "The signaling procedure was aborted due to interfering S1AP Initial UE Message [79]". This error involves the interruption of the signaling process by an initial message to the MME, which includes critical user information and service requests. This issue affects 2% of intra 4G/5G-NSA HOs and 3.4% of HOs to 3G, but not HOs to 2G.

• **Cause #3**: "Signaling procedure was rejected due to invalid target cell ID" occurs when the target cell ID is not recognized or if there are configuration issues with the

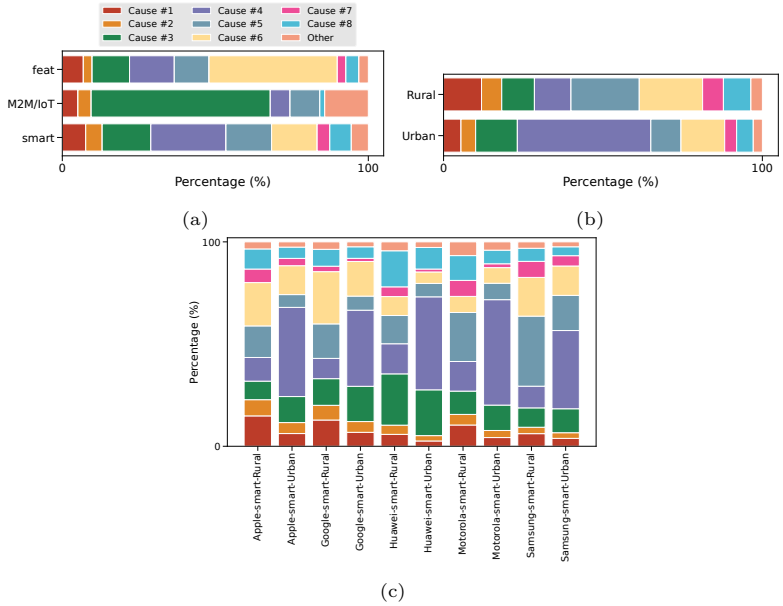


Figure 2.14: Stacked bars showing the percentages of HOF causes (each bar adds to 100%) vs (a) device types, (b) urban/rural areas, and (c) top-5 smartphone manufacturers.

MME pool area (i.e., a collection of MMEs configured to serve any common part of a radio network). This is the main reason for failure in intra 4G/5G-NSA HOs, accounting for an average of 17.2% of the failures, and reaching up to 41.3%. From this cause, 59% of M2M/IoT devices fail (see Figure 2.14).

• **Cause #4**: "Load on target cell is too high" indicates that the target cell cannot accommodate the HO due to resource constraints. It is the most common reason for failure in HOs to 3G (up to 42.3% of all HOFs), affecting 25% of the failures per day, on average. It happens mainly during peak hours in dense urban areas (see Figure 2.6), causing 42% of the total HOFs there (see Figure 2.14).

• **Cause #5**: "MME detects a HO-related failure in the target MME, SGW, PGW, cell, or system"; these types of infrastructure-related outages occur for 14–23% of HOs to 3G, and for 0.8–1.6% of intra 4G/5G-NSA HOs. This cause does not pinpoint precisely the reason that the HOF occurred; however, it is important to note that this is the extent of information that is available to the MNO.

Causes #6, #7 #8 are specific to HOs from 4G/5G-NSA to 3G. We provide more information in the sequel.

• **Cause #6**: "The Single Radio Voice Call Continuity (SRVCC) service is not subscribed by the UE" affects 15.2% of HOs to 3G on average, peaking at 24.1%. SRVCC is a scheme used with VoLTE (Voice over LTE) and ensures seamless handovers of voice calls from packet-switched (PS), like 4G, to circuit-switched (CS) networks, like 2G and 3G [97], [98]. We note that this failure occurs primarily in rural areas and in feature phones, where the MNO still relies mostly on 3G to ensure the support of voice services (see Figure 2.14).

• **Cause #7**: Like Cause #6, Cause #7 is associated with SRVCC HOs, and it occurs when "the MSC responds with PS to CS Response with cause indicating failure" during SRVCC HO preparation; it affects about 4.2% of all HOs [97]. We note that it affects almost no M2M/IoT device and occurs twice as often in rural than in urban areas (see Figure 2.14).

• **Cause #8**: "No Forward Relocation Complete or Notification was received before the max time for waiting for the relocation completion expires", affecting 7.1% of HOs. Forward Relocation Complete message is sent to the source MME/SGSN to indicate the HO has been successful. We observe this cause ×3 in M2M/IoT devices w.r.t. smartphones and feature phones (see Figure 2.14).

HOF Duration per Cause. Figure 2.13b complements the HOF analysis by delineating the HO duration associated with the 8 causes of failure. Causes #3 and #6 result in failures that prevent the initiation of the HO (i.e., signaling time equals 0 ms), with the former attributed to an invalid target cell ID and the latter to SRVCC service not being permitted for the UE. Cause #4, linked to insufficient resources in the target cell, exhibits a median duration of 81 ms and a 95th percentile of 97 ms. More prolonged delays in HO signaling are caused by Causes #1 and #2, where the HO is halted due to cancellation by the source or interference, respectively, leading to medians of 1–2s and 95th percentiles of 5–6s. Notably, Cause #8, associated with timeout failures, demands the most extended signaling, with a median >10s and 95% of cases occurring in < 10.2s. Our study aligns with existing works, such as [64], [99], in demonstrating the increased duration involved in HOFs.

Key takeaways: *(i) Despite the prevalence of 4G and 5G, 75% of all failures concern HOs from 4G/5G-NSA to 3G, and 25% of them happen due to high load in the target cell. (ii) 59% of M2M/IoT UEs and 42% of feature phones fail due to Cause #3 and Cause #6, respectively. (iii) 42% of HOFs in urban areas occur due to Cause #4, while Causes #5 and #6 account for 20% each, in rural areas. (iv) The duration of HOs that fail due to timeouts (Cause #8) or cancellations (Cause #1) exceeds on average 2s (for the former case it reaches 10s). These numbers highlight the noticeable outage duration caused by HOFs in the network.*

2.5.3. Statistical Modeling

We aim to understand which cell-level features contribute to HOFs, by isolating and combining the effect of various parameters and ensuring our findings in the previous sections are robust against potential biases or unaccounted variables. We reorganize the dataset using as dependent variable the *daily* HOF rate of each source cell, and use as covariates the cell-level features in Table 2.3. This creates 6.7M observations. Our hypothesis is that the HO type is the primary factor influencing HOF rates. Note that while HOs to 3G amount to only 5.86% of the total HOs (Section 2.4.2), they are responsible for 75% of all HOFs (Section 2.5.2).

A first look into the data demonstrates that HOs to 2G and 3G are associated with substantially higher failure rates with medians of 21.42% and 5.85% respectively, compared to 0.04% for HOs to 4G/5G-NSA; and this disparity persists even when we focus on the failed HOs and filter outliers, see Appendix 2A. We further perform an analysis of variance (ANOVA) test [100] (log-transforming the HOF

Table 2.3: Regression covariates.

Feature	Values
Number of HOs per day	≥ 0
RATs	4G/5G-NSA, 3G, 2G
District population	≥ 0
Cell Region	West, South, North, Capital area
Area Type	Rural / Urban
Antenna Vendor	4 vendors (V1, V2, V3, V4)

Table 2.4: Linear model coefficients for $\log(HOF\,rate)$.

Feature	Coef.	95% CI	P-value
Intra 4G/5G-NSA	-2.77	-2.77, -2.76	0
4G/5G-NSA→3G	5.12	5.12, 5.13	0
4G/5G-NSA→2G	6.82	6.76, 6.88	0

rates) which verifies the significance of this effect ($p < 0.001$); and the same conclusion is reached using the Kruskal-Wallis test [101]. We repeat these tests, with the same findings, even when controlling for variations in the area and antenna vendor.

Accordingly, we use a generalized linear regression model (with log transformation) to quantify the effect of RAT on HOFs. We first run a univariate model to facilitate interpretation. We find that HOs to 3G (2G) increase the HOF rate by 166% (915%, respectively) compared to HOs to 4G/5G-NSA, see Table 2.4. We repeat this analysis while controlling for the other covariates and filtering the outliers (HOF rate<50%, number of HOs per day in $[50, 30k]$), finding the same result with slightly smaller intensity (coefficients of 5.48 and 4.77 instead of 6.82 and 5.12), as can be seen in Table 2.5. From the remaining covariates, the antenna vendor has a significant but smaller effect, which we also verify with an ANOVA test. These findings are also robust to alternative models (step-wise covariate selection and removing HOs to 2G), including also a quantile linear regression model. The details of these additional tests are deferred to Appendix 2A.

Key takeaways: *By modeling HOFs and investigating different covariates (see Table 2.3), we verify our hypothesis that the HO type is the main factor shaping the observed HOF rates: HOs to 3G (2G) increase the HOF rate by 166% (915%, respectively) compared to HOs to 4G/5G-NSA.*

2.6. Related Work

In terms of measurement approaches, the vast majority of studies rely on (mainly rooted) UEs and collect traces from their cellular modems [30], [32], [63], [64], [66]–[73]. For instance, [63] and [68] build their mobility analysis upon Mobile Insight [102] with rooted phones, while [66], [71] use the G-NetTrack Pro monitoring tool [103]. These solutions are confined to certain chipset manufacturers and have limited data collection granularity (orders of seconds, instead of milliseconds as in the current study). Other works study mobility patterns in one [65] or a few

Table 2.5: Regression Summary: Linear Model, All Covariates.

Feature	Coeff.	Std Err	t value	Pr(> |t|)
(Intercept)	−3.10	0.0217	−143	0
HO type: 4G/5G-NSA→2G	5.48	0.118	46.4	0
HO type: 4G/5G-NSA→3G	4.77	0.00150	3169	0
Number of daily HOs	$-2.84 \cdot 10^{-5}$	0	−331	0
Area Type: Rural	0.260	0.00272	95.5	0
Area Type: Urban	0.190	0.00258	73.4	0
Antenna Vendor: V2	0.115	0.00173	66.7	0
Antenna Vendor: V3	0.719	0.0203	35.3	0
Antenna Vendor: V4	0.0629	0.0222	2.84	0.49
Cell Region: North	−0.0728	0.0216	−3.57	$4.05 \cdot 10^{-6}$
Cell Region: South	−0.0168	0.00166	−10.1	$2.28 \cdot 10^{-6}$
Cell Region: West	0.398	0.0204	19.5	$3.89 \cdot 10^{-66}$
District population	$-1.75 \cdot 10^{-7}$	0	−61.6	0
$N = 3857074$, RMSE=1.023, $R^2 = 0.8265$, AIC=11121590				

cities [32], such as Minneapolis [69], [104], Chicago [69], [70], Atlanta [69], and Rome [105]. These studies provide valuable information, yet their spatial focus does not facilitate insights across larger scales (e.g., countrywide) and in varied settings (e.g., urban/rural areas). The works of [64], [67], [99] conduct extensive 4G performance measurements on high-speed rails in China, and [30], [72] study mobility management policies in 4G/5G. The collected data in these relevant works are related to certain mobility patterns, and a subset of users, and do not contain network-side data.

Our study, on the other hand, records *all* mobility events from a commercial MNO network with ≈40M UEs connected, with millisecond granularity, during 4 weeks, and for the entire territory of a European country; it is not limited to specific routes, cities, mobility modes, or user types. To date, only a few works study HOs from the operator's perspective, as this involves technical challenges [63] and requires in-network measurements (see Figure 2.2). Namely, [106] suggests an approach to categorize and minimize undesired Ping-Pong (PP)[6] HOs based on a restricted dataset with 1.7k UEs; and [31] investigates PP HOs using 13 days of data from a network operator in a Mediterranean area. Our study differs from these works due to the scale, coverage, and granularity of measurements (all active connections of a top-tier MNO at the country level; see Table 2.1), and due to the fusion of different datasets (about UEs and population) that allows drawing fresh insights, e.g., about the impact of HOs and HOFs on different RATs, device types, and areas.

Specifically, in terms of measurement results, our findings about the HO duration are on par with previous studies, e.g. [30], [64], [72], and provide additional insights, e.g., about the effect of RAT, finding that inter-RAT HOs are the most impactful. Several studies measured the volume of HOs [32], [69], [91], finding, e.g., horizontal HOs to be more frequent in 5G-SA and 4G and vertical HOs in 5G-NSA [91]. Here,

[6]PP HO occurs when a UE is handovered from a source to a target cell, and then back to the source, under a short, predefined time.

we enrich these results by dissecting the HOs per RAT and UE manufacturer/type, analyzing their temporal pattern over 4 weeks, and their relation to the demographic distribution over the studied country, with district granularity (300+ districts), thus refining the typical urban/rural categorization of prior studies [31].

Furthermore, leveraging our unique network-side dataset, we characterize HOFs (cause *and* duration) using detailed antenna vendor-specific information. Prior studies, inhibited by their UE-side data, have mainly studied the effect of user speed on HOFs [68] or used coarser categorization, e.g., 2 possible causes [64], or analyzed general connectivity failures for specific devices [73]. Given that HOs were found to affect significantly the user-perceived network performance, our work can inform the design of new HO policies, such as [91], [107], [108], and guide the optimization of network deployment and RAT upgrades.

2.7. Conclusion

This chapter provides a countrywide analysis of the heterogeneity of modern cellular networks and their mobility management, leveraging data from a top-tier MNO in a European country. By tracking ≈40M users over four weeks, our findings highlight the critical impact of spatio-temporal factors, RATs, device types, and manufacturers on network complexity. Based on these, we study horizontal and vertical HOs and HOFs, specifying the impact of the latter, and modeling them using statistical methods.

These findings are crucial for understanding and developing new HO mechanisms, as well as identifying groups of UEs and areas that require enhanced support. In this way, our analysis lays the groundwork for the next chapter, which focuses on improvements in handover performance, ensuring that the promise of 5G and subsequent generations of cellular technologies can be fully realized.

3

Mobility Management through Smooth Handovers

The previous chapter revealed the complexity of modern cellular networks, where heterogeneous radio access technologies (RATs), users, and regions impact mobility management and specifically, handovers (HOs). In this chapter, we take the next step: we model HOs and design an algorithmic framework to optimize them.

To inform our modeling, we extend the Mobile Network Operator (MNO) datasets analyzed earlier with targeted measurements and crowdsourced signal data. This new data allows us to (i) quantify the impact of HOs and HO failures (HOFs) on key user and network performance indicators (KPIs), such as packet loss and throughput, (ii) identify and measure which cell- and user-specific features affect HO delays, which we later incorporate into our modeling, and (iii) evaluate the performance of our proposed algorithm in real-world settings. Leveraging the framework of Smoothed Online Learning (SOL) that maximizes a performance criterion while explicitly discouraging frequent decision changes, we approach the HO optimization task as a user-cell association problem. In this setting, changes in association decisions translate into HOs that may incur prolonged delays. Unlike prior works, our formulation incorporates user- and cell-specific features on HOs and their induced delays, and does not assume future knowledge of signal quality or user mobility/trajectories. Our online meta-learning algorithm, aligned with the O-RAN paradigm, provides robust dynamic regret guarantees even in challenging environments and demonstrates superior performance in multiple scenarios using synthetic and real-world data.

The content of this chapter has been published in:

M. Kalntis, A. Lutu, J. O. Iglesias, F. A. Kuipers, and G. Iosifidis, "Smooth Handovers via Smoothed Online Learning," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2025.

3.1. Challenges and Contributions

The goal of this chapter is to design a novel UE-cell association and smooth HO control mechanism that is both *effective* and *robust*, overcoming key limitations of previous works (see discussion in Section 3.7) that might rely on heuristics or strong assumptions; thus, filling a key gap in the literature and paving the road for the next generation of mobility management.

Building on the countrywide analysis presented in the previous chapter, we revisit and extend the MNO's datasets with targeted 1-week measurements (see Table 3.1) and crowdsourced signal data for the source cells. This enriched view enables us to *(i)* quantify the impact of HOs and HOFs on KPIs such as packet loss and throughput, *(ii)* identify which cell- and user-specific features drive HO delays, and *(iii)* evaluate the performance of our algorithm in real-world settings, with the crowdsourced signal measurements we collect.

Leveraging these insights, we introduce a realistic system model that captures the impact of the different HOs and mitigates them while maximizing the network throughput. Our model aligns with recent works [33], [34], [36]–[38], [109], which we extend substantially by accounting for the network and HO diversity, and importantly, by dropping requirements for access to future signal-to-interference-plus-noise ratios (SINRs) and UE mobility patterns. It is commonly accepted that this is a strict condition that limits the applicability of such solutions. Additionally, we do not assume that the relevant UE/cell parameters are stationary-perturbed since rapid and unpredictable channel fluctuations are becoming increasingly common in heterogeneous networks and mobile services. In fact, our perturbation model is an adversarial one, where an attacker can even select the various random parameters, and still, all results and guarantees remain valid.

With this in mind, we turn the problem on its head and study HOs through the lens of online convex optimization (OCO) [16]. Namely, we model the UE-cell associations as dynamic decisions that the network controller updates in a time-slotted fashion, where successive (de)associations induce undesirable (sometimes necessary) HOs, or equivalently, switching costs. A natural framework for this setting is that of *smoothed online learning* (SOL) [110], which maximizes a performance criterion while reducing the decision changes (i.e., switching cost). This enables the controller to be oblivious to the SINRs for each UE-cell pair when deciding the associations, and the throughput these will achieve; indeed, it is challenging to predict accurately or know the SINRs over a time window of several ms/s [111]. Still, following a rigorous analysis, we show that our learning algorithm ensures *sublinear dynamic regret*, i.e., its gap w.r.t. an ideal oracle that has full information about the future diminishes with time [25].

Our model is informed by, and aligned with, the O-RAN paradigm, and the proposed algorithm can be implemented as xApp in *near-real-time* (running time 10ms–1s) [12]; the reader is kindly referred to [111], [112] for the involved interfaces and detailed steps. Finally, to verify the robustness of our proposed solution, we evaluate its performance on simple and extreme (i.e., adversarial) synthetic scenarios in accordance with related work, as well as in real scenarios using signal measurement from crowdsourced data (i.e., measured on the field).

Table 3.1: Mobility Dataset Size for European Country.

# of cell sites, # of radio cells	> 26k, > 370k
# of UEs measured	≈ 40M
# of handovers (daily)	> 1.7B
Measurement period	24-Jun-2024 to 30-Jun-2024 (1 week, 7 days)
Trace size (daily)	≈ 8 TB

In summary, the contributions of this chapter are:

• We present HO and network statistics by using a 1-week dataset from a tier-1 MNO. We highlight the network heterogeneity and identify key factors impacting smooth HOs. We are the first to optimize HOs from the MNO's perspective, which sets the basis for realistic HO optimization models.

• We model the HO optimization as a *smoothed online learning problem* where the HO delays depend on the RAT and UE type, and assume no prior information for the channels and mobility patterns. This approach departs from related work and its (often simplifying) modeling assumptions.

• We design a scalable (near-real-time) algorithm that achieves sublinear dynamic regret, and we characterize its performance w.r.t. system parameters. We also propose extensions for the case of time-varying HO delays and the case of available (untrusted) forecasting tools.

• We create a simulator using our real, crowdsourced radio signal quality measurements, as well as actual cell and UE information, and evaluate our solution against meaningful benchmarks; e.g., we find up to ×79.6 lower HO cost than previous works, without sacrificing throughput.

3.2. Data Collection and Analysis

Datasets. To demonstrate the network's heterogeneity and identify key factors that affect seamless handovers (and thus are essential for our system model), we collect passive measurements from 24-June-2024 to 30-June-2024 (Table 3.1) using commercial tools available within the MNO's infrastructure, as described in Chapter 2. Apart from collecting more recent datasets than those in the previous chapter, we also capture new datasets that reflect the impact of HOs and HOFs on user and network KPIs, such as packet loss and throughput, and commercial crowdsourced data with ms granularity of radio signal measurements to assess the performance of our algorithm. At the time all these datasets were captured, the majority of traffic for the studied MNO used the 5G-Non-Standalone (NSA) deployment [77]. From a mobility management perspective, 5G-NSA and 4G are identical, as the former relies on the 4G Evolved Packet Core (EPC); thus, we use the term "4G/5G-NSA".

Effect of HOs on KPIs. We first study how HOs and HOFs impact some key network KPIs, which the operational team uses to assess the network's health and evaluate the quality of service to end-users. For confidentiality, we normalize each KPI by its median from all days. Also, we discard the outliers, i.e., cells with daily HOFs >140 (0.1% of data). Figure 3.1 depicts the daily HOFs per cell and their impact on downlink (DL) packet loss and user throughput per day. We observe a decrease in normalized average user throughput with increasing HOFs; e.g., (0,

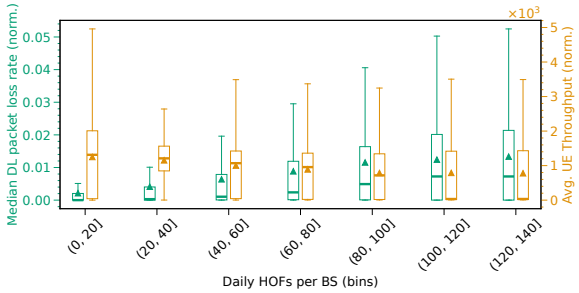


Figure 3.1: Median normalized DL packet loss (left, green y-axis) and normalized average UE throughput (right, orange y-axis) vs binned daily HOFs. Triangles and bold horizontal lines show the mean and median, respectively, in each boxplot.

20] HOFs lead to a normalized mean throughput of 1.25k (normalized value, i.e., does not mean Kbps or Mbps), which declines by 37.6% to 0.78k for (120, 140] HOFs per day. Similar studies (e.g., [32], [69], [72]) have reported UE throughput but only for specific UE types (smartphones) and manufacturers. Simultaneously, the normalized median DL packet loss rises from a mean of around 0.002 to 0.013 when daily HOFs increase from (0, 20] to (120, 140], respectively. The higher mean compared to the median in each boxplot indicates a few UEs with larger losses compared to the majority.

We verified the findings of Figure 3.1 with a generalized linear regression model using as dependent variable the DL packet loss rate and as independent variables the number of HOs and HOFs while controlling for all key covariates; namely, the HO type (inter/intra RAT), district population, cell type/vendor/transmission power, and the district. This ensures the packet loss effects are indeed due to HOs and HOFs, and not to some latent factor. We find that 1% increase of HOs in a cell, increases by 0.02% the packet loss rate, on average, for the served users; and even worse, 1% increase in HOFs increases by 0.6% this loss. A similar model found that, all else being equal (including total uplink/DL volume and physical resource blocks), a HOF increase of 1% reduces by 0.008% the average user throughput. And this drop is more pronounced in cells with few HOFs, as Figure 3.1 shows.

Network & UE Heterogeneity. Using the GSMA devices database, we discern the eight most crucial *UE types* (dongle, IoT, feature phone, modem, smartphone, tablet, WLAN router and wearable), and infer their connectivity capabilities (i.e., up to what RAT they support) from their frequency bands. Given the dominance of 4G/5G capable models (98.5%), depicted in Figure 3.2, we study the HOs from 4G/5G-NSA to the same or older RATs. In these *HO types* (i.e., Intra 4G/5G-NSA, 4G/5G-NSA to 3G, 4G/5G-NSA to 2G), approximately 94% are Intra 4G/5G-NSA, caused by a wide range of devices. HOs to 3G and 2G hold an important 6%, mainly from smartphones, modems, and tablets, magnifying the heterogeneity in these dimensions as well. As we show in the sequel, the HO delay in the older RATs is ×5–40 higher. Network heterogeneity is also extensively analyzed in [113].

Figure 3.3 and Figure 3.4 illustrate the histograms and probability densities of different HOs and device types, respectively, w.r.t. the HO delay. From Figure 3.3,

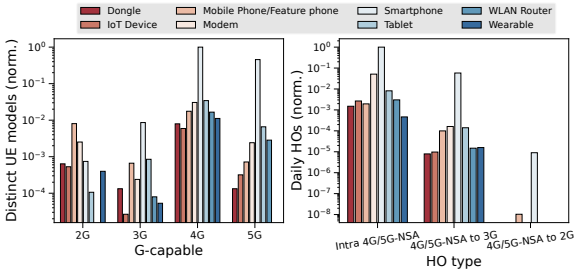


Figure 3.2: **Left**: number (norm. by max) of different UE models and their RAT capabilities (up to 2G, 3G, 4G, 5G). **Right**: HOs (norm. by max) per day each of the UE model executes, and to what RAT.

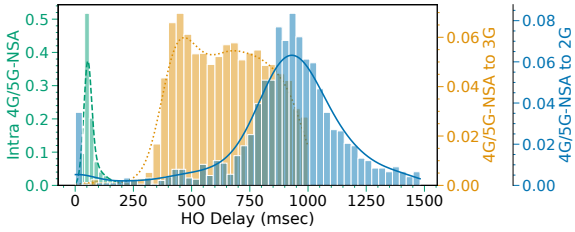


Figure 3.3: Histogram (bars) and distribution (line) of the HO delays for each HO type (all same-colored bars sum to 1).

we observe that the median Intra 4G/5G-NSA HOs require approximately 50 ms, while the HOs to 3G range from 400 to 950 ms and to 2G from 750 to 1100 ms; thus, especially in the older RATs, the distribution is significantly more spread. Moreover, from Figure 3.4, we deduce the different distributions of the HO delays for each device type. It is interesting to observe the different means and variances of each UE type; for instance, modems and IoT devices require, on average, 75 ms and 73 ms, respectively, but around the same number of these start from 50 ms and reach 110 ms; smartphones, on the other hand, need 50–62 ms. Consequently, *accounting for both the UE and HO type is essential when optimizing the HO delay.*

3.3. System Model and Problem Statement

We consider a heterogeneous cellular network comprising a set $\mathcal{J}$ of J cells serving a set $\mathcal{I}$ of I users (UEs) in the downlink. Each cell is characterized by an array of features such as its operating frequency, RAT generation (2G-5G), location, etc. Similarly, the set of UEs comprises smartphones, feature phones, IoT devices, and so on. The network is managed centrally by a network controller, in the spirit of recent O-RAN architecture proposals [12]. The system operation is time-slotted where we index the time slots with t and without loss of generality assume the slots have unitary length. These slots refer to the UE association intervals, which subsume other resource scheduling time slots (e.g., for power control). We study the system for a set $\mathcal{T}$ of T slots. The key metric for the association decisions is the SINR for

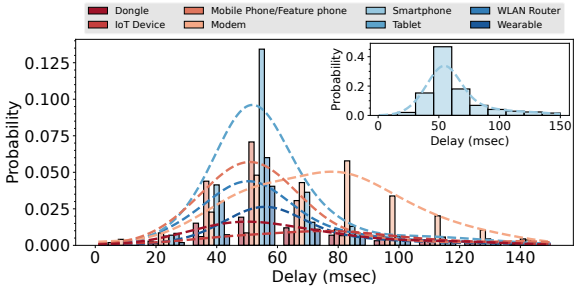


Figure 3.4: Histogram (bars) and distribution (line) of the HO delays for each UE model (all same-colored bars sum to 1).

the signal delivered by cell j to user i:

$$s_{ij}(t) = \frac{p_j \phi_{ij}(t)}{\sum_{k \in \mathcal{B}_j} p_k \phi_{ik}(t) + w_j \sigma^2},$$

where p_j is the transmit power of cell j, $\mathcal{B}_j$ the set of cells that operate in the same frequency as j, $\phi_{ij}(t)$ the channel gain (including pathloss, shadowing, and antenna gains), w_j the bandwidth of cell j, and σ^2 the power spectral density. In line with previous works [33], [34], [36], [109], $s_{ij}(t)$ is the average SINR over the slot t (the UEs report multiple SINRs during each slot [33]).

The rate that each user $i \in \mathcal{I}$ associated with cell $j \in \mathcal{J}$ achieves during slot t, can be expressed as:

$$r_{ij}(t) = \frac{c_{ij}(t)\big(1 - d_{ij}(t)\big)}{y_j(t)}, \tag{3.1}$$

where $y_j(t)$ is the total number of UEs that cell j needs to serve during t, and $c_{ij}(t) = w_j \log_2\big(1 + s_{ij}(t)\big)$ is the maximum possible rate for i if it was using cell j exclusively. The rate $r_{ij}(t)$ is discounted by the service disruption time $d_{ij}(t)$, which models any HO delay associated with the assignment of UE i to cell j. Clearly, this delay is negligible if i was already associated with cell j; is pronounced when the target cell is of different RAT; and is even larger when there is a HO failure (see Section 3.2). We normalize $d_{ij}(t) \leq 1, \forall i, j$, to express the *portion* of slot the UE was not receiving service. Equation (3.1) assumes the cell resources are allocated fairly across the active users via, e.g., a round robin or proportional-fair scheduler [114]. As will become clear, our analysis also holds for other scheduling schemes.

We denote with $x_{ij}(t) \in \{0, 1\}$ the association of user i with cell j in the beginning of slot t and define the vector $\boldsymbol{x}_t = \big(x_{ij}(t) \in \{0,1\}, i \in \mathcal{I}, j \in \mathcal{J}\big)$. Then, the problem the network controller wishes to solve can be expressed as:

$$\begin{aligned} \mathbb{P}: \quad & \max_{\{\boldsymbol{x}_t\}_t} \quad \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij}(t) \log r_{ij}(t) \\ & \text{s.t.} \quad \sum_{j \in \mathcal{J}} x_{ij}(t) = 1, \qquad \forall i \in \mathcal{I}, \end{aligned} \tag{3.2}$$

$$y_j(t) = \sum_{i \in \mathcal{I}} x_{ij}(t), \quad \forall j \in \mathcal{J}, \tag{3.3}$$
$$\boldsymbol{x}_t \in \{0,1\}^{I \cdot J}, \qquad \forall t \in \mathcal{T}.$$

The logarithmic utility function is selected to maximize the product of rates so as to balance network sum-rate and fairness [33], [34], [115]; constraint (3.2) ensures each UE is assigned to one cell, and (3.3) calculates the assigned UEs to each cell.

The solution to this problem at the beginning of horizon T is hindered by several factors. First, at $t = 1$, the controller does not have access to all future SINR values for each UE-cell pair. In fact, the SINRs during each slot t are practically unknown *even at the beginning* of that slot. Secondly, the HO delays $\{d_{ij}(t)\}$ depend on the associations $\boldsymbol{x}_t$, but also on the previous associations $\boldsymbol{x}_{t-1}$, since these two vectors determine if there is a HO or not; a HO is triggered for user i if $\boldsymbol{x}_{ij}(t-1) \neq \boldsymbol{x}_{ij}(t)$. In other words, there is a memory effect in the system, thus the problem cannot be decomposed on per-slot basis. With these challenges in mind, our goal is to design an online association algorithm that is oblivious to these time-varying unknown parameters, and which nevertheless maximizes the throughput while minimizing the HO delays, compared to a meaningful (i.e., competitive) benchmark.

3.4. Learning Dynamic Associations

Reformulation & Benchmark. We approach $\mathbb{P}$ as a *smoothed online learning* problem and tackle it via meta-learning based on the *experts framework* [18], [116]. The main idea is to deploy a set of parallel learning algorithms with different learning rates, and a meta-learner that tracks their performance and discerns on-the-fly the best-performing one. We start by reformulating the objective:

$$\begin{aligned} f_t(\boldsymbol{x}_t) &\triangleq \sum_{i=1}^{I}\sum_{j=1}^{J}\Big[x_{ij}(t)\log c_{ij}(t) - x_{ij}(t)\log y_j(t) \\ &+ x_{ij}(t)\log\big(1-d_{ij}(t)\big)\Big] \overset{(\alpha)}{=} \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}(t)\log c_{ij}(t) \\ &- \sum_{j=1}^{J} y_j(t)\log y_j(t) + \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}(t)\log\big(1-d_{ij}(t)\big) \end{aligned}$$

where (α) follows from $y_j(t) = \sum_{i \in \mathcal{I}} x_{ij}(t), \forall j, t$. The last term corresponds to the performance cost due to HO delays. Extending the rationale of prior works [33], [36], [109], and based on our data analysis, we will capture this cost using:

$$h\big(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}\big) = -\gamma \left\| \boldsymbol{A}\big(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\big) \right\| \triangleq -\gamma \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_A$$

where $\boldsymbol{A} = \mathrm{diag}(a_n > 0, n = 1, \ldots, I \cdot J)$ is a positive definite matrix where each element a_n, $n = i \cdot j$ models the delay when UE i is associated (de-associated) to (from) cell j, and $\|\cdot\|_A$ is its induced norm, i.e., $\|\boldsymbol{x}\|_A^2 = \sum_n a_n x_n^2$ and $\|\boldsymbol{x}\|_{A*}^2 = \sum_n x_n^2/a_n$ [117]. For instance, for $I = 3$ and $J = 2$, we have: $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_A^2 =$

$\sum_{i=1}^{3}\sum_{j=1}^{2} a_{ij}(x_{ij}(t)-x_{ij}(t-1))^2$; if UE $i=1$ moves from cell $j=1$ to $j=2$, it yields a total HO delay $a_{11}+a_{12}$, where a_{11} (a_{12}) is the delay due to the de-association (association) from (to) $j=1$ ($j=2$). The scalarization parameter γ is used to normalize units and prioritize one criterion (i.e., throughput) over the other (i.e., HO delays), based on the preferences of each operator.

This HO model departs from previous works, e.g., [33], [36] and references therein, that merely count the association changes as if all HOs had the same impact on performance. Clearly, the analysis in Section 3.2 showed this assumption not to be accurate in today's heterogeneous networks. Hence, we opt here instead to modulate the HO costs with parameters reflecting the potentially-different HO time for each UE-cell pair, and additionally with the tunable γ weight. In particular, based on Section 3.2, we will be using in our numerical evaluations the cell RAT and UE type as the main features for the HO delays. Finally, we define the decision set:

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \{0,1\}^{I\cdot J} \;\middle|\; \sum_{j\in\mathcal{J}} x_{ij} = 1, i \in \mathcal{I} \right\},$$

and its convex hull $\mathcal{X}^c = \text{co}(\mathcal{X})$ that relaxes the integrality, i.e., $\boldsymbol{x} \in [0,1]^{I\cdot J}$. We will use $\boldsymbol{x}$ when referring to the discrete associations, and denote with $\boldsymbol{x}^m \in \mathcal{X}^c$ the respective relaxed vector. Putting these together, we have the next result.

Proposition 1. The throughput and HO cost function is concave on $\mathcal{X}^c$: $f_t(\boldsymbol{x}) \triangleq g_t(\boldsymbol{x}) + h(\boldsymbol{x}, \boldsymbol{x}_{t-1}) =$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{ij}(t)\log c_{ij}(t) - \sum_{j\in\mathcal{J}} y_j(t)\log y_j(t) - \gamma\|\boldsymbol{x}-\boldsymbol{x}_{t-1}\|_A.$$

The concavity of $g_t(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}^c$ is proved in [33], and it follows that subtracting the A-norm preserves the property.

The performance of the algorithm will be assessed using the *Expected Dynamic Regret*, defined as:

$$\mathbb{E}\left[\mathcal{R}_T\right] \triangleq \sum_{t=1}^{T} f_t(\boldsymbol{x}_t^\star) - \sum_{t=1}^{T} \mathbb{E}\Big[f_t(\boldsymbol{x}_t)\Big], \tag{3.4}$$

where $\{\boldsymbol{x}_t\}_t$ are the algorithm decisions, $\{\boldsymbol{x}_t^\star\}_t$ is the benchmark, and the expectation captures any randomization in the algorithm. Our goal is to design an algorithm that ensures this gap diminishes with time, $\lim_{T\to\infty} \mathbb{E}[\mathcal{R}_T]/T = 0$ for any possible benchmark sequence $\{\boldsymbol{x}_t^\star\}_t$. In other words, we compare our algorithm with the best oracle that has full information at $t=1$ for the SINR and HO delays, for all users and cells, for the entire horizon T. Clearly, this is a very competitive benchmark, going beyond static and (per-slot) dynamic regret (see discussion in [118]), and thus a sublinear-regret algorithm in this context is highly desirable.

Online Algorithm. The proposed learning mechanism is summarized in Figure 3.5. There is a meta-learner that receives suggestions for the association policy from K agents (the *experts*); creates accordingly a weighted policy; and learns gradually how much to trust each expert based on its performance. In turn, each expert learns

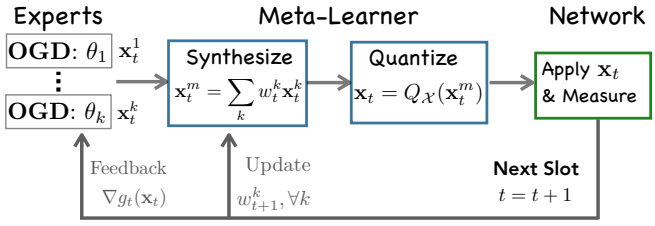


Figure 3.5: Learning mechanism.

dynamically the best association strategy, based on feedback from the meta-learner. The experts use different learning steps $\boldsymbol{\theta} = (\theta_k, k \in \mathcal{K})$, that are selected so as to ensure at least one of them will perform optimally w.r.t. the yet-to-be-encountered problem conditions. The details of the mechanism are provided in Algorithm 1 (LDA). At the beginning of each slot, each expert k shares its suggestion $\boldsymbol{x}_t^k$, and the meta-learner synthesizes them:

$$\boldsymbol{x}_t^m = \sum_{k \in \mathcal{K}} w_t^k \boldsymbol{x}_t^k, \tag{3.5}$$

where the weights $\boldsymbol{w}_t = (w_t^k, k \in \mathcal{K})$, with $\boldsymbol{w}_t^\top \mathbf{1}_K = 1$, is what the meta-learner needs to learn. It follows that if $\boldsymbol{x}_t^k \in \mathcal{X}^c, \forall k$, then $\boldsymbol{x}_t \in \mathcal{X}^c$. Next, the meta-learner creates a binary decision vector $\boldsymbol{x}_t \in \mathcal{X}$ (so as to be implementable) using the quantization routine $Q_{\mathcal{X}}$. For this operation, one can use any unbiased sampling technique, as long as $\mathbb{E}[\boldsymbol{x}_t] = \boldsymbol{x}_t^m$. For instance, Madow's sampling [119] ensures this condition which, for the structure of $\mathcal{X}$, simply picks an element from $\boldsymbol{x}_i(t)$ with probabilities $\boldsymbol{x}_i^m(t)$, for each $i \in \mathcal{I}$.

At the end of the slot, the controller observes the system parameters[1] and calculates the gradient $\nabla g_t(\boldsymbol{x}_t)$, which is sent to all experts. Then, the meta-learner updates the weights:

$$w_{t+1}^k = \frac{w_t^k e^{\beta \ell_t(\boldsymbol{x}_t^k)}}{\sum_{k \in \mathcal{K}} w_t^k e^{\beta \ell_t(\boldsymbol{x}_t^k)}} \tag{3.6}$$

using the surrogate (i.e., partially linearized) loss:

$$\ell_t(\boldsymbol{x}_t^k) = \langle \nabla g_t(\boldsymbol{x}_t), \boldsymbol{x}_t^k - \boldsymbol{x}_t \rangle - \gamma \|\boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k\|_A.$$

It is interesting to observe that all experts use the same gradient and not the gradient of their own action. This is possible due to the concavity of g_t, which can be upper bounded by the linearized loss at *any* information point; see also [120].

Finally, the experts perform an online gradient ascent:

$$\boldsymbol{x}_{t+1}^k = \Pi_{\mathcal{X}}\Big(\boldsymbol{x}_t^k + \theta_k \nabla g_t(\boldsymbol{x}_t)\Big), \tag{3.7}$$

[1] Since UEs report their channel gains with all reachable base stations (and not only with the one they are associated), we have a full-information model.

