User-Centric Slicing with Functional Splits in 5G Cloud-RAN

Salma Matoussi

DOCTOR OF PHILOSOPHY
SORBONNE UNIVERSITÉ

Specialization

Computer Science & Telecommunications

École doctorale Informatique, Télécommunications et Électronique de Paris

Presented by

Ms Salma MATOUSSI

Title

User-Centric Slicing with Functional Splits in 5G Cloud-RAN

Defended on January 22, 2021, in front of the committee composed of:

Mr Marceau COUPECHOUX	Reviewer	Full Professor, TELECOM Paris, IPP
Mr Thierry TURLETTI	Reviewer	Research Director, INRIA
Mr Tijani CHAHED	Examiner	Full Professor, TELECOM SudParis
Mr Marcelo DIAS DE AMORIM	Examiner	Research Director, Sorbonne University
Mr Adlen KSENTINI	Examiner	Full Professor, Eurecom, Sophia Antipolis
Mr Nadjib AITSAADI	Examiner	Full Professor, UVSQ Paris-Saclay
Ms Ilhem FAJJARI	Advisor	Researcher, Orange Labs, Châtillon
Mr Rami LANGAR	Supervisor	Full Professor, Gustave Eiffel University

ACKNOWLEDGEMENTS

I would like to thank my supervisor Pr. Rami Langar, who has given me the opportunity to do this thesis. He helped me in doing a lot of Research and I came to know about many new things. I would also like to show my gratitude to my co-supervisor Pr. Nadjib Aitsaadi, for his assistance and dedicated involvement in every step throughout the process. His enthusiasm and wonderful energy for work is contagious.

I have been extremely lucky to have Dr. Ilhem Fajjari as an advisor. She is my primary resource for getting my science questions answered. I am especially indebted for her advice and insightful discussions and suggestions. I am continually amazed by her integrity, professionalism and passion for science.

I would also like to thank all the jury members for reading my thesis, for offering their valuable time and for providing constructive feedback.

My sincere thanks go to my former college Salvatore Costanzo for his valuable support, encouragement and advice. The impact of his work on my dissertation is obvious.

I will forever be thankful to my friends and colleagues in PHARE, MLIA and SoC teams from LIP6, Sorbonne University, SNR/LRT from LIGM, Gustave Eiffel University and LiSSi, University of Paris-Est Créteil. They have been so supportive and helpful along the way of doing my thesis. Thanks for all the great times that we have shared in these last years.

Getting through my dissertation required more than an academic support, and I have many people to thank for listening to and, at times, having to tolerate me over the past years. I express my gratitude and appreciation especially to my friend Nour.

I must express my gratitude to my wonderful family and husband Ramy. This dissertation stands as a testament to them unconditional love and encouragement. A very special word of thanks goes for my parents; this thesis is dedicated to them.

ABSTRACT

5G Radio Access Network (RAN) aims to evolve new technologies spanning the Cloud infrastructure, virtualization techniques and Software Defined Network capabilities. Advanced solutions are introduced to split the RAN functions between centralized and distributed locations to improve the RAN flexibility. However, one of the major concerns is to efficiently allocate RAN resources, while supporting heterogeneous 5G service requirements.

In this thesis, we address the problematic of the user-centric RAN slice provisioning, within a Cloud RAN infrastructure enabling flexible functional splits. Our research aims to jointly meet the end users' requirements, while minimizing the deployment cost. The problem is NP-hard. To overcome the great complexity involved, we propose a number of heuristic provisioning strategies and we tackle the problem on four stages. First, we propose a new implementation of a cost efficient C-RAN architecture, enabling on-demand deployment of RAN resources, denoted by `AgilRAN`. Second, we consider the network function placement sub-problem and propound a new scalable user-centric functional split selection strategy named `SPLIT-HPSO`. Third, we integrate the radio resource allocation scheme in the functional split selection optimization approach. To do so, we propose a new heuristic based on Swarm Particle Optimization and Dijkstra approaches, so called `E2E-USA`. In the fourth stage, we consider a deep learning based approach for user-centric RAN Slice Allocation scheme, so called `DL-USA`, to operate in real-time. The results obtained prove the efficiency of our proposed strategies.

Keywords :

Cloud Radio Access Network (C-RAN), 3GPP Functional Split, Radio Resource Allocation, 5G RAN Slicing, Software Defined RAN (SD-RAN), ETSI-NFVI, C-RAN resource orchestration, Multi-objective optimization, Machine Learning.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

Contents

Mobile broadband services have notably proliferated over the last few years leading to the democratization of smart devices of which the number has grown into billions [1]. In order to meet such a huge demand, next generation mobile networks need to support a scaling system capacity. This issue is especially relevant for the Radio Access Network (RAN), which is considered as the most resource-demanding part of mobile networks [2].

In this context, the Cloud RAN (C-RAN) architecture [2] has been proposed as a promising technology, aiming at leveraging the Cloud infrastructure capabilities to make the RAN ecosystem more agile. Indeed, the virtualization of network functions is the cornerstone of a successful on-demand resource

deployment. With this scope, C-RAN is expected to respond to User Equipment's (UE) demands, while reducing the energy consumption and the resource provisioning cost [2].

In spite of its great success, several issues surrounding C-RAN remain open such as: the increasing bandwidth demand on the transport connection [3], virtualization of RAN functions [4], Cloud resource management [5], management of radio resources [6] and implementation of real-time processing algorithms [7]. Thereby, C-RAN should leverage the Cloud infrastructure, while considering the stringent UE's requirements and the specificity of RAN applications. To overcome the aforementioned issues, C-RAN needs to enable the deployment of flexible and scaling RAN solutions. In this thesis, we address the problematic of optimizing the resource provisioning within the Cloud Radio Access Network to fulfill UE QoS, while reducing the deployment cost.

This Chapter is organized as follows. First, the concept of slicing within the 5G Radio Access Network is introduced. Secondly, we put forward the Cloud Radio Access Network (C-RAN) as a promising 5G RAN architecture. Thirdly, we detail the major challenges of C-RAN applications. Finally, we summarize our contributing work addressing the aforementioned issues.

1.1 Slicing in 5G