Algorithm 1: Learning Dynamic Associations (LDA)

Input: Horizon T; K experts with steps $\{\theta_k\}_{k\in\mathcal{K}}$; Meta-learning step β
Initialize: set $w_1^k = \frac{1+(1/K)}{k(k+1)}, \forall k \in \mathcal{K}$; draw $\boldsymbol{x}_1^k \in \mathcal{X}, \forall k \in \mathcal{K}$; sort $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_K$;
for $t = 1, 2, \ldots, T$ **do**
  Each expert $k \in \mathcal{K}$ shares its decision $\boldsymbol{x}_t^k$;
  The controller synthesizes $\boldsymbol{x}_t^m$ using (3.5);
  The controller implements $\boldsymbol{x}_t = Q_{\mathcal{X}}(\boldsymbol{x}_t^m)$;
  The controller sends $\nabla g_t(\boldsymbol{x}_t)$ to experts;
  The controller updates its weights using (3.6);
  Each expert updates its decision using (3.7);
end

so as to produce their next suggestion. The number of experts, their rates, initial weights $\boldsymbol{w}_1$, and meta-learning step β, are selected to ensure the regret convergence, as explained next.

Performance Analysis. We first introduce some key parameters that are used below. Defining $a_{\max} = \max_{n\leq I\cdot J} a_n$ we obtain the bounds:

- $\|\boldsymbol{x}-\boldsymbol{x}'\|_2 \leq D$, $\|\boldsymbol{x}-\boldsymbol{x}'\|_A \leq \sqrt{a_{max}} D \triangleq D_A$, $\|\boldsymbol{x}-\boldsymbol{x}'\|_{A*} \leq D/\sqrt{a_{max}} \triangleq D_{A*}$ with $D = \sqrt{2I}$.
- $\|\nabla g_t(\boldsymbol{x})\|_2 \leq G$, $\|\nabla g_t(\boldsymbol{x})\|_A \leq \sqrt{a_{max}} G \triangleq G_A$, with $G = \sqrt{IJ(\log J + 1/\ln 10)^2}$.

To characterize the performance of Algorithm LDA, we proceed in two steps. First, we bound the regret of the relaxed (continuous) performance and switching cost, with the following lemma that we prove in the Appendix 3A:

Lemma 3.1 (Performance of LDA)**.** *Using the following parameters:*

- $K = \lceil \log_2 \sqrt{1+2T} \rceil + 1$.
- $\theta_k = 2^{k-1} \left[\frac{D_A^2}{T(G^2+2G_A)} \right]^{1/2}$, $k = 1, \ldots, K$.
- $\beta = 1/\sqrt{T\nu}$, *with* $\nu \triangleq (2GD+D_A)^2(D_A+(1/8))$.

The continuous decisions $\{\boldsymbol{x}_t^m\}_t$, *where* $\boldsymbol{x}_t^m \in \mathcal{X}^c$ *ensure:*

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_t^\star) - \sum_{t=1}^{T} f_t(\boldsymbol{x}_t^m) \leq \mathcal{O}\left(\sqrt{T(1+P_T)}\right)$$

with the benchmark's total HO delay $P_T = \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|_A$.

As is expected in dynamic regret, the bound depends on the variability of the benchmark which here, interestingly, has the physical meaning of HO delays. In

any case, the meta-learner follows the benchmark and gradually decreases the gap. These decisions refer to a continuous-valued association strategy ($\boldsymbol{x}_t^m \in \mathcal{X}^c$) similarly to [33], [36] and others, and can be interpreted as probabilistic associations. Unlike these prior works, here we make an extra step to provably bound the expected dynamic regret of the implementable *discrete* associations. The observation we utilize is that $\mathbf{0} \preceq \boldsymbol{x}_t, \boldsymbol{x}_t^m \preceq \mathbf{1}$, and are related through an unbiased sampling, i.e., $\mathbb{E}[\boldsymbol{x}_t] = \boldsymbol{x}_t^m$. Based on this, we obtain the following result, proved in Appendix 3A.

Theorem 3.2. *Algorithm* `LDA` *ensures the following bound against* any *benchmark sequence* $\{\boldsymbol{x}_t^*\}_{t=1}^T$*:*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \mathcal{O}\left(\sqrt{T(1+P_T)}\right) + G_f T\sqrt{I-(I/J)}$$

where $\|f_t(\boldsymbol{x})\|_2 \leq G_f, \forall \boldsymbol{x} \in \mathcal{X}, \forall t$.

Discussion. The theorem shows that the regret of `LDA` is sublinearly dependent on the oracle's HO delay, which marks its learning capability. There is also an unavoidable non-diminishing error term due to discretization. In fact, this error can be eliminated by normalizing properly the step β and using the doubling trick. Due to lack of space we kindly refer the reader to [121, Lemma 3] for further details. The doubling trick will also eliminate the need to know in advance the horizon T. Besides, even without the step normalization, we find the error to be only 1.1–1.3% of the objective value, when tested on static and volatile scenarios (see Section 3.6).

Moreover, it is important to stress that Algorithm `LDA` is oblivious to information such as the SINR during the slot, which, in practice, is unknown at the beginning of each slot [111]. This is a key difference of the proposed approach compared to prior works such as [33], [36]. Further, `LDA` is scalable to the number of UEs and cells and is amenable to near-real-time execution as it has relatively lightweight operations and thus, can be implemented in O-RAN [111], [112].

3.5. Time-varying Delays and Forecasters

Finally, we discuss two key extensions: *(i)* when HO delays are unknown, where we show that `LDA` can adapt to them dynamically; and *(ii)* when there is an ML mechanism that proposes associations based on SINR/mobility forecasts [33], [38], and prove that `LDA` can seamlessly benefit from them.

Time-varying HO delays. Algorithm `LDA` does not require knowing in advance the elements of A, i.e., the HO delays. To see this, first, observe that the experts do not use the HO delay when they update their decisions (only the throughput $\nabla g_t(\boldsymbol{x}_t)$). And secondly, the meta-learner uses the HO delays when it calculates $\ell_t(\boldsymbol{x}_t^k), \forall k \in \mathcal{K}$ to update the weights $\boldsymbol{w}_t$, which takes place *after* the HOs are realized. This flexibility allows to tackle cases where the HO delays for each type of UEs and cells, change with time. From a technical point of view, instead of the fixed A-norm, `LDA` can use a time-varying norm $\|\cdot\|_{A_t}$, where each A_t captures the delay that was observed (a posteriori) at each slot t. This information is then used to calculate the weights $\boldsymbol{w}_{t+1}$ and association $\boldsymbol{x}_{t+1}$. The proofs of Lemma 3.1 and

Theorem 3.2 follow nearly verbatim, with the modification of changing the fixed norm to the time-varying norms, and redefining $P_T = \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|_{A_t}$.

Encompassing Forecasters. Unlike our approach that learns from runtime data, several recent works [33], [38] proposed to decide the HOs leveraging forecasters for SINRs. The output of such tools, let us denote it $\{\boldsymbol{x}_t^p\}_t$, can be very close to $\{\boldsymbol{x}_t^\star\}_t$ if the forecasters are accurate, but very suboptimal otherwise (e.g., if there is a distribution shift). Our framework, on the other hand, can benefit from such tools in a robust fashion, by assessing their accuracy in real-time. The meta-learner can include $\{\boldsymbol{x}_t^p\}_t$ as the $(K+1)$th expert, and assess its performance in real-time so as to discard it if proved inaccurate. And if the forecasting tool is effective, the regret will improve significantly. This can be seen by revisiting the proof of Lemma 3.1 that utilizes the Hedge algorithm (see [120, Lem. 1]), which bounds the regret of the meta-learner from the best expert (thus, also from the forecaster p) as:

$$\max_{k \in \mathcal{K} \cup p} \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^k) - \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^m) \overset{(\alpha)}{=}$$
$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^p) - \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^m) \leq \frac{\beta T c^2}{8} + \frac{1}{\beta} \ln \frac{1}{w_1^k}$$

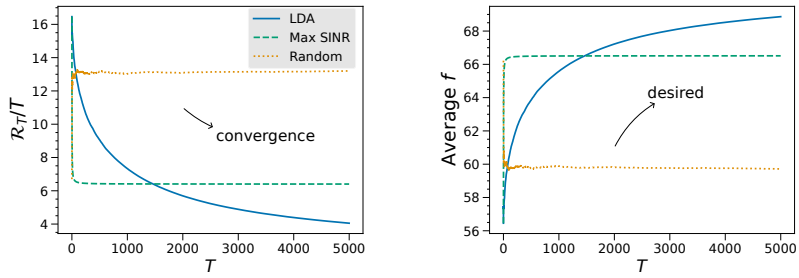
where $c = 2GD + D_A$ and (α) holds when the forecaster is the best expert. Defining its error $\sum_t \ell_t(\boldsymbol{x}_t^\star) - \ell_t(\boldsymbol{x}_t^p) = \epsilon_T$ we get:

$$\mathbb{E}[\mathcal{R}_T] \leq \frac{\beta T c^2}{8} + \frac{1}{\beta} \ln \frac{1}{w_1^k} + \epsilon_T + G_f T \sqrt{I - (I/J)}$$

which is optimized when $\beta = \mathcal{O}(1/\sqrt{T})$ (as in Theorem 3.2). Comparing this result with Theorem 3.2, we see that when the forecaster is successful, the overall performance improves by dropping an entire term, and does not depend on P_T; while, when the forecaster is found to be inaccurate, `LDA` maintains the previous performance as it relies on a different expert. The idea of combining forecasters with online learning is often referred to as *optimistic learning* [21], [22], and we apply it, for the first time, in the context of SOL with dynamic regret.

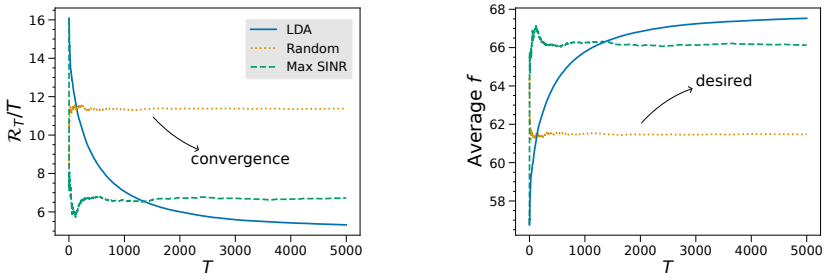
3.6. Performance Evaluation

`LDA` undergoes rigorous evaluation across multiple scenarios that encompass both real-world conditions (actual users, cells, and SINRs) and synthetic ones, verifying that its efficacy is broadly applicable. We utilize these scenarios to investigate the algorithm's learning convergence, and compare its performance with different benchmarks in terms of *(i)* attained dynamic regret, *(ii)* accumulated objective function, throughput, and HO cost, and *(iii)* impact of incorporating $\boldsymbol{A}$ versus using a simple $L2$ norm, or none at all. The employed benchmarks are *(i)* an `LDA`-based algorithm using the Euclidean norm in the HO cost, called `LDA 2-norm`, aiming to highlight the significance of the A-norm; *(ii)* an advanced greedy algorithm, `Max SINR`, that assigns UEs to cells based on the maximum SINR from the previous slot



(a) (b)

Figure 3.6: Static scenario for $\gamma = 20$: (a) average dynamic regret and (b) average obtained objective function f.



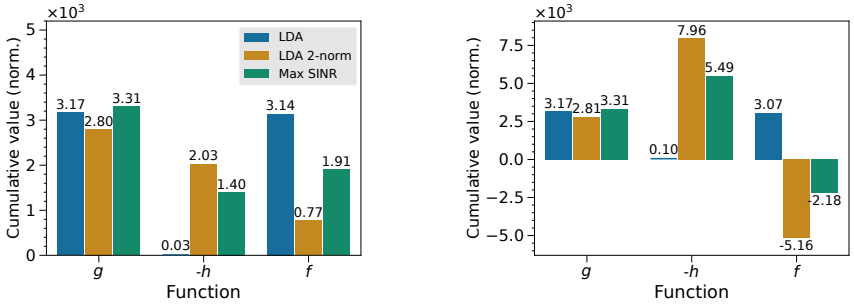
(a) (b)

Figure 3.7: Volatile scenario for $\gamma = 20$: (a) average dynamic regret and (b) average obtained objective function f.

(the current slot's SINR is only known post-association), disregarding HO delays; *(iii)* an optimal `Oracle`, used in the definition of dynamic regret in (3.4), which has complete knowledge of future; *(iv)* a basic `Random` algorithm that makes the associations randomly, serving as a minimal benchmark.

Synthetic scenarios. We select two synthetic scenarios that are in line with those in prior work [60], [111], [112], and which we define as follows, for $T = 5k$ slots: (*i*) *static*: the SINR $s_{ij}(t)$ remains constant across all slots, and (*ii*) *volatile*: $s_{ij}(t)$ fluctuates every 5 slots within the range of [10, 30]dB [122], encompassing poor to excellent values. In both scenarios, we randomly select bandwidths $w_j \in \{5, 10, 15, 20\}$ MHz [123], while $\boldsymbol{A}$ takes random values within [0, 1], and $\gamma = 20$ (penalizes more HO cost, without sacrificing throughput). We choose a smaller case study with $I = 100$ UEs and $J = 10$ cell to facilitate the calculation of average regret, as determining the best oracle is computationally intensive. However, it is important to note that the best oracle is not necessary for running `LDA`.

Figures 3.6 and 3.7 show that `LDA` identifies the best benchmark (diminishing regret) just in solely $T = 5k$ slots. This verifies our claim (see Section 3.4) that the error due to discretization is small. Conversely, `Max SINR` fails to converge, even though the SINR does not change, with the regret remaining constant at 6.5 (exploiting sub-optimal associations). This trend is also evident in the average



(a) (b)

Figure 3.8: Comparison of throughput (g), negative HO cost ($-h$), and total (f) values for the real case, when (a) $\gamma = 5$, (b) $\gamma = 20$.

objective function, where `LDA` outperforms `Max SINR` after $t = 1.4k$. As anticipated, `Random` algorithm performs poorly.

Real-world scenarios. To test `LDA` in real-world scenarios, we use crowdsourced measurements that contain various signal metrics (e.g., SINR and SNR) along with their precise latitude and longitude information and the concerned cell, at every second. This dataset presents a highly competitive scenario as SINRs can vary arbitrarily, and occasionally exhibit adversarial behavior. The crowdsourced measurements are anonymized, hence we lack specific UE information (e.g., its type or exact mobility pattern). Therefore, $\boldsymbol{A}$ is chosen from the distribution of our data, as shown in Figures 3.2, 3.3, and 3.4, ensuring a variety of UE types and HO delays are considered. Moreover, we adopt the Gauss-Markov mobility model [124], in line with prior works [33], with randomness parameter $a = 0.5$ ($a = 0$ being totally random and $a = 1$ being linear motion) and consider velocities in [1, 28] meters per second, and velocity variances in [0, 14]; thus having from pedestrians to fast-moving UEs.

At $t = 1$, 1k UEs are placed randomly on the map at a different location with recorded SINR measurements. At each subsequent slot $t = 2, ..., T$, where $T = 10k$, each user moves in accordance with their individual parameters, to their own new location. To incorporate the signal measurements, we map each new location to the nearest one with available measurements and continue similarly for the next slot. We utilize measurements from an urban district and choose 25 of the cell with the most data. For the bandwidths of each cell, we use the real ones from our dataset.

We consider two different cases for γ: (i) $\gamma = 5$, and (ii) $\gamma = 20$ (chosen in accordance with throughput and HO cost values). Both cases prioritize high throughput, but the latter penalizes HO costs more. From Figure 3.8, we observe the cumulative g values (normalized by the maximum value) are the same for each algorithm, in both $\gamma = 5$ and $\gamma = 20$ cases; ensuring no throughput is sacrificed in turn of lower HO delays. At the same time, `LDA 2-norm`, which does not consider the HO delays but solely the number of HOs, achieves $\times 67.7$ and $\times 79.6$ higher HO cost than `LDA` for $\gamma = 5$ and $\gamma = 20$, respectively. Similarly, although `Max SINR` achieves 4.4% higher throughput, it does not take into account the HO costs, resulting in 39% (171%) lower total values for $\gamma = 5$ ($\gamma = 20$). Therefore, `LDA`

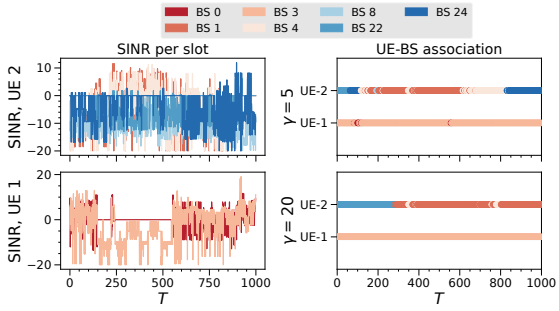


Figure 3.9: SINR per slot of two real users (upper and lower left) for $T = 1k$, and the UE-cell associations that `LDA` chooses, for $\gamma = 5$ (upper right) and $\gamma = 20$ (lower right).

performs optimally in terms of HO costs, without sacrificing throughput.

To further deep-dive into how `LDA` responds to γ, we show in Figure 3.9 the SINR of two UEs for the first $t = 1k$ slots, and the UE-cell associations for $\gamma = 5$ and $\gamma = 20$. In the former case, we observe that both users change cells more frequently, e.g., for $\gamma = 5$, UE-1 changes 5 times between BS/cell-3 and BS/cell-0 even though the SINR of the latter becomes, for a few slots, slightly better than the former, while for $\gamma = 20$, UE-1 stays constantly in BS/cell-3. These results show that `LDA` can be easily adjusted by changing γ, to react more to SINR changes (i.e., maximize throughput), or take more into account the HO cost. The latter is a crucial aspect in mobile networks, where HOs may lead to the ping-pong effect [31].

3.7. Related Work

Measurements. HOs have mainly been studied using traces from UEs, which are inevitably limited to certain manufacturers [30], services [91], or user types [72], [99]. Our findings on HO duration align with these prior works [30], [64], [72] and offer new insights into the impact of RAT and UE type on HOs. While several studies have measured HO volume [32], [69], [91], noting that horizontal HOs are more frequent in 5G-Standalone (5G-SA) and 4G, and vertical HOs in 5G-NSA [91], we enhance these findings by revealing the geographic heterogeneity of HOs and measuring their effect on important KPIs. Importantly, our work utilizes a large-scale dataset measuring HOs from the network-side of a top-tier operator with 40M UEs. This contrasts sharply with the few network-side studies, e.g., [31] has less than 0.7% of our UEs. While [113] is the first countrywide study of cellular HOs, it focuses solely on measurements, without providing a HO solution or focusing on the effect of HO failures (HOFs) on KPIs.

Handover Optimization. In user-centric mobility schemes, UE associations (and thus HOs) are often approached as a bandit learning problem where each UE explores which cell offers better throughput [125]–[130]. Conversely, using SINR measurements from all UEs, the network can make more effective centralized decisions. This idea is regaining momentum, but is also more challenging (due to scale, among others) to implement. For instance, a recent thread of studies employs (Deep)

Reinforcement Learning to decide HOs or HO rules (e.g., SINR thresholds) [34]–[37], [131], [132]. Despite its modeling appeal, these approaches essentially rely on heuristics. On the other hand, our method comes with performance guarantees, even under adversarial conditions.

Similarly, the joint optimization of throughput and HO delays is considered in [33], [34], [36], [37], [109]. Our goal is very close to these works, but we enrich their model to capture the different delay effects of the various HO types, RATs, and UE types. Also, we drop several of their assumptions and, in particular, their need to have accurate SINR information. Clearly, in today's volatile and non-stationary network conditions, this assumption is impractical. Finally, some recent papers leverage the ML-provisions of O-RAN, and propose forecasting-assisted HOs [33], [37], [38], [133]. Our solution is also aligned with O-RAN, but unlike these works, it learns on the fly the network/UE parameters and does not require offline training nor the availability of a reliable forecaster. At the same time, when such predictions are available, our algorithm can directly benefit from them to expedite its learning, building on the idea of optimistic learning [22]. For additional discussion on HOs, see [26], [27].

Smoothed Online Learning. In terms of solution, unlike the model-predictive control of [33], the RL strategies in [34], [36], [37], [134], or the heuristics in [109], we approach this problem, for the first time, through the lens of *smoothed online learning* [25], [110], [120]. SOL enriches the online convex optimization toolbox [16], accounting for costs induced by decision changes, and has recently found applications in caching [135], [136], network selection [127], and service deployment [137], among others. Here, the decisions express the UE-cell associations, and the switching cost captures the HO delays in a natural way. Yet, unlike the above works, we leverage the more competitive dynamic regret benchmark, as HOs are unavoidable, and the switching cost models the delay rather than merely the HO count. In general, SOL should not be confused with mathematical smoothing [138], nor with the stochastic and full-information framework of decisions with reconfiguration delays [139].

3.8. Conclusion

This chapter addresses the problem of (traditional) HO optimization in cellular networks using SOL. By extending the MNO's datasets introduced in the previous chapter with targeted measurements and crowdsourced signal data, we identify key correlations between HO failures/delays and the characteristics of radio cells and devices. Based on these insights, we develop a realistic dynamic model for UE-to-cell associations that incorporates these user/cell features, and propose an online meta-learning, O-RAN-compatible algorithm to tackle the problem. Our solution does not require knowledge of future signal quality or user mobility/trajectories, and demonstrates robust performance in both real-world and synthetic scenarios, providing a solid foundation for future advancements in network performance.

Until now, all our contributions around handovers have centered on the traditional (THO) approach, which is inherently triggered reactively; or in other words, only after signal conditions have already degraded (significantly). Even though THOs have long served as the backbone of mobility management, their reactive na-

ture may still pose challenges, especially in environments with dense deployments and high-frequency bands, where signal quality might drop too quickly for reactive mechanisms to respond. This naturally raises the following question: What if the network and users could prepare before the signal has degraded considerably, when needed? In the next chapter, we explore the proactive Conditional Handovers (CHOs), investigating how learning-based control for both THO and CHO can unlock their full potential in the next generation of cellular networks.

4

From Reactive to Proactive Handovers

In the previous chapters, we focused solely on traditional handovers (THOs), the long-standing backbone of mobility management. While we leveraged online meta-learning for optimizing THOs, they remain fundamentally reactive, triggered after signal conditions have deteriorated.

To mitigate these challenges, 3GPP introduced Conditional Handovers (CHOs), which, despite promising lower delays and fewer failures than THO by proactively reserving resources in multiple target cells and delegating the final handover decision to the user, bring new challenges: how many and which cells to prepare, as well as how to avoid unnecessary signaling and wasted resources. To address these challenges, we revisit the Mobile Network Operator (MNO) datasets from the previous chapter and extend them with additional information on both source and target cells (e.g., frequency, vendor changes). These allow us to capture the unique dynamics of CHOs, which depend not only on the serving cell but also on the characteristics of multiple target cells. Taking these characteristics into account, we develop online meta-learning solutions that jointly optimize THO and CHO within the O-RAN architecture. Our approach is oblivious to time-varying and unknown future system conditions (e.g., user trajectories, signal strength), adapts to runtime observations, and offers robust guarantees and comparable performance to an oracle that has perfect knowledge of these future conditions. Our solution surpasses 3GPP-compliant and Reinforcement Learning (RL) baselines in dynamic and real-world scenarios.

The content of this chapter has been published in:

M. Kalntis, G. Iosifidis, J. Suárez-Varela, A. Lutu, and F. A. Kuipers, "Meta-Learning-Based Handover Management in NextG O-RAN," in *IEEE Journal on Selected Areas in Communications (JSAC)*, 2026.

M. Kalntis, F. A. Kuipers, and G. Iosifidis, "CHOMET: Conditional Handovers via Meta-Learning," in *Proc. of International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2025.

4.1. Challenges and Contributions

In this chapter, we argue that next-generation (NextG) networks will benefit from AI-based solutions that jointly optimize traditional and conditional HOs and adapt the HO strategy to the current user requirements. In fact, 6G early releases are expected to include both network-triggered and user-triggered HO solutions [50]. At the same time, the rise of *O-RAN* provides a practical vehicle for deploying such intelligence since HO logic can now be implemented as near-real-time xApps, enabling a flexible and programmable AI-native radio access network intelligent controller (near-RT RIC); see Figure 4.1. This controller communicates with the O-RAN-compliant central, distributed, and radio units (O-CU, O-DU, and O-RU, respectively), delegating the decisions to multiple cells and users. This architectural openness and AI-based control creates a timely opportunity to rethink mobility management for NextG networks. In summary, in this chapter, we answer the question: *How can we design a robust and principled AI-driven framework for jointly optimizing traditional and conditional handovers in NextG O-RAN architectures?*

Methods & Contributions. To answer this question, we first analyze new countrywide datasets of mobility events from a top-tier mobile network operator (MNO) in a European country. Unlike prior studies that focus on limited end-user measurement campaigns or confined regions (see Section 4.8), our dataset spans 13.5K cells and 40M users, and provides comprehensive visibility into handover failures (HOFs), HO delays, signal fluctuations, and cell heterogeneity. Employing a statistical analysis, we quantify the conditional effect on HOFs and HO delays as a function of cell's frequency band, vendor, and location/time. We observe a Pareto-like distribution in signal fluctuation: for 20% of movements, the Signal-to-Interference-plus-Noise Ratio (SINR) varies by 100% within 1 s, which makes them ideal candidates for CHOs. These insights confirm the need for adaptive and robust HO control at user-level granularity and motivate the solution of this work.

We next introduce a first-of-its-kind *unified modeling framework* that jointly captures both traditional (explicit) and conditional (implicit) HO processes. Our model represents the decisions of a controller [53] in an O-RAN setting and accounts for user throughput, cell load, signaling overhead, and service delay due to HOs (switching costs). We study two variants of `CONTRA`: one with a priori HO type assignment per user, reflecting distinct service or user-specific requirements, and another where the controller determines on-the-fly the HO type based on system dynamics. The key novelty lies in treating HO type selection, candidate cell preparation, and cell association as a single online learning problem without relying on knowledge from underlying stochastic processes (e.g., SINR fluctuations, cost of signaling). We model the resource trade-offs and performance dependencies between HO types and across users, enabling per-user decision-making at the granularity of O-RAN near-real-time control loops.

To solve this problem, we propose `CONTRA`, a meta-learning-based algorithm (see Figure 4.1) that maintains a pool of learners (i.e., experts) tuned for different signal conditions and mobility patterns, and employs a meta-learner to track their performance. This design enables *adaptation* to changing network environments, types of user equipment (UE) and cells, without prior statistics, and ensures per-

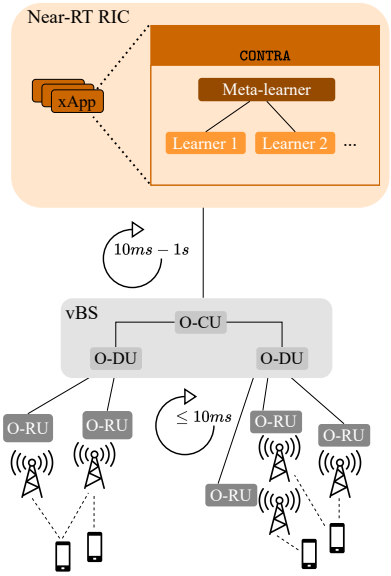


Figure 4.1: CONTRA deployed as an xApp in the near-RT RIC of O-RAN.

formance commensurate with that of an oracle with knowledge of the future. Our approach is based on the theory of online convex optimization (OCO) [16] and our formulation builds on recent advances in *smoothed online learning* (SOL) [25]. The proposed algorithm provides theoretical guarantees in terms of *expected dynamic regret*, a rigorous measure of how well the algorithm performs compared to the oracle, under a vast range of scenarios for the changing parameters. Finally, we evaluate CONTRA using the datasets from the top-tier MNO and actual users. Benchmarking it against 3GPP-compliant and Reinforcement Learning (RL) based THO and CHO mechanisms, we find that it provides higher user throughput, lower signaling cost, and improved robustness, particularly in volatile and real SINR environments. In summary, our contributions are:

• We present and analyze countrywide mobility datasets to shed light on parameters, such as frequencies, vendors, and location/time, that affect HO failures and delays.

• Motivated by the 6G vision and the analyzed datasets, we model HO control as a learning problem and propose CONTRA, the first unified THO/CHO orchestration framework for real-time and robust HO control in NextG O-RAN. We study two variants of CONTRA: a static one with predefined HO types per user, and a dynamic one, where the controller decides the HO type on-the-fly. Our solution relies on a meta-learner that is oblivious to UE types, mobility, and network conditions, offering performance guarantees (expected dynamic regret).

• We evaluate CONTRA using crowdsourced data and multiple scenarios against 3GPP-compliant and RL benchmarks. The experiments highlight CONTRA's *efficacy, deployability, and alignment with 6G goals* of intelligent Radio Access Network (RAN) functions.

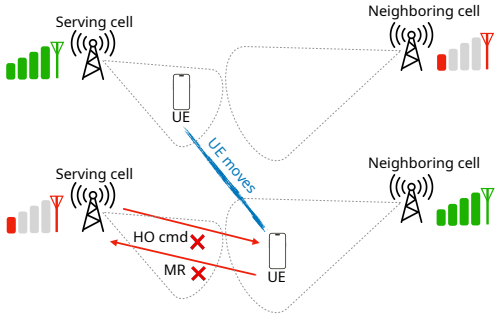


Figure 4.2: A failed traditional HO, where the MR or HO command may not reach the serving cell or UE, respectively, as the signal has dropped significantly.

4.2. Trigger of Handovers

Traditional & Conditional HOs. THOs are triggered when the SINR of the serving cell degrades (A2 event) and/or another neighboring cell's signal is higher than the serving cell (A3 event) [78], [80], [81]. This *reactive* approach, where THOs are initiated *after* the signal conditions deteriorate, often leads to increased HO delay and HOF rates, particularly due to the small cell density and the rapid signal fading of high frequency bands ($>$ 24 GHz, FR2) [113], [140]. Specifically, as illustrated in Figures 4.2 and 4.3(a), it is common for HOFs to occur when a user attempts to send a measurement report (MR) under degrading signal conditions; or even if the MR is successfully sent, the worsening signal conditions may prevent the user from receiving the HO command from the serving cell.

CHOs, designed first as part of 3GPP Rel. 16, address these limitations by offloading part of the HO decision-making to the user, before signal conditions deteriorate [39]. As can be seen in Figure 4.3(b), a CHO decision is taken while signal quality is still adequate (step 2), and the source cell can pre-configure (i.e., prepare) multiple *candidate* target cells (steps 3, 4, 5) based on the MR of the UE (step 1). To conclude the *preparation phase*, the source cell provides monitoring conditions (step 6), such as hysteresis, offset, and time-to-trigger (TTT) parameters, and the *execution phase* starts: the user applies these conditions continuously (step 7) to evaluate the signal of the source and candidate target cells.

If any of the predefined conditions are met (step 8), the UE executes the stored HO command (as if it had just been received) without an MR or reply from the serving cell. If more than one cell meets the execution condition, the UE decides which to access; typically, the one with the highest SINR. CHO is finalized in the *completion phase*, where resources are released from the serving cell and a new path to the target cell is established (steps 9, 10, and beyond). Once resources are reserved for a UE through admission control (step 4), they are not released until a cancellation is sent (step 10b).

Conversely, the HO decision in traditional HOs dictates the *one* target cell the UE should connect to; and this transition (steps 6, 7) is executed *immediately* after the UE receives the command. In other words, the execution phase in THOs occurs immediately after preparation, in contrast to CHOs, where the gap between the two

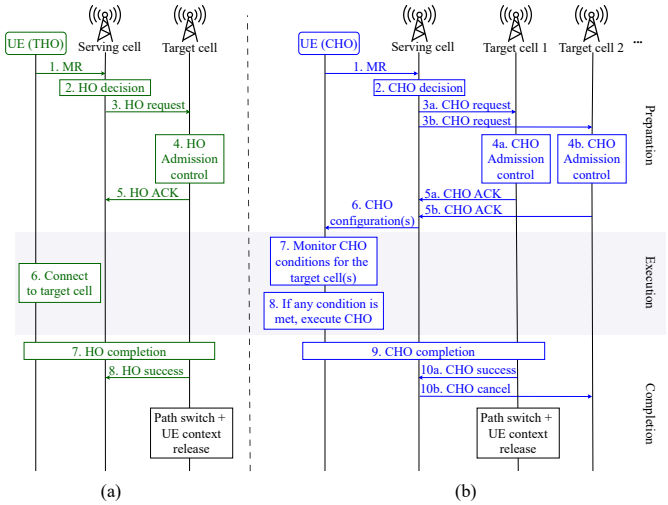


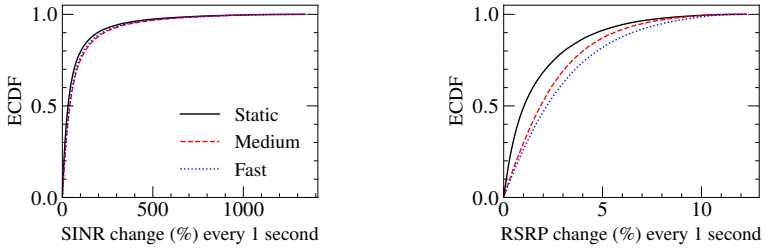
Figure 4.3: Basic steps/procedures of UEs performing (a) traditional (green) and (b) conditional (blue) HOs, requiring 8 and 10 main steps, respectively.

phases can be as large as 9–10 s [45].

An essential aspect of any intelligent mechanism is the granularity at which THO/CHO decisions are made (step 2 of Figure 4.3). Typically, these decisions are taken in hundreds of milliseconds (e.g., 100–640 ms), driven by the TTT parameter; namely, the duration for which the signal of a neighboring cell should be constantly better than that of the serving cell [36], [46], [80], [81]. This granularity aligns with the decisions of a central controller in the near-real-time of O-RAN, which could even handle multiple cells simultaneously [12].

CHO Key Trade-Offs & Selection Criteria. Determining which cells to prepare is critical in CHOs for balancing resource efficiency and mobility robustness. Ideally, the MNO, whose goal is to economize cell resources, would allow the preparation of the single cell to which the UE will connect in the next slot (i.e., the "correct" target cell) and whose signal strength is high, hence maximizing the UE's throughput. However, this cannot always be predicted and multiple candidate cells often need to be prepared, e.g., because the user is located at the edge of a cell and its trajectory is unknown. At the same time, a long list of prepared cells does not *necessarily* increase the likelihood of including the "correct" target cell, especially if all prepared cells exhibit low signal quality. Conversely, while preparing fewer cells may save resources, it increases the risk of falling back on traditional HOs; thus, larger HO delays and HOF probability (see Section 4.3).

Another aspect to consider when optimizing CHOs is the *signaling cost* of preparing cells. For instance, in environments such as FR2, the small cell density and rapid signal fluctuations already result in more frequent HOs, leading to significant signaling overhead. The additional burden of continuously preparing and releasing cells due to constant signal variations further exacerbates this overhead [45], [141]. For that reason, although initial 3GPP releases specify that UEs should release CHO candidate cells after any successful HO completion to conserve resources [39], subse-



(a) (b)

Figure 4.4: (a) SINR and (b) RSRP changes per second.

quent studies suggest that this approach is not always optimal for minimizing HOFs and signaling overhead [140].

These trade-offs emphasize that the selection process for cell preparations should focus on finding a balance between identifying the "best" (in terms of signal quality) potential target cells and keeping the number of prepared cells small (i.e., limited reserved resources).

4.3. Data Collection and Analysis

To highlight the potential benefits of CHOs, we illustrate the problems of traditional HOs, extending the datasets of the previous chapter that omit the impact of target cell features, apart from the change in radio access technology. Our goal is to identify the features (e.g., vendor, location) in the source and target cells (not only in the former), as well as in the UEs, that contribute to increased HOFs and delays.

Signal Quality Metrics & Mobility. In Figure 4.4, we examine temporal signal variations, namely SINR and RSRP, across various speeds using our crowdsourced datasets, which offer measurements at 1-second intervals. We classify samples into three mobility categories: *static*, *medium* (3–10 km/h, likely walking/running), and *high* speed movements (>80 km/h, likely vehicular). In general, SINR is sensitive to short-term effects such as fast fading, interference, and noise floor variations, while RSRP reflects a more stable signal strength from the serving cell.

Our findings reveal that SINR exhibits considerable fluctuations, even at the second-level granularity, with more than 20% of movements experiencing a 100% or greater change in SINR in one second, regardless of speed. Given that our passive measurement dataset covers cell deployments in the frequency bands below 3.5 GHz, we highlight that SINR variability does not increase with mobility due to, for instance, greater Doppler spread and more dynamic interference conditions. Doppler effects typically become relevant at higher frequencies, such as mmWave and above [142], and become more noticeable in very high-speed scenarios, such as trains traveling at speeds of up to 500 km/h and Vehicle-to-Vehicle (V2V) communications [142], [143]. In contrast, our high-mobility tier mainly captures vehicular UEs moving at speeds above 80 km/h (and rarely exceeding 130 km/h) in an urban environment. On the other hand, RSRP is more stable, and speed plays a bigger role; e.g., 20% medium-speed movements have their RSRP changing more than 4%.

It is worth recalling that in HO procedures, a typical threshold for initiating the

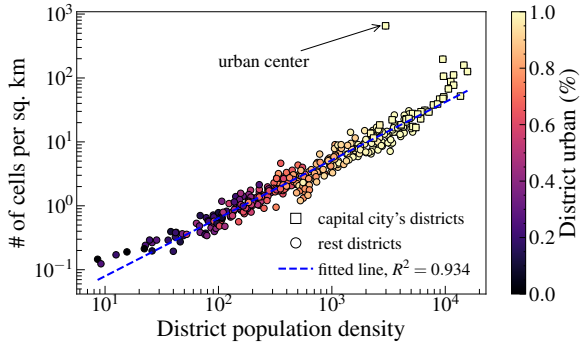


Figure 4.5: Number of cells per sq. km in the country (district level). Each point is colored based on the ratio of postcodes classified as urban (others are rural).

HO based on the A3 event is commonly set around 3 dB [36]. This threshold provides a useful reference when interpreting Figure 4.4, as it highlights how frequently such conditions can be met.

Key Takeaways: The SINR fluctuates by more than 100% for over 20% of UEs within short time intervals of 1 s. Such rapid signal variations increase the likelihood of suboptimal HO triggering, and thus, of HO delays and HOFs. The proactive cell preparation approach of CHOs can mitigate the impact of such fluctuations and improve user performance.

Spatial Heterogeneity. From the official census data of the analyzed country, we dissect the 300+ defined districts and 2.5M+ postcodes, and combine them with our collected cell-level datasets from the MNO to understand how cells are placed across different environments, thus affecting the number of cells that can be prepared in CHOs. Postcodes have been classified as *urban* or *rural*, based on whether they are allocated to an area with a population of more or less than 10k residents, respectively.

Figure 4.5 displays the district-level cell density (number of cells per sq. km) compared to the population density. Each point corresponds to one of the 300+ districts in the studied country and is color-coded based on the percentage of postcodes classified as urban within the district. As expected, there is a high correlation between these two factors (Pearson correlation of 0.967). In districts where more than 90% of postcodes are urban, shown mainly in yellow, we observe 10 to 100 cells per sq. km, with areas near the capital city exceeding this range. It is interesting to note that the urban center of the capital city (highest yellow rectangle) attracts a substantial influx of non-resident users due to being a major administrative and economic hub; consequently, operators deploy more than 650 cells per sq. km to accommodate the increased demand. On the other hand, areas with less than 20% urban postcodes exhibit as few as 0.12 cells per sq. km, which is sufficient to meet the reduced user demand there.

In Figure 4.6, we further delve into a dense urban district to illustrate the spatial heterogeneity of the cell deployment. We partition the district into one sq. km tiles, revealing 2.2k+ total deployed cells. The per-tile cell count spans over two orders of magnitude, ranging from a single cell in sparsely covered areas to more than

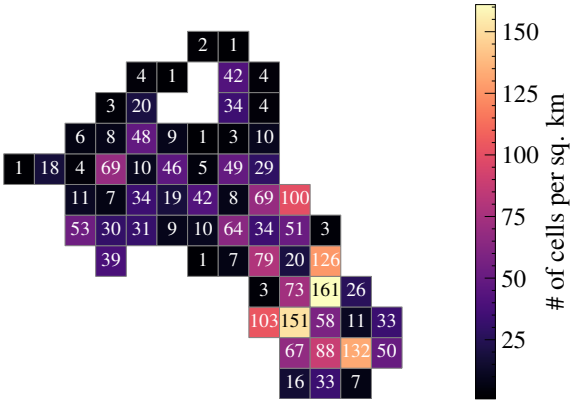


Figure 4.6: Number of cells per sq. km in an urban area with >2.2k cells.

160 per sq. km in highly saturated hotspots. Notably, some adjacent tiles exhibit substantial disparities, with cell counts increasing from as few as 3 cells in one tile to as many as 103 in a neighboring one. In CHOs, this means that a UE traversing these areas must dynamically account for a highly variable and often large set of candidate cells.

Key Takeaways: Cellular network deployments are highly heterogeneous in terms of density, which is correlated with population. In our dataset, there are 650+ cells per sq. km in the urban center of the capital city, while in remote areas, there are as few as 0.12 per sq. km. In urban areas, the number of candidate cells for CHO can be very large and depends on cell density. There are even significant differences in cell counts between adjacent tiles (1 sq. km). This means that the signaling cost for preparing target cells is not uniform across the network, as it depends on the cell deployment density and properties (distance, frequency, etc). Therefore, the system model in Section 4.4 considers cell-dependent signaling costs.

HO Failures & Delays. We analyze factors that increase HOFs and delays, thereby highlighting the scenarios in which CHOs offer clear benefits.

Figure 4.7 presents the hourly evolution of successful HO delays (in ms) and HOF rate (in %) for 1 week in the entire country. Boxplots aggregate cell-level data from the same hour. The two metrics exhibit a diurnal pattern, reaching a max of 0.08% HOF rate and 62 ms latencies (median values) at 15:00, which is the hour when most HOs occur and the network is most congested [113]. Notably, from the pre-dawn median minimum (06:00) to the max peak at 15:00, there is an increase of 252% in the HOF rate; at the same time, HO delay increases by 28%. Also, during the night hours 00:00–06:00, even though the HOF rate decreases significantly, the delay of successful HOs increases by approximately 3 ms. This is partially due to the MNO applying dynamic energy-saving policies that switch off cells acting as capacity boosters when they are not needed to meet the demand [144]. Clearly, existing HO mechanisms offer room for improvement, especially during peak hours, to reduce HOF rates and delays, which can be achieved by leveraging a dynamic and robust THO/CHO algorithm.

We further analyze which features of the source and target cells mainly affect

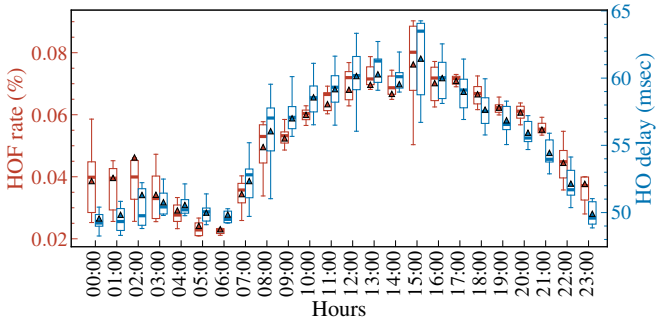


Figure 4.7: HOF rate (left, red y-axis) and delay of successful HOs (right, blue y-axis) per hour for a 1-week period. Triangles in boxplots depict the mean.

the HOF rate. For that, we focus on source-target cell pairs with at least one HOF registered per day (excluding approximately 3.4% that are outliers), and study the effect of: (*i*) vendor, (*ii*) frequency band, (*iii*) model name, (*iv*) transmit power, (*v*) cell type (macro, micro, etc.), (*vi*) region (north, south, etc.) and (*vii*) area type (urban/rural).

We start this analysis with a non-parametric Kruskal-Wallis test, which confirms that all examined factors affect the HOF rate ($p < 0.05$).[1] To determine the importance of each feature, we run multiple Machine Learning (ML) models. We discard features with high multicollinearity (VIF > 3) and Pearson correlation (>0.5), and include the number of HOs as an input feature to account for spatio-temporal variations (Figures 4.6 and 4.7). The HOF rate is log-transformed, given its heavy-tailed distribution. Among the evaluated models (linear/lasso/ridge regression, k-nearest neighbours, and random forest), the random-forest regressor achieved the best performance, reaching an R^2 of 0.8 (compared to at most 0.58 for the other models) and RMSE of 0.66 (compared to at most 0.97 for the other models). As a reference, a naive model that always predicts, in the test set, the mean HOF rate from the training set, achieves an RMSE of 1.47. Permutation-based importance reveals that the number of HOs is by far the strongest predictor, followed by the cell frequency band and the vendor; excluding these last two features drops the performance to R^2=0.6, confirming that frequency and vendor still contribute meaningfully.

Figure 4.8 shows the effect of these latter features on HOs and HOFs. The x-axis indicates whether the HO occurred in the same (intra) or different (inter) frequency and antenna vendor, as well as if the frequencies of the source and target cells lie in the low ($\leq$ 2GHz) or mid-high ($>$ 2GHz) spectrum. The y-axis shows the HOF rate in ascending order, the delay of successful HOs and the percentage of total HO and HOF count within each category. We observe that the mean HOF rate in inter-frequency HOs, mainly when the source and/or target cells operate in low frequencies, is $\approx 0.26\%$, i.e., $\times 2.4$ more than the 0.11% of the intra-frequency ones. However, the HO delay for the former is no more than 1.5 ms higher. Intra-frequency

[1] Normality and homoscedasticity assumptions are violated; hence, analysis of variance (ANOVA) [100] was not tested.

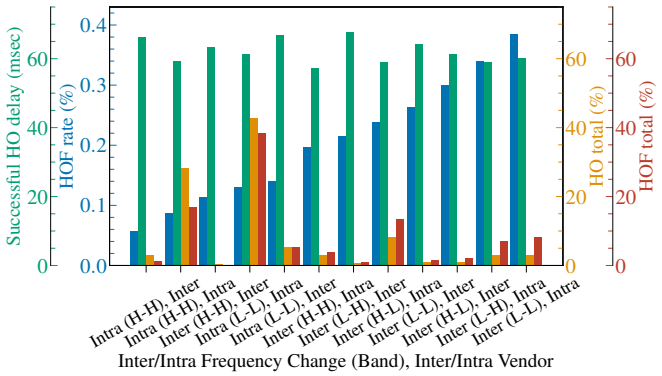


Figure 4.8: (left to right) HOF rate, delays of successful HOs, percentage of total HOs, and HOFs based on the source-target frequency bands and vendors.

HOs in the same low-spectrum frequency handle the majority of HOs (i.e., 42%), but their contribution to failures closely matches this number (38.4% of all HOFs). Lastly, note that HO delay is mainly affected by the change in vendor: even though 88% of HOs are intra-vendor with a delay of 59–60 ms, inter-vendor HOs exhibit a delay that is 5–6 ms higher. These inter-vendor HOs are prevalent at regional borders (e.g., west-east border), given that a single vendor predominantly serves each region in the studied country.

This analysis demonstrates that accounting for the previously mentioned cell-level features is crucial for HO optimization, particularly for minimizing the impact of increased HO delays and HOFs.

Key Takeaways: HO failures and delays are strongly influenced by: (*i*) time of day, exhibiting diurnal patterns that peak at 15:00 with a 0.08% HOF rate and a 62 ms HO delay (median values); (*ii*) frequency changes between the source and target cells, with inter-frequency HOs having ×2.4 higher HOFs than intra-frequency HOs, although with a slight difference in HO delay, and (*iii*) vendor transitions, where inter-vendor HOs exhibit 5–6 ms higher delay than intra-vendor. This evidences the potential benefit of accounting for the previous cell-level characteristics for HO optimization, as we consider in the design of our algorithm.

4.4. System Model and Problem Statement

Modeling Components. We consider a heterogeneous cellular network comprising a set $\mathcal{J}$ of J cells that serve a set $\mathcal{I}$ of I UEs. We assume that a central controller takes decisions for multiple UEs/cells [53] in a time-slotted manner for a set $\mathcal{T}$ of T slots [33], [111] with each slot lasting hundreds of milliseconds (see Section 4.2), as enabled by near-RT RIC in O-RAN[12].[2]

These decisions concern both THOs and CHOs, and are taken on the *same time-scale*. The set $\mathcal{I}$ is partitioned into the set $\mathcal{I}_{\text{THO}}$ of I_{THO} UEs and $\mathcal{I}_{\text{CHO}}$ of I_{CHO} UEs, who follow THOs (*THO-enabled*) and CHOs (*CHO-enabled*), respectively, where

[2]The modeling and optimization framework of this section can be extended and implemented by other RAN systems, not solely O-RAN, as long as there is support for centralized decision-making that is fed by signal measurements.

$\mathcal{I}_{\text{THO}} \cap \mathcal{I}_{\text{CHO}} = \emptyset$ and $\mathcal{I}_{\text{THO}} \cup \mathcal{I}_{\text{CHO}} = \mathcal{I}$. This distinction can arise due to service-level agreements (SLAs), network slicing policies [145], or vertical-specific requirements, in which different classes of UEs are managed separately; e.g., enhanced-reliability or low-latency UEs may be assigned to CHO. Alternatively, an MNO may apply a rule-based classification, informed by prior data (e.g., historical patterns), to assign UEs to either *HO type*.

For the THO-enabled UEs, we introduce the decisions:

$$\boldsymbol{x}_t = \big(x_{ij}(t) \in \{0,1\}, i \in \mathcal{I}_{\text{THO}},\ j \in \mathcal{J}\big),$$

where $x_{ij}(t) \in \{0,1\}$ defines the *explicit one-cell association decision* with $x_{ij}(t) = 1$ if user i is assigned from the controller to cell j in slot t. For the CHO-enabled UEs, we introduce:

$$\boldsymbol{y}_t = \big(y_{ij}(t) \in \{0,1\}, i \in \mathcal{I}_{\text{CHO}},\ j \in \mathcal{J}\big),$$

where $y_{ij}(t) \in \{0,1\}$ is the *preparation decision* (*implicit multi-cell association decision*): $y_{ij}(t) = 1$ means that cell j is prepared from the controller in slot t for user i and the user can decide whether to connect to this cell (as explained later). These decisions are drawn from the sets:

$$\mathcal{X} = \Big\{ \boldsymbol{x} \in \{0,1\}^{I_{\text{THO}} \cdot J} \ \Big|\ \sum_{j \in \mathcal{J}} x_{ij} = 1, i \in \mathcal{I}_{\text{THO}} \Big\},$$

where naturally each $i \in \mathcal{I}_{\text{THO}}$ is assigned to one cell, and:

$$\mathcal{Y} = \Big\{ \boldsymbol{y} \in \{0,1\}^{I_{\text{CHO}} \cdot J} \ \Big|\ 1 \leq \sum_{j \in \mathcal{J}} y_{ij} \leq b_i, i \in \mathcal{I}_{\text{CHO}} \Big\},$$

where no more than b_i cells can be prepared for $i \in \mathcal{I}_{\text{CHO}}$.

The key metric is the SINR for the signal delivered by cell $j \in \mathcal{J}$ to any user $i \in \mathcal{I}$ in slot $t \in \mathcal{T}$:

$$s_{ij}(t) = \frac{q_j \phi_{ij}(t)}{W_j \sigma^2 + \sum_{k \in \mathcal{B}_j} q_k \phi_{ik}(t)},$$

where q_j is the transmit power of cell j, $\mathcal{B}_j$ the set of cells that operate in the same frequency as j, $\phi_{ij}(t)$ the channel gain (including pathloss, shadowing, and antenna gains), W_j is the bandwidth, and σ^2 the power spectral density. In line with previous works [33], [34], [36], [109], $s_{ij}(t)$ is the average SINR in t, since UEs report multiple values during each slot.

We distinguish two cases for the *maximum throughput* a UE can receive from a cell, based on whether it is THO- or CHO-enabled: For the former [33], [34], [146], this throughput is expressed as:

$$c_{ij}(t) = W_j \log\big(1 + s_{ij}(t)\big), i \in \mathcal{I}_{\text{THO}},$$

while for the latter, we introduce:

$$c'_{ij}(t) = \begin{cases} W_j \log\big(1 + s_{ij}(t)\big), \text{ if } j = \underset{k:\, y_{ik}(t)=1}{\arg\max}\ s_{ik}(t) \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{4.1}$$

where $i \in \mathcal{I}_{\text{CHO}}$. The throughput of the CHO-enabled UEs in eq. (4.1) depends on their own decision, which, in turn, depends on the measured signals and the list of prepared cells it receives from the network. We consider here the standard cell-selection rule, in which the CHO-enabled UE connects to the best-SINR cell among those prepared for it. [3]

Finally, we define the load of each cell $j \in \mathcal{J}$:

$$\ell_j(t) = \sum_{i \in \mathcal{I}_{\text{THO}}} x_{ij}(t) + \sum_{i \in \mathcal{I}_{\text{CHO}}} y_{ij}(t) \leq C_j, \tag{4.2}$$

as the total number of associated users (both THO- and CHO-enabled), which cannot exceed the upper bound C_j, in lieu of the actual spectrum budget capacity. Note that for the CHO-enabled users, the cell resources are reserved independently of whether the UEs will actually select the cell, and hence CHOs are likely to induce unnecessary resource waste. This is a key CHO issue that our model tackles.

To streamline the presentation, we define $\boldsymbol{z}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ and the associated set:

$$\mathcal{Z} = \Big\{ \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \mid \ell_j \leq C_j,\ \forall j \in \mathcal{J},\ \boldsymbol{x} \in \mathcal{X},\ \boldsymbol{y} \in \mathcal{Y} \Big\}.$$

In our framework, the network controller makes *joint* decisions for the THO- and CHO-enabled users, whose performance is coupled through the load of each cell, as can be seen in eq. (4.2). With this in mind, we introduce the *utility* function the controller needs to maximize:

$$g_t(\boldsymbol{z}_t) = \sum_{j \in \mathcal{J}} \Bigg(\underbrace{\sum_{i \in \mathcal{I}_{\text{THO}}} x_{ij}(t) \log \frac{c_{ij}(t)}{\ell_j(t)}}_{\substack{\text{explicitly assigned UEs} \\ \text{(can do THO)}}} + \underbrace{\sum_{i \in \mathcal{I}_{\text{CHO}}} y_{ij}(t) \log \frac{c'_{ij}(t)}{\ell_j(t)}}_{\substack{\text{implicitly assigned UEs} \\ \text{(can do CHO)}}} \Bigg) \tag{4.3}$$

where the two terms define the throughput of the THO-enabled and CHO-enabled UEs. [4] We assume the cell resources are allocated fairly across the users via, e.g., a round robin or a proportional-fair scheduler [114]. The logarithmic transformation balances throughput across users to achieve fairness [33]; however, other mappings or schedulers can be used as easily.

Using the association decision $x_{ij}(t)$ for user $i \in \mathcal{I}_{\text{THO}}$, the THO is modeled with the change $x_{ij}(t) \neq x_{ij}(t-1)$. This way, the total number of THOs can be captured with the norm $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|$, as in [33]. Nevertheless, we are interested in the *THO delay (switching) cost* and not merely in the number of THOs. Following our findings in Section 4.3, we propose using the weighted norm $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{A_t}$, where $\boldsymbol{A}_t = \text{diag}(a_n(t) > 0)$ is a positive definite matrix with its diagonal weights $a_n(t) \in [0, 1]$, $n = 1, \ldots, I_{\text{THO}} \cdot J$, penalizing differently the THOs for each UE-cell pair; a penalty that may even change over time.[5]

[3] To avoid often CHOs and ping-pong effects [31] when finding the highest-SINR prepared cell, it is possible to subtract a cell-specific offset, as happens in the A3 event [46], [80], [81].

[4] We consider that there is no idle cell, and in case of zero throughput, $c_{ij}(t) + 1$ or $c'_{ij}(t) + 1$ can be used inside the logarithm.

[5] E.g., 3G UEs have higher cost for changing cells as they are more prone to HO failures [146]. In this way, the model accounts for the UE's HO capabilities.

On the other hand, a CHO is executed from the UE based on the prepared cells by the controller captured through $\boldsymbol{y}_t$: the user's decision in the CHO case is to be assigned to the highest-SINR prepared cell. Clearly, frequent cell preparations and releases may lead to increased signaling [44], [46], especially in dense cell deployments (can be up to 161 cells, see Section 4.3). Hence, we introduce the *CHO signaling (switching) cost* $\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_{B_t}$. In this case, $\boldsymbol{B}_t = \text{diag}(b_{n'}(t) > 0, n' = 1, \ldots, I_{\text{CHO}} \cdot J)$ is a diagonal positive definite matrix with weights $b_{n'}(t) \in [0, 1]$, to penalize differently the preparations and releases of cells for each UE-cell pair[6] in slot t. Given that THOs are more prone to failures and delays than CHOs, it holds that $a_n(t) > b_{n'}(t),\ \forall t \in \mathcal{T}$.

These THO and CHO switching costs can be combined as:

$$\|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|^2_{C_t} = \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2_{A_t} + \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|^2_{B_t},$$

where $\boldsymbol{C}_t$ is the full block diagonal matrix:

$$\boldsymbol{C}_t = \begin{bmatrix} \boldsymbol{A}_t & 0 \\ 0 & \boldsymbol{B}_t \end{bmatrix},$$

and the induced norm and its dual [117], are:

$$\|\boldsymbol{x}_t\|^2_{A_t} = \sum_{n=1}^{I_{\text{THO}} \cdot J} a_n(t) x_n(t)^2, \|\boldsymbol{x}_t\|^2_{A_t *} = \sum_{n=1}^{I_{\text{THO}} \cdot J} x_n(t)^2 / a_n(t),$$

and similarly for $\boldsymbol{y}_t$.

Putting the above together, the problem that the network controller wishes to address is the following:

$$\begin{aligned} \mathbb{P}_1 : \max_{\{\boldsymbol{z}_t\}_t} & \sum_{t=1}^{T} \Big(g_t(\boldsymbol{z}_t) - \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t} \Big) \\ \text{s.t.} \quad & \boldsymbol{z}_t \in \mathcal{Z},\ \forall t \in \mathcal{T}, \end{aligned}$$

where $g_t(\boldsymbol{z}_t) - \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t}$ is the *objective* function. Next, we explain in detail why $\mathbb{P}_1$ cannot be solved offline.

Optimization Challenges. Solving $\mathbb{P}_1$ offline is impossible since, at $t = 1$, the controller lacks knowledge of the future SINRs of UEs as it has no access to, or influence over, their mobility. Thus, $\mathbb{P}_1$ must be tackled in an online fashion. In fact, in increasingly-many scenarios, the problem parameters (SINRs, loads, switching costs, etc.) are unknown even at the beginning of each slot; they are only revealed *after* the association and preparation decisions are made. For example, while it may seem the controller could observe the current SINRs before making a decision, in practice, there is a non-negligible delay between the time a UE measures and reports these values and the time the network processes this information. And for fast-moving users or highly-volatile environments, this delay will yield outdated

[6] E.g., the signaling cost might be more detrimental during peak-hours.

information. This means that we need an online *learning* approach, where both the explicit (for THO) and implicit (for CHO) association decisions are made based on historical UE, network, and environment data, without presuming knowledge of their future values.

What is more, the signaling and switching overheads depend on successive decision changes: a user that remains in acell or a cell that remains prepared does not incur additional costs. This introduces a memory effect in the optimization, as past decisions influence current decisions. We also stress that the A_t-norm and B_t-norm imply the costs change over time; however, this cost does not depend on previous decisions. Finally, $\mathbb{P}_1$ is further compounded by the discreteness of the variables $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ which prevents the application of off-the-shelf OCO techniques.

Despite these challenges, the goal is to design an online learning algorithm that determines UE-cell associations and preparations and is oblivious to all time-varying and unknown *system conditions*, including future SINR, load, switching costs, and HO delays for each UE-cell pair.

4.5. Traditional-Conditional Handover Learning

To leverage the OCO/SOL toolbox for tackling the problem at hand, the following two main issues need to be addressed: *(i)* the discreteness of decision variables, and *(ii)* the (convex) max operator in the definition of the throughput concerning the CHO-enabled UEs, as can be seen in eqs. (4.1) and (4.3). In a nutshell, our solution strategy involves relaxing the discreteness of the decision variables, transforming the resulting continuous problem into a concave one, solving the new concave problem, and then mapping the obtained continuous solution back to the discrete domain via careful rounding.

Relaxation & Transformation. First, we define the convex hulls $\mathcal{X}^{\mathrm{c}} = \mathrm{co}(\mathcal{X})$, $\mathcal{Y}^{\mathrm{c}} = \mathrm{co}(\mathcal{Y})$, $\mathcal{Z}^{\mathrm{c}} = \mathrm{co}(\mathcal{Z})$ that relax the integrality of the decision variables:

$$\mathcal{X}^{\mathrm{c}} = \Big\{ \boldsymbol{x} \in [0,1]^{I_{\mathrm{THO}} \cdot J} \;\Big|\; \sum_{j \in \mathcal{J}} x_{ij} = 1, i \in \mathcal{I}_{\mathrm{THO}} \Big\},$$

$$\mathcal{Y}^{\mathrm{c}} = \Big\{ \boldsymbol{y} \in [0,1]^{I_{\mathrm{CHO}} \cdot J} \;\Big|\; 1 \leq \sum_{j \in \mathcal{J}} y_{ij} \leq b_i, i \in \mathcal{I}_{\mathrm{CHO}} \Big\},$$

$$\mathcal{Z}^{\mathrm{c}} = \Big\{ \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \mid \ell_j \leq C_j,\ \forall j \in \mathcal{J},\ \boldsymbol{x} \in \mathcal{X}^{\mathrm{c}},\ \boldsymbol{y} \in \mathcal{Y}^{\mathrm{c}} \Big\}.$$

Then, using the properties of logarithm, eq. (4.3) becomes:

$$g_t(\boldsymbol{z}_t) = \underbrace{\sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_{\mathrm{THO}}} x_{ij}(t) \log c_{ij}(t)}_{g_t^{\mathrm{THO}}(\boldsymbol{x}_t)} + \underbrace{\sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_{\mathrm{CHO}}} y_{ij}(t) \log c'_{ij}(t)}_{g_t^{\mathrm{CHO}}(\boldsymbol{y}_t)}$$
$$\underbrace{- \sum_{j \in \mathcal{J}} \ell_j(t) \log \ell_j(t)}_{g_t^{\mathrm{load}}(\boldsymbol{x}_t,\ \boldsymbol{y}_t)},$$

where the throughput for the THO-enabled UEs, $g_t^{\text{THO}}(.)$, is linear in $\boldsymbol{x}_t \in \mathcal{X}^{\text{c}}$, and the load penalty, $-g_t^{\text{load}}(.)$, describes the entropy; thus, both are *concave*. On the other hand, the throughput for the CHO-enabled UEs, $g_t^{\text{CHO}}(.)$, is not concave even when $\boldsymbol{y}_t \in \mathcal{Y}^{\text{c}}$, due to the influence of preparation decisions $\boldsymbol{y}_t$ on $c'_{ij}(t)$ through the max operator, see eq. (4.1). To illustrate this more clearly, we can equivalently write this part as follows:

$$g_t^{\text{CHO}}(\boldsymbol{y}_t) \triangleq \sum_{i \in \mathcal{I}_{\text{CHO}}} \max_{j:\, y_{ij}(t)=1} \log c_{ij}(t), \tag{4.9}$$

where this transformation achieves the same/desired behavior: the CHO-enabled UE obtains throughput from the cell with the best SINR *among the prepared ones*. Given that the max operator of eq. (4.9) is nonsmooth and convex, we introduce a linear masking mechanism that leads to a *concave* surrogate, inspired by the standard LogSumExp function [147] used in deep neural networks (DNNs):

$$\tilde{g}_t^{\text{CHO}}(\boldsymbol{y}_t) \triangleq \sum_{i \in \mathcal{I}_{\text{CHO}}} \frac{1}{\alpha} \log\left(\sum_{j \in \mathcal{J}} y_{ij}(t)\, c_{ij}(t)^{\alpha} \right), \tag{4.10}$$

where $\alpha > 0$ controls the tightness of the approximation.

Lemma 4.1 (Approximation of CHO Throughput)**.** *For $\boldsymbol{y}_t \in \mathcal{Y}^c$ and $t \in \mathcal{T}$,*

$$\lim_{\alpha \to \infty} \tilde{g}_t^{CHO}(\boldsymbol{y}_t) = g_t^{CHO}(\boldsymbol{y}_t).$$

Proof. Let $d_i^*(t) = \max_{j:\, y_{ij}(t)=1} \log c_{ij}(t)$ and $d_{ij}(t) = \log c_{ij}(t)$. Then, eq. (4.10) becomes:

$$\tilde{g}_t^{\text{CHO}}(\boldsymbol{y}_t) = \sum_{i \in \mathcal{I}_{\text{CHO}}} \frac{1}{\alpha} \log\left(\sum_{j \in \mathcal{J}} y_{ij}(t) e^{\alpha d_{ij}(t)} \right) =$$

$$\sum_{i \in \mathcal{I}_{\text{CHO}}} \frac{1}{\alpha} \log\left(e^{\alpha d_i^*(t)} \left(1 + \sum_{j:\, d_{ij}(t) \neq d_i^*(t)} y_{ij}(t) e^{\alpha \left(d_{ij}(t) - d_i^*(t) \right)} \right) \right) =$$

$$\sum_{i \in \mathcal{I}_{\text{CHO}}} \left(d_i^*(t) + \frac{1}{\alpha} \log\left(1 + \sum_{j:\, d_{ij}(t) \neq d_i^*(t)} y_{ij}(t) e^{\alpha \left(d_{ij}(t) - d_i^*(t) \right)} \right) \right)$$

Since $d_{ij}(t) < d_i^*(t)$ by definition, then $e^{\alpha \left(d_{ij}(t) - d_i^*(t) \right)} \to 0$ as $\alpha \to \infty$, and indeed, converges to the max, namely, $d_i^*(t)$.

□

From Lemma 4.1, eq. (4.10) approximates eq. (4.9), and the transformed utility:

$$\tilde{g}_t(\boldsymbol{z}_t) = g_t^{\text{THO}}(\boldsymbol{x}_t) + \tilde{g}_t^{\text{CHO}}(\boldsymbol{y}_t) - g_t^{\text{load}}(\boldsymbol{x}_t, \boldsymbol{y}_t) \tag{4.11}$$

is concave, as desired.

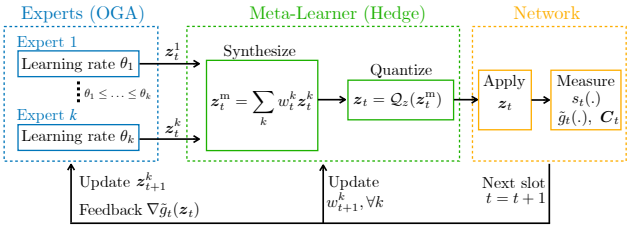


Figure 4.9: Learning mechanism. *(i)* Each expert uses a different learning rate and proposes a HO policy to the meta-learner (blue box). *(ii)* The meta-learner combines the proposals, using the experts' weights, and implements the result after discretizing it (green box). *(iii)* The mechanism observes the performance and assesses how good each expert's proposal was (orange box). *(iv)* The experts adapt their decisions and the meta-learner adapts their weights.

4

The Meta-Learning Approach. We approach $\mathbb{P}_1$ as a *smoothed online learning* problem and solve it using a *meta-learning* approach based on the *experts* framework [18], [116], as can be seen Figure 4.9. This is particularly well-suited to our setting because switching costs are incorporated into the formulation, and system conditions can vary significantly across time slots due to, e.g., unknown user mobility. As a result, deploying a single learner with a fixed learning rate may perform well in some regimes but poorly in others.

We deploy a set $\mathcal{K}$ of K learning agents, called *experts*, or simply *learners*, each with a different learning step/rate $\theta = (\theta_k, k \in \mathcal{K})$ that is applied to online gradient ascent (OGA) [15]. An expert with a larger learning rate puts more emphasis on the latest gradient and hence adapts more quickly to rapid changes in SINRs and signaling costs, making it more suitable for highly volatile scenarios. In contrast, experts with a smaller learning rate update their decisions more carefully (do not move fast from previous decisions) and hence are less affected by parameter fluctuations. This makes them better suited for relatively stationary scenarios as they induce less handovers (decision changes).

Given that the volatility of the environment and UEs' mobility patterns are unknown in advance, we use a large enough set $\mathcal{K}$ of experts, each tuned with a different learning rate. The rationale is that, with careful selection of the rates (number of experts and steps), at least one expert is guaranteed to perform well for the encountered scenario. We identify the best expert for each scenario at runtime by using a meta-learning algorithm, which implements Hedge [148] (and not OGA as the experts) and learns how much weight to put on each expert's proposal. These weights are dynamically updated based on observed expert performance.

Formally, at the beginning of each slot $t = 1, ..., T$, each expert $k \in \mathcal{K}$ shares its suggestion $\boldsymbol{z}_t^k = (\boldsymbol{x}_t^k, \boldsymbol{y}_t^k)$, where we initialize $z_0^k = 0$ for all k. The meta-learner combines these suggestions into a single decision $\boldsymbol{z}_t^{\mathrm{m}} = (\boldsymbol{x}_t^{\mathrm{m}}, \boldsymbol{y}_t^{\mathrm{m}})$ as follows:

$$\boldsymbol{z}_t^{\mathrm{m}} = \sum_{k \in \mathcal{K}} w_t^k \boldsymbol{z}_t^k, \tag{4.12}$$

where $\boldsymbol{w}_t = (w_t^k, k \in \mathcal{K})$, with $\boldsymbol{w}_t^\top \mathbf{1}_K = 1$, are the meta-learner's adaptive weights for each expert. The goal is to assign higher weights to experts that perform better.

Algorithm 2: CONditional-TRAditional HOs (`CONTRA`)

Required: Step η for meta-learner and $\{\theta_k\}_{k\in\mathcal{K}}$ for experts
Initialize: Sort $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_K$ and set $w_1^k = \frac{1+1/K}{k(k+1)}$
for $t = 1, 2, \ldots, T$ **do**
  Each expert $k \in \mathcal{K}$ shares its decision $\boldsymbol{z}_t^k$
  Combine all decisions $\boldsymbol{z}_t^k$ into $\boldsymbol{z}_t^{\mathrm{m}}$ using (4.12)
  Create an implementable discrete decision $\boldsymbol{z}_t = Q_{\mathcal{Z}}(\boldsymbol{z}_t^{\mathrm{m}})$
  Observe SINRs, $\tilde{g}_t(.)$ and switching costs C_t
  Update the weights of experts using (4.13) and (4.14)
  Send $\nabla g_t(\boldsymbol{z}_t)$ to each expert
  Each expert updates its decision using (4.15)
end

Since the decision $\boldsymbol{z}_t^{\mathrm{m}}$ of the meta-learner takes continuous values due to $\boldsymbol{w}_t$, we apply a quantization function $Q_{\mathcal{Z}}$ to project it back to a valid (i.e., implementable) discrete decision $\boldsymbol{z}_t$. We require that the quantization is unbiased, i.e., $\mathbb{E}[\boldsymbol{z}_t] = \boldsymbol{z}_t^{\mathrm{m}}$, which holds by: (i) picking a cell j to assign a THO-enabled UE $i \in \mathcal{I}_{\mathrm{THO}}$ with probabilities $\boldsymbol{x}_i^{\mathrm{m}}(t)$, creating $\boldsymbol{x}_t \in \mathcal{X}$, and (ii) deciding whether to prepare a cell j for a CHO-enabled UE $i \in \mathcal{I}_{\mathrm{CHO}}$ through sampling from a Bernoulli distribution with probabilities $\boldsymbol{y}_{ij}^{\mathrm{m}}(t)$, creating $\boldsymbol{y}_t \in \mathcal{Y}$.

Once the decision $\boldsymbol{z}_t$ is implemented, the controller observes the SINR values $s_{ij}(t)$ for all UEs and cells, and calculates $\tilde{g}_t(.)$ through eq. (4.11), as well as the switching costs (by observing C_t). We pad with zeros the entries $s_{ij}(t)$ for which no SINR is received, because these cells are unreachable. As a next step, the meta-learner evaluates the decision $\boldsymbol{z}_t^k$ of each expert using the surrogate (i.e., partially linearized) loss:

$$l_t(\boldsymbol{z}_t^k) = -\langle \nabla \tilde{g}_t(\boldsymbol{z}_t), \boldsymbol{z}_t^k - \boldsymbol{z}_t \rangle + \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t} \tag{4.13}$$

and updates its weights using the step η (defined later) as:

$$w_{t+1}^k = \frac{w_t^k e^{-\eta l_t(\boldsymbol{z}_t^k)}}{\sum_{k\in\mathcal{K}} w_t^k e^{-\eta l_t(\boldsymbol{z}_t^k)}}. \tag{4.14}$$

Lastly, it sends the gradient $\nabla g_t(\boldsymbol{z}_t)$ in the implementable decision $\boldsymbol{z}_t$ to all experts, and they update their choices via OGA:

$$\boldsymbol{z}_{t+1}^k = \Pi_{\mathcal{Z}}\Big(\boldsymbol{z}_t^k + \theta_k \nabla \tilde{g}_t(\boldsymbol{z}_t)\Big). \tag{4.15}$$

The projection $\Pi_{\mathcal{Z}}(.)$ ensures that the proposed decisions $\boldsymbol{z}_{t+1}^k$ lie in the feasible space, namely, $\boldsymbol{z}_{t+1}^k \in \mathcal{Z}$. A summary of the steps can be seen in Algorithm 2 (`CONTRA`).

Performance Guarantees. To assess the performance of our algorithm, we compare it against a powerful benchmark (i.e., *oracle*) that has full a priori knowledge of

the system conditions (i.e., SINR, load, costs etc) and can choose the best possible sequence of decisions over the entire horizon to maximize the cumulative objective of $\mathbb{P}_1$. This serves as a highly competitive benchmark, exceeding both static (single best solution for all time slots) and dynamic ones (best solution for each time slot independently) [16], [24].

For that, we leverage the *Expected Dynamic Regret* [25]:

$$\mathbb{E}\left[\mathcal{R}_T\right] \triangleq \sum_{t=1}^{T} \left(\tilde{g}_t(\boldsymbol{z}_t^\star) - \|\boldsymbol{z}_t^\star - \boldsymbol{z}_{t-1}^\star\|_{C_t}\right) - \sum_{t=1}^{T} \mathbb{E}\Big[\tilde{g}_t(\boldsymbol{z}_t) - \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t}\Big], \tag{4.16}$$

where $\{\boldsymbol{z}_t\}_t$ and $\{\boldsymbol{z}_t^\star\}_t$ are the algorithm's and powerful-oracle's decisions, respectively. The latter is the solution of $\mathbb{P}_1$, and the expectation captures the randomization in `CONTRA`'s decisions due to quantization $Q_{\mathcal{Z}}$. Our goal is to design an algorithm that ensures this gap diminishes with time, $\lim_{T\to\infty} \mathbb{E}[\mathcal{R}_T]/T = 0$ for any possible benchmark sequence $\{\boldsymbol{z}_t^\star\}_t$. It is important to highlight that achieving close-to-zero expected dynamic regret is particularly challenging.

Ideally, designing an effective algorithm in such settings requires prior knowledge of how much optimal decisions can vary over time, commonly characterized by the *path length*:

$$P_T = \sum_{t=1}^{T} \|\boldsymbol{z}_t^\star - \boldsymbol{z}_{t-1}^\star\|_{C_t}.$$

However, this information is not available in practice. To overcome this limitation, we have relied on a meta-learning algorithm that adaptively learns from a pool of experts with different learning rates, each tailored to perform well under different system conditions. To bound the expected dynamic regret, we first prove the following lemma.

Lemma 4.2 (Bound of Domain and Gradients)**.** *By considering:* $a_n(t) \le a_{\max}$, $b_{n'}(t) \le b_{\max}$, $c'_n(t) \le c_n(t) \le c_{\max}$ *and* $M = \max\{\log c_{\max} - 1,\ \log I + 1\}$, *with* $t \in \mathcal{T}$, *it holds that:*

- $\|\boldsymbol{z} - \boldsymbol{z}'\|_2 \le \sqrt{2\, I_{THO}} + \sqrt{I_{CHO}\,(J-1)} \triangleq D$,
- $\|\boldsymbol{z} - \boldsymbol{z}'\|_{C_t} \le \sqrt{2\, I_{THO}\ a_{\max}} + \sqrt{I_{CHO}\,(J-1)\ b_{\max}} \triangleq D_C$,
- $\|\boldsymbol{z} - \boldsymbol{z}'\|_{C_t *} \le \sqrt{2\, I_{THO}/a_{\max}} + \sqrt{I_{CHO}\,(J-1)/b_{\max}} \triangleq D_{C^*}$,
- $\|\nabla \tilde{g}_t(\boldsymbol{z})\|_2 \le M\sqrt{I\, J} \triangleq G$,
- $\|\nabla \tilde{g}_t(\boldsymbol{z})\|_{C_t} \le M\sqrt{(I_{THO}\ a_{\max} + I_{CHO}\ b_{\max})\, J} \triangleq G_C$.

Proof. We provide the full proof in the Appendix. □

The main result for the performance guarantee of `CONTRA` is presented in the following lemma.

Lemma 4.3 (Performance / Optimality Guarantee)**.** *Using the parameters:*

- $K = \left\lceil \log_2 \sqrt{1+2T} \right\rceil + 1,$ (4.17)
- $\theta_k = 2^{k-1} \sqrt{\dfrac{D_C^2}{T(G^2+2G_C)}},\ k = 1, \ldots, K,$ (4.18)
- $\eta = 1/\sqrt{T\nu}$ *with*
  $\nu \triangleq (D_C{+}1/8)(GD{+}2D_C)^2$

the discrete decisions $\{\boldsymbol{z}_t\}_t$ *of our algorithm* `CONTRA` *ensure:*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \sqrt{T}\Big(\sqrt{\nu}\left(1+\ln(1/w_1^k)\right) + \sqrt{(G^2+2G_C)(D_C^2+2D_{C*}P_T)}\ \Big) + T\Big(G+\sqrt{a_{\max}+b_{\max}}\Big)\sqrt{I_{CHO}J/4 + I_{THO}(1-1/J)}, \quad (4.19)$$

Proof. We provide the full proof in the Appendix. □

With Lemma 4.3, we bound the expected regret of the implementable decisions, in contrast to previous works [33], [36]. Even though the continuous preparation variables achieve sublinear dynamic regret, the discretization introduces an unavoidable non-diminishing error. As shown in Section 4.7, this error is small in practical scenarios, and the algorithm converges towards the optimal solutions.

Computational Complexity. The primary source of complexity in our algorithm arises from the projection step performed by each learner $k \in \mathcal{K}$ during OGA, as shown in eq. (4.15); all other computations are executed in constant time $\mathcal{O}(1)$. The projection difficulty depends on *(i)* the structure of the feasible set, which essentially depends on the constraints of our problem, and *(ii)* the size of the problem, which in practice corresponds to the area monitored by each controller (i.e., the number of UEs and cells). We note that for convex sets in general, there might be no closed-form solution, with the projection requiring $\mathcal{O}(d^3)$, where d is the set's dimension.

Our current implementation indeed computes the projections directly by solving the concave subproblems, which have been found sufficient for the studied problems. Specifically, and despite relying on a general-purpose solver rather than specialized projection routines and a modest hardware (default) setup of a MacBook Pro equipped with an Apple M1 chip (8-core CPU), the per-slot inference (execution) time remains below the near-RT threshold required in O-RAN operation when up to 150 UEs and 25 cells are considered.

More precisely, the inference time increases from 18 ms ($I = 1$ user) to 62 ms ($I = 150$ users) when $J = 5$ cells. For $J = 15$ ($J = 25$), it begins near 18 ms (18 ms) for $I = 1$ user and reaches up to 384 ms (1 s) for $I = 150$ users. It is important to note that the exact runtime may vary depending on the specifications of different machines. For larger search spaces, one can employ tailored projection algorithms, such as [149], [150], and/or execute the algorithm on GPUs or high-performance computing servers.

4.6. Dynamic Handover Type Selection

Problem Formulation. In contrast to the framework of Section 4.4, we now consider a more flexible and dynamic formulation in which users are not a priori assigned to a specific HO type. Instead, the network controller can decide on-the-fly (i.e., at each slot t), whether a UE should be THO- or CHO-enabled, based on current system conditions and performance trade-offs, enabling a more responsive and efficient HO strategy, as envisioned by 6G and NextG [51]. Specifically, we allow each UE to be in either HO mode and decide this dynamically: when a UE moves fast, it may benefit from CHO due to the rapid signal fluctuations that lead to higher THO failures and delays, while it might find THO suitable when it moves slowly.

In this case, the controller problem becomes:

$$\mathbb{P}_2 : \max_{\{\boldsymbol{z}_t\}_t} \sum_{t=1}^{T} \Big(g_t(\boldsymbol{z}_t) - \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t} \Big)$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}} x_{ij}(t) \leq 1, \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.20}$$

$$y_{ik}(t) \leq 1 - \sum_{j\in\mathcal{J}} x_{ij}(t), \ \forall i \in \mathcal{I}, k \in \mathcal{J}, t \in \mathcal{T} \tag{4.21}$$

$$\sum_{j\in\mathcal{J}} y_{ij}(t) \leq b_i, \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.22}$$

$$\sum_{j\in\mathcal{J}} x_{ij}(t) + \sum_{j\in\mathcal{J}} y_{ij}(t) \geq 1, \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.23}$$

$$\sum_{i\in\mathcal{I}} x_{ij}(t) + \sum_{i\in\mathcal{I}} y_{ij}(t) \leq C_j, \ \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{4.24}$$

$$\boldsymbol{z}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t) \in \{0,1\}^{I\cdot J} \times \{0,1\}^{I\cdot J}, \ \forall t \in \mathcal{T}. \tag{4.25}$$

Eq. (4.20) allows a UE $i \in \mathcal{I}$ to be THO-enabled (i.e., explicitly assigned to a cell if $x_{ij}(t) = 1$), or CHO-enabled (i.e., implicitly assigned through CHOs to a cell, otherwise). Eq. (4.21) complements the previous one, ensuring that a UE can be explicitly or implicitly assigned to a cell, but not both; eq. (4.22) prevents each CHO-enabled UE from having more than b_i prepared cells, similarly to Section 4.4. Eq. (4.23) ensures that a UE will be explicitly or implicitly assigned, preventing blocking completely some UEs, and eq. (4.24) enforces each cell's capacity constraints using the total number of users. Lastly, eq. (4.25) defines the binary decisions.

Performance Analysis. The analysis and proofs follow nearly verbatim those of Section 4.4, with the modification of the sampling technique $Q_{\mathcal{Z}}$. We highlight that the analysis of Section 4.4 relies on fixed sets $\mathcal{I}_{\text{THO}}$ and $\mathcal{I}_{\text{CHO}}$, and therefore, the unbiased estimator defined earlier cannot be used in the case of dynamic HO type selection, where $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are coupled with the controller deciding the HO type of each UE in every slot. For that reason, we create a new unbiased estimator.

Let $\boldsymbol{z}_t^{\text{m}} = \left(\boldsymbol{x}_t^{\text{m}}, \boldsymbol{y}_t^{\text{m}}\right)$ denote the *continuous* meta-learner output at slot t, where

$\boldsymbol{x}_t^{\mathrm{m}} \in [0,1]^{I \cdot J}$ and $\boldsymbol{y}_t^{\mathrm{m}} \in [0,1]^{I \cdot J}$. The definition of $\boldsymbol{z}_t^{\mathrm{m}}$ resembles eq. (4.12), with the distinction that in that earlier formulation, the users were partitioned into two disjoint sets $\mathcal{I}_{\mathrm{THO}}$ and $\mathcal{I}_{\mathrm{CHO}}$.

For the implementable binary decision $\boldsymbol{z}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$, we start by defining for each $i \in \mathcal{I}$ the probability to be explicitly assigned (THO-enabled) at slot t:

$$\pi_i(t) \triangleq \sum_{j \in \mathcal{J}} x_{ij}^{\mathrm{m}}(t),$$

where eq. (4.20) ensures that $\pi_i(t) \in [0,1]$. To obtain an unbiased estimator, i.e., $\mathbb{E}[\boldsymbol{z}_t] = \boldsymbol{z}_t^{\mathrm{m}}$, we sample:

$$z_i(t) \sim \mathrm{Bernoulli}\big(\pi_i(t)\big),$$

and if $z_i(t) = 1$, the UE $i \in \mathcal{I}$ is THO-enabled at slot t, else CHO-enabled. Lastly, the associations/preparations are decided with the normalized quantities: $x_{ij}(t) = x_{ij}^{\mathrm{m}}(t)/\pi_i(t)$ and $y_{ij}(t) = y_{ij}^{\mathrm{m}}(t)/(1-\pi_i(t))$.

From the law of total expectation, and as $x_{ij}(t) = 0$ when $z_i(t) = 0$, we get $\mathbb{E}[x_{ij}(t)] = x_{ij}^{\mathrm{m}}(t)$ and similarly $\mathbb{E}[y_{ij}(t)] = y_{ij}^{\mathrm{m}}(t)$. This concludes that $\mathbb{E}[\boldsymbol{z}_t] = \boldsymbol{z}_t^{\mathrm{m}}$.

4.7. Performance Evaluation

We compare `CONTRA` with various *(i)* 3GPP-compliant, threshold-based THO and CHO algorithms, as well as *(ii)* RL benchmarks. The former are the algorithms currently used by MNOs and antenna vendors [39], [81], solving either solely the THO or CHO problem (i.e., not jointly). Given that these benchmarks treat THO and CHO as independent procedures, we adapt and extend *three* existing foundational *RL* algorithms from the literature that were originally designed for other tasks [36], [151], [152], such as only THO optimization. In this way, it becomes possible to compare the proposed algorithm with more advanced (i.e., not relying on pre-defined thresholds) benchmarks for precisely the problem at hand, namely, the joint optimization of traditional and conditional handovers.

The algorithms are evaluated in two main scenarios: *(i) volatile* SINR scenario, in which $s_{ij}(t)$ fluctuates every 10 slots within the range of [0, 30]dB [122], reflecting a wide spectrum of signal conditions (i.e., from poor to excellent values), and *(ii)* real SINR scenario from crowdsourced countrywide measurements. For the latter, we focus on an area of approximately 160 sq. kms, where 73,303 signal measurements in a second granularity are taken for 100 cells (20 cell sites). These cell sites are shown in Figure 4.10, marked in red, and the measurement areas are indicated in green. This dataset presents a highly competitive scenario as SINRs can vary arbitrarily and occasionally exhibit adversarial behavior (see Section 4.3). Given that these measurements are anonymized, we consider 100 fast-moving UEs, with random velocities ranging from 20 to 28 m/s and variances in [0, 5], placed at $t = 1$ in a random location with measurement (green). We adopt the Gauss-Markov mobility model [124], in line with prior works [33], with 0.9 randomness parameter to find the location of each UE in each subsequent slot $t = 2, ..., T$, for $T = 1$k.

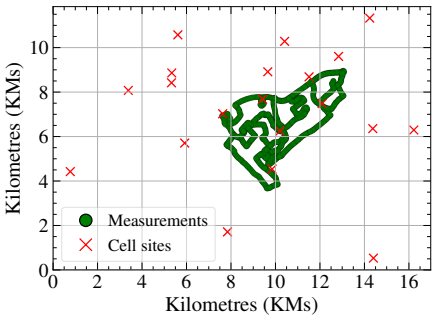
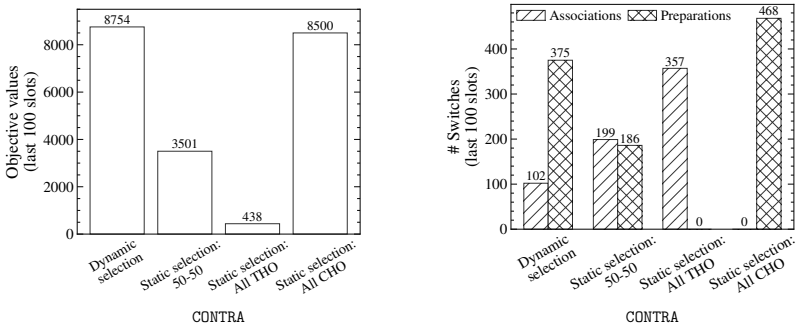


Figure 4.10: Cell sites (red) and SINR measurement locations (green) used for the evaluation of our algorithms and the benchmarks.



(a) (b)

Figure 4.11: Real SINR scenario with $I = 100$ UEs. Over the last 100 slots of the $T = 1$k simulation, (a) shows the objective function values (i.e., throughput and switching costs), and (b) the number of switches in associations (i.e., $\boldsymbol{x}_t$) and preparations (i.e., $\boldsymbol{y}_t$), for the dynamic and three variants of static HO type selection of `CONTRA`.

To incorporate the signal measurements, we map each new location to the nearest location with available measurements.

Understanding the Proposed Algorithm. First and foremost, it is imperative to showcase the differences between the static (Section 4.4) and dynamic (Section 4.6) HO type selection. For that reason, we compare four variants of `CONTRA`: *(i)* dynamic HO type selection, where the controller decides if and when each user should operate in THO or CHO mode according to our framework, and three variants of static HO type selection, namely, *(i)* a random split with half of the users predefined as THO-enabled and the other half as CHO-enabled, as well as a configuration in which all users are *(ii)* THO-enabled, and *(iii)* CHO-enabled.

In detail, Figure 4.11 compares the performance of these four `CONTRA` variants over the last 100 slots of the real-SINR scenario. When all users are in THO mode, the objective function values are the lowest (i.e., 438), primarily due to the large number (357) and high cost of THOs. The 50–50 split between THO- and CHO-enabled users improves performance by nearly $\times 8$ compared to the all-THO config-

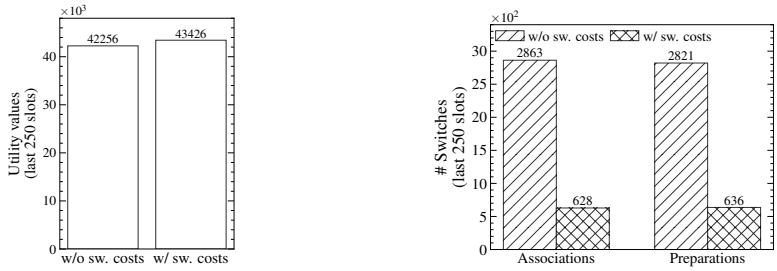


Figure 4.12: Real SINR scenario with $I = 100$ UEs. Over the last 250 slots of the $T = 1$k simulation, (a) shows the utility function values (i.e., $\tilde{g}_t$), and (b) the number of switches in associations (i.e., $\boldsymbol{x}_t$) and preparations (i.e., $\boldsymbol{y}_t$), for ignoring and taking into account the switching costs.

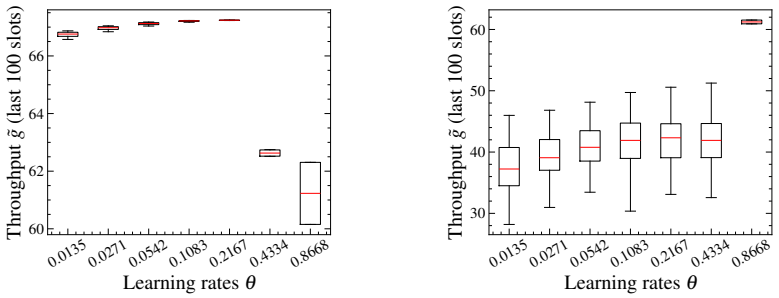
4

uration; yet it remains roughly $\times 2.5$ lower than both the automatic mode-selection and all-CHO cases. The automatic mode achieves higher performance ($\approx 3\%$) compared to the all-CHO configuration, as it executes fewer CHO preparations, even though it requires 102 THOs. From here onward, `CONTRA` is assessed in the dynamic HO formulation.

Secondly, we focus on the trade-off between throughput and the two switching cost terms to understand the importance of the latter for the studied problem. Figure 4.12 shows the utility function and the number of association and preparation changes when switching costs are *(i)* excluded, and *(ii)* taken into account, for the real-SINR scenario. Focusing on the last 250 slots of $T = 1$k, we note that taking into account the switching cost reduces the number of associations (and thus, THOs) and preparation changes by approximately 77%, while achieving a 2.7% higher throughput. Hence, solely maximizing the utility/throughput (i.e., without considering the switches) may not be the optimal strategy, given that comparable, or even better, performance can be achieved with fewer switches, thereby reducing signaling overhead, resource utilization, and HO delays.

Thirdly, to show the importance of combining learners with different rates, Figure 4.13 shows the attained throughput $\tilde{g}(.)$ in a case where the SINRs are *stationary* (changing only every 200 slots) and real (changing significantly between slots). Given that $T = 1$k, eq. (4.17) determines that 7 experts will be used, with learning rates 0.0135, 0.0271, 0.0542, 0.1083, 0.2167, 0.4334, and 0.8668 (combining eq. (4.18) and Lemma 4.2). In the stationary case, the OGA experts with small learning rates obtain up to 12% better throughput than the highest-learning-rate one. On the other hand, in the real case, the highest-learning-rate expert achieves 65% more throughput. These findings support the claims that in more stationary scenarios, smaller-learning-rate experts perform better as they update their decisions more carefully, while in volatile/real cases, it is the opposite, as higher-learning-rate experts adapt more quickly to the rapid changes of the environment.

Regret Analysis. Section 4.5 shows that the regret guarantees hold for any benchmark; however, computing the benchmark that has full knowledge of the future is computationally intensive for mixed-integer programs [36], [153]. To facilitate,



(a) (b)

Figure 4.13: Cumulative throughput $\tilde{g}$ for the last 100 of 1k slots for different learners in (a) stationary and (b) real SINR case.

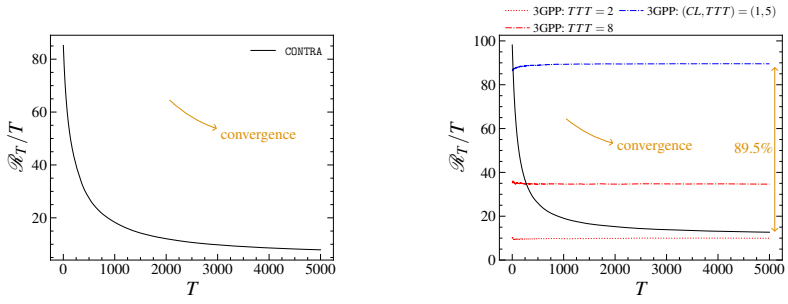
4

therefore, the average dynamic regret calculation, the optimal `Oracle` considered solves the optimization problem *in every step* using *CVXPY* [154], and a smaller number of $I = 20$ users. Apart from the `CONTRA`, we show the average dynamic regret of 3GPP threshold-based benchmarks. In the sequel, we refer to the 3GPP CHO benchmarks as (CL, TTT), where the first element denotes the number of cells that are prepared in each slot, provided they are the highest-SINR cells for more than TTT consecutive slots (similar to the A3 event); while for the 3GPP THO benchmarks, only TTT is used. For instance, the benchmark $(1, 2)$ prepares one cell for each user if it remains the highest-SINR for more than two slots.

Figure 4.14 shows the average dynamic regret of our proposed algorithm and the 3GPP-compliant competitors in the volatile case. In Figures 4.14a and 4.14b, we show the continuous, $\boldsymbol{z} \in \mathcal{Z}^{\mathrm{c}}$, and discrete, $\boldsymbol{z} \in \mathcal{Z}$, decisions, respectively. In both cases, we observe that the average dynamic regret converges towards zero for $T = 5$k slots; and in the latter case, where the decisions are actually implementable in practice, `CONTRA` outperforms the best-performing CHO-only 3GPP benchmark by 89.5%. The average dynamic regret of the THO-only benchmarks stays almost constant for all slots ("stuck" in sub-optimal decisions). Lastly, we observe a relatively small discretization error of 16.2% measured between the continuous and discrete decision plots at $t = 5$k.

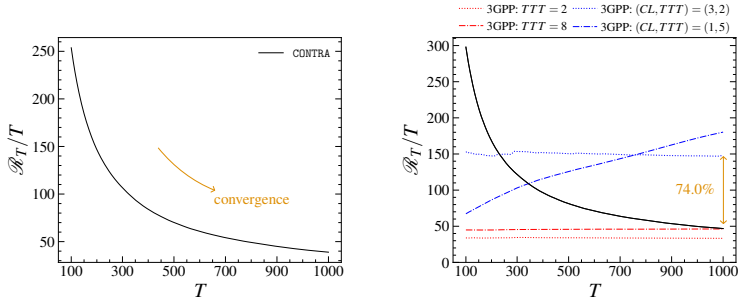
From Figure 4.15, the average dynamic regret converges again towards zero for the real SINR cases, showing that `CONTRA` is adaptable in all scenarios, with the gap between the continuous and discrete decisions being solely 1.83%. On the other hand, the best CHO-only (THO-only) 3GPP-compliant benchmark attains 74% (28.2%) higher (lower) dynamic regret at $t = 5$k. Yet, the THO-only benchmark is stuck in suboptimal (non-diminishing regret) decisions.

Comparison with Reinforcement Learning. To compare `CONTRA` with RL approaches, we need to adjust the definitions of *RL state* and *RL action*. We define the *RL state* with the SINRs of the previous time slots, see e.g., [36], [152]. We intentionally avoid incorporating other UE features, such as velocity or ping-pong counters, to remain as aligned as possible with our problem formulation. We also define an *RL action* that determines whether each user $i \in \mathcal{I}$ is THO- or CHO-enabled and how many cells to prepare in the latter case. In principle, the CHO decisiona should be modeled as the selection of any subset of candidate cells. This formula-



(a) (b)

Figure 4.14: Volatile scenario (SINR changes every 10 slots), for $T = 5$k slots, for the (a) continuous and (b) discrete decisions.



(a) (b)

Figure 4.15: Real SINR scenario from crowdsourced data, for $T = 1$k (skipping first 100 slots for visibility), for the (a) continuous and (b) discrete decisions.

tion captures the "true" decision space of CHO, but in the case of RL, it leads to an exponential growth in the number of possible actions; specifically, $J + (2^J - 1)$ per user. The first term, J, corresponds to actions regarding THOs: a user can be assigned to any of the available J cells. The second term (exponential), $2^J - 1$, corresponds to any non-empty subset of the J cells that can get prepared (minus the empty set).

Even when viewed purely from the search-space perspective, this exponential growth makes RL infeasible for moderate J, as exploration becomes prohibitively expensive. In contrast, `CONTRA` maintains a tractable search-space, avoiding the combinatorial explosion inherent to this RL formulation. To avoid the exponential explosion, we adopt a more compact parameterization for the RL approaches with only $2J$ actions per user. Here, the first J terms still correspond to THOs, while the rest J to CHOs, where the user prepares up to the top-J strongest cells in terms of SINR in the previous slots. The reward/objective remains as in `CONTRA` (see Section 4.4), namely, throughput minus switching costs.

The three RL benchmarks are: *(i)* policy-gradient methods without a critic, through `REINFORCE` algorithm [151], *(ii)* value-based methods through Deep Q-Network (`DQN`) [36], and *(iii)* actor-critic methods with proximal updates through

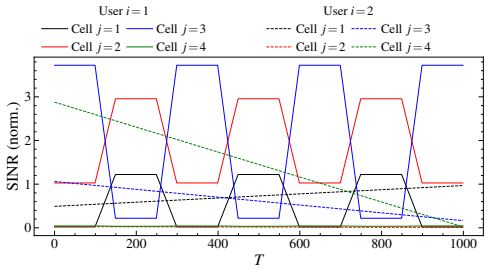


Figure 4.16: Simple deep-dive example for $I = 2$ UEs, $J = 4$ cells, and $T = 1$k slots, with two SINR modes: one with more dynamic/abrupt changes (continuous lines) and one with more gradual changes (dotted lines).

4

Proximal Policy Optimization (`PPO`) [152].

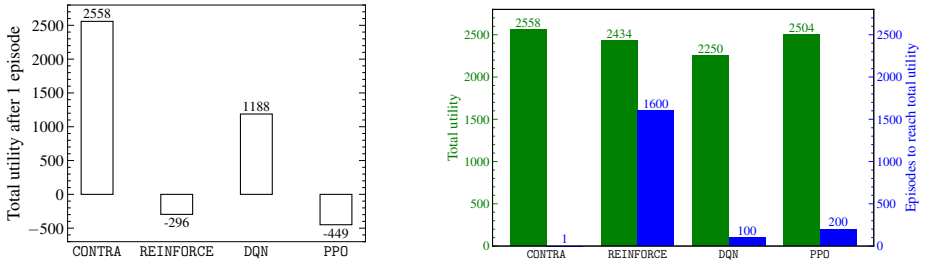
`REINFORCE` is a simpler method, that tries actions, observes total rewards, and adjusts the policy toward those that performed well. It does not estimate how good a state is (no critic), which makes it simple but noisy and data-hungry. More precisely, it implements a Monte Carlo policy-gradient method without a critic. Its policy network is a two-layer multilayer perceptron (MLP) with 128 neurons per layer and Tanh activations, followed by a softmax output layer producing a categorical distribution over all possible actions. After each episode, the algorithm computes discounted returns and updates the policy parameters using gradients of the log-probabilities weighted by returns. The optimizer is Adam with a learning rate of 3×10^{-4}.

`DQN` does not directly learn a policy; instead, it learns an action–value function $Q(state, action)$ that shows how good each action is. The agent then picks the action with the highest Q, allowing though a random exploration. More specifically, `DQN` consists of two hidden layers with 128 neurons and ReLU activations. Exploration follows an ε-greedy strategy with $\varepsilon_{\text{start}} = 1.0$, $\varepsilon_{\text{end}} = 0.05$, and exponential decay constant 200. A replay buffer of size 50k and minibatch size 64 are used, with soft target-network updates ($\tau = 0.01$) to improve stability. The loss is the mean-squared temporal-difference error, optimized with Adam at a learning rate of 10^{-3}.

`PPO` combines both ideas. It learns an *actor* (policy) and a *critic* (value estimate), and ensures stable updates by clipping how much the policy can change each step. Both the actor and critic share a two-layer MLP backbone (128 neurons per layer, Tanh activations). The actor outputs a categorical distribution over all actions, while the critic outputs a scalar state-value estimate. Training uses generalized advantage estimation with $\lambda = 0.95$ and the clipped objective parameter $\epsilon = 0.2$; also, minibatches of 64 samples for 3 epochs per cycle are used. Both networks are optimized using Adam with rate 3×10^{-4}.

Unlike `CONTRA`, which learns and adapts online with a single exposure to the environment, these RL methods are trained over hundreds of simulated episodes, effectively granting them repeated access to the same system dynamics. This allows RL algorithms to asymptotically approximate the optimal policy under repeated trials, whereas `CONTRA` must adapt on-the-fly.

To showcase the disadvantages of RL even in a simple setting with four cells,



(a) (b)

Figure 4.17: Total utility for (a) one episode and (b) multiple episodes, of `CONTRA` vs three RL benchmarks for a simple deep-dive example with $I = 2$ UEs, $J = 4$ cells, and $T = 1$k slots, with the SINRs of Figure 4.16.

we consider only two users with different types of mobility/SINR dynamics as can be seen in Figure 4.16: User 1 frequently switching between good and bad cells (high variability), and User 2 steadily moving toward a better SINR environment (gradual change). More precisely, the SINR of User 1 remains constant for 100 slots, then changes during a 50-slot transition period, and subsequently stabilizes again for another 100 slots, and so on.

In Figure 4.17(a), we run each algorithm once (i.e., one episode), since our proposed method requires only a single pass to produce its decisions and learn. As expected, RL algorithms, and specifically `REINFORCE` and `PPO`, perform poorly (negative utility) as they require many interactions over the same conditions to learn. On the other hand, `DQN` can improve its decision in the same episode, achieving, however, 53.6% lower utility than `CONTRA`. In Figure 4.17(b), we allow the RL benchmarks to learn this simple scenario by running them for multiple episodes and reporting their final utilities. Compared to our algorithm that requires a single episode/pass to reach $\approx$ 2.5k total utility, `REINFORCE`, `DQN`, and `PPO` reach until 95.2%, 88.0%, and 97.9% of its performance, by running for 1.6k, 100, and 200 episodes, respectively.

Finally, in the real SINR scenario, the performance gap becomes even more pronounced. From Figure 4.18, and compared to our algorithm that requires a single pass to reach $\approx$175k total utility, `REINFORCE`, `DQN`, and `PPO` reach until 54.9%, 52.6%, and 76.6% of its performance, by running for 2k, 500, and 500 episodes, respectively. The gap thus widens significantly in realistic network conditions, showing that `CONTRA` attains substantially higher performance without the extensive training required by RL-based approaches. Even with this simplified assumption, our approach outperforms the RL in the experiments executed with real SINR/UEs. This poor performance of RL approaches is expected, as they do not adapt to volatile or adversarial environments (known to converge under stationarity only), and do not offer performance guarantees as those we provide in Lemma 4.3.

4.8. Related Work

Measurements & Traditional Handovers. HOs are mainly studied with traces from UEs, which are inevitably limited to certain manufacturers, areas and devices

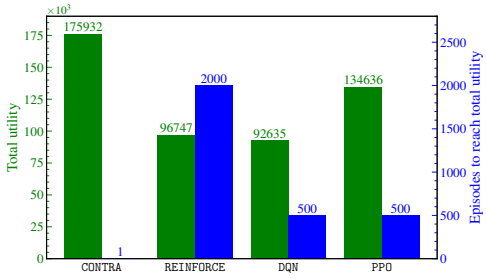


Figure 4.18: Total utilities (i.e., $\tilde{g}_t$) and number of episodes required to reach these values, for the real SINR scenario with $I = 100$ UEs and $T = 1$k slots.

4

[30]. Recent large-scale, network-side studies [113] provide more visibility, but do not propose models or solutions. Prior works have addressed the joint optimization of throughput and HOs [33], [34], [36], [109] [152], offering important insights into association strategies under delay constraints. Our objective aligns with these studies, and we extend their models to capture the delay characteristics and costs associated with THOs and CHOs, and heterogeneity across cells and UEs, by analyzing countrywide datasets. Separately, recent work has addressed HOs under minimal assumptions via online learning [146], but without considering CHOs. Our approach builds on these works by incorporating, for the first time, THO and CHO characteristics into a unified, adaptive learning framework.

Conditional Handovers. CHOs have been proposed as a solution for non-terrestrial networks [42], [155], [156], 5G NR-unlicensed systems [44], fast-moving users [44], as well as in beamforming and contention-free random access [41], [157]. Proposals for improving the CHO mechanism include [46], [158] which tweak the CHO thresholds to decrease HOFs; [159], [160] which use historical handover data to decide the cell preparations; and [161] which employs UE trajectory prediction to optimize the CHO decisions. Machine Learning-assisted approaches include [162], [163] which predict ideal association strategies based on SINR data or predict the SINRs values. These important works, however, propose heuristic solutions (no optimality guarantees) and/or rely on historical data for offline training of models, which in practice may be unavailable or non-representative of encountered conditions.

In contrast, motivated by our measurements that show this problem is dynamic across UEs, cells, and time, we leverage an adaptive optimization framework. Unlike other works that explicitly avoid decision changes in cell preparations [45], [141] or propose static optimization formulations [43] which, unavoidably, rely on heavy assumptions (static and known parameters), we propose a tunable approach to the network's preference switching model that can cope with time-varying and unknown cost values. In [164], a meta-learning approach with minimal assumptions is used but focuses only on CHOs and omits a fair scheduler for allocation; see eq. (4.3).

Meta-Learning. Meta-learning is finding increasing applications in communication systems due to its robustness to distribution shifts and fast-adaptation [20]. We refer to [165] for merging proposed actions for management and orchestration (MANO) operations; [166] for beamforming adaptation and MIMO systems; [167] for user-level traffic prediction over a short time horizon; [168] for unmanned aerial

vehicle (UAV) networks; [169] for IoT devices learning together; [170] for load balancing; and [171] for handovers in vehicle-to-network communications. Here, we utilize the dynamic version of this tool, combined with online learners, so that optimal CHO and THO decisions can be learned on-the-fly, achieving rigorous theoretical and practical performance guarantees.

4.9. Conclusion

This chapter is motivated by the demanding mobility support requirements in 6G, technological advances in handover (HO) mechanisms, and the availability of AI-based management solutions in O-RAN. Its goal is to address a challenge that is critical as 5G deployments mature and 6G systems emerge: executing HOs rapidly and reliably, especially in dense deployments and high-frequency bands, where (*i*) traditional HO (THO) mechanisms exhibit high failure rates and thus, increased delays, and (*ii*) the newly introduced by 3GPP, Conditional HOs (CHOs), which tackle these issues by enabling proactive cell reservations, raise intricate trade-offs in signaling and resource utilization. To better understand these issues, we analyze countrywide datasets from a major operator in Europe; and the findings underscore the need for more adaptive and robust HO control. Motivated by this, we propose `CONTRA`, a meta-learning, provably-optimal, minimal-assumption, O-RAN-compatible algorithm that jointly optimizes THO and CHO and adapts to runtime observations, demonstrating significant improvements in real-world and synthetic scenarios. While our work focuses on O-RAN, the core ideas can be applied to other 3GPP-compliant RAN architectures that support real-time measurements and centralized decision making.

Having thoroughly addressed mobility management by optimizing both traditional and conditional handover strategies across diverse users, cells, and system conditions, ensuring that users are connected to the "right" cells/base stations, we now turn to the other side of the link to complete the end-to-end control vision of next-generation O-RAN systems: how these (virtualized) base stations (vBSs) adapt their available resources to serve connected users efficiently.

5

Resource Allocation for Virtualized Base Stations in Non-Real-Time

Once traditional and conditional handovers are optimized, attention must shift to the other side of the link: how virtualized base stations (vBSs) utilize their resources to serve users efficiently. While proportional-fair or round-robin schedulers, used in the previous chapters, remain a common baseline, they are not always sufficient to meet the dynamic performance and energy requirements of modern networks.

Despite offering numerous advantages, including increased flexibility and reduced costs, vBSs introduce new challenges due to their volatile operation and the dynamic network conditions they are called to support. In this chapter, we leverage the O-RAN multi-tier control architecture to propose a class of robust and effective non-real-time policies for vBS resource allocation. First, we introduce an online learning algorithm that is suitable even for non-stationary or adversarial environments. However, this robustness often comes at the cost of conservatism when the environment is static/stationary or predictable. To address this, we develop a meta-learning framework that ensures effectiveness across a wide range of conditions by dynamically selecting among a pool of learners, including specialized ones designed for static/stationary or adversarial environments (as our first algorithm). We establish strong (sub-linear) regret guarantees and demonstrate, through extensive experiments on real-world data collected from our testbed, that our proposed approach achieves up to 64.5% energy savings compared to state-of-the-art baselines.

The content of this chapter has been published in:

M. Kalntis, G. Iosifidis, and F. A. Kuipers, "Adaptive Resource Allocation for Virtualized Base Stations in O-RAN with Online Learning," in *IEEE Trans. on Communications (TCOM)*, vol. 73, no. 3, pp. 1787–1800, 2025.

M. Kalntis and G. Iosifidis, "Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2022.

5.1. Challenges and Contributions

This chapter focuses on O-RAN policies that determine thresholds (upper bounds) for key vBS operation knobs, namely for the vBS transmission power, the eligible MCS, and the Physical Resource Blocks (PRB), in the Uplink (UL) and Downlink (DL). Each policy is updated at a non-RT scale, based on observations of past performance, cost, and context (including conditions and demands), and is subsequently fed to real-time schedulers that assign the vBS radio resources. The question this chapter addresses is the following: *how to design robust vBS non-RT policies that offer performance/cost guarantees without relying on strong assumptions and avoid sub-optimal operation points?*

Our *first contribution* is the design and evaluation of a robust *adversarial* bandit algorithm, cf. [112], which: *(i)* identifies effective policies without relying on assumptions about the environment; *(ii)* offers tight performance guarantees; *(iii)* is oblivious to the (unknown and possibly time-varying) vBS performance; and *(iv)* has minimal and constant (in observations and time) memory requirements, as it uses closed-form expressions that can be calculated even in real-time and in resource-constrained platforms. The performance is quantified using a combined metric of effective throughput modulated by the traffic demands, and energy consumption, where the latter can be prioritized via a weight parameter. It is important to note that no assumption (e.g., convexity) is made on the performance function (i.e., we follow a black-box approach). For the optimality criterion, we use *regret*, where we compare the time-aggregated performance of the algorithm with that of a hypothetical benchmark that is designed with the help of an oracle providing access to all future/necessary information.

The *second contribution* is the expansion of this learning algorithm with a *meta-learning* scheme, which boosts the performance whenever possible. Namely, the robustness of the algorithm described above means it might be conservative when the environment is *easy*, e.g., when the network has access to context information, or if the channel qualities and traffic demands are stationary or exhibit periodicity [17]. For these cases, data-efficient solutions such as [60] can leverage the available information to identify optimal policies faster. Hence, the question that arises naturally is how to combine the required robustness without compromising learning performance (in terms of convergence speed) whenever the environment is easy. To address this, we introduce a *meta-learner* that selects intelligently among policies proposed by different algorithms that rely on, and perform better under, different assumptions. A key challenge is that learning occurs on two levels: the meta-learner must learn which algorithm is the best-performing, and each algorithm must learn which policy is the best-performing, while receiving partial (i.e., bandit) feedback on both levels. Our approach addresses this challenge through a framework that guarantees the network performs as well as the best-performing algorithm.

In summary, the main technical contributions of this chapter are the following:

• We study the vBS resource allocation problem in its most general form, i.e., in non-stationary/adversarial environment and without knowledge of vBS throughput/cost functions. Our proposed scheme achieves sub-linear regret and has minimal computation and memory overhead [112]. This is the first work applying *adversarial*

bandits to vBS resource allocation.
• We devise a meta-learning strategy that entails the use of algorithms tailored to different environments and obtains sub-linear regret with respect to the best algorithm, in each case.
• We use real-world traffic traces and testbed measurements to demonstrate the weaknesses of prior works [60], as well as the efficacy of the proposed learning algorithm in a battery of representative scenarios. Upon publication of this article, we will release all the source code to foster further research on this important topic.

5.2. System Model and Problem Statement

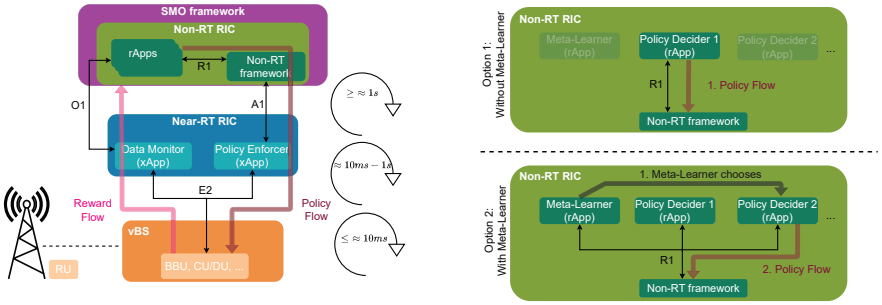
O-RAN Placement. Our model follows the O-RAN architecture, and the proposed algorithms can be implemented as rApps at the Non-RT RIC, aiming to learn energy-efficient threshold radio policies [10]–[12]. These policies are essentially *threshold rules* regarding the maximum MCS, PRB, and transmission power that each vBS, in real-time, is allowed to use. Specifically, these rules are communicated to the vBSs, guiding their RRM schedulers to allocate radio resources in real-time accordingly, as shown in Figure 1.4. This approach aligns with a recent stream of papers [12], [60]–[62] proposing threshold policies and exploits the multi-tier (multi-timescale) architecture of O-RAN to offer centralized control of multiple vBS, without intervening in their (often proprietary/hardcoded) real-time schedulers.

This tiered control approach can be seen in Policy Flow, Figure 5.1a and 5.1b top. At each round, with typical duration$\sim$1s, the *Policy Decider* (i.e., algorithm) devises the threshold policy which is communicated (via the A1 interface) to the Near-Real-Time (Near-RT) RIC, where an xApp (termed *Policy Enforcer*) forwards it to the different vBSs. This makes a two-timescale system where the policy is devised at each round (s) and the vBSs schedulers update their typical RRM decisions every slot (ms), based on these rules.

O-RAN's flexibility enables the usage of O1 to receive/forward the policy directly from/to the real-time scheduler [172]. Nevertheless, our decision to involve xApps through the Near-RT RIC stems from providing a more general framework, where, e.g., another xApp could take the thresholds we provide, save them to a database, and perform additional actions to ensure that those thresholds are respected or make any other inference. Our modular architecture is designed to be adaptable to accommodate this, and is in accordance with recent works [12], [60].

The Policy Flow changes when including a meta-learner as another rApp (Figure 5.1b bottom), whose goal is to discern the best Policy Decider among the employed ones. This is achieved by selecting at each round one of the available Policy Deciders, which, in turn, chooses the threshold policy. At the end of each round, the Near-RT RIC's Data Monitor computes a *reward* by aggregating the performance and cost measurements (for all slots) received via the E2 and feeds them to the selected Policy Decider via the O1 interface (Reward Flow in Figure 5.1a). The terms Policy Decider, Policy Enforcer, and Data Monitor are introduced in this work to clarify the role of each rApp/xApp, as these last terms are generic.

vBS Policies. We optimize the system operation over a time horizon of $t = 1, \ldots, T$



(a) (b)

Figure 5.1: O-RAN-compliant architecture & policy workflow. **(a)** The key building block is the Non-RT RIC, hosted by the Service Management and Orchestration (SMO) framework, and the Near-RT RIC. The system has three control loops: (*i*) Non-RT, which involves large-timescale operations with execution time $\geq$1 s, (*ii*) Near-RT (>10 ms), and (*iii*) RT ($\leq$10 ms). **(b)** Policy Flow for the Non-RT RIC with (bottom) and without (top) an rApp implementing a meta-learner.

5

rounds. For the DL, we define the set of the maximum allowed *vBS transmission powers*, $\mathcal{P}^{\mathrm{d}} = \{p_i^{\mathrm{d}}, \forall i \in \{1,\dots,P^{\mathrm{d}}\}\}$, the set of highest eligible *MCS*, $\mathcal{M}^{\mathrm{d}} = \{m_i^{\mathrm{d}}, \forall i \in \{1,\dots,M^{\mathrm{d}}\}\}$, and the set of maximum *PRB ratio*, $\mathcal{B}^{\mathrm{d}} = \{b_i^{\mathrm{d}}, \forall i \in \{1,\dots,B^{\mathrm{d}}\}\}$, where P^{d}, M^{d}, and B^{d} denote the number of possible transmission power, MCS, and PRB ratio levels in DL, respectively.[1] The PRB ratio corresponds to the portion of the available PRBs the channel supplies, e.g., $b_i^{\mathrm{d}} = 0.2$ leads to utilization of 20 % (10 out of 50 PRBs). The DL policy for round t is denoted with $x_t^{\mathrm{d}} \in \mathcal{P}^{\mathrm{d}} \times \mathcal{M}^{\mathrm{d}} \times \mathcal{B}^{\mathrm{d}}$. Similarly, for the UL we introduce the sets $\mathcal{M}^{\mathrm{u}} = \{m_i^{\mathrm{u}}, \forall i \in \{1,\dots,M^{\mathrm{u}}\}\}$ and $\mathcal{B}^{\mathrm{u}} = \{b_i^{u}, \forall i \in \{1,\dots,B^{\mathrm{u}}\}\}$, where M^{u}, B^{u} are the available MCS and PRB ratio levels in UL[2] and denote with $x_t^{\mathrm{u}} \in \mathcal{M}^{\mathrm{u}}$ the UL policy. Putting these together, the t-round threshold policy is:

$$x_t = (x_t^{\mathrm{d}}, x_t^{\mathrm{u}}) \in \mathcal{X}, \text{ where } \mathcal{X} = \mathcal{P}^{\mathrm{d}} \times \mathcal{M}^{\mathrm{d}} \times \mathcal{B}^{\mathrm{d}} \times \mathcal{M}^{\mathrm{u}} \times \mathcal{B}^{\mathrm{u}}.$$

Rewards & Costs. The first goal of the learner is to maximize the *effective* DL and UL throughputs, which depend on the aggregate transmitted data and the backlog in each direction. In line with prior works (see [60] and references therein), we use the *utility* function:

$$U_t(x_t) = \log\left(1 + \frac{R_t^{\mathrm{d}}(x_t^{\mathrm{d}})}{d_t^{\mathrm{d}}}\right) + \log\left(1 + \frac{R_t^{\mathrm{u}}(x_t^{\mathrm{u}})}{d_t^{\mathrm{u}}}\right), \tag{5.1}$$

where $d_t^{\mathrm{d}}, d_t^{\mathrm{u}} > 0$, with $U_t(x_t) = 0$ otherwise. $R_t^{\mathrm{d}}(\cdot)$ and $R_t^{\mathrm{u}}(\cdot)$ denote the DL and UL *transmitted data* during round t; and d_t^{d} and d_t^{u} are the respective backlogs, i.e.,

[1] The MCS values are predetermined, and similarly, one can quantize the power and PRB ratio values; see, e.g. [55].

[2] A maximum allowed UE transmission power is not defined since the users' transmission power has less impact on the vBS power than the MCS and PRBs in the UL. However, it can be included in x_t^{u} if deemed relevant for another application.

the traffic *demands* during t. The logarithmic transformation balances the system utility across each stream (i.e., DL and UL), but we note that other mappings (e.g., linear) might be used to capture the specifics of different applications. We have divided the transmitted data by the actual traffic demands in the respective stream (UL or DL), since the reward should naturally be defined w.r.t. the needs of the system. Similarly, one can readily extend the utility function to capture various QoS metrics, e.g., by measuring only the throughput above a certain threshold. We refrain from making assumptions about how $x_t^{\mathrm{u}}, x_t^{\mathrm{d}}$ affect the transmitted data, $R_t^{\mathrm{d}}, R_t^{\mathrm{u}}$; similarly, the traffic demands, $d_t^{\mathrm{d}}, d_t^{\mathrm{u}}$, are also considered unknown and can vary arbitrarily.[3] In this black-box approach, each threshold policy x_t (i.e., *bandit arm*) yields a reward, which we calculate a posteriori, and corresponds to the reward of the respective bandit arm. The goal of our algorithms is to learn progressively which bandit arm leads to the highest possible reward.

The second goal of the learner is to minimize the vBS energy costs. To that end, we introduce the time-varying *power cost* function $P_t(x_t)$, which depends on policy x_t in a possibly unknown fashion. Our decision to refrain from making assumptions about this function is rooted in the complexities involved in characterizing the power consumption and costs of such virtualized base stations [59]. Also, this black-box approach allows us to capture a range of factors that might affect the consumed energy (e.g., retransmissions due to interference or time-varying electricity prices).

Putting these together, the *learner*'s criterion is the *reward* function $\tilde{f}_t \colon X \to \mathbb{R}$ defined as:

$$\tilde{f}_t(x_t) = U_t(x_t) - \delta P_t(x_t), \tag{5.2}$$

where parameter $\delta > 0$ is set by the network operator to tune the relative priority of utility and energy costs. Parameter δ serves as a metric transformation, enabling a meaningful scalarization of U_t and P_t. Furthermore, we introduce, for technical reasons, the *scaled* reward function $f_t : \mathcal{X} \to [0, 1]$, since our learning algorithms (see Section 5.3 and 5.4) operate on that interval. An easy-to-implement mapping that ensures this normalization is:

$$f_t(x_t) = \big(\tilde{f}_t(x_t) - \tilde{f}_{min}\big)/\big(\tilde{f}_{max} - \tilde{f}_{min}\big). \tag{5.3}$$

Parameters $\tilde{f}_{min}$ and $\tilde{f}_{max}$ can be determined based on δ, the min/max value of power cost function, the min/max vBS transmission power, PRB ratio, MCS and traffic demands.

Environment. We refer to the "external" information, i.e., $\{c_t^{\mathrm{d}}, c_t^{\mathrm{u}}, d_t^{\mathrm{d}}, d_t^{\mathrm{u}}\}_{t=1}^{T}$ as *environment*, and it is responsible for shaping the function f_t. It is crucial to note that both reward components, U_t and P_t, vary with time t, an effect that is attributed to several factors. First, the traffic demands, i.e., d_t^{d} in DL and d_t^{u} in UL, change, sometimes drastically, in every round t, e.g., in small-cell networks where user churn is high, which affects U_t, see (5.1). The demands also impact the choice of MCS and PRB, leading to different processing times and, thus, different power costs. Second, the channel qualities (i.e., CQIs) in DL and UL, denoted as c_t^{d} and

[3] Kindly refer to Section 5.5 for details on their calculation during the evaluation of the proposed algorithms.

c_t^{u}, respectively, might vary (in slow, fast, or mixed timescales), and this affects the transmitted data R_t^{d} and R_t^{u} (hence, U_t changes even for fixed x_t) and the energy cost P_t (low CQI induces more BBU processing [59]).[4]

Importantly, we consider the environment to be *unknown* at the beginning of each scheduling round t. It is often challenging to predict the traffic demands, energy availability, channel qualities, wireless interference and other performance-related impairments that each vBS might encounter over the time window of several seconds that these threshold policies will be enforced. This, in turn, means that when we decide x_t in each round, we do not have access to f_t; and this is in notable contrast to the typical real-time radio management solutions that require accurate context information. Our model is hence oblivious to this information and this renders our solution applicable to a range of practical scenarios, such as those involving highly volatile environments and small cells where demands are non-stationary [173].

5.3. Policy Learning for Adversarial Environments

Objectives & Approach. The goal of our rApp (see Policy Decider, Figure 5.1) is to find a sequence of policies $\{x_t\}_{t=1}^T$ that induce rewards approaching the cumulative reward of the single best policy (*benchmark*). Formally, we employ the metric of *static expected regret*:

$$\mathcal{R}_T = \max_{x\in\mathcal{X}} \left\{ \sum_{t=1}^{T} f_t(x) \right\} - \mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t) \right], \tag{5.4}$$

where the first term is the aggregate performance of the benchmark (ideal) policy that can be selected only with knowledge of all future reward functions until T; and the second term measures the aggregate performance of the algorithm. The expectation in the second term is induced by any possible randomization in $\{f_t\}_{t=1}^T$ and in the selection of $\{x_t\}_{t=1}^T$ by the learner. Eventually, our objective is to devise a rule that decides the policies in such a way that the average regret, for any possible realization of rewards $\{f_t\}_{t=1}^T$, diminishes asymptotically to zero, i.e., $\lim_{T\to\infty} \mathcal{R}_T/T = 0$. Importantly, we wish to ensure this condition: *(i)* without knowing f_t when deciding x_t, and *(ii)* by observing only $f_t(x_t)$ when applying x_t, and not the complete function $f_t(x), \forall x \in \mathcal{X}$, as only *one* policy x_t in each round t can be deployed to the vBS.

The proposed scheme, named Bandit Scheduling for vBS (`BSvBS`), builds upon the *Exp3* algorithm [174], and its underlying idea is to learn the correct probability distribution y_t^{B} (B refers to `BSvBS`) from which we can sample x_t for each round t:

$$x_t \sim \mathbb{P}(x_t = x') = y_t^{\mathrm{B}}(x'), \forall x' \in \mathcal{X}.$$

The distributions $\{y_t^{\mathrm{B}}\}_{t=1}^T$ belong to the probability simplex:

$$\mathcal{Y}^{\mathrm{B}} = \left\{ y^{\mathrm{B}} \in [0,1]^{|\mathcal{X}|} \;\middle|\; \sum_{x\in\mathcal{X}} y^{\mathrm{B}}(x) = 1 \right\},$$

[4] The operation cost of the vBS hosting platform is subject to variations in external computing loads (e.g. when co-hosting other services or other vBS/DUs), changes in the monetary cost (or availability) of the energy price, and so on.

and are calculated in each round using the following explore / exploit rule:

$$y_t^{\mathrm{B}}(x) = \frac{\gamma}{|\mathcal{X}|} + (1-\gamma)\frac{w_t^{\mathrm{B}}(x)}{\sum_{x'\in\mathcal{X}} w_t^{\mathrm{B}}(x')}, \quad \forall\, x \in \mathcal{X}. \tag{5.5}$$

This formula includes three components: (i) the exploration part, $1/|\mathcal{X}|$ which selects a policy randomly, (ii) the exploitation part, $w_t^{\mathrm{B}}(x)/\sum_{x'\in\mathcal{X}} w_t^{\mathrm{B}}(x')$, which chooses a threshold policy based on its performance up until $t-1$, where the weight $w_t^{\mathrm{B}}(x)$ tracks the reward of each policy $x \in \mathcal{X}$, and (iii) parameter $\gamma \in (0, 1]$, which prioritizes the former (explore) or the latter part (exploit).

For the latter, we employ the weight vector $w_t = \big(w_t(x) : x = 1, \ldots, |\mathcal{X}|\big)$ that tracks the success of each tested policy, which is updated at the end of each round:

$$w_{t+1}^{\mathrm{B}}(x) = w_t^{\mathrm{B}}(x) \exp\left(\frac{\gamma \Phi_t^{\mathrm{B}}(x)}{|\mathcal{X}|}\right), \quad \forall\, x \in \mathcal{X}, \tag{5.6}$$

which assigns a probability exponentially proportional to the cumulative reward $\Phi_t^{\mathrm{B}}(x)$, that accounts for the selection of each policy, namely:

$$\Phi_t^{\mathrm{B}}(x) = \begin{cases} f_t(x_t)/y_t^{\mathrm{B}}(x_t), & \text{if } x = x_t, \\ 0, & \text{otherwise.} \end{cases} \tag{5.7}$$

By dividing each observed reward, $f_t(x_t)$ with the selection probability of the threshold-policy, $y_t^{\mathrm{B}}(x_t)$, we ensure the conditional expectation of $\Phi_t^{\mathrm{B}}(x)$ is the actual reward $f_t(x), \forall x \in \mathcal{X}$, meaning that Φ_t^{B} is an unbiased function estimator of the rewards [23]. Intuitively, this compensates the reward of thresholds that are unlikely to be chosen. The steps of the learning scheme are summarized in Algorithm 3, which takes as input γ and devises the ideal selection probability for each policy based on its expected reward.

The performance of Algorithm 3 is characterized in the following lemma, which holds for any possible sequence of functions $\{f_t\}_{t=1}^T$:

Lemma 5.1. *Let $T > 0$ be a fixed time horizon. Set input parameter $\gamma = \min\Big\{1, \sqrt{|\mathcal{X}| \ln |\mathcal{X}| / \big((e-1)T\big)}\Big\}$. Then, running Algorithm 1 ensures that the expected regret is:*

$$\mathcal{R}_T \leq 2\sqrt{(e-1)}\sqrt{T|\mathcal{X}| \ln |\mathcal{X}|} \tag{5.8}$$

Proof. The proof follows by tailoring the main result of [174], which provides an upper bound to (5.4), namely:

$$\mathcal{R}_T \leq (e-1)\,\gamma \max_{x\in\mathcal{X}} \left\{\sum_{t=1}^{T} f_t(x)\right\} + \frac{|\mathcal{X}| \ln |\mathcal{X}|}{\gamma}. \tag{5.9}$$

The number of *bandit arms* in our case corresponds to the eligible policies; hence it is equal to $|\mathcal{X}|$. Given that: *(i)* the horizon T can be known in advance, and *(ii)* the

Algorithm 3: Bandit Scheduling for vBS (`BSvBS`)

Parameters: $\gamma = (0, 1]$
Initialize: at $t = 1$, $w_1^{\mathrm{B}}(x) \leftarrow 1,\ \forall x \in \mathcal{X}$
for $t = 1, 2, \ldots, T$ **do**
  Define the probability $y_t^{\mathrm{B}}(x),\ \forall x \in \mathcal{X}$ using (5.5).
  Sample next policy: $x_t \sim y_t^{\mathrm{B}}$.
  Receive & scale reward $f_t(x_t)$ using (5.2) and (5.3).
  Calculate weighted feedback $\Phi_t^{\mathrm{B}}(x),\ \forall x \in \mathcal{X}$ using (5.7).
  Update $w_t^{\mathrm{B}}(x),\ \forall x \in \mathcal{X}$ using (5.6).
end

rewards $f_t(x_t)$ for each chosen policy x_t at round t cannot be greater than 1 (due to the normalization described in Section 5.2), we determine an upper bound g of $\max_{x\in\mathcal{X}} \left\{\sum_{t=1}^{T} f_t(x)\right\}$ equal to T, i.e., $g = T$. By choosing the suggested γ, (5.9) leads to (5.8). □

5

Discussion. As `BSvBS` operates with bandit feedback, it is guaranteed to achieve the same performance as the (unknown) single best policy without imposing any conditions on system operation, channel qualities, or traffic demands (Lemma 5.1).

Regarding the overheads of this algorithm, `BSvBS` depends on the number of policies $|\mathcal{X}|$ and the number of rounds T. Each round of the algorithm involves updating the probability distribution over the policies, see equation (5.5), which requires $\mathcal{O}(|\mathcal{X}|)$ time. Additionally, the algorithm updates the weights for each eligible threshold policy based on the reward, which again takes $\mathcal{O}(|\mathcal{X}|)$ time, see equations (5.6) and (5.7). Thus, for T rounds, the time complexity is generally $\mathcal{O}(T|\mathcal{X}|)$. Also, its space complexity is $\mathcal{O}(|\mathcal{X}|)$, as it needs to store only the weights and the probabilities for each policy. In other words, the algorithm is both robust and lightweight in terms of implementation, especially compared to its main competitor, `BP-vRAN` [60], which has $\mathcal{O}(T^3)$ time complexity and $\mathcal{O}(T^2)$ space complexity. Nevertheless, the robustness of `BSvBS` is achieved via a conservative approach that prevents the system from performing better when the conditions allow it. We tackle this issue in the following section.

5.4. Universal Policy Learning via a Meta-Learner

Modeling & Challenges. The analysis in Section 5.3 demonstrates the effectiveness of the proposed adversarial scheme in *all* environments, whether challenging or easy. However, in the latter case, alternative schemes that leverage the knowledge of the environment can achieve faster learning convergence [60]. *Our goal here is to devise a meta-learning scheme that leverages multiple algorithms, each tailored to a specific environment, and chooses dynamically the optimal one.* This idea is leveraged in online learning [19]; however, to the best of the author's knowledge, it is hitherto unexplored for resource allocation in RAN.

In practice, the implementation of such a meta-learning algorithm can be realized in the non-RT RIC, i.e., co-located with the Policy Deciders. Namely, we deploy A rApps, i.e., *algorithms* $a^j, j \in \mathcal{A} = \{1, \ldots, A\}$, each associated with a set of policies

Algorithm 4: Meta-learning for vBS (`MetBS`)

Parameters: $\eta = (0, 1]$
Initialize: at $t = 1$, $w_1^{\mathrm{M}}(j) \leftarrow 1$ and $h_0^{j,\mathrm{S}} \leftarrow \emptyset$, $\forall j \in \mathcal{A}$
for $t = 1, 2, \ldots, T$ **do**
  Define the probability $y_t^{\mathrm{M}}(j)$, $\forall j \in \mathcal{A}$ using (5.10).
  Sample algorithm a^{i_t} according to: $a^{i_t} \sim y_t^{\mathrm{M}}$.
  Algorithm a^{i_t} recommends policy $x_t^{i_t}$ based on $h_t^{i_t,\mathrm{S}}$.
  Receive & scale reward $f_t(x_t^{i_t})$ using (5.2) and (5.3).
  Calculate weighted feedback $\Phi^{\mathrm{M}}(j)$, $\forall j \in \mathcal{A}$ using (5.11).
  Update $w_t^{\mathrm{M}}(j)$, $\forall j \in \mathcal{A}$ using (5.12).
  Sample ξ_t using (5.13).
  if $\xi_t = 0$ **then**
    block feedback of algorithm a^{i_t}, i.e., $h_t^{i_t,\mathrm{S}} \leftarrow h_{t-1}^{i_t,\mathrm{S}}$.
  else
    allow feedback of algorithm a^{i_t}, i.e., $h_t^{i_t,\mathrm{S}} \leftarrow h_{t-1}^{i_t,\mathrm{S}} \cup \left(x_t^{i_t}, f_t(x_t^{i_t})\right)$.
  end
end

$\mathcal{X}^j$; and another rApp for the *meta-learner* that observes their performances over a time horizon of $t=1,\ldots,T$ rounds via the R1 interface (see Figure 5.1). At a time t, an algorithm $a^j, j \in \mathcal{A}$ takes as input the *full* history $h_t^i = \left\{\left(x_\tau^j, f_\tau(x_\tau^j)\right)\right\}_{\tau=1}^{t-1}$ of its previously proposed policies and their respective rewards, and proposes a policy $x_t^j = a^j(h_t^j)$. The objective of the meta-learner is to find the best performing algorithm a^{i^*}, $i^* \in \mathcal{A}$. The challenge lies in the fact that the algorithms are learning entities that update their proposed threshold policies based on bandit feedback, which in turn depends on whether they are selected by the meta-learner. In other words, at round t, the meta-learner chooses one algorithm $i_t \in \mathcal{A}$, denoted as a^{i_t}, which, in turn, proposes one policy $x_t^{i_t} \in \mathcal{X}^{i_t}$ that is deployed in the vBS; and thus, reward $f_t(x_t^{i_t})$ is returned,[5] cf. (5.2). Lastly, a^{i_t} updates its learning state by updating its history $h_t^{i_t} \leftarrow h_{t-1}^{i_t} \cup \left(x_t^{i_t}, f_t(x_t^{i_t})\right)$. All other algorithms, i.e., $\forall j \in \mathcal{A}\colon j \neq i_t$, observe no feedback and do not update their learning state at time t.

This downward spiral creates a challenging situation where the partial feedback reduces the learning capability of the meta-learner, which is further compounded by the limited chances of obtaining feedback for each policy. Without coordination between the meta-learner and the algorithms in the bandit setting, it is proven that the meta-learner will achieve linear regret, even if each of the algorithms obtains sub-linear regret if it were run on its own (and thus obtain feedback in every round) [175], [176]. To surmount this challenge, effective coordination between the algorithms and the meta-learner becomes essential. The approach we employ, inspired by the ideas presented in [176], aims to minimize the interaction required between the algorithms and the meta-learner. Other existing meta-algorithms such as [175] and [177] require feeding unbiased estimates of rewards to the algorithms, meaning that

[5] This is a natural approach for our problem setting, as each algorithm proposes possibly different policies at each round, but only the policy of one algorithm can be deployed to the vBS and return a reward.

the meta-learner has access to the rewards of the algorithms and can modify them; an assumption that we want to drop in our setting.

In our case, the meta-learner can allow or block the chosen algorithm a^{i_t} from learning at round t by sending a corresponding bit (0 or 1). This means that each algorithm $a^j, j \in \mathcal{A}$ has access to *sparse* history $h_t^{j,\mathrm{S}} = \left\{\left(x_\tau^j, f_\tau(x_\tau^j)\right) | \, \xi_\tau = 1\right\}_{\tau=1}^{t-1}$, where ξ_τ is a Bernoulli random variable, i.e., $\xi_\tau \sim \mathcal{B}(\rho_\tau)$, defined by the meta-learner. More precisely, with probability $\rho_t \in (0, 1]$ at each round t, the meta-learner sends bit 1, allowing the chosen algorithm a^{i_t} to learn, i.e., update its history $h_t^{i_t,\mathrm{S}} \leftarrow h_{t-1}^{i_t,\mathrm{S}} \cup \left(x_t^{i_t}, f_t(x_t^{i_t})\right)$; otherwise, $h_t^{i_t,\mathrm{S}} \leftarrow h_{t-1}^{i_t,\mathrm{S}}$. Obviously, it is true that if $\rho_t = 1$, for $t = 1, \ldots, T$, then $h_t^{j,\mathrm{S}} \equiv h_t^j$. Intuitively, this prevents a situation where algorithms that initially find a good policy, but later experience a decline in performance, are continuously selected by the meta-learner over algorithms that explore more extensively in the early stages but achieve superior performance later. By choosing ρ_t accordingly in every round t (see the following analysis), all algorithms could observe feedback in an equal number of rounds (although the best-performing algorithms will be chosen more often) and thus have equal learning steps to improve their performance.

Approach. Following this rationale, the second proposed scheme, named *Meta-Learning for vBS* (`MetBS`), builds upon [176]. Due to its similarity with Algorithm 3, we elaborate next only on its most crucial and distinct steps. The concept lies in learning the sequence of distributions $\{y_t^{\mathrm{M}}\}_{t=1}^T$ (M refers to MetBS), which enables the selection of an algorithm $i_t \in \mathcal{A}$, denoted as a^{i_t} at round t based on the following explore-exploit criteria with parameter η:

$$y_t^{\mathrm{M}}(j) = \frac{\eta}{A} + (1-\eta)\frac{w_t^{\mathrm{M}}(j)}{\sum_{j' \in \mathcal{A}} w_t^{\mathrm{M}}(j')}, \quad \forall\, j \in \mathcal{A}. \tag{5.10}$$

Based on its history $h_t^{i_t,\mathrm{S}}$ and its internal mechanism of using it (e.g., `BSvBS` uses (5.5)), a^{i_t} outputs a policy $x_t^{i_t} \in \mathcal{X}^{i_t}$. The meta-learner observes only the reward $f_t(x_t^{i_t})$ that a^{i_t} produced, and thus, similarly to `BSvBS`, calculates an unbiased estimator for the rewards[6] of all the algorithms (even the unchosen ones):

$$\Phi_t^{\mathrm{M}}(j) = \begin{cases} f_t(x_t^{i_t})/y_t^{\mathrm{M}}(i_t), & \text{if } j = i_t, \\ 0, & \text{otherwise,} \end{cases}, \forall j \in \mathcal{A} \tag{5.11}$$

The weights, which determine the meta-learner's choices in each t, are updated according to:

$$w_{t+1}^{\mathrm{M}}(j) = w_t^{\mathrm{M}}(j) \exp\left(\frac{\gamma \Phi_t^{\mathrm{M}}(j)}{A}\right), \quad \forall\, j \in \mathcal{A}. \tag{5.12}$$

Before `MetBS` proceeds to the next round, it has the ability to block algorithm a^{i_t} from acquiring feedback (i.e., learning) at this particular round t. Consequently,

[6] We recall that no assumptions are made about the sequence of rewards $\{f_t\}_{t=1}^T$, which can even be chosen from an adversary, as described analytically in Section 5.2.

`MetBS` uses the following Bernoulli random variable to allow or block the feedback of a^{i_t}:

$$\xi_t \sim \mathcal{B}\left(\frac{\eta}{A\, y_t^{\mathrm{M}}(j)}\right), j = i_t. \tag{5.13}$$

More specifically, with probability $\rho_t = \eta/(A\, y_t^{\mathrm{M}}(j))$, $j = i_t$ at each round t, the selected algorithm a^{i_t} updates its learning state, while with the remaining probability, its feedback gets blocked. The selection of this random variable ensures that the feedback of each algorithm is allowed, on average, with constant probability $\rho = \eta/A$ over the whole horizon T. The analytical steps of this learning scheme are shown in Algorithm 4.

It is crucial to stress that the regret of the meta-learner w.r.t. the best algorithm, cf. (5.16), is uninformative on its own in the bandit setting. The reason can be attributed to the indirect association between rewards at any given time t and the algorithms the meta-learner previously selected. The past selections define the current learning state of the algorithms, which, in turn, impacts the rewards [177]. Therefore, the evaluation should contain a comparison to an ideal policy that consistently selects the best algorithm, which obtains feedback in every t and performs well with respect to the single best policy. Formally, we are interested in minimizing the regret of the meta-learner w.r.t. the single best policy, which is equal to:

$$\mathcal{R}_T^{\mathrm{M}} = \underbrace{\max_{x \in \mathcal{X}^{i^*}} \left\{ \sum_{t=1}^{T} f_t(x) \right\}}_{\text{best policy}} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} f_t\big(a^{i_t}(h_t^{i_t,\mathrm{S}})\big) \right]}_{\text{meta-learner}}. \tag{5.14}$$

The aggregate reward of the best algorithm a^{i^*} achieved until round t is:

$$\max_{j \in \mathcal{A}} \left\{ \sum_{t=1}^{T} \mathbb{E}\left[f_t\big(a^{j}(h_t^{j,\mathrm{S}})\big) \right] \right\} \equiv \sum_{t=1}^{T} \mathbb{E}\left[f_t\big(a^{i^*}(h_t^{i^*,\mathrm{S}})\big) \right]. \tag{5.15}$$

We add and subtract (5.15) from (5.14), and we derive:

$$\mathcal{R}_T^{\mathrm{M}} = \mathcal{R}_T^{\mathrm{M}_1} + \mathcal{R}_T^{\mathrm{M}_2},$$

where $\mathcal{R}_T^{\mathrm{M}_1}$ corresponds to the regret of the meta-learner with respect to the best algorithm:

$$\mathcal{R}_T^{\mathrm{M}_1} = \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[f_t\big(a^{i^*}(h_t^{i^*,\mathrm{S}})\big) \right]}_{\text{best algorithm}} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} f_t\big(a^{i_t}(h_t^{i_t,\mathrm{S}})\big) \right]}_{\text{meta-learner}}, \tag{5.16}$$

and $\mathcal{R}_T^{\mathrm{M}_2}$ corresponds to the regret of the best algorithm w.r.t. to the best policy:

$$\mathcal{R}_T^{\mathrm{M}_2} = \underbrace{\max_{x \in \mathcal{X}^{i^*}} \left\{ \sum_{t=1}^{T} f_t(x) \right\}}_{\text{best policy}} - \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[f_t\big(a^{i^*}(h_t^{i^*,\mathrm{S}})\big) \right]}_{\text{best algorithm}}.$$

If a^{i^*} had access to its full history $h_t^{i^*}$, we denote as $\beta^{i^*} \in [0,1]$ the exponent of the upper bound of its regret, namely[7]:

$$\max_{x \in \mathcal{X}^{i^*}} \left\{ \sum_{t=1}^{T} f_t(x) \right\} - \sum_{t=1}^{T} \mathbb{E}\Big[f_t\big(a^{i^*}(h_t^{i^*})\big) \Big] \leq \mathcal{O}(T^{\beta^{i^*}}).$$

However, in the considered analysis, it has access to its partial history $h_t^{i^*,\mathrm{S}}$. For proving a non-trivial upper bound on $\mathcal{R}_T^{\mathrm{M}}$ in this case, the best performing algorithm a^{i^*} should satisfy the following:

$$\max_{x \in \mathcal{X}^{i^*}} \left\{ \sum_{t=1}^{T} f_t(x) \right\} - \sum_{t=1}^{T} \mathbb{E}\Big[f_t\big(a^{i^*}(h_t^{i^*,\mathrm{S}})\big) \Big] \leq \mathcal{O}\left(\frac{(\rho T)^{\beta^{i^*}}}{\rho} \right), \tag{5.17}$$

where $\rho = \eta / A$, as defined beforehand. A rich class of online learning algorithms, including Exp3 (and thus, `BSvBS`), satisfy (5.17), which, in turn, quantifies the robustness of an online learning algorithm w.r.t. the sparsity of the history [176].

The performance of Algorithm 4 is captured by the following lemma:

Lemma 5.2. *Let $T > 0$ be a fixed time horizon, and assume the best algorithm, a^{i^*}, satisfies* (5.17) *with β^{i^*}. Set input parameter $\eta = \Theta\big(T^{-\frac{1-\beta}{2-\beta}} A^{\frac{1-\beta}{2-\beta}} (\log A)^{\frac{1}{2}\mathbb{1}\{\beta=0\}}\big)$, where $\beta \geq \beta^{i^*}$. Then, running Algorithm 4 ensures that the expected regret is sub-linear:*

$$\mathcal{R}_T^M \leq \mathcal{O}\big(T^{\frac{1}{2-\beta}} A^{\frac{1}{2-\beta}} (\log A)^{\frac{1}{2}\mathbb{1}\{\beta=0\}}\big) \tag{5.18}$$

Proof. The proof follows by tailoring the main result of [176]; we therefore provide a brief but sufficient explanation. By applying Lemma 5.1, (5.16) gives:

$$\mathcal{R}_T^{\mathrm{M}_1} \leq c\,\eta\,T + \frac{A \log A}{\eta}, \tag{5.19}$$

where $c > 0$ is a constant. Adding (5.17) and (5.19), results in:

$$\mathcal{R}_T^{\mathrm{M}} \leq \mathcal{O}\big(\eta\,T + \frac{A \log A}{\eta} + \frac{T^{\beta^{i^*}} A^{1-\beta^{i^*}}}{\eta^{1-\beta^{i^*}}}\big). \tag{5.20}$$

Setting $\eta \sim T^{-z}$ and finding the z that minimizes the power of T in (5.20), leads to (5.18). □

Discussion. When interacting with *learning algorithms* in the *bandit* setting, Algorithm 4 is guaranteed to achieve the same performance as the best algorithm if it ran on its own (and thus, acquiring feedback in every round). Hence, `MetBS` attains reward as the (unknown) single best algorithm without making assumptions for the

[7] For instance, if `BSvBS` is the best algorithm a^{i^*}, then $\beta^{i^*} = 1/2$, see Lemma 5.1.

environment (see Lemma 5.2). This accomplishment is made possible through minimum coordination between the meta-learner and the algorithms, as described in lines 10-11 of Algorithm 4.

In terms of implementation, `MetBS` can be implemented as another rApp, which also facilitates its coordination with the co-located rApps implementing the different algorithms; see also Figure 5.1b. Regarding its overheads, due to its similarity with `BSvBS`, its complexity depends on the number of algorithms that it chooses from, i.e., $\mathcal{O}(T|\mathcal{A}|)$ for T rounds. However, as it chooses between different algorithms (where each of them selects policies and has its own complexity), the overall time complexity of `MetBS` depends on the worst-case scenario of the most time-complex algorithm. Similarly, its space complexity is equal to $\mathcal{O}(|\mathcal{A}|)$; however, an important factor is the complexity of the algorithms that it chooses from, and especially, the most space-complex algorithm.

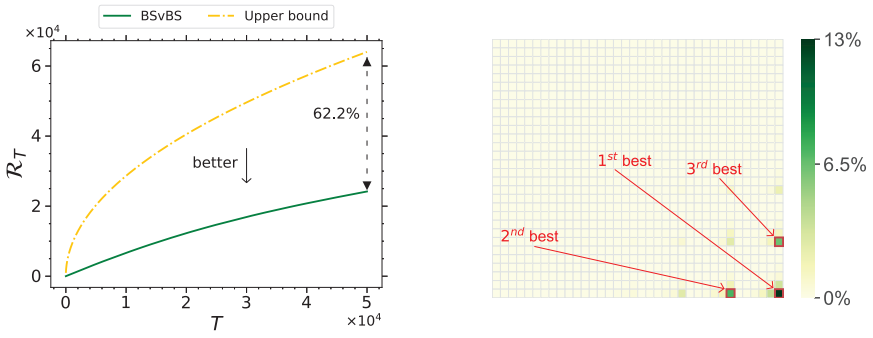
5.5. Performance Evaluation

Experimental Setup & Scenarios. The solutions are assessed under different traffic and environment scenarios using our recent publicly-available dataset [60] with power consumption and throughput measurements from an O-RAN compatible testbed. This experimental setup includes a vBS and a UE[8], implemented as srseNB and srsUE from the srsRAN suite [55]. The RUs of the vBS and UE are composed of an Ettus Research USRP B210, and their BBUs and near-RT RICs are implemented on general-purpose computers (Intel NUC BOXNUC8I7BEH). The power consumption of the BBU and RU is measured with the GW-Instek GPM-8213. A 10 MHz band is selected, supplying a maximum capacity of approximately 32 Mbps and 23 Mbps for the downlink and uplink operation, respectively. The non-RT threshold policies are calculated in a programming language, emulating the operation of rApps; the real-time scheduling decisions are made by the default srsRAN scheduler that has been amended to comply with the MCS, PRB, and power thresholds that are provided to them in each round.

The dataset contains 32 797 measurements for $|\mathcal{X}|=1080$ policies corresponding to $\mathcal{B}^{\mathrm{d}}=\{0,0.2,0.6,0.8,1\}$, $\mathcal{B}^{\mathrm{u}}=\{0.01,0.2,0.4,0.6,0.8,1\}$, $\mathcal{M}^{\mathrm{d}}=\{0,5,11,16,22,27\}$, $\mathcal{P}^{\mathrm{d}}=\{3\}$[9] and $\mathcal{M}^{\mathrm{u}}=\{0,5,9,14,18,23\}$. The random perturbations, as explained in Section 5.2, emanate due to time-varying UL and DL demands, $\{d_t^{\mathrm{u}},d_t^{\mathrm{d}}\}_{t=1}^{T}$, measured in Mbps, and time-varying CQIs, $\{c_t^{\mathrm{u}},c_t^{\mathrm{d}}\}_{t=1}^{T}$, which are dimensionless. The transmitted data, $\{R_t^{\mathrm{u}},R_t^{\mathrm{d}}\}_{t=1}^{T}$, are calculated by multiplying the values of $\mathcal{B}^{\mathrm{d}}$ ($\mathcal{B}^{\mathrm{u}}$) with the transport block size (TBS); the latter is determined by mapping the $\mathcal{M}^{\mathrm{d}}$ ($\mathcal{M}^{\mathrm{u}}$) with the TBS index [178]. W.l.o.g., we have assumed 50 PRBs. The power cost function is set to $P_t(x_t)=V_t$, where V_t is the total power consumed by

[8] The usage of one UE is not limiting for our study, since the algorithm devises each vBS's thresholds based on the average (across users) throughput and energy, and the average CQI and traffic, i.e., the UE emulates the load of multiple users.

[9] The DL transmission power is determined through the transmission gain of the USRP implementing the BS. The RU of the testbed is equipped with a fixed power amplifier that consumes 3 W and a variable attenuator for power calibration. To account for this limitation, the dataset power measurements are post-processed using linear modeling [60].



(a) (b)

Figure 5.2: (a) R_T achieved from `BSvBS` in Scenario A (static) and its upper bound; (b) heatmap for the choices of `BSvBS` in Scenario A, showing the probability that each policy is chosen at $t = 50k$.

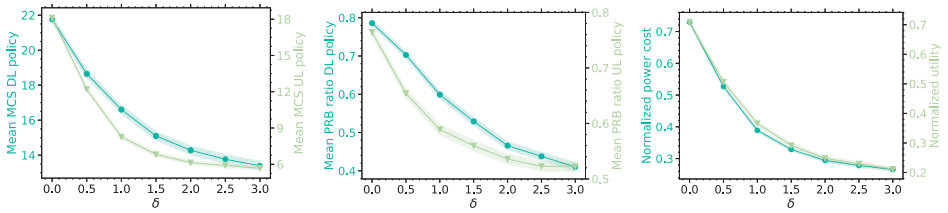
the vBS, and the utility function as stated in (5.2). We also scale both components of the reward function to $[0, 1]$ and choose $\delta = 1.5$ to prioritize power consumption unless stated otherwise. We set $\gamma = 0.29$ for `BSvBS` and $\eta = 0.04$ for `MetBS`, and use $T = 50k$. All results are averaged over 10 independent experiments.

For the ensuing analysis, we assess three scenarios which represent a static environment (fixed, time-invariant parameters); a stationary stochastic environment (i.i.d. parameters); and an adversarial scenario. The latter, clearly, is an extreme case (e.g., can appear under high mobility conditions, heavy interference or attacks) that we use to demonstrate the robustness of the learning algorithms. On the other hand, the first two scenarios are in line with those typically considered by prior benchmarks, e.g., [60], [62]. In detail:

- *Scenario A (static)*: the demands and CQIs take the highest possible values according to our testbed, i.e., $d_t^{\mathrm{d}} = 32$, $d_t^{\mathrm{u}} = 23$, $c_t^{\mathrm{d}} = 15$, $c_t^{\mathrm{u}} = 15$.
- *Scenario B (stationary)*: the demands and CQIs are drawn randomly from fixed uniform distributions in each round, i.e., $d_t^{\mathrm{d}} \sim \mathcal{U}(29, 32)$, $d_t^{\mathrm{u}} \sim \mathcal{U}(20, 23)$, c_t^{d}, $c_t^{\mathrm{u}} \sim \mathcal{U}(1, 3)$, where $\mathcal{U}(a, b)$ denotes the uniform distribution over the interval $[a, b]$.
- *Scenario C (adversarial)*: the demands and CQIs are drawn randomly in a *ping-pong* way; namely, in *odd* rounds according to $d_t^{\mathrm{d}} \sim \mathcal{U}(29, 32)$, $d_t^{\mathrm{u}} \sim \mathcal{U}(20, 23)$, c_t^{d}, $c_t^{\mathrm{u}} \sim \mathcal{U}(13, 15)$, and in *even* rounds from d_t^{d}, $d_t^{\mathrm{u}} \sim \mathcal{U}(0.01, 1)$, $c_t^{\mathrm{d}}, c_t^{\mathrm{u}} \sim \mathcal{U}(1, 3)$. [10] We note note that the learner does not have access to this information, and is oblivious to when the switches happen.

Scenario C resembles dynamic environments, where the parameters might change drastically every round. This corresponds to the most challenging-to-learn *adversarial* schemes in regret analysis, cf. [179]. Clearly, an algorithm that performs well under this case is guaranteed to perform well in all other scenarios. In the sequel, we use these scenarios to explore the convergence of the proposed learning and meta-learning algorithms, and compare them with selected state-of-the-art competitors in terms of *(i)* regret, *(ii)* vBS power savings, and *(iii)* inference time.

[10] CQI 13 and 15 correspond to SNR of 25 dB and 29 dB, while CQI 1 and 3 to SNR of 1.95 dB and 6 dB, respectively.



(a) (b) (c)

Figure 5.3: Scenario A (static) for `BSvBS`: a) MCS in DL (left) / UL (right); (b) PRB ratio in DL (left) / UL (right); c) power (left) and utility (right) w.r.t. δ, with 0.95-CI. In each plot, the blue and green lines correspond to the left and right y-axis, respectively.

Static & Stationary Scenarios. Figure 5.2a shows the expected regret in Scenario A when prioritizing the utility function (small δ). The attained regret is sub-linear and 62.2 % smaller than the upper bound (which is itself sub-linear), cf. (5.8). To complement the analysis, Figure 5.2b shows a grid with 1080 cells, each mapping a different policy. The cells are colored based on the probability `BSvBS` selects each policy at $t = 50k$, where darker colors indicate higher probabilities. The red squares indicate the three best policies chosen 25 % of the rounds, where the top-performing one is selected twice as frequently. This outcome can be attributed to the small δ, which favors the policy with the highest MCS and PRB ratio in both DL and UL, as the demands and CQIs are high. For the second and third-best policies, the MCS in UL and DL take the highest values, except for the PRB ratios, which are fixed at 0.8, namely, the second-best UL and DL PRB ratios.

Figure 5.3a and 5.3b delineate the effect of δ on the MCS DL/UL, and PRB ratio DL/UL, respectively (i.e., the chosen policies), for the static scenario. The solid lines in the plots represent the mean values averaging 100 rounds after running `BSvBS` for $t = 50k$ rounds, and the shadowed areas are the 0.95-confidence intervals. Moreover, the blue and green lines correspond to the left and right y-axis, respectively. We observe that smaller δ leads to higher MCS and PRB ratio choices in DL and UL. This is justified by the high CQI values considered in this scenario, as they enable using higher MCS, which allows more data transmission and larger decoding computational load [180]. Furthermore, larger δ in Scenario A effectuates the selection of lower MCS and PRB values in order for the vBS to save resources by diminishing the turbo decoding iterations.

Similarly, Figure 5.3c illustrates the impact of δ on the reward function, where its two components are normalized, see (5.2). Higher δ boosts the usage of policies that minimize the consumed power, forcing the utility function to decrease, whereas lower δ leads to policies that maximize the utility but increase the power consumption. Values $\delta > 2$ have less effect on the power and utility functions, as there is a limit in the consumed power that can be saved.

Figure 5.4 depicts the average regret over time for stationary Scenario B, which converges towards zero as time elapses. We also plot the average regret of a typical benchmark that randomly selects policies with equal probability; we call this benchmark `Random`. `BSvBS` explores policies with probability 29 % (since $\gamma = 0.29$) and exploits the best-performing ones with probability 71 %. Therefore, in the first

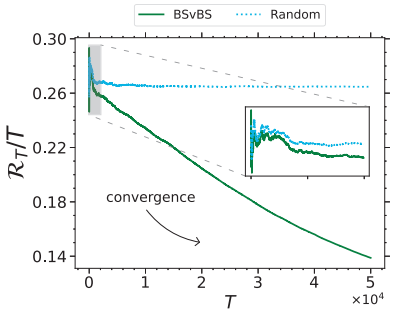


Figure 5.4: R_T/T for `BSvBS` in Scenario B (stationary), together with `Random`, a naive algorithm that selects policies randomly.

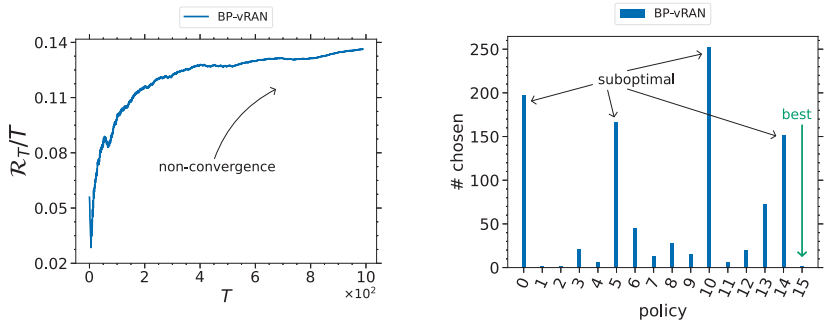
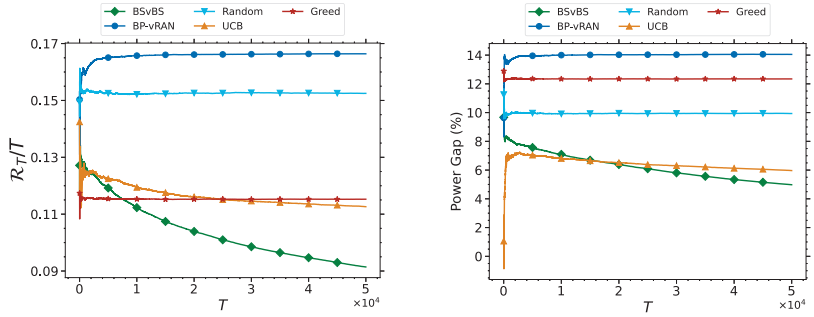


Figure 5.5: `BP-vRAN` executed for $T = 1000$ rounds in dynamic Scenario C, in a subset of the policy space: (a) R_T/T; (b) number of times each policy was chosen.

800 rounds, `BSvBS` obtains similar regret as the benchmark algorithm, but their performance difference grows gradually, reaching 33.3 % in round $t = 50k$, as `BSvBS` opts for the best-performing policies with higher probability at latter stages.

Key takeaways: *(i)* The measured regret is sub-linear in static and stationary scenarios and substantially smaller (up to 62.2 %) than the theoretical bound. *(ii)* The network can adjust δ to trade certain power consumption with commensurate losses in utility; yet, increasing δ more than a specific value ($\delta = 2$ in our case) does not provide further substantial savings.

Gap in Prior Work. The primary objective is to showcase how state-of-the-art techniques perform inadequately in challenging environments. To delineate this effect, we focus on a smaller set of policies, i.e., $|\mathcal{M}_d| = |\mathcal{M}_u| = |\mathcal{B}_u| = |\mathcal{B}_d| = 2$ and $|\mathcal{P}_d| = 1$, yielding $|\mathcal{X}| = 16$ policies. The performance of the `BP-vRAN` algorithm [60], which constitutes, to the best of the authors' knowledge, the only existing work designed to configure such threshold policies in vBS, is assessed in adversarial Scenario C. `BP-vRAN`, which is based on the seminal GP-UCB algorithm [181], models the traffic demands and CQIs as *context*, which are observed before the policy is decided. Given that the context directly impacts the selection of policies, it will



(a) (b)

Figure 5.6: Comparison of BSvBS with several competitors in adversarial Scenario C: (a) R_T/T; (b) power saving of each algorithm with respect to the ideal-minimum energy of the benchmark.

be shown how abrupt changes in CQI values and traffic demand deteriorate the algorithm performance. We present an example where the context differs between its observation and application to the system. This case appears quite often in practice, given that the rounds of reference are of several seconds. For the plots in this section, the reward function $f_t(x_t)$ is unbounded.[11]

As indicated in Figure 5.5a, the average regret in the adversarial Scenario C does not decrease (in fact, it increases) after $T = 1k$ rounds, which is more than $33\times$ of the advertised convergence time. This happens because the algorithm takes decisions in each t by assuming perfect knowledge of f_t, which might take arbitrary values depending on the environment. Clearly, due to the system's volatility, the policy for each t should be selected based on past values $\{f_\tau(x_\tau)\}_{\tau=1}^{t-1}$; yet, as Figure 5.5b corroborates, BP-vRAN selects sub-optimal policies for most rounds and fails to explore efficiently even this small space.

Evaluation of the Bandit Algorithm. Figure 5.6a compares the average regret over time of BSvBS for Scenario C, in relation to several competitor algorithms, namely: the BP-vRAN, a naive algorithm that selects thresholds uniformly randomly (Random); the classical UCB algorithm that is designed for stationary environments [182]; and a greedy algorithm that prioritizes exploitation (Greed, selects the best solution found until now) [183]. We consider $T = 50k$ rounds and use the complete policy space (i.e., $|\mathcal{X}| = 1080$), and all results are averaged over 10 independent experiments. We observe that BSvBS is superior, acquiring 45.1% less regret w.r.t BP-vRAN, and 22% less w.r.t Greed and UCB at $t = 50k$. It is worth noting that Random performs better than BP-vRAN in this case, by approximately 9%.

In Figure 5.6b, we present the vBS power gains that each algorithm achieves in the same scenario, w.r.t. the ideal-minimum-energy of the benchmark, where the power consumption of the idle user is subtracted. It can be seen that with BSvBS, the network operator can save up to 64.5% of energy if the algorithm runs for $t = 50k$ rounds in contrast to BP-vRAN. Moreover, it can be seen that UCB also chooses policies

[11] When BSvBS is depicted in the same plot as BP-vRAN, the reward function of BP-vRAN is scaled too.

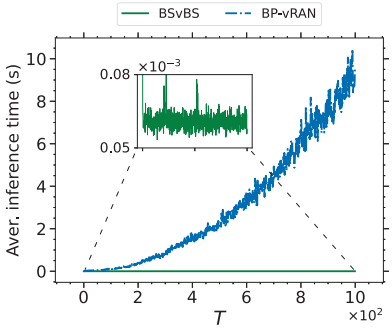


Figure 5.7: Average time needed to infer a policy in each round, for our algorithm BSvBS, and its main competitor, BP-vRAN.

that allow for saving energy, but again, attains less energy saving than BSvBS. These plots also showcase that the Greed algorithm, which does not explore new policies, is not competitive and is stuck in exploiting sub-optimal policies (straight line in the regret plot).

Another key advantage of BSvBS is its low inference time, i.e., the time to deduce a policy in each round. Figure 5.7 exhibits the average inference time and compares it with BP-vRAN. Using standard kernel-based methods (as BP-vRAN does) is widely recognized to result in a high computational cost of $\mathcal{O}(t^3)$ with respect to the number of data points t [184]. This is a significant limitation as it delays the vBS operation to more than 10 s after $t = 1k$ when tested on an Apple M1 chip with 8-core CPU@3.2 GHz. Clearly, this hinders the vBS operation, which will then have to rely on stale information. On the other hand, we notice that BSvBS requires no more than 0.08 ms to decide a policy, which remains constant throughout.

Key takeaways: In challenging (i.e., non-stationary / adversarial) environments, decisions for configuring the vBS should be taken based on past performance. Requiring *perfect* knowledge of the environment could lead to sub-optimal policies, increasing power costs up to 64.5 % for operators. BSvBS's performance is robust to such adversarial scenarios and outperforms a state-of-the-art algorithm in terms of: *(i)* the average regret (up to 45.1 % superiority), *(ii)* the power gap w.r.t. the minimum vBS energy consumption (up to 64.5 % superiority), and *(iii)* inference time (solely 0.08 ms). We recall that BSvBS does not have access to how and when the demands and CQI change.

Evaluation of the Meta-Learning Algorithm. We consider $A = 2$ with BP-vRAN, and BSvBS that select policies from $\mathcal{X}$. On the one hand, if the context is not available at the beginning of each round, as happens in several real-world applications, BSvBS is superior and BP-vRAN fails, as seen in Section 5.5. Hence, MetBS opts mainly for BSvBS. The attained regret is by 61.7 % less than the upper bound, which implies the desired sub-linear regret. The algorithms that MetBS chooses can be verified in Figure 5.8b, where BSvBS is selected in approximately $47k$ rounds, while the sub-optimal BP-vRAN in the remaining $3k$ rounds ($T = 50k$). On the other hand, if the environment is easy, BP-vRAN is expected to converge faster than BSvBS; and, as a consequence, to be preferred by the meta-learner. Indeed, the regret of MetBS is 96 % lower than the upper bound stated in (5.8), which clearly indicates the ex-

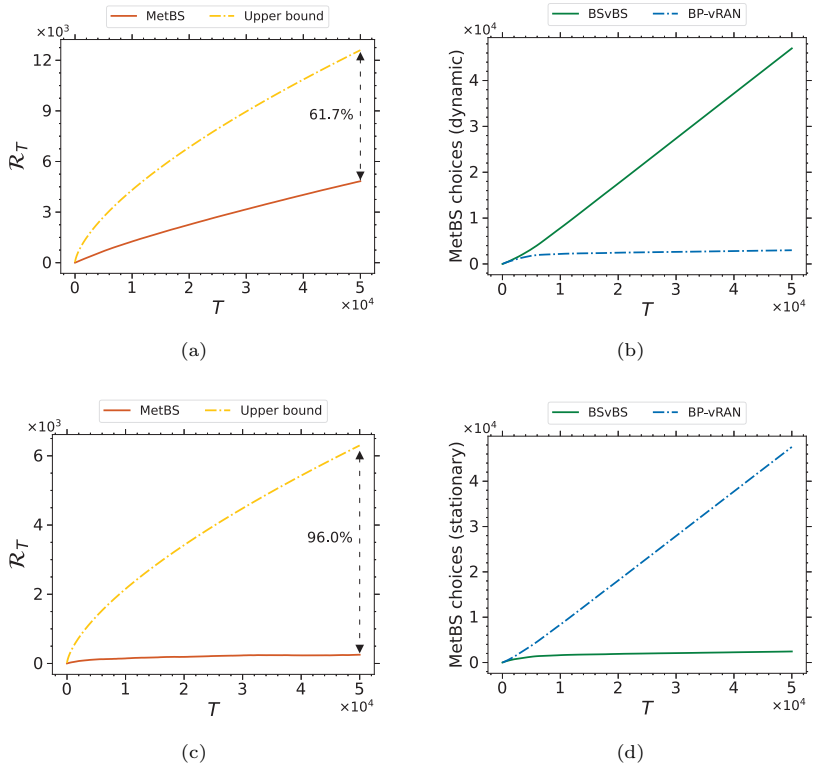
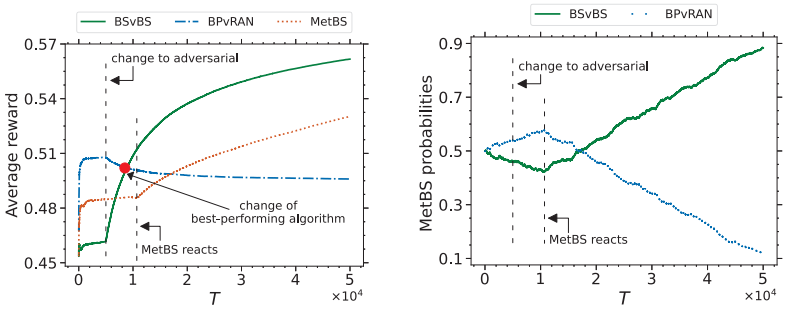


Figure 5.8: Meta-learning algorithm: R_T and the upper bound for dynamic (a) and stationary (c) scenarios; number of times `BSvBS` and `BP-vRAN` were chosen in $T = 50k$ rounds for dynamic (b) and stationary (d) scenarios.

pected sub-linear regret has been achieved. `MetBS` selects `BP-vRAN` in roughly $46k$ rounds, while `BSvBS` in $4k$ rounds, Figure 5.8d. It is important to heed that `BSvBS` converges as well to the optimal policy but slower (see Figure 5.2 and Figure 5.4), an unavoidable side-effect of its robustness under any environment (even adversarial).

Finally, we test the meta-learner in a "mixed" environment, where, in the first $5k$ rounds the demands and CQIs are drawn from Scenario B (stationary), and in the remaining $45k$ rounds from Scenario C (adversarial). Figure 5.9a depicts the average rewards of `MetBS`, `BSvBS`, and `BP-vRAN`. It can be viewed that before the change of the environment, the average reward of the meta-learner follows closer to the reward of `BP-vRAN`; the orange dotted line is 3.8 % lower than the blue dash-dotted line. The same can be verified from Figure 5.9b, where `BP-vRAN` is chosen with higher probability, 58 %, before $t = 5k$. When the change occurs, `MetBS` does not opt immediately for `BSvBS`, as the average reward of `BP-vRAN` is still higher, until the change-point at roughly $t = 8k$, which is shown with a red dot in Figure 5.9a. After this round, `BSvBS` experiences larger reward values on average, and within less than $1k$ rounds (i.e., 2 %), `MetBS` starts indeed selecting `BSvBS` more frequently (up to 88.2 %).



(a) (b)

Figure 5.9: (a) Average reward and (b) probabilities that `BSvBS` and `BP-vRAN` are chosen by the `MetBS` when the environment changes (leftmost dashed line, *change to adversarial*) from stationary to adversarial at $t = 5k$. The red dot (*change of best-performing algorithm*) shows the change of the best-performing algorithm (from `BP-vRAN` to `BSvBS`), and the rightmost dashed line `MetBS` reacts) depicts the round after which `MetBS` starts choosing the best-performing algorithm, `BSvBS`, more often.

Key takeaways: `MetBS` chooses the best-performing algorithm for each scenario. When the demands and CQIs are drawn from a stationary distribution, it prioritizes `BP-vRAN` (92 % of rounds), while in adversarial scenarios, `BSvBS` (94 % of rounds). In mixed scenarios, `MetBS` tracks and applies the changes after only 2 % of rounds.

5.6. Related Work

Resource management for *softwarized* networks can be broadly classified into models that relate policies to performance functions, model-free approaches, and Reinforcement Learning (RL) techniques. Model-based examples include [58] and [185], which maximize the served traffic subject to vBS computing capacity, but do not capture the impact that the hosting platform, the environment, or user demands may have on the vBS's operation [59]. Model-free approaches employ Neural Networks to approximate the performance functions of interest [186], yet, their efficacy depends on the availability of representative training data. Finally, RL solutions [187] use runtime observations and have been used, for example, in interference management [188], vBS function splitting [189], and handover optimization [35]. The disadvantages of all these works are the curse of dimensionality and the lack of robust convergence guarantees [190]. Following an akin approach, contextual bandit algorithms are employed to optimize video streaming rates [191] or handover decisions [192]; assign CPU time to virtualized BSs [61], and control millimeter-Wave networks [193]. Unfortunately, these works require access to *context* information. More recently, Bayesian learning has been used for RRM, see [194] and references therein, but these solutions also need access to context information and converge only under stationary conditions.

We take here a different approach, based on *adversarial* bandits, cf. [179], which is robust to adversarial or non-stationary contexts (channel qualities and traffic demands), and has low memory and computation requirements. This latter feature is in stark contrast to RL (with sizeable memory space required to store all

space-actions combinations) and Bayesian techniques [60], [194] which involve slow matrix inversions [195]. Such adversarial/non-stationary environments are increasingly common due to highly volatile network conditions [196] and traffic demands [173]. Furthermore, we draw ideas from the expert-learning paradigm [18] and enrich our policy decisions with a meta-learning scheme that combines our adversarial learning algorithm (that can be at times conservative) with any other algorithm (e.g., [60]) that performs better on more *easy* scenarios where the environment is predictable (e.g., when stationary). This meta-algorithm obtains the best of both approaches, and succeeds in being both fast-learning and robust; an idea that has been used in online learning [19], but not in network management.

We employ the above method to tackle a joint performance and energy cost optimization problem. Similar (in scope) formulations have been extensively studied in the literature. For instance, [197] considered a joint user association, spectrum, and power allocation model for throughput optimization; [198] focused on spectrum and energy efficient beamforming; and [199] optimized the spectrum and power assignment using genetic algorithms. Nevertheless, such approaches assume the system and user-related parameters to be fixed and known. On the other hand, many dynamic formulations rely on RL to optimize energy and performance, e.g., [200]; or on variants of the seminal CGP-UCB algorithm [60]. The main limitation of these works is the need to know contextual information (channels, user demands, etc.), and the lack of optimality guarantees, mainly in non-stationary conditions. Our approach is instead tailored to handle the inherent performance and cost volatility of O-RAN systems without access to context and provides optimality guarantees against competitive oracles.

Finally, a key difference between our work and the above RRM literature lies in our emphasis on non-RT RAN policies. These policies serve as operational thresholds for the real-time vBS (i.e., RRM) schedulers and are facilitated by the O-RAN architecture, which has provisions for such tiered control loops [10], [11]. This approach enables centralized management of multiple BSs without disrupting their RRM functionality. Recent works [172], [201] use RL for selecting the slicing and scheduling policies in O-RAN (i.e., RRM schedulers, see Section 5.2). Nevertheless, our policy thresholds operate on a higher timescale (i.e., Non-RT) and are fed to these vBS schedulers, which make RRM decisions subject to our provided thresholds; also, they learn in an online (not offline) manner and adapt to environment changes, even if these changes happen drastically.

A recent stream of works has followed this path to design such non-RT operation thresholds. Namely, [60] uses CGP-UCB to identify thresholds for the maximum allowed transmission power and MCS to reduce the energy consumption of base stations; [62] follows a similar approach but focuses on different performance KPIs; and [61] decides maximum MCS and duty cycles through deep learning. Unlike these works, our solution is the first to provide optimality guarantees for non-stationary environments and without requiring access to context, while through the proposed meta-learner we can combine and benefit from other algorithms (e.g., [60]) when they perform well.

5.7. Conclusion

The virtualization of base stations and the design of O-RAN systems are instrumental for the success of the next generation of mobile networks. Allocating resources for these vBSs by choosing policies that operate on a longer time scale and do not require intervention in (often proprietary) vBS node implementations are a new and promising network control approach. However, in order to be practical and successful, the proposed solutions should have low overhead and require no assumption about the future channel qualities and traffic demands (i.e., the environment).

The first proposed scheme possesses exactly these properties, building on a tailored adversarial learning algorithm that has minimal overhead and can run in submilliseconds. In line with prior works, we focus on the important metrics of throughput and energy consumption and explore their trade-offs in a range of scenarios with experimental datasets. As this robustness comes at a cost for convergence speed, we aim to increase the latter in easy scenarios, where the environment is known beforehand (or changes slowly), through a meta-learning scheme that combines a mix of algorithms, including our own, and delineates the best-performing one at runtime. This creates a best-of-both-worlds solution. Our extensive data-driven experiments demonstrate energy savings up to 64.5 % compared to state-of-the-art competitors.

6

Conclusion

6.1. Looking Back

Modern cellular systems operate in dynamic conditions/environments with heterogeneous users and use cases, ranging from autonomous vehicles and augmented reality to remote healthcare and massive IoT deployments. These scenarios, enabled by 5G and envisioned under 6G and NextG roadmaps, impose diverse and often conflicting requirements on the network, including ultra-low latency, high reliability, extreme data rates, and support for high-mobility users. Traditional control decisions, while stable, are brittle under changing conditions, and classical optimization techniques often rely on static assumptions that do not hold in practice. On the other end of the spectrum, offline-trained ML models offer performance in known/observed conditions but fail to generalize under distribution shifts or with limited data.

The work presented in this thesis revolves around a central challenge: making robust and effective mobility and resource management decisions within the emerging O-RAN ecosystem based on real-world insights from countrywide datasets and testbed traces, by using online meta-learning. The emerging Open RAN architecture, embodied primarily through the O-RAN Alliance, provides a flexible and programmable foundation by disaggregating control from hardware and enabling AI, data-driven decision-making through open interfaces and intelligent controllers.

Specifically, we focused on two core problems: *(i) user-cell association*, with induced *handovers*, a main mobility management problem where the network, and sometimes in cooperation with users, decides which cell should serve them over time, and *(ii) resource allocation*, where virtualized base stations manage and assign their resources to connected users; and in both problems, without knowledge of future conditions. These decisions were inherently sequential and coupled, as assigning a user to a cell directly influences how and to which users that cell allocates its resources, shaping performance at both the user and network level.

We proposed online learning algorithms that offer robust performance guarantees under both static and dynamic regret, obtaining as good (user and network)

performance as an oracle algorithm with full information for the future would have achieved. Moreover, we extended these approaches via meta-learning to enable the usage of other offline and/or online learners, which can be tailored for different environments. Importantly, all algorithms were compatible with the principles of O-RAN, ensuring that they were not only theoretically grounded but also practically deployable in the next generation of cellular networks. Our methodologies were supported by empirical insights drawn from large-scale datasets, including countrywide mobility and handover traces. These datasets revealed the structural complexity and variability in networks nowadays, motivating the need for learning-based control. We also leveraged a novel dataset for resource allocation decisions, utilizing a testbed to further ground our algorithms in real-world scenarios.

This thesis represents, to our knowledge, the first comprehensive integration of countrywide empirical network analysis and O-RAN-compatible testbeds with online meta-learning into the end-to-end control stack of O-RAN: from UE-cell associations and handover decisions to virtualized base station resource allocation. Our results establish that learning-based control under uncertainty is not only feasible but essential for modern mobile networks and the NextGs to come. The architectural and algorithmic abstractions introduced here extend beyond the specifics of cellular control. They suggest a more general vision: that modern networked systems can (and should) adapt their operation (i.e., learn) continuously and be robust.

6.2. Limitations

In this dissertation, datasets and actual, unnormalized numbers in figures related to the real-world data could not be published openly due to privacy guidelines of the MNO (see Appendix 2A). Moreover, at the time of capturing all datasets, the 5G-SA deployment of the MNO was still in its early stages, with a limited range of (mostly test) UEs actively using it. Thus, we focused on 5G-NSA, which relies on the 4G EPC for mobility management. In other words, we could not explicitly capture the HOs to/from 5G-SA radio cells, since the EPC only sees their corresponding 4G radio cell anchor. In addition, the studied HOs had 4G/5G-NSA as the source RAT, and 4G/5G-NSA, 3G, or 2G as the target. In other words, apart from the horizontal HOs in 4G/5G-NSA, we focused on the specifics of how/when/why users downgrade to older RATs, and not the other way around, given that users spend more than 82% of their time and 94.5% of their traffic in 4G/5G (see Section 2.3).

Furthermore, (*i*) we did not have access to HO configuration parameters and policies, which are dynamically configured by proprietary solutions from equipment vendors, (*ii*) our analysis on HOFs and mobility was limited to the use of mobility metrics (number of cells and radius of gyrations) at daily intervals, which may hide correlations that occur at finer time scales, and (*iii*) CHOs were still not deployed at the studied MNO (no logs available).

In addition, the algorithms proposed (*i*) assumed a centralized controller, taking decision for multiple users and cells, (*ii*) were tested with real-world data, as well as synthetic traces covering from static to even adversarial conditions, but were not deployed in live environments.

6.3. Summary of Contributions

In this section, we synopsize the main accomplishments of this thesis.

Chapter 2: Heterogeneity and Mobility Management of Cellular Networks. This chapter offered the foundation for the thesis by presenting a large-scale (countrywide) empirical study of modern cellular networks and their mobility management through the lens of a top-tier MNO. Prior studies relied on small-scale measurement campaigns or limited devices, resulting in a partial view of operational networks. To overcome these limitations, we captured various 4-week datasets and leveraged open-source census records involving millions of users and thousands of cells, to uncover the complexity of modern deployments along three particularly interesting axes, from the network's perspective: the heterogeneity of (*i*) RATs and (*ii*) UEs, as well as the (*iii*) geodemographic diversity. We further delved into HOs, the cornerstone of mobility management in cellular networks. Specifically, we investigated their frequencies, duration, and types, as well as uncovered temporal and spatial correlations and causes of failures, which we modeled using statistical tools. Our analysis revealed inefficiencies in HO realization, motivating the need for learning-based control in these complex cellular systems.

Chapter 3: Mobility Management through Smooth Handovers. To ground our HO optimization strategies in real-world scenarios, we leveraged and extended the visions presented in the previous chapter, providing fresh insights into the effects of HOs on KPIs such as packet loss and throughput, as well as crowdsourced signal data. In this way, we identified key correlations between HO failures/delays and the characteristics of radio cells and devices. Subsequently, we formulated the user-cell association problem as an instance of OCO, where changes in association decisions between two consecutive timeslots cause handovers, inducing measurable switching costs that depend on the users and cells themselves. We proposed a meta-learning framework that optimizes handover decisions by balancing and prioritizing, if needed, the throughput of users and the (often) increased delay cost of frequent handovers. The algorithm integrated cell and device features into its decision-making and eliminated the need for knowledge of future external conditions, measurements, or trajectory information. Our experimental results demonstrated that this approach improved performance in a battery of scenarios, including real-world ones from our crowdsourced datasets, achieving robust dynamic regret guarantees and enabling practical deployment in O-RAN. Nevertheless, THOs alone may still pose challenges in dense deployments or high-frequency bands due to their reactive nature. This motivated the following step: we explored how CHOs, which delegate part of the decision-making to users before the signal has degraded, and together with the traditional approach, can be key enablers in the next generations of cellular networks.

From Reactive to Proactive Handovers. In this chapter, we began by exploring CHOs as a new form of proactive mobility control and showed how they can complement the (reactive) THOs that were extensively analyzed in the previous two chapters. In contrast to previous chapters that focused on the impact of source cell characteristics on HOs, we extended these datasets to include both source and target cell features, providing the necessary information to analyze how CHOs interact.

Based on these fresh insights, we proposed online meta-learning solutions aligned with O-RAN to optimize THO and CHO jointly. These models did not require prior knowledge of the conditions and demonstrated robust performance, as well as significant improvements in both real-world and synthetic scenarios. Having addressed the dual challenge of optimizing reactive and proactive HOs by associating users with cells optimally, it remained open how these cells / (virtualized) base stations should allocate their available resources to serve the users efficiently.

Chapter 5: Resource Allocation for Virtualized Base Stations in Non-Real-Time. With proactive and reactive mobility control in place, allocating users to the right base stations, attention naturally turned to how these base stations can learn to serve (i.e., allocate their resources) their users more intelligently. For that, in this chapter, we developed a novel control framework for vBS operating under O-RAN, where resource allocation non-real-time (i.e., threshold) policies, such as MCS and PRB, must be set at coarse time scales without access to future network conditions/environments. While we proved that our proposed algorithm outperformed the state-of-the-art in non-stationary or adversarial conditions, its learning process was slower in static/stationary conditions, precisely because of its robustness and cautiousness for the former conditions. For this purpose, we also proposed a meta-learning scheme that leveraged other available learners (possibly tailored for static/stationary conditions), by dynamically selecting the best-performing algorithm; thus enhancing the system's effectiveness and speed. Both of our solutions achieved strong theoretical static regret guarantees and demonstrated substantial energy savings under real-world conditions.

Brief Summary. This thesis provided a rigorous study of mobility and resource management in modern cellular networks. By leveraging multiple countrywide MNO datasets, our work focused on (traditional and conditional) handover optimization and resource allocation non-RT policies for virtualized base stations; two tightly linked problems. Altogether, the contributions of this thesis enhanced the robustness, adaptability, and intelligence of O-RAN control in both current and next generations of cellular networks.

6.4. Future Directions

6.4.1. Joint Handover-Resource Allocation Optimization

In this thesis, handover decisions (Chapters 2-4) and resource allocation (Chapter 5) were modeled and optimized as separate layers. However, they are deeply intertwined: the result of a handover affects the load at the target cell, while scheduling constraints may, in turn, influence the feasibility or quality of handovers, even when they succeed. Although the proposed models for optimizing handovers considered an important aspect of the cell in the decision-making process, namely its load, they relied on standard resource allocation schedulers, such as round-robin or proportional-fair. This means that the resources of each cell were divided equally among all connected users, regardless of their traffic demands or signal qualities.

A natural next step is to develop a learning-based control framework that, in addition to maximizing users' throughput and reducing the delays/costs associated

with traditional or conditional handovers while considering the cells' load, incorporates additional cell restrictions or schedulers. For instance, it could be beneficial to replace the "typical" schedulers of Chapters 3-4 with the scheduler proposed in Chapter 5; which, in turn, as a meta-learner, can incorporate different schedulers (i.e., learners) to choose from, based on unknown network conditions. In this way, we would also provide an end-to-end solution in the O-RAN architecture: from policies in non-RT RIC to handover decisions in near-RT RIC.

6.4.2. Extension to Non-Terrestrial Networks

The proposed frameworks have been developed and evaluated in terrestrial cellular settings, but the same principles are increasingly relevant for Non-Terrestrial Networks (NTNs), including LEO satellite constellations and hybrid terrestrial-satellite systems. NTNs, in conjunction with terrestrial networks (TNs), provide ubiquitous wireless access to an unprecedented number of users [202]. Nevertheless, they introduce even more intricate challenges than TNs, such as long propagation delays and changing topology due to satellite movement or cloud coverage, which result in extremely dynamic network conditions [203].

All these create an opportunity to apply the online meta-learning tools developed in this thesis, especially for dynamic UE-cell association and robust base station control, to the NTN domain. For example, our HO optimization frameworks, which make no assumptions about future dynamic network conditions, can be extended to the case of LEO satellites, offering robustness and adaptability without relying on stationary models or offline training. NTNs also open the door for additional research questions: how to incorporate the unique satellite mobility models into learning dynamics if these movements are predetermined or repeated; how to balance power-aware HO policies for low-battery IoT devices in remote areas; and how devices could switch between TN and NTN infrastructures for coverage and better quality of experience.

6.4.3. Joint Service Migration and Network Control

As edge computing becomes a key enabler for the increasing computation demands from the mobile users, a new dimension emerges: the co-migration of services and users [13], [204]. Given the resource-constrained devices of the users and new, diversified latency-sensitive applications, such as XR/AR or vehicles in intelligent transportation systems, it might no longer be sufficient to only hand over connectivity; the computational services bound to those connections must migrate in tandem. This introduces tightly coupled decision problems: when a user is handed over, should the associated service be migrated as well? And under what criteria?

Extending our framework to support joint optimization of service placement and handover/resource control is a rich and practically relevant direction. Meta-learning could be employed to balance migration overheads with gains in quality of service, while online control algorithms may operate over multi-timescale decisions, e.g., fast handovers coupled with slower service migration.

2A

Appendix of Chapter 2

2A.1. Ethics

The collected datasets are protected under Non-Disclosure Agreements (NDAs) that explicitly forbid the dissemination of information to unauthorized parties and public repositories. The procedures for data collection and storage within the network's infrastructure strictly follow the guidelines set forth by the MNO, and are in full compliance with local regulations. Moreover, while some metrics are computed on the user-level, our data-handling processes strictly focus on generating aggregated, anonymized insights, without access to the exact locations/trajectories of the users. No personal and/or contract information was available for this study and none of the authors participated in the extraction and/or encryption of the raw data. Ultimately, our datasets and research do not involve risks for the mobile subscribers, while they provide new knowledge about the dynamics of mobility management and handovers.

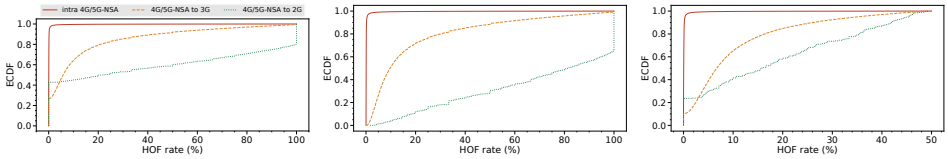
2A.2. Regression Analysis Details

Here, we complement the main regression models presented in Section 2.5.3 with additional models, which have comparable performance in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) with Random Forest (RF) [205]. The results are aligned and support the reported findings. We remind the reader that the analysis is performed on a dataset that records the daily percentage of failed HOs (i.e., HOF rate) during the studied 4-week interval.

We start by plotting the main statistics (boxplots with mean and median values) for the effect of HO type, antenna vendor, and cell area on the HOF rates. We also plot the ECDFs for the first two cases in Figure 2A.1, while the summary statistics can be seen in Table 2A.1. Performing a one-way ANOVA test we find that the effect of HO type on HOF rate is statistically significant and large ($F(2, 3857071) = 8.01{\cdot}10^6$, $p < .001$; $\eta^2 = 0.81, 95\%\text{CI}\ [0.82, 1.00]$), and Post-hoc pairwise comparisons

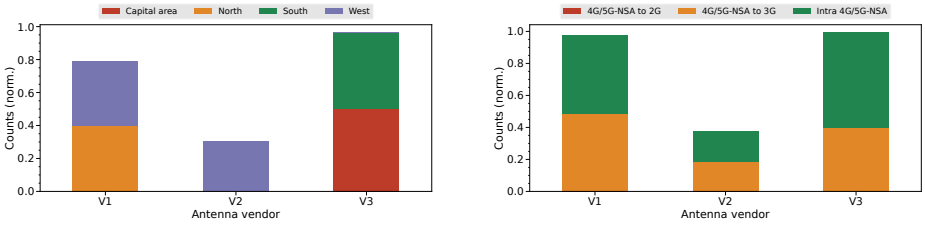
Table 2A.1: Summary Stats of Dataset.

Feature	Min	1st Qu	Median	Mean	3rd Qu	Max
Daily HOs	1	76	1989	6431	8591	953287
HOF rate	0.0	0.0	0.069	6.131	4.191	100.0



(a) (b) (c)

Figure 2A.1: ECDF of HOF rates for HO type: (a) all HOF rates; (b) non-zero HOF rates, (c) HOF rates without outliers.



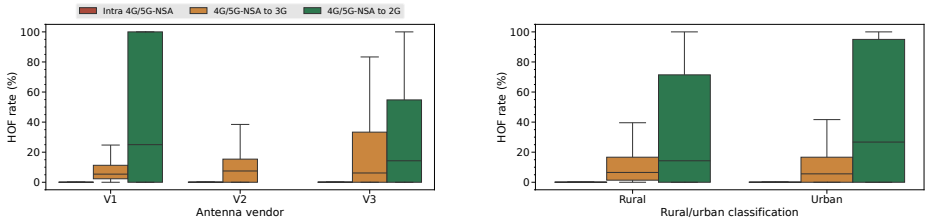
(a) (b)

Figure 2A.2: Antenna vendor per (a) region and (b) HO type.

(Tukey's HSD) verify that this effect is significant for all HO types. A Kruskal-Wallis test also supports this hypothesis ($p=0$).

Next, we turn our attention to the vendor of the source cell (i.e., antenna vendor). Due to confidentiality issues, we refer to the 4 vendors with the codes V1, V2, V3, and V4, instead of using their actual names. First, we note that different vendors are used in cells in different regions (North, South, West, Capital area), Figure 2A.2a; while all but one vendors are involved in similar proportions in intra 4G/5G-NSA HOs and HOs to 3G, Figure 2A.2. In Figure 2A.3a, we present the boxplots for the effect of the antenna vendor on HOF rates. In this case, we create one plot for each type of RAT and focus on HOF rates $< 1\%$ for 4G/5G-NSA, since the values are concentrated in the low-end of the spectrum. ANOVA tests for each HO type and for all HO types concurrently verify this effect is statistically significant but small $((F(3, 4911927) = 30524.85, p < .001; \eta^2 = 0.02, 95\%CI[0.02, 1.00]))$. Finally, Figure 2A.3b studies the effect of the area type, where this feature takes two values: rural and urban. We observe a small effect of the area type and indeed, performing an ANOVA test, we find it statistically significant but small ($F(2, 4664505) = 18559.77$, $p < .001$, $\eta^2 = 7.90 \cdot 10^{-3}$, $95\%CI[7.76 \cdot 10^{-3}, 1.00]$), even when we subset per HO type.

After this first level of analysis, we proceed with regression models that com-



(a) (b)

Figure 2A.3: Boxplots of HOF rates vs (a) antenna vendor and (b) urban/rural areas.

Table 2A.2: Regression Summary: Linear Model w/o 2G HOs.

Feature	Coeff.	Std Err	t value	$\Pr(>\|t\|)$
(Intercept)	-3.64	0.0185	-196	0
HO type: 4G/5G-NSA→3G	5.23	0.00120	4348	0
Number of daily HOs	$-1.02 \cdot 10^{-5}$	0	-215	0
Area Type: Rural	0.416	0.00273	153	0
Area Type: Urban	0.365	0.00259	141	0
Antenna Vendor: V2	0.0241	0.00166	14.5	0
Antenna Vendor: V3	1.00	0.0183	54.6	0
Antenna Vendor: V4	0.227	0.0199	11.4	0
Cell Region: North	-0.107	0.0184	-5.81	$6.14 \cdot 10^{-9}$
Cell Region: South	-0.0527	0.00160	-32.9	0
Cell Region: West	0.577	0.0184	31.5	0
District population	$-1.52 \cdot 10^{-7}$	0	-54.7	0
$N = 4892154$, RMSE=1.072901, $R^2 = 0.8502$, AIC=14571839				

plement those presented in Section 2.5.3. Table 2.5 reports the results for a linear regression model, after log-transforming the dependent variable and excluding outliers (i.e., removing entries with HOF rates exceeding 50%, less than 10 HOs per day or more than 30k HOs per day) that includes all main features of the dataset. In line with the simpler univariate model in Section 2.5.3, we see that the HO type remains the main contributing factor on HOF, even when accounting for all other covariates. On the other hand, the rest of the features are significant, yet have a much smaller, often negligible, effect. To further delineate the effect of the other covariates, we repeat the analysis after excluding HOs to 2G since they represent only 0.04% of dataset entries and are skewed towards much higher HOFs (see boxplots). The results are summarized in Table 2A.2 where we see that the HO type (only related to 3G in this case) is pronounced, the rural/urban feature is significant but the two values have a similar effect, as well as a significant and large effect of the vendor and the region (West). We note these latter findings (effect of Vendor V2 and West) remain significant even if we exclude the HOs to 2G and 3G, and regress only over the intra 4G/5G-NSA HOs.

As a final robustness test and based on the (near) bimodal distribution of the log-

Table 2A.3: Quantile Regression w/o Outliers.

Feature; Quantile	Coeff.	Std Err	t value	$\Pr(>\|t\|)$
(Intercept); $\tau = 0.2$	−3.59	0.00072	−5000.50	0
HO type: 4G/5G-NSA→2G	5.80	0.07401	78.37	0
HO type: 4G/5G-NSA→3G	4.86	0.00113	4297.03	0
(Intercept); $\tau = 0.4$	−2.99	0.00077	−3865.27	0
HO type: 4G/5G-NSA→2G	5.880	0.07951	73.95	0
HO type: 4G/5G-NSA→3G	4.79	0.00122	3935.15	0
(Intercept); $\tau = 0.6$	−2.56	0.00066	−3874.20	0
HO type: 4G/5G-NSA→2G	5.84	0.06822	85.74	0
HO type: 4G/5G-NSA→3G	4.83	0.00104	4632.57	0
(Intercept) $\tau = 0.8$	−2.09	0.00092	−2281.89	0
HO type: 4G/5G-NSA→2G	5.72	0.09450	60.57	0
HO type: 4G/5G-NSA→3G	4.97	0.00145	3437.48	0

Table 2A.4: Quantile Regression – All HOFs.

	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$	$\tau = 0.8$
(Intercept)	−3.62	−3.00	−2.58	−2.11
HO type: 4G/5G-NSA→2G	7.13	7.20	7.13	6.72
HO type: 4G/5G-NSA→3G	5.03	4.99	5.15	5.51

transformed HOF rate variable, we perform quantile regression on 5 intervals ($\tau \in \{0.2, 0.4, 0.6, 0.8\}$), using the HO type as the only feature. Table 2A.3 summarizes the results for the case we filter outliers as before, and Table 2A.4 presents the coefficients for the entire dataset of non-zero HOF rates. These results reinforce the findings of the previous models, verifying the significant and large effect of the HO type on HOFs across the entire spectrum of observed values.

3A

Appendix of Chapter 3

Proof of Lemma 3.1. The proof follows the rationale in [25, Th. 4], [120, Th. 3]. First, we bound the regret of each expert w.r.t. benchmark, and then the regret of the meta-learner w.r.t. any expert. W.l.o.g., we set $\gamma = 1$ and define:

$$s_t(\boldsymbol{x}) = \langle \nabla g_t(\boldsymbol{x}_t), \boldsymbol{x} - \boldsymbol{x}_t \rangle, \text{ and } \bar{\boldsymbol{x}}_{t+1}^k = \boldsymbol{x}_t^k + \theta_k \nabla g_t(\boldsymbol{x}_t).$$

Then, we can write: $s_t(\boldsymbol{x}_t^\star) - s_t(\boldsymbol{x}_t^k) = \langle \nabla g_t(\boldsymbol{x}_t), \boldsymbol{x}_t^\star - \boldsymbol{x}_t^k \rangle$

$$\begin{aligned}
&\overset{(\alpha)}{\leq} \frac{1}{2\theta_k}(\|\boldsymbol{x}_t^\star - \boldsymbol{x}_t^k\|_2^2 - \|\bar{\boldsymbol{x}}_{t+1}^k - \boldsymbol{x}_t^\star\|_2^2 + \theta_k^2 G^2 \\
&+ \|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star\|_2^2 - \|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star\|_2^2) \\
&\overset{(\beta)}{=} \frac{1}{2\theta_k}\big(\|\boldsymbol{x}_t^k - \boldsymbol{x}_t^\star\|_2^2 - \|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star\|_2^2 + \\
&+ (\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star + \boldsymbol{x}_{t+1}^k - \boldsymbol{x}_t^\star)^\top (\boldsymbol{x}_t^\star - \boldsymbol{x}_{t+1}^\star)\big) + \frac{\theta_k}{2} G^2 \\
&\overset{(\gamma)}{\leq} \frac{1}{2\theta_k}\left(\|\boldsymbol{x}_t^k - \boldsymbol{x}_t^\star\|_2^2 - \|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star\|_2^2\right) + \\
&+ \frac{D_{A*}}{\theta_k}\|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t+1}^\star\|_A + \frac{\theta_k}{2} G^2.
\end{aligned}$$

(α) uses the identity $\langle x, y \rangle = (\|x\|_2^2 + \|y\|_2^2 - \|x-y\|_2^2)/2$, the definition of $\bar{\boldsymbol{x}}_{t+1}^k$ and $\|\nabla g_t(\boldsymbol{x}_t)\|_2 \leq G$ as well as adds/subtracts $\|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_{t+1}^\star\|_2^2$; (β) uses the projection non-expansiveness and $\|x\|_2^2 - \|y\|_2^2 = (x-y)^\top (x+y)$; and (γ) uses Cauchy-Schwartz, triangle inequality, and D_{A*}. Telescoping to T and using $\|\boldsymbol{x}_1^k - \boldsymbol{x}_1^\star\|_A^2 \leq D_A$, gives

$$\sum_{t=1}^{T} s_t(\boldsymbol{x}_t^\star) - s_t(\boldsymbol{x}_t^k) \leq \frac{D_A}{2\theta_k} + \frac{D_{A*} P_T}{\theta_k} + \frac{\theta_k T G^2}{2}.$$

Next, we bound the switching cost of (any) expert k:

$$\sum_{t=1}^{T} \|\boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k\|_A = \sum_{t=0}^{T-1} \|\boldsymbol{x}_{t+1}^k - \boldsymbol{x}_t^k\|_A \leq$$
$$\sum_{t=0}^{T-1} \|\bar{\boldsymbol{x}}_{t+1}^k - \boldsymbol{x}_t^k\|_A = \sum_{t=0}^{T-1} \| - \theta_k \nabla g_t(\boldsymbol{x}_t)\|_A \leq \theta_k T G_A,$$

and combining with the previous result, we get:

$$\sum_{t=1}^{T} s_t(\boldsymbol{x}_t^\star) - \sum_{t=1}^{T} \left(s_t(\boldsymbol{x}_t^k) - \|\boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k\|_A \right) \leq$$
$$\frac{D_A^2}{2\theta_k} + \frac{D_{A*} P_T}{\theta_k} + \theta_k T \left(\frac{G^2}{2} + G_A \right). \tag{3A.1}$$

Next, we bound the gap of the meta-learner from all experts. Following [25, Lem. 3] and using the A-norm:

$$\|\boldsymbol{x}_t^m - \boldsymbol{x}_{t-1}^m\|_A = \Big\| \sum_{k\in\mathcal{K}} w_t^k \boldsymbol{x}_t^k - \sum_{k\in\mathcal{K}} w_{t-1}^k \boldsymbol{x}_{t-1}^k \Big\|_A$$
$$\leq \Big\| \sum_{k\in\mathcal{K}} w_t^k (\boldsymbol{x}_t^k - \boldsymbol{x}) - \sum_{k\in\mathcal{K}} w_t^k (\boldsymbol{x}_{t-1}^k - \boldsymbol{x}) \Big\|_A$$
$$+ \Big\| \sum_{k\in\mathcal{K}} w_t^k (\boldsymbol{x}_{t-1}^k - \boldsymbol{x}) - \sum_{k\in\mathcal{K}} w_{t-1}^k (\boldsymbol{x}_{t-1}^k - \boldsymbol{x}) \Big\|_A$$
$$\leq \sum_k w_t^k \left\| \boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k \right\|_A + |w_t^k - w_{t-1}^k| \left\| \boldsymbol{x}_{t-1}^k - \boldsymbol{x} \right\|_A$$
$$= \sum_{k\in\mathcal{K}} w_t^k \left\| \boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k \right\|_A + D_A \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1.$$

3A

The relative loss of meta-learner is: $\sum_{t=1}^{T} \left(s_t(\boldsymbol{x}_t^k) - \|\boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k\|_A \right) - \sum_{t=1}^{T} \left(s_t(\boldsymbol{x}_t^m) - \|\boldsymbol{x}_t^m - \boldsymbol{x}_{t-1}^m\|_A \right)$

$$\leq \sum_{t=1}^{T} \Big(\sum_k w_t^k \left\| \boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k \right\|_A + \langle \nabla g_t(\boldsymbol{x}_t), \boldsymbol{x}_t^k - \boldsymbol{x}_t \rangle$$
$$- \left\| \boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k \right\|_A + D_A \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1 \Big) =$$
$$\sum_{t=1}^{T} \left(\sum_k w_t^k \ell_t(\boldsymbol{x}_t^k) - \ell_t(\boldsymbol{x}_t^k) \right) + D_A \sum_{t=1}^{T} \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1.$$

The first term is bounded noting that $d \leq \ell_t(\boldsymbol{x}) \leq d + c$, with $d = -DG - D_A$, $c = 2GD + D_A$ and using the Hedge bound [23, Th. 2.2] (see also [120, Lem. 1]):

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^m) - \min_{k\in\mathcal{K}} \left(\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t^k) + \frac{1}{\beta} \ln \frac{1}{w_1^k} \right) \leq \frac{\beta T c^2}{8}.$$

Using the value of c and the definition of $\boldsymbol{x}_t^m$, we get:

$$\sum_{t=1}^{T}\left(\sum_k w_t^k \ell_t(\boldsymbol{x}_t^k) - \ell_t(\boldsymbol{x}_t^k)\right) \leq \frac{1}{\beta}\ln\frac{1}{w_1^k} + \frac{\beta T(2GD+D_A)^2}{8}.$$

Next, we use the strong convexity of entropic FTRL [206, Lem. 7], as the basis for Hedge, to arrive at:

$$\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1 \leq \beta \left\| \left[\ell_{t-1}(\boldsymbol{x}_t^k)\right]_k \right\|_\infty \leq \beta(GD+D_A).$$

Combining the above, we prove that $\forall k \in \mathcal{K}$, it is:

$$\sum_{t=1}^{T} s_t(\boldsymbol{x}_t^k) - \|\boldsymbol{x}_t^k - \boldsymbol{x}_{t-1}^k\|_A - s_t(\boldsymbol{x}_t^m) + \|\boldsymbol{x}_t^m - \boldsymbol{x}_{t-1}^m\|_A \leq \frac{1}{\beta}\ln\frac{1}{w_1^k} + \beta T \underbrace{\left[(2GD+D_A)^2(D_A+(1/8))\right]}_{\nu}. \tag{3A.2}$$

and set $\beta = 1/\sqrt{T\nu}$ to balance the RHS (omitting ln).

Finally, considering the max and min values of P_T, the expert step that minimizes the RHS of (3A.1) lies in:

$$\sqrt{\frac{D_A^2}{T(G^2+2G_A)}} \leq \theta^\star \leq \sqrt{\frac{D_A^2 + 2D_{A*}D_A T}{T(G^2+2G_A)}}$$

and inspecting the experts' steps $\{\theta_k\}_k$ in Lemma 3.1, we see that at least one expert step lies in that range. Thus, we obtain:

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{x}_t^m) \leq \sqrt{T}\Big[\sqrt{\nu}\left(1+\ln(1/w_1^k)\right) + (G^2+2G_A)^{1/2}(D_A^2+2D_{A*}P_T)^{1/2}\Big].$$

Proof of Theorem 3.2. It holds:

$$\mathbb{E}[\mathcal{R}_T] \leq \mathbb{E}\left[\sum_{t=1}^{T}\Big(f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{x}_t) + f_t(\boldsymbol{x}_t^m) - f_t(\boldsymbol{x}_t^m)\Big)\right]$$
$$= \mathbb{E}\left[\sum_{t=1}^{T} f_t(\boldsymbol{x}_t^m) - f_t(\boldsymbol{x}_t)\right] + \sum_{t=1}^{T}\Big(f_t(\boldsymbol{x}_t^\star) - f_t(\boldsymbol{x}_t^m)\Big).$$

The second term is bounded by Lemma 3.1. The first term is the discretization *error*, and we bound it as follows:

$$\mathbb{E}\left[\sum_{t=1}^{T}\Big(f_t(\boldsymbol{x}_t^m) - f_t(\boldsymbol{x}_t)\Big)\right] \overset{(\alpha)}{\leq} \sum_{t=1}^{T}\Big(G_f \mathbb{E}\left[|\boldsymbol{x}_t - \boldsymbol{x}_t^m|\right]\Big)$$

$$\overset{(\beta)}{\leq} G_f \sum_{t=1}^{T} \sqrt{\mathbb{E}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(x_{ij}(t) - \mathbb{E}[x_{ij}(t)]\Big)^2\right]}.$$

In (α) we used that expectation is a linear operator and the Lipschitz constant of f_t, and in (β) Jensen's inequality and the unbiased sampling. This is the variance of the random binary output of $Q_{\mathcal{X}}$. Since the binary vector is subject to a simplex per user, each sum w.r.t. j is bounded by the variance $1-(1/J)$, and the overall term by $\mu \triangleq \sqrt{I-(I/J)}$. Using (3A.2):

$$\mathbb{E}[\mathcal{R}_T] = G_f T\mu + \frac{1}{\beta}\ln\frac{1}{w_1^k} + \beta T\nu + (G^2 + 2G_A)^{1/2}(D_A^2 + 2D_{A*}P_T)^{1/2}.$$

4A

Appendix of Chapter 4

4A.1. Proof of Lemmas 4.2 and 4.3

Proof of Lemma 4.2. First, we observe that the diameter of the domain is bounded, i.e., $\|\boldsymbol{z}-\boldsymbol{z}'\|_2 \leq \|\boldsymbol{x}-\boldsymbol{x}'\|_2+\|\boldsymbol{y}'-\boldsymbol{y}'\|_2 \leq \sqrt{2I_{\text{THO}}} + \sqrt{I_{\text{CHO}}(J-1)} = D$, since the maximum distance for the former occurs when assignments $\boldsymbol{x}$ and $\boldsymbol{x}'$ are all different, and for the latter, when all cells are prepared in $\boldsymbol{y}$ and one cell is prepared per UE in $\boldsymbol{y}'$ (i.e., minimum preparation). Similarly, we prove the bound for the C_t-norm and its dual.

Finally, we compute the gradients component-wise and, for simplicity, we omit the time t index. For $i \in \mathcal{I}_{\text{THO}}, j \in \mathcal{J}$ (similarly for $i' \in \mathcal{I}_{\text{CHO}}$ and the $y_{i'j}$), and since $\partial \ell_j / \partial x_{ij} = 1$:

$$\frac{\partial \tilde{g}_t}{\partial x_{ij}} = \log c_{ij} - \log \ell_j - 1.$$

Using $c_n \leq c_{\max}$ and $\ell_j \leq I_{\text{THO}} + I_{\text{CHO}} = I$, each component can be bounded as:

$$\left| \frac{\partial g_t}{\partial x_{ij}} \right| \leq \max\left\{\log c_{\max} - 1,\ \log I + 1\right\} = M,$$

The proof is finalized, as there are $I\,J$ total components in the gradient vector; and likewise for the C_t-norm.

Proof of Lemma 4.3. The proof follows by tailoring the main results of [25], [120], [146], which, however, solve the UE-cell association problem (i.e., THO only). Eq. (4.16) can be rewritten as: $\mathbb{E}\left[\mathcal{R}_T\right] =$

$$\sum_{t=1}^{T}\Big(\big(\tilde{g}_t(\boldsymbol{z}_t^\star) - \|\boldsymbol{z}_t^\star - \boldsymbol{z}_{t-1}^\star\|_{C_t}\big) - \big(\tilde{g}_t(\boldsymbol{z}_t^{\text{m}}) - \|\boldsymbol{z}_t^{\text{m}} - \boldsymbol{z}_{t-1}^{\text{m}}\|_{C_t}\big)\Big)$$
$$+\mathbb{E}\left[\sum_{t=1}^{T}\Big(\big(\tilde{g}_t(\boldsymbol{z}_t^{\text{m}}) - \|\boldsymbol{z}_t^{\text{m}} - \boldsymbol{z}_{t-1}^{\text{m}}\|_{C_t}\big) - \big(\tilde{g}_t(\boldsymbol{z}_t) - \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_{C_t}\big)\Big)\right],$$

where we begin by bounding the expected dynamic regret of the relaxed (continuous) preparation decisions $\{\boldsymbol{z}_t^{\mathrm{m}}\}_t$ and, afterwards, the error of the (implementable) discrete decisions $\{\boldsymbol{z}_t\}_t$ as can be seen from the first and second terms, respectively. From the first term of the result in eq. (4.19), sublinear dynamic regret can be achieved for the relaxed preparation decisions. It follows by bounding the regret of each expert w.r.t. the benchmark and then the regret of the meta-learner w.r.t. any expert. For the extra cost/error introduced due to the discretization of the decisions through the quantization routine Q_z, we bound the term as follows:

$$\mathbb{E}\left[\sum_{t=1}^{T}\Big(\tilde{g}_t(\boldsymbol{z}_t^{\mathrm{m}})-\|\boldsymbol{z}_t^{\mathrm{m}}-\boldsymbol{z}_{t-1}^{\mathrm{m}}\|_{C_t}-\tilde{g}_t(\boldsymbol{z}_t)+\|\boldsymbol{z}_t-\boldsymbol{z}_{t-1}\|_{C_t}\Big)\right]\overset{(*)}{\leq}$$

$$\Big(G+\sqrt{a_{\max}+b_{\max}}\Big)\sum_{t=1}^{T}\sqrt{\mathbb{E}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(z_{ij}(t)-\mathbb{E}[z_{ij}(t)]\Big)^2\right]}\overset{(**)}{\leq}$$

$$T\Big(G+\sqrt{a_{\max}+b_{\max}}\Big)\sqrt{I_{\mathrm{CHO}}J/4+I_{\mathrm{THO}}(1-1/J)},$$

leading to eq. (4.19). Specifically, for $(*)$, we use Jensen's inequality, the linearity of expectation, and the Lipschitz continuity. For the latter, consider $\boldsymbol{z}_1,\boldsymbol{z}_2\in\mathcal{Z}^{\mathrm{c}}$, use the triangle inequality and the Lipschitz constant G of g_t and of the Euclidean norm, and define $\boldsymbol{z}_{t-1}=C$, to get:

$$\begin{aligned}&\Big|\big(\tilde{g}_t(\boldsymbol{z}_1)-\|\boldsymbol{z}_1-C\|_{C_t}\big)-\big(\tilde{g}_t(\boldsymbol{z}_2)-\|\boldsymbol{z}_2-C\|_{C_t}\big)\Big|\leq\\&\big|\tilde{g}_t(\boldsymbol{z}_1)-\tilde{g}_t(\boldsymbol{z}_2)\big|+\big|\|\boldsymbol{z}_1-C\|_{C_t}-\|\boldsymbol{z}_2-C\|_{C_t}\big|\leq\\&\Big(G+\sqrt{a_{\max}+b_{\max}}\Big)\|\boldsymbol{z}_1-\boldsymbol{z}_2\|.\end{aligned}$$

Lastly, we notice that $(**)$ calculates the variance of the binary $\boldsymbol{z}_t$. For $\boldsymbol{y}_t\in\{0,1\}^{I_{\mathrm{CHO}}\cdot J}$, the maximum variance each component can obtain is $1/4$ (the variance per element is $q(1-q)$ due to the Bernoulli trial with probability q, and maximizes for $q=0.5$); hence, we can upper bound the part for CHO-enabled UEs expression in the square root with the constant $I_{\mathrm{CHO}}J/4$. For $\boldsymbol{x}_t\in\{0,1\}^{I_{\mathrm{THO}}\cdot J}$, the binary vector is subject to a simplex per user; thus, each sum of THO-enabled users w.r.t. j in the square root is upper bounded by variance of $1-1/J$, and the overall part of the term in the square root has upper bound of $I_{\mathrm{THO}}(1-1/J)$.

4A.2. Conditional Handovers with a General Scheduler

Here, we study CHOs in the most general case, namely, with a general scheduler for the cell resources, and reintroduce the notation in the sequel. We denote by $x_{ij}(t)\in\{0,1\}$ the preparation decision: $x_{ij}(t)=1$ means that cell j is prepared in slot t for user i, and $x_{ij}(t)=0$ otherwise. Also, we define the vector $\boldsymbol{x}_t=\big(x_{ij}(t)\in$

$\{0,1\}, i \in \mathcal{I}, j \in \mathcal{J}\big)$, with the decision set:

$$\mathcal{X} = \Big\{ \boldsymbol{x} \in \{0,1\}^{I \cdot J} \;\Big|\; \sum_{j \in \mathcal{J}} x_{ij} \leq J, i \in \mathcal{I} \Big\},$$

and its convex hull $\mathcal{X}^c = \text{co}(\mathcal{X})$ that relaxes the integrality, i.e., $\boldsymbol{x} \in [0,1]^{I \cdot J}$. For $p_{ij}(t)$ being the probability that cell j will be the highest-rate cell for UE i during slot t, then this CHO will be realized with probability $p_{ij}(t)x_{ij}(t)$, where $\boldsymbol{p}_t = \big(p_{ij}(t) \in \{0,1\}, i \in \mathcal{I}, j \in \mathcal{J}\big)$ and $p_{ij}(t) = \mathbb{1}\{j = \arg\max_k \big(s_{ik}(t) - o_k(t)\big)\}$, with $o_j(t)$ being the cell-specific offsets that can vary in each slot t, resembling the offsets used in, e.g., the A3 event [46], [80], [81]. We also note that a user can be served from only one cell at a time, which is inferred from the definition of $\boldsymbol{p}$.

With these in mind, we introduce the *utility* function that the network controller wishes to maximize as follows:

$$g_t(\boldsymbol{x}) \triangleq \sum_{i=1}^{I} \sum_{j=1}^{J} \Big(x_{ij}(t) p_{ij}(t) u_j(t) \log c_{ij}(t) \\ - \beta_t \; x_{ij}(t)\big(1 - p_{ij}(t)\big) - \gamma_t \; p_{ij}(t)\big(1 - x_{ij}(t)\big) \Big), \boldsymbol{x} \in \mathcal{X}.$$

The first term of g_t defines the rate of a user, which is non-zero only for its served cell and is discounted by u_j due to exogenous (i.e., independent of the preparation decisions) effects; e.g., other UEs that executed traditional HO. The logarithmic transformation balances the sum-rate across all users to achieve fairness [33]; however, we note that other mappings (e.g., linear) can be used to capture the specifics of different applications. The second term of g_t introduces a penalty if cells other than the highest-rate ones are prepared to reduce resource waste, but it does not assume a proportional-fair or round-robin scheduler as in Section 4.4; the third term represents an additional cost if the highest-rate cell is not prepared. Moreover, we introduce explicit scalarization parameters β_t and γ_t to normalize units and prioritize one criterion over the other according to the preferences of each MNO that can even change over time or be different for each UE-cell pair due to different slices [207]. Implicitly, these parameters are incorporated inside the matrices $\boldsymbol{A}_t, \boldsymbol{B}_t$ and $\boldsymbol{C}_t$ in Section 4.4.

At the same time, the goal of the network controller is to minimize the signaling/switching overheads (or, similarly, maximize their negation), which are captured using $-\delta_t \; \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{B_t}$, with $\boldsymbol{x}_t, \boldsymbol{x}_{t-1} \in \mathcal{X}$. This presents the *switching cost* induced by the signaling of preparing and releasing cells (see Figure 4.3) scaled by $b_{ij}(t)$, as each cell j might have different costs for preparing and releasing cells for each user in slot t due to fluctuating traffic demands. More precisely, we define with $\boldsymbol{B}_t = \text{diag}(\boldsymbol{b}_n(t) > 0)$ a positive definite matrix which has on its diagonal the signaling weights $b_n(t) \in [0,1]$, $n = i \cdot j$ when UE i prepares and releases cell j, and $\|\cdot\|_{B_t}$ is its induced norm that can change over time t, i.e., $\|\boldsymbol{x}\|^2_{B_t} = \sum_n b_n(t) x_n^2$ and its dual $\|\boldsymbol{x}\|^2_{B_t *} = \sum_n x_n^2 / b_n(t)$ [117]. The role of the parameter δ_t is similar to β_t and γ_t. Thus, the overall problem that the network controller wishes to solve is:

$$\mathbb{P}_3 : \max_{\{\boldsymbol{x}_t\}_t} \sum_{t=1}^{T} \Big(g_t(\boldsymbol{x}_t) - \delta_t \; \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{B_t} \Big)$$

$$\text{s.t.} \quad \boldsymbol{x}_t \in \{0,1\}^{I\cdot J}, \ \forall t \in \mathcal{T},$$

where $g_t(\boldsymbol{x}_t) - \delta_t \, \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{B_t}$ is the *objective* function. Solving the problem at the beginning of the horizon T is challenging, as problems $\mathbb{P}_1$ and $\mathbb{P}_2$. To design an algorithm that is oblivious to the time-varying and unknown parameters and maximizes the users' rate, while keeping the signaling costs and the amount of wasted resources to a minimum, we follow the steps of Section 4.5 nearly verbatim. The bounds of the domain and gradients are bounded differently, namely, $\|\nabla g_t(\boldsymbol{x})\|_2 \leq G$, $\|\boldsymbol{x}-\boldsymbol{x}'\|_2 \leq D$, where $D = \sqrt{IJ}$ and $G = \sqrt{I(J-1)\beta_{\max}^2 + I(\log c_{\max} + \gamma_{\max})^2}$, respectively, with $\beta_{\max} = \max_{t\in\mathcal{T}}\{\beta_t\}$, $\gamma_{\max} = \max_{t\in\mathcal{T}}\{\gamma_t\}$, $c_{\max} = \max_{t\in\mathcal{T}}\{c_t\}$. Therefore, defining $b_{\max} = \max_{n\leq I\cdot J}\{b_n(t)\}$ for $t \in \mathcal{T}$, it holds that $\|\boldsymbol{x}-\boldsymbol{x}'\|_{B_t} \leq D\sqrt{b_{\max}} \triangleq D_B$, $\|\boldsymbol{x}-\boldsymbol{x}'\|_{B_t *} \leq D/\sqrt{b_{\max}} \triangleq D_{B*}$, and $\|\nabla g_t(\boldsymbol{x})\|_{B_t} \leq G\sqrt{b_{\max}} \triangleq G_B$.

Lemma 4A.1 (Performance Analysis / Optimality Guarantee)**.** *Similar optimality guarantees hold as Lemmas 4.2 and 4.3, but using the parameters:*

- $K = \lceil \log_2 \sqrt{1+2T} \rceil + 1$,
- $\theta_k = 2^{k-1}\sqrt{\frac{D_B^2}{T(G^2+2G_B)}}$, $k = 1, \ldots, K$,
- $\eta = 1/\sqrt{T\nu}$, *with* $\nu \triangleq (D_B+1/8)(GD+2D_B)^2$,
- $P_T = \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|_{B_t}$ *(path length).*

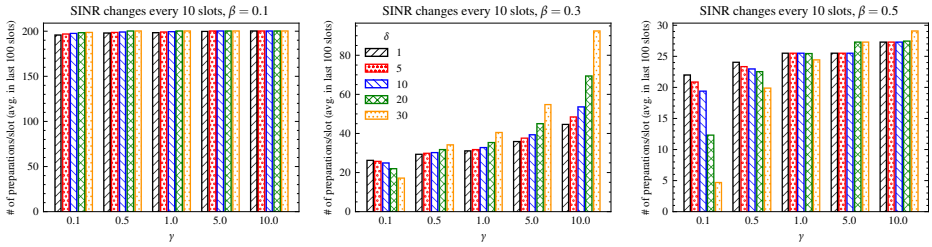
Then, the discrete decisions $\{\boldsymbol{x}_t\}_t$*, where* $\boldsymbol{x}_t \in \mathcal{X}$ *ensure:*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \sqrt{T}\Big(\sqrt{\nu}\left(1 + \ln(1/w_1^k)\right) + \sqrt{(G^2+2G_B)(D_B^2 + 2D_{B*}P_T)}\ \Big) + T(G+\sqrt{b_{\max}})\sqrt{IJ}/2.$$

The algorithm follows the same steps as `CONTRA`; but to diassociate them, we call the CHO-only algorithm `CHOMET` (Conditional HOs via METa-learning).

Performance Evaluation. We assess `CHOMET` under different synthetic scenarios to verify its robustness and showcase its learning convergence. For that, we compare our algorithm against the baseline 3GPP-compliant HO/CHO algorithm using the A3 event, which is the algorithm currently being used by MNOs and antenna vendors [39], [81]. We use the tuple notation `(# Best BS, TTT)` to refer to these comparators, where the first argument refers to the number of top-N best cells (i.e., highest SINR) that are prepared for each user in each slot, while the second is the time-to-trigger (TTT, i.e., number of consecutive slots a cell must have been in the top-N before is prepared), resembling the A3 event. For example, algorithm `(3, 8)` prepares the 3 highest-SINR cells for each user only if these cells remain the highest ones for at least 8 slots. Moreover, we compare `CHOMET` with an optimal `Oracle` that solves the optimization problem in every step using *CVXPY* [154]. We note that the comparison with the oracle that has complete knowledge of the future is computationally intensive for mixed-integer programs [36], [153]; however, our dynamic regret guarantees hold, as can be verified from our previous analysis.

In line with prior works [111], [146], we select two synthetic scenarios as follows, for $T = 5k$ slots: (i) *stationary*: the SINR $s_{ij}(t), i \in \mathcal{I}, j \in \mathcal{J}$ remains almost constant across all slots $t \in \mathcal{T}$, changing only once every 600 slots, and (ii) *volatile*: $s_{ij}(t)$ fluctuates every 10 slots within the range of [10, 30]dB [122], encompassing



(a) (b) (c)

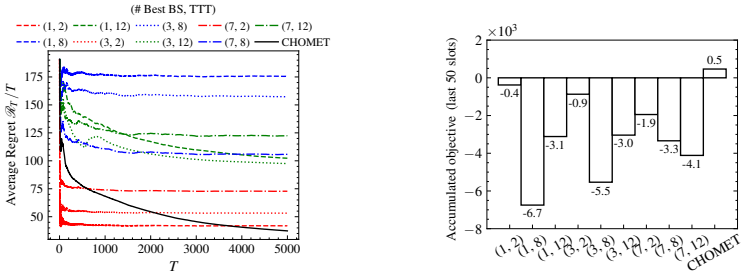
Figure 4A.1: Effect of parameters $\beta_t \equiv \beta, \gamma_t \equiv \gamma,$ and $\delta_t \equiv \delta$, for $T = 5k$ slots in the volatile scenario (SINR changes ever 10 slots) when running CHOMET: (a) $\beta = 0.1$, (b) $\beta = 0.3$, and (c) $\beta = 0.5$. Y-axis symbolizes the average number (last 100 slots) of preparations per slot.

poor to excellent values. In both scenarios, we randomly select the bandwidths $W_j \in \{5, 10, 15, 20\}$ MHz [123], while $\boldsymbol{B}_t$ takes random values within [0, 1], as the offsets $o_j(t), \forall j \in \mathcal{J}$. We select $I = 20$ UEs and $J = 10$ cells to facilitate the calculation of the average regret, as determining the best oracle is computationally intensive. However, we underline that the best oracle is not needed to run CHOMET.

First, Figure 4A.1 shows the effect of the parameters $\beta_t, \gamma_t,$ and δ_t on CHOMET, which, for simplicity, are assumed to be constant for the entire duration of the experiments, namely $\beta_t \equiv \beta, \gamma_t \equiv \gamma, \delta_t \equiv \delta,$ and $u_t = 1, \forall t \in \mathcal{T}$, where $T = 5k$ and SINR changes every 10 slots (volatile scenario). It is important to note that due to the different values that each component of the objective function takes, we focus on the comparison of the values within the same parameter (i.e, they act as scalarization values too). For example, $\beta = 0.5 > \gamma = 5$ does not imply that more importance is given to γ; however, choosing $\beta = 0.5$ instead of $\beta = 0.1$ does. We choose $\beta = \{0.1, 0.3, 0.5\}$, $\gamma = \{0.1, 0.5, 1, 5, 10\}$ and $\delta = \{1, 5, 10, 20, 30\}$. The y-axis of Figure 4A.1 shows the average number of preparations per slot, for the last 100 slots; due to considering 20 UEs and 10 cells, this number cannot exceed 200.

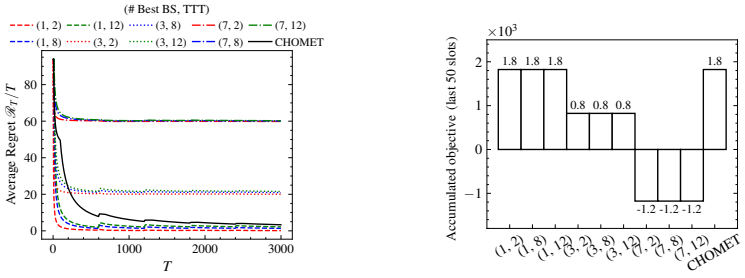
For a specific γ, the effect of β and δ on the number of prepared cells depends on the interplay of these parameters and should be carefully examined. If γ, the penalty for not preparing the best cell, is small (e.g., $\gamma = 0.1$) and β is high (e.g., $\beta = 0.5$ so fewer cells can be prepared), then as δ increases, the switching cost becomes more significant, making it less beneficial to prepare different cells to determine the best one. Consequently, the number of prepared cells *decreases*; e.g., setting $\beta = 0.5$ and $\gamma = 0.5$, we observe 24 and 19 preparations on average for $\delta = 1$ vs $\delta = 30$, respectively. Conversely, if γ is large, the penalty for not preparing the best cell is substantial, and as δ grows and switching becomes more costly, the algorithm decides to keep more cells prepared for more slots. As a result, the number of prepared cells *increases* even up to $\times 2$ times, as we can see for $\beta = 0.3$, $\gamma = 10$, and $\delta = 1$ vs $\delta = 30$.

For a fixed δ, the number of prepared cells *increases* with γ, as higher γ imposes a greater penalty for not preparing the highest-SINR cell. To mitigate this penalty, the algorithm prepares more cells to ensure that the best is found. This trend is consistent for all three β values considered, although a higher β limits the number of cells other than the highest-SINR that can be prepared, leading to differences in absolute values. For instance, our algorithm makes approximately 200 preparations



(a) (b)

Figure 4A.2: Volatile scenario (SINR changes ever 10 slots) for $\beta_t = 0.5, \gamma_t = 10,$ and $\delta_t = 5, \forall t \in \mathcal{T}$ with $T = 5k$ slots: (a) average dynamic regret and (b) total objective values for the last 50 slots, of `CHOMET` and benchmarks.



(a) (b)

Figure 4A.3: Stationary scenario (SINR changes every 600 slots) for $\beta_t = 0.5, \gamma_t = 10,$ and $\delta_t = 5, \forall t \in \mathcal{T}$ with $T = 3k$ slots: (a) average dynamic regret and (b) total objective values for the last 50 slots, of `CHOMET` and benchmarks.

on average per slot in Figure 4A.1a, meaning that all cells for all users are prepared in each slot, while we observe at most 29 preparations per slot (i.e., 1–2 cells per UE) in Figure 4A.1c; for that reason, lower values than $\beta = 0.1$ or higher than $\beta = 0.5$ are not considered in the evaluation. We underline that the interplay of these parameters also changes significantly depending on the conditions (i.e., SINR).

In the volatile case, Figure 4A.2a shows the average dynamic regret of our proposed algorithm and the 3GPP-compliant competitors, with the former surpassing all `(q, r)`, for $q = \{1, 3, 7\}$ and $r = \{8, 12\}$ (blue and green lines) by up to 375% in slot $t = 5k$. Even though at first glace competitors `(1, 2)`, `(3, 2)` and `(7, 2)` seem to have comparable performance with `CHOMET`, we underline that their average dynamic regret stays almost constant for all slots ("stuck" in sub-optimal decisions). This claim can be verified from Figure 4A.2b, where the accumulated objective in the last 50 slots of the best comparator, namely `(1, 2)`, is 180% less than `CHOMET`. Lastly, in a stationary (almost static) case, such as this of Figure 4A.3 where SINR changes very slowly (5 times in the total $T = 3k$), comparators `(1, 2)`, `(1, 8)` and `(1, 12)` behave similarly to `CHOMET` in terms of average regret and total objective values. This is reasonable, as preparing only the single highest-SINR cell (which is the one that the user is allocated) is the best policy to follow when conditions stay the same. Thus, we verify that `CHOMET` is adaptive to both volatile and stationary cases, approaching the behavior of an omniscient benchmark.

Bibliography

[1] G. L. Stüber and G. L. Steuber, *Principles of Mobile Communication*, 4th. Springer, 2017.

[2] M. Rahnema, "Overview Of the GSM System and Protocol Architecture", *IEEE Communications Magazine*, vol. 31, no. 4, pp. 92–100, 1993.

[3] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. United Kingdom: Wiley, 2002.

[4] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. United Kingdom: Wiley, 2011.

[5] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, *et al.*, "Cloud RAN for Mobile Networks—A Technology Overview", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[6] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges", *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[7] F. Haysom and R. Curran, "Open RAN Orchestration & Management Automation", Appledore Research, White Paper, 2022.

[8] O-RAN Software Community, `https://o-ran-sc.org/`.

[9] Small Cells Forum, "Small Cell Open RAN: A catalyst for new 5G business models", White Paper, 2022.

[10] O-RAN Alliance, *O-RAN Architecture-Description 6.0*, Technical Specification, 2022.

[11] O-RAN Alliance, *O-RAN Cloud Architecture and Deployment Scenarios for O-RAN vRAN 2.02 (O-RAN.WG6.CAD-v02.02)*, Technical Spec. 2021.

[12] A. Garcia-Saavedra and X. Costa-Pérez, "O-RAN: Disrupting the Virtualized RAN Ecosystem", *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 96–103, 2021.

[13] L. Niu, X. Chen, N. Zhang, Y. Zhu, R. Yin, *et al.*, "Multiagent Meta-Reinforcement Learning for Optimized Task Scheduling in Heterogeneous Edge Computing Systems", *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 519–10 531, 2023.

[14] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks", *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[15] M. Zinkevich, "Online Convex Programming and Generalized Infinitesimal Gradient Ascent", in *Proc. of ICML*, 2003.

[16] E. Hazan, "Introduction to Online Convex Optimization", *Foundations and Trends in Optimization*, vol. 2, no. 3–4, 2016.

[17] C. Márquez, M. Gramaglia, M. Fiore, A. Banchs, and Z. Smoreda, "Identifying Common Periodicities in Mobile Service Demands with Spectral Analysis", in *Proc. of MedComNet*, 2020.

[18] N. Littlestone, and M. Warmuth, "The Weighted Majority Algorithm", *Information & Comp.*, vol. 108, no. 2, 1994.

[19] A. Sani, G. Neu, and A. Lazaric, "Exploiting easy data in online optimization", in *Proc. of NeurIPS*, 2014.

[20] L. Chen, S. T. Jose, I. Nikoloska, S. Park, T. Chen, *et al.*, "Learning with Limited Samples: Meta-Learning and Applications to Communication Systems", *Foundations and Trends® in Signal Processing*, vol. 17, no. 2, pp. 79–208, 2023.

[21] N. Mhaisen, G. Iosifidis, and D. Leith, "Online Caching With no Regret: Optimistic Learning via Recommendations", *IEEE Trans. on Mobile Computing*, vol. 23, no. 5, 2024.

[22] A. Rakhlin and K. Sridharan, "Online Learning with Predictable Sequences", in *Proc. of COLT*, 2013.

[23] N. Cesa-Bianchi and G. Lugosi, "Prediction, Learning, and Games", *Cambridge University Press*, 2006.

[24] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.

[25] L. Zhang, W. Jiang, S. Lu, and T. Yang, "Revisiting Smoothed Online Learning", in *Proc. of NeurIPS*, 2021.

[26] H. Tabassum, M. Salehi, and E. Hossain, "Fundamentals of Mobility Aware Performance Characterization of Cellular Networks", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, 2019.

[27] S. Alraih, R. Nordin, A. Abu-Samah, I. Shayea, and N. F. Abdullah, "A Survey on Handover Optimization in B5G Networks: Challenges & Solutions", *IEEE Access*, vol. 11, 2023.

[28] M. Tayyab, X. Gelabert, and R. Jäntti, "A Survey on Handover Management: From LTE to NR", *IEEE Access*, vol. 7, pp. 118 907–118 930, 2019.

[29] X. Lin, R. K. Ganti, P. J. Fleming, and J. G. Andrews, "Towards Understanding the Fundamentals of Mobility in Cellular Networks", *IEEE Trans. on Wireless Communications*, vol. 12, no. 4, 2013.

[30] A. Hassan, A. Narayanan, A. Zhang, W. Ye, R. Zhu, *et al.*, "Vivisecting Mobility Management in 5G Cellular Networks", in *Proc. of SIGCOMM*, 2022.

[31] D. Zidic, T. Mastelic, I. Nizetic Kosovic, M. Cagalj, and J. Lorincz, "Analyses of Ping-Pong Handovers in Real 4G Telecommunication Networks", *Comput. Netw.*, vol. 227, no. C, 2023.

[32] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, *et al.*, "A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications", in *Proc. of SIGCOMM*, 2021.

[33] M. Gupta, S. Chinchali, P. P. Varkey, and J. G. Andrews, "Forecaster-Aided User Association and Load Balancing in Multi-Band Mobile Networks", *IEEE Trans. on Wireless Communications*, vol. 23, no. 5, 2024.

[34] M. Gupta, R. M. Dreifuerst, A. Yazdan, P.-H. Huang, S. Kasturia, *et al.*, "Load Balancing and Handover Optimization in Multi-band Networks using Deep Reinforcement Learning", in *Proc. of GLOBECOM*, 2021.

[35] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User Access Control in Open Radio Access Networks: A Federated Deep Reinforcement Learning Approach", *IEEE Trans. on Wireless Communications*, vol. 21, no. 6, pp. 3721–3736, 2022.

[36] A. Prado, F. Stöckeler, F. Mehmeti, P. Krämer, and W. Kellerer, "Enabling Proportionally-Fair Mobility Management With Reinforcement Learning in 5G Networks", *IEEE JSAC*, vol. 41, no. 6, 2023.

[37] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, *et al.*, "Programmable and Customized Intelligence for Traffic Steering in 5G Networks Using Open RAN Architectures", *IEEE Trans. on Mobile Computing*, vol. 23, no. 4, 2024.

[38] I. Panitsas, A. Mudvari, A. Maatouk, and L. Tassiulas, *Predictive Handover Strategy in 6G and Beyond: A Deep and Transfer Learning Approach*, 2024. arXiv: `2404.08113`.

[39] 3rd Generation Partnership Project (3GPP), "5G; NR; NR and NG-RAN Overall description; Stage-2", Technical Specification (TS) 38.300, 2024.

[40] Ericsson, *This is the Key to Mobility Robustness in 5G Networks*, [Online], `https://www.ericsson.com/en/blog/2020/5/the-key-to-mobility-robustness-5g-networks` [Accessed: 2025-06-01], 2020.

[41] S. Bin Iqbal, U. Karabulut, A. Awada, A. Noll Barreto, P. Schulz, *et al.*, "Mobility Performance Analysis of RACH Optimization Based on Decision Tree Supervised Learning for Conditional Handover in 5G Beamformed Networks", in *Proc. of EW*, 2023.

[42] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Performance Evaluation of the 5G NR Conditional Handover in LEO-based Non-Terrestrial Networks", in *Proc. of WCNC*, 2022.

[43] A. Prado, F. Mehmeti, and W. Kellerer, "Cost-Efficient Mobility Management in 5G", in *Proc. of WoWMoM*, 2023.

[44] J. Stanczak, U. Karabulut, and A. Awada, "Conditional Handover in 5G - Principles, Future Use Cases and FR2 Performance", in *Proc. of IWCMC*, 2022.

[45] S. Bin Iqbal, A. Awada, U. Karabulut, I. Viering, P. Schulz, *et al.*, "On the Modeling and Analysis of Fast Conditional Handover for 5G-Advanced", in *Proc. of PIMRC*, 2022.

[46] H. Martikainen, I. Viering, A. Lobinger, and T. Jokela, "On the Basics of Conditional Handover for 5G Mobility", in *Proc. of PIMRC*, 2018.

[47] W. Chen, P. Gaal, J. Montojo, and H. Zisimopoulos, *Fundamentals of 5G Communications: Connectivity for Enhanced Mobile Broadband and Beyond*, 1st. New York: McGraw Hill, 2021.

[48] V. W. S. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, *Key Technologies for 5G Wireless Systems*. Cambridge University Press, 2017.

[49] P. Jain, R. Borsato, P. Schmitt, *3GPP workshop on 6G; Chairman's Summary*, Mar. 2025. [Online]. Available: `https://www.3gpp.org/technologies/6gworkshop-2025`.

[50] H. Bergström, L. Bostrom, J. Sachs, S. Sorrentino, J. Vikberg, and R. Wang, *Dependable Networks: from Best-Effort to Guaranteed Performance*, Ericsson Technology Review, 2025.

[51] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, *et al.*, "Massive Access for 5G and Beyond", *IEEE JSAC*, vol. 39, no. 3, pp. 615–637, 2021.

[52] W. Chen, X. Lin, J. Lee, A. Toskala, S. Sun, *et al.*, "5G-Advanced Toward 6G: Past, Present, and Future", *IEEE JSAC*, vol. 41, no. 6, pp. 1592–1619, 2023.

[53] Vestel Electronics, *6GWS-250097: A Sensing-Mode Handoff Mechanism for High-Resolution and Accurate Sensing in 6G Networks*, Discussion, 6G Workshop, Mar. 2025.

[54] J. Vikberg, G. Hall, T. Cagenius, R. Wang, and J. Schultz, *Robustness Evolution: Building Robust Critical Networks with the 5G System*, Ericsson Technology Review, 2021.

[55] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, *et al.*, "SrsLTE: An Open-Source Platform for LTE Evolution and Experimentation", in *Proc. of WinTECH*, 2016.

[56] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, *et al.*, "OpenAirInterface: A Flexible Platform for 5G Research", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, 2014.

[57] A. Alnoman, G. H. S. Carvalho, A. Anpalagan, and I. Woungang, "Energy efficiency on fully cloudified mobile networks: Survey, challenges, and open issues", *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1271–1291, 2018.

[58] P. Rost, A. Maeder, M. C. Valenti, and S. Talarico, "Computationally Aware Sum-Rate Optimal Scheduling for Centralized Radio Access Networks", in *Proc. of GLOBECOM*, 2015.

[59] J. A. Ayala-Romero, I. Khalid, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Experimental Evaluation of Power Consumption in Virtualized Base Stations", in *Proc. of ICC*, 2021.

[60] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Bayesian Online Learning for Energy-Aware Resource Orchestration in Virtualized RANs", in *Proc. of INFOCOM*, 2021.

[61] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, *et al.*, "Vrain: A deep learning approach tailoring computing and radio resources in virtualized rans", in *Proc. of Mobicom*, 2019.

[62] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "EdgeBOL: Automating Energy-Savings for Mobile Edge AI", in *Proc. of CoNEXT*, 2021.

[63] H. Deng, C. Peng, A. Fida, J. Meng, and Y. C. Hu, "Mobility Support in Cellular Networks: A Measurement Study on Its Configurations and Implications", in *Proc. of IMC*, 2018.

[64] J. Wang, Y. Zheng, Y. Ni, C. Xu, F. Qian, *et al.*, "An Active-Passive Measurement Study of TCP Performance over LTE on High-Speed Rails", in *Proc. of MobiCom*, 2019.

[65] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, *et al.*, "Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption", in *Proc. of SIGCOMM*, 2020.

[66] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics", in *Proc. of MMSys*, 2018.

[67] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "TCP Performance over Mobile Networks in High-Speed Mobility Scenarios", in *Proc. of ICNP*, 2014.

[68] Y. Li, Q. Li, Z. Zhang, G. Baig, L. Qiu, *et al.*, "Beyond 5G: Reliable Extreme Mobility Management", in *Proc. of SIGCOMM*, 2020.

[69] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, *et al.*, "A First Look at Commercial 5G Performance on Smartphones", in *Proc. of WWW*, 2020.

[70] A. Narayanan, M. I. Rochman, A. Hassan, B. S. Firmansyah, V. Sathya, *et al.*, "A Comparative Measurement Study of Commercial 5G mmWave Deployments", in *Proc. of INFOCOM*, 2022.

[71] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond Throughput, The Next Generation: A 5G Dataset with Channel and Context Metrics", in *Proc. of MMSys*, 2020.

[72] M. Ghoshal, I. Khan, Z. J. Kong, P. Dinh, J. Meng, *et al.*, "Performance of Cellular Networks on the Wheels", in *Proc. of IMC*, 2023.

[73] Y. Li, H. Lin, Z. Li, Y. Liu, F. Qian, *et al.*, "A Nationwide Study on Cellular Reliability: Measurement, Analysis, and Enhancements", in *Proc. of SIGCOMM*, 2021.

[74] NGMN 5G Initiative, "5G White Paper, Version 2", NGMN, White Paper, 2020.

[75] M. Polese, M. Dohler, F. Dressler, M. Erol-Kantarci, R. Jana, *et al.*, "Empowering the 6G Cellular Architecture With Open RAN", *IEEE JSAC*, vol. 42, no. 2, pp. 245–262, 2024.

[76] A. Mahimkar, A. Sivakumar, Z. Ge, S. Pathak, and K. Biswas, "Auric: Using Data-driven Recommendation to Automatically Generate Cellular Configuration", in *Proc. of SIGCOMM*, 2021.

[77] G. Liu, Y. Huang, Z. Chen, L. Liu, Q. Wang, *et al.*, "5G Deployment: Standalone vs. Non-Standalone from the Operator Perspective", *IEEE Communications Magazine*, vol. 58, no. 11, pp. 83–89, 2020.

[78] 3rd Generation Partnership Project (3GPP), "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", Technical Specification (TS) 23.401, 2023.

[79] 3rd Generation Partnership Project (3GPP), "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)", Technical Specification (TS) 36.413, 2023.

[80] 3rd Generation Partnership Project (3GPP), "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification", Technical Specification (TS) 36.331, 2023.

[81] 3rd Generation Partnership Project (3GPP), "NR; Radio Resource Control (RRC); Protocol specification", Technical Specification (TS) 38.331, 2023.

[82] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. F. Kaijage, M. Kisangiri, *et al.*, "A Survey of Machine Learning Applications to Handover Management in 5G and Beyond", *IEEE Access*, vol. 9, pp. 45 770–45 802, 2021.

[83] 3rd Generation Partnership Project (3GPP), "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", Technical Specification (TS) 29.274, 2023.

[84] A. Lutu, B. Jun, A. Finamore, F. E. Bustamante, and D. Perino, "Where Things Roam: Uncovering Cellular IoT/M2M Connectivity", in *Proc. of IMC*, 2020.

[85] T. Bilen, B. Canberk, and K. R. Chowdhury, "Handover Management in Software-Defined Ultra-Dense 5G Networks", *IEEE Network*, vol. 31, no. 4, pp. 49–55, 2017.

[86] L. C. Gimenez, P. H. Michaelsen, K. I. Pedersen, T. E. Kolding, and H. C. Nguyen, "Towards Zero Data Interruption Time with Enhanced Synchronous Handover", in *Proc. of VTC*, 2017.

[87] D. Han, S. Shin, H. Cho, J.-m. Chung, D. Ok, *et al.*, "Measurement and Stochastic Modeling of Handover Delay and Interruption Time of Smartphone Real-Time Applications on LTE Networks", *IEEE Communications Magazine*, vol. 53, no. 3, pp. 173–181, 2015.

[88] L. C. Gimenez, M. C. Cascino, M. Stefan, K. I. Pedersen, and A. F. Cattoni, "Mobility Performance in Slow- and High-Speed LTE Real Scenarios", in *Proc. of VTC*, 2016.

[89] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns", *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[90] Ericsson, "Voice and SMS transformation following 2G/3G Sunset", Ericsson, White Paper, 2023.

[91] X. Yuan, M. Wu, Z. Wang, Y. Zhu, M. Ma, *et al.*, "Understanding 5G Performance for Real-World Services: A Content Provider's Perspective", in *Proc. of SIGCOMM*, 2022.

[92] R. Ahas, S. Silm, O. Järv, E. Saluveer, and M. Tiru, "Using Mobile Positioning Data to Model Locations Meaningful to Users of Mobile Phones", *Journal of Urban Technology*, vol. 17, no. 1, pp. 3–27, 2010.

[93] S. Phithakkitnukoon, Z. Smoreda, and P. Olivier, "Socio-Geography of Human Mobility: A Study Using Longitudinal Mobile Phone Data", *PLoS One*, vol. 7, no. 6, 2012.

[94] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications", *IEEE Access*, vol. 7, pp. 802–822, 2018.

[95] R. Tan, Y. Shi, Y. Fan, W. Zhu, and T. Wu, "Energy Saving Technologies and Best Practices for 5G Radio Access Network", *IEEE Access*, vol. 10, pp. 51 747–51 756, 2022.

[96] A. De Domenico, D. López-Pérez, W. Li, N. Piovesan, H. Bao, *et al.*, "Modeling User Transfer During Dynamic Carrier Shutdown in Green 5G Networks", *IEEE Trans. on Wireless Communications*, vol. 22, no. 8, pp. 5536–5549, 2023.

[97] 3rd Generation Partnership Project (3GPP), "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Single Radio Voice Call Continuity (SRVCC); Stage 2", Technical Specification (TS) 23.216, 2022.

[98] 3rd Generation Partnership Project (3GPP), " Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Organization of subscriber data", Technical Specification (TS) 23.008, 2022.

[99] Y. Ni, F. Qian, T. Liu, Y. Cheng, Z. Ma, *et al.*, "POLYCORN: Data-driven Cross-layer Multipath Networking for High-speed Railway through Composable Schedulerlets", in *Proc. of NSDI*, 2023.

[100] R. A. Fisher, *Statistical Methods for Research Workers*, 14th. Oliver and Boyd, 1970.

[101] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis", *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[102] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, *et al.*, "Mobileinsight: extracting and analyzing cellular network information on smartphones", in *Proc. of MobiCom*, 2016.

[103] G. Solutions, *G-NetTrack Pro*, `https://gyokovsolutions.com/g-nettrack/`, Accessed: 2024-05-15, 2024.

[104] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, *et al.*, "Lumos5G: Mapping and Predicting Commercial mmWave 5G Throughput", in *Proc. of IMC*, 2020.

[105] K. Kousias, M. Rajiullah, G. Caso, O. Alay, A. Brunstrom, *et al.*, "Implications of Handover Events in commercial 5G Non-Standalone Deployments in Rome", in *Proc. of SIGCOMM Workshop 5G-MeMU*, 2022.

[106] Z. Feher, A. Veres, and Z. Heszberger, "Ping-Pong Reduction Using Sub Cell Movement Detection", in *Proc. of VTC*, 2012.

[107] H. Deng, Q. Li, J. Huang, and C. Peng, "ICellSpeed: Increasing Cellular Data Speed with Device-Assisted Cell Selection", in *Proc. of MobiCom*, 2020.

[108] Z. Luo, S. Fu, M. Theis, S. Hasan, S. Ratnasamy, *et al.*, "Democratizing Cellular Access with CellBricks", in *Proc. of SIGCOMM*, 2021.

[109] J. Choi, W.-H. Lee, Y.-H. Kim, J.-H. Lee, and S.-C. Kim, "Throughput Estimation Based Distributed Base Station Selection in Heterogeneous Networks", *IEEE Trans. on Wireless Communications*, vol. 14, no. 11, 2015.

[110] N. Chen, G. Goel, and A. Wierman, "Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent", in *Proc. of COLT*, 2018.

[111] M. Kalntis, G. Iosifidis, and F. A. Kuipers, "Adaptive Resource Allocation for Virtualized Base Stations in O-RAN With Online Learning", *IEEE Trans. on Communications*, vol. 73, no. 3, pp. 1787–1800, 2025.

[112] M. Kalntis, and G. Iosifidis, "Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning", in *Proc. of GLOBECOM*, 2022.

[113] M. Kalntis, J. Suárez-Varela, J. O. Iglesias, A. K. Bhattacharjee, G. Iosifidis, *et al.*, "Through the Telco Lens: A Countrywide Empirical Study of Cellular Handovers", in *Proc. of IMC*, 2024.

[114] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas", *IEEE Trans. Information Theory*, vol. 48, no. 6, 2002.

[115] R. Srikant and L. Ying, "Communication Networks: An Optimization, Control and Stochastic Networks Perspective", *Cambridge University Press*, 2014.

[116] S. Arora, E. Hazan, and S. Kale, "The Multiplicative Weights Update Method: a Meta-algorithm and Applications", *Theory of Computing*, vol. 8, no. 6, 2012.

[117] A. Beck, "First-Order Methods in Optimization", *MOS-SIAM Series on Optimization*, 2017.

[118] N. Cesa-Bianchi, O. Dekel, and O. Shamir, "Online Learning with Switching Costs and Other Adaptive Adversaries", in *Proc. of NIPS*, 2013.

[119] W. G. Madow, "On the Theory of Systematic Sampling", *The Annals of Mathematical Statistics*, vol. 20, no. 3, 1949.

[120] L. Zhang, S. Lu, and Z.-H. Zhou, "Adaptive Online Learning in Dynamic Environments", in *Proc. of NeurIPS*, 2018.

[121] A. Lesage-Landry, J. A. Taylor, and D. S. Callaway, "Online Convex Optimization with Binary Constraints", *IEEE Trans. on Automatic Control*, vol. 66, no. 12, 2021.

[122] 3rd Generation Partnership Project (3GPP), "Requirements for support of radio resource management", Technical Specification (TS) 36.133, 2024.

[123] 3rd Generation Partnership Project (3GPP), "User Equipment (UE) radio transmission and reception", Technical Specification (TS) 36.101, 2024.

[124] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communications and Mobile Computing*, vol. 2, no. 5, 2002.

[125] C. Shen, C. Tekin, and M. van der Schaar, "A Non-Stochastic Learning Approach to Energy Efficient Mobility Management", *IEEE JSAC*, vol. 34, no. 12, 2016.

[126] C. Shen and M. van der Schaar, "A Learning Approach to Frequent Handover Mitigations in 3GPP Mobility Protocols", in *Proc. of WCNC*, 2017.

[127] Y. Zhou, C. Shen, and M. van der Schaar, "A Non-Stationary Online Learning Approach to Mobility Management", *IEEE Trans. on Wireless Communications*, vol. 18, no. 2, 2019.

[128] Y. Li, E. Datta, J. Ding, N. B. Shroff, and X. Liu, "Can Online Learning Increase the Reliability of Extreme Mobility Management?", in *Proc. of IEEE/ACM IWQoS*, 2021.

[129] L. Sun, J. Hou, and T. Shu, "Spatial and Temporal Contextual Multi-Armed Bandit Handovers in Ultra-Dense mmWave Cellular Networks", *IEEE Trans. on Mobile Computing*, vol. 20, no. 12, 2021.

[130] S. K. Singh, V. S. Borkar, and G. S. Kasbekar, "User Association in Dense mmWave Networks as Restless Bandits", *IEEE Trans. on Vehicular Technology*, vol. 71, no. 7, 2022.

[131] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover Control in Wireless Systems via Asynchronous Multiuser Deep Reinforcement Learning", *IEEE Internet of Things Journal*, vol. 5, no. 6, 2018.

[132] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, *et al.*, "Efficient Handover Mechanism for Radio Access Network Slicing by Exploiting Distributed Learning", *IEEE Trans. on Network & Service Management*, vol. 17, no. 4, 2020.

[133] A. Masri, T. Veijalainen, H. Martikainen, S. Mwanje, J. Ali-Tolppa, *et al.*, "Machine-Learning-Based Predictive Handover", in *Proc. of IFIP/IEEE IM*, 2021.

[134] J. J. Alcaraz, F. Losilla, A. Zanella, and M. Zorzi, "Model-Based Reinforcement Learning With Kernels for Resource Allocation in RAN Slices", *IEEE Trans. on Wireless Communications*, vol. 22, no. 1, 2023.

[135] S. Mukhopadhyay and A. Sinha, "Online Caching with Optimal Switching Regret", in *Proc. of ISIT*, 2021.

[136] G. S. Paschos, A. Destounis, and G. Iosifidis, "Online Convex Optimization for Caching Networks", *IEEE/ACM Trans. on Networking*, vol. 28, no. 2, pp. 625–638, 2020.

[137] J. Steiger, B. Li, B. Ji, and N. Lu, "Constrained Bandit Learning with Switching Costs for Wireless Networks", in *Proc. of INFOCOM*, 2023.

[138] A. Block, Y. Dagan, N. Golowich, and A. Rakhlin, "Smoothed Online Learning is as Easy as Statistical Learning", in *Proc. of COLT*, 2022.

[139] G. D. Celik and E. Modiano, "Scheduling in Networks With Time-Varying Channels and Reconfiguration Delay", *IEEE/ACM Trans. on Networking*, vol. 23, no. 1, 2015.

[140] 3rd Generation Partnership Project (3GPP), *Consecutive Conditional HO, R2-1909862*, 2019.

[141] S. B. Iqbal, S. Nadaf, A. Awada, U. Karabulut, P. Schulz, *et al.*, "On the Analysis and Optimization of Fast CHO With Hand Blockage for Mobility", *IEEE Access*, vol. 11, 2023.

[142] C. Ballesteros, A. Pfadler, L. Montero, J. Romeu, and L. Jofre-Roca, "Adaptive beamwidth optimization under doppler ici and positioning errors at mmwave bands", *Veh. Comm.*, vol. 34, p. 100 456, 2022.

[143] H. Elgendi, et al., "Uplink performance of lte and nr with high-speed trains", in *Proc. of IEEE VTC*, 2021, pp. 1–5.

[144] Y. S. Soh, T. Q. S. Quek, M. Kountouris, and H. Shin, "Energy Efficient Heterogeneous Cellular Networks", *IEEE JSAC*, vol. 31, no. 5, pp. 840–850, 2013.

[145] J. Dai, L. Li, R. Safavinejad, S. Mahboob, H. Chen, *et al.*, "O-RAN-Enabled Intelligent Network Slicing to Meet Service-Level Agreement (SLA)", *IEEE Trans. on Mobile Computing*, vol. 24, no. 2, pp. 890–906, 2025.

[146] M. Kalntis, A. Lutu, F. Kuipers, and G. Iosifidis, "Smooth Handovers via Smoothed Online Learning", in *Proc. of IEEE INFOCOM*, 2025.

[147] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[148] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[149] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions", in *Proc. of ICML*, 2008.

[150] G. S. Paschos, Georgios, A. Destounis, and G. Iosifidis, "Online convex optimization for caching networks", *IEEE/ACM Trans. on Networking*, vol. 28, no. 2, pp. 625–638, 2020.

[151] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning", *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992.

[152] J. Voigt, P. J. Gu, and R. M. Rost, "A deep rl-based approach for adaptive handover protocols", in *Proc. of SCC*, 2025.

[153] M. A. Bragin and E. L. Tucker, "Surrogate "Level-Based" Lagrangian Relaxation for mixed-integer linear programming", *Scientific Reports*, vol. 12, no. 1, p. 22 417, Dec. 2022.

[154] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization", *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[155] M. I. Saglam, "Conditional Handover for Non-Terrestrial Networks", in *Proc. of WINCOM*, 2023.

[156] L. Yang, X. Yang, and Z. Bu, "A Conditional Handover Strategy Based on Trajectory Prediction for High-Speed Terminals in LEO Satellite Networks", in *ICC Workshops*, 2024.

[157] J. Stańczak, U. Karabulut, and A. Awada, "Conditional Handover Modelling for Increased Contention Free Resource Use in 5G-Advanced", in *Proc. of PIMRC*, 2023.

[158] S. C. Sundararaju, S. Ramamoorthy, D. P. Basavaraj, and V. Phanindhar, "Advanced Conditional Handover in 5G and Beyond Using Q-Learning", in *Proc. of WCNC*, 2024.

[159] C. Fiandrino, D. J. Martínez-Villanueva, and J. Widmer, "A study on 5G performance and fast conditional handover for public transit systems", *Computer Communications*, vol. 209, pp. 499–512, 2023, ISSN: 0140-3664.

[160] J. Stanczak, U. Karabulut, A. Awada, and P. Spapis, "Signalling-efficient CFRA Resource Updating for Conditional Handover in 5G-Advanced", in *Proc. of CSCN*, 2023.

[161] A. Prado, H. Vijayaraghavan, and W. Kellerer, "ECHO: Enhanced Conditional Handover boosted by Trajectory Prediction", in *Proc. of GLOBECOM*, 2021.

[162] C. Lee, H. Cho, S. Song, and J.-M. Chung, "Prediction-Based Conditional Handover for 5G mm-Wave Networks: A Deep-Learning Approach", *IEEE Vehicular Tech. Mag.*, vol. 15, no. 1, pp. 54–62, 2020.

[163] H.-S. Park, H. Kim, C. Lee, and H. Lee, "Mobility Management Paradigm Shift: From Reactive to Proactive Handover Using AI/ML", *IEEE Network*, vol. 38, no. 2, pp. 18–25, 2024.

[164] M. Kalntis, G. Iosifidis, and F. A. Kuipers, "CHOMET: Conditional Handovers via Meta-Learning", in *Proc. of WiOpt*, 2025.

[165] A. Collet, A. Bazco-Nogueras, A. Banchs, and M. Fiore, "Automanager: A meta-learning model for network management from intertwined forecasts", in *Proc. of IEEE INFOCOM*, 2023.

[166] J. Zhang, Y. Yuan, G. Zheng, I. Krikidis, and K.-K. Wong, "Embedding Model-Based Fast Meta Learning for Downlink Beamforming Adaptation", *IEEE Trans. on Wireless Communications*, vol. 21, no. 1, pp. 149–162, 2022.

[167] Q. He, A. Moayyedi, G. Dán, G. P. Koudouridis, and P. Tengkvist, "A Meta-Learning Scheme for Adaptive Short-Term Network Traffic Prediction", *IEEE JSAC*, vol. 38, no. 10, pp. 2271–2283, 2020.

[168] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed Multi-Agent Meta Learning for Trajectory Design in Wireless Drone Networks", *IEEE JSAC*, vol. 39, no. 10, pp. 3177–3192, 2021.

[169] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, *et al.*, "Efficient Federated Meta-Learning Over Multi-Access Wireless Networks", *IEEE JSAC*, vol. 40, no. 5, pp. 1556–1570, 2022.

[170] A. Feriani, D. Wu, Y. T. Xu, J. Li, S. Jang, *et al.*, "Multiobjective Load Balancing for Multiband Downlink Cellular Networks: A Meta- Reinforcement Learning Approach", *IEEE JSAC*, vol. 40, no. 9, pp. 2614–2629, 2022.

[171] R. M. Sohaib, O. Onireti, K. Tan, Y. Sambo, R. Swash, *et al.*, "Meta-Transfer Learning-Based Handover Optimization for V2N Communication", *IEEE Trans. on Vehicular Technology*, vol. 73, no. 11, pp. 17 331–17 346, 2024.

[172] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO-RAN: Developing Machine Learning-Based xApps for Open RAN Closed-Loop Control on Programmable Experimental Platforms", *IEEE Trans. on Mobile Computing*, vol. 22, no. 10, pp. 5787–5800, 2023.

[173] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: technical misconceptions and business barriers", *IEEE Communications Magazine*, vol. 54, no. 8, 2016.

[174] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The Nonstochastic Multiarmed Bandit Problem", *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[175] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire, "Corralling a Band of Bandit Algorithms", in *Proc. of COLT*, 2017.

[176] A. Singla, H. Hassani, and A. Krause, "Learning to Interact With Learning Agents", *Proc. of AAAI*, 2018.

[177] M. Odalric and R. Munos, "Adaptive Bandits: Towards the best history-dependent strategy", in *Proc. of AISTATS*, 2011.

[178] 3rd Generation Partnership Project (3GPP), "Physical layer procedures", Technical Specification (TS) 36.213, 2024.

[179] S. Bubeck and N. Cesa-Bianchi, *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*, 2012.

[180] P. Rost, S. Talarico, and M. C. Valenti, "The Complexity–Rate Tradeoff of Centralized Radio Access Networks", *IEEE Trans. on Wireless Communications*, vol. 14, no. 11, 2015.

[181] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design", in *Proc. of ICML*, 2010.

[182] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem", *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[183] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. The MIT Press, 2018.

[184] S. Vakili, J. Scarlett, D.-S. Shiu, and A. Bernacchia, "Improved Convergence Rates for Sparse Approximation Methods in Kernel-Based Learning", in *Proc. of ICML*, 2022.

[185] D. Bega, A. Banchs, M. Gramaglia, X. Costa-Pérez, and P. Rost, "CARES: Computation-aware Scheduling in Virtualized Radio Access Networks", *IEEE Trans. on Wireless Communications*, vol. PP, pp. 1–1, Oct. 2018.

[186] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[187] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?", *IEEE Trans. on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[188] J. Alcaraz, J. Ayala Romero, J. Vales-Alonso, and F. Losilla-López, "Online Reinforcement Learning for Adaptive Interference Coordination", *Trans. on Emerging Telecommunications Technologies*, vol. 31, Oct. 2020.

[189] F. W. Murti, J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Pérez, and G. Iosifidis, "An Optimal Deployment Framework for Multi-Cloud Virtualized Radio Access Networks", *IEEE Trans. on Wireless Communications*, vol. 20, no. 4, pp. 2251–2265, 2021.

[190] A. Barto and S. Mahadevan, "Recent Advances in Hierarchical Reinforcement Learning", *Discrete Event Dynamic Systems: Theory and Applications*, vol. 13, Dec. 2002.

[191] B. Alt, T. Ballard, R. Steinmetz, H. Koeppl, and A. Rizk, "CBA: Contextual Quality Adaptation for Adaptive Bitrate Video Streaming", in *Proc. of INFOCOM*, 2019.

[192] J. Chuai, Z. Chen, G. Liu, X. Guo, X. Wang, *et al.*, "A Collaborative Learning Based Approach for Parameter Configuration of Cellular Networks", in *Proc. of INFOCOM*, 2019.

[193] M. Qureshi and C. Tekin, "Fast Learning for Dynamic Resource Allocation in AI-Enabled Radio Networks", *IEEE Trans. on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 95–110, 2020.

[194] L. Maggi, A. Valcarce, and J. Hoydis, "Bayesian Optimization for Radio Resource Management: Open Loop Power Control", *IEEE JSAC*, vol. 39, no. 7, pp. 1858–1871, 2021.

[195] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization", *Proc. of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[196] M. A. Abdul Careem and A. Dutta, "Real-time Prediction of Non-stationary Wireless Channels", *IEEE Trans. on Wireless Communications*, vol. 19, no. 12, pp. 7836–7850, 2020.

[197] S. Zarandi, A. Khalili, M. Rasti, and H. Tabassum, "Multi-Objective Energy Efficient Resource Allocation and User Association for In-Band Full Duplex Small-Cells", *IEEE Trans. on Green Communications and Networking*, vol. 4, no. 4, pp. 1048–1060, 2020.

[198] Y. Wu, F. Zhou, W. Wu, Q. Wu, R. Q. Hu, *et al.*, "Multi-Objective Optimization for Spectrum and Energy Efficiency Tradeoff in IRS-Assisted CRNs With NOMA", *IEEE Trans. on Wireless Communications*, vol. 21, no. 8, pp. 6627–6642, 2022.

[199] X. Qi, S. Khattak, A. Zaib, and I. Khan, "Energy Efficient Resource Allocation for 5G Heterogeneous Networks Using Genetic Algorithm", *IEEE Access*, vol. 9, pp. 160 510–160 520, 2021.

[200] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches", *IEEE Trans. on Wireless Communications*, vol. 19, no. 10, pp. 6255–6267, 2020.

[201] M. Tsampazi, S. D'Oro, M. Polese, L. Bonati, G. Poitau, *et al.*, "A Comparative Analysis of Deep Reinforcement Learning-Based xApps in O-RAN", in *Proc. of GLOBECOM*, 2023.

[202] 3rd Generation Partnership Project (3GPP), "Study on using satellite access in 5G", Technical Specification (TS) 22.822, 2018.

[203] A. A. R. Alsaeedy and E. K. P. Chong, "A Survey of Mobility Management in Non-Terrestrial 5G Networks: Power Constraints and Signaling Cost", *IEEE Access*, vol. 12, pp. 107 529–107 551, 2024.

[204] S. Li, N. Zhang, H. Chen, S. Lin, O. A. Dobre, *et al.*, "Joint Road Side Units Selection and Resource Allocation in Vehicular Edge Computing", *IEEE Trans. on Vehicular Technology*, vol. 70, no. 12, pp. 13 190–13 204, 2021.

[205] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[206] H. B. McMahan, "A survey of Algorithms and Analysis for Adaptive Online Learning", *Journal of Machine Learning Research*, vol. 18, no. 90, pp. 1–50, 2017.

[207] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, *et al.*, "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks", *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, 2017.

Acknowledgements

Looking back, “Mobility and Resource Management”, apart from technical aspects as the title of this thesis points out, proved to be important pillars of my PhD as a whole. As this journey comes to an end, it becomes clear that “mobility” was all about learning how to move, sometimes slow, sometimes fast, between ideas and expectations; while managing limited “resources”: time, energy, and mental capacity. The people and places acknowledged here shaped how I navigated this journey and, ultimately, made it one worth remembering.

Coming from a big city, the transition to Delft was not easy at first. The slower rhythm felt unfamiliar in the beginning, but very soon, I learned to appreciate it. When it is not raining, the city is genuinely beautiful, with long bike rides and canals around every corner. Even the bad weather became part of a routine that I am grateful for, a reminder that progress often happens without spectacle, in moments that feel unremarkable at the time. Alongside the city, I am grateful to TU Delft for providing an environment where my work could grow. Beyond infrastructure, it offered a place where ideas were always encouraged, and discussions were open.

While the city and the institution made the work possible, it was the people within them, and first and foremost my supervisor, George, who made the journey meaningful. From the very beginning, he showed me how a supervisor can create a remarkable PhD experience: by listening carefully, by giving honest feedback, by setting the bar high while remaining supportive, and by caring about progress without losing sight of the person behind the work. I still remember that after preparing the draft of my first work with him, he smiled and said, “Now comes the nice part: choosing where you want to travel.” George has a remarkable ability to make things feel amusing and light when needed, having, however, ensured that the work has met the highest academic standards. There were no few moments when I sent a draft to him, proud of my progress, and he returned it with “a few comments from his side”, as he would say; a phrase that meant most of the paper was annotated. I am grateful for the example he set as a mentor and person.

My deepest thanks also go to my promotor/supervisor Fernando, who shaped the way I approached research. Apart from his academic integrity, what stood out most was his guidance and trust. I was given responsibility early on, along with the space to grow into it. I learned how to think critically about my own work and how to stand behind my results. This dissertation exists because of his support, honesty, clarity, and ability to ask the right questions at the right time. I am grateful for having the chance to work alongside him.

I am also especially thankful to Andra, my collaborator at Telefónica. She welcomed me for a research visit that turned out to be far more impactful than I initially imagined, extending our collaboration from a few months to approximately 2 years. She placed trust in me and gave me freedom to explore ideas. Moreover, I

would like to extend my sincere thanks to José and Jesús from Telefónica for their constant availability, ideas, excitement, and (sometimes) sleepless nights close to important deadlines. Working with Andra, José, and Jesús proved how rewarding collaborative research can be.

From my group in TU Delft, I am especially thankful to my friend Naram. From the beginning and increasingly as time passed, our academic discussions helped shape the way I thought about problems and ideas, more than he has probably realized. Beyond research and during breaks (despite separate offices at the end), travels (sometimes like "kings"), or while discovering new restaurants, we constantly exchanged stories and viewpoints, enriching each other's understanding of the world. This PhD would have been very different without him. I would also like to sincerely thank Adrian, Anup, Fatih, Gabe, and Kees (valuable help for the cover) from my group at TU Delft. Our shared coffee breaks, (un)common lunches, and casual conversations made the days truly enjoyable and memorable.

In addition, I am thankful for my lifelong friends Akis, Dimitris, Giorgos, Kimonas, and Orestis, with whom I share more than 23 years of memories. Distance has, unfortunately, separated us over time, but it never really changed anything that mattered. Knowing that you are there has been grounding throughout these years. I am equally grateful for my dear ECE-NTUA friends Aggelos, Dimitris, Eleni, Elissaios, Giannis, Giorgos, Kostas, Mary, Nikos, and Vasiliana. We eventually scattered across different continents, but the connection and support are there, remembering always who we were when everything was just beginning. To the friends I met in Delft, Christiana, Giannis, Giorgos, Marina, Najib, Sofia, Stelios, and Thanos, thank you for making these years rich. From day-one in Delft, sharing conversations and everyday moments outside the university are a large part of why this period of my life will always stand out.

Before closing, I would like to extend my gratitude to some people who have been the constant foundation beneath everything I have done. There are no words to describe the amount of unwavering support and love that my mother, Adriana, my father, Dimitris, my brother, Nikos, and my grandmother, Dimitra, have shown me. Being away from home meant missing everyday moments that, unfortunately, cannot be recovered. Still, their presence never felt far. Even our dog, Nala, patiently waiting at home, represents the comfort and familiarity I always return to. Their sacrifices and steady presence since the day I was born made it possible for me to grow and become the person I am today. Simultaneously, this PhD journey held surprises. Some of the most important moments in life arrive without any warning. A simple day trip to Antwerp and Rotterdam was one of those cases. It was then that I met my girlfriend, Eline, who has since been present in all moments of my life. She could sense when things were not quite right, and she supported me unconditionally in countless ways. Her presence brought balance and stability to a process that can often feel demanding and inward-looking. I am deeply grateful for her love, patience, energy, smartness, and the life we are building together. She was not only the most important part of this journey, but the person with whom I am excited to share everything that comes next.

List of Publications

6. **M. Kalntis**, G. Iosifidis, J. Suárez-Varela, A. Lutu, and F. A. Kuipers, "Meta-Learning-Based Handover Management in NextG O-RAN," in *IEEE Journal on Selected Areas in Communications (JSAC)*, 2026.

5. **M. Kalntis**, F. A. Kuipers, and G. Iosifidis, "CHOMET: Conditional Handovers via Meta-Learning," in *Proc. of International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2025.

4. **M. Kalntis**, A. Lutu, J. O. Iglesias, F. A. Kuipers, and G. Iosifidis, "Smooth Handovers via Smoothed Online Learning," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2025.

3. **M. Kalntis**, G. Iosifidis, and F. A. Kuipers, "Adaptive Resource Allocation for Virtualized Base Stations in O-RAN with Online Learning," in *IEEE Trans. on Communications (TCOM)*, vol. 73, no. 3, pp. 1787–1800, 2025.

2. **M. Kalntis**, J. Suárez-Varela, J. O. Iglesias, A. K. Bhattacharjee, G. Iosifidis, F. A. Kuipers, and A. Lutu, "Through the Telco Lens: A Countrywide Empirical Study of Cellular Handovers," in *Proc. of ACM Internet Measurement Conference (IMC)*, 2024.

1. **M. Kalntis** and G. Iosifidis, "Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2022.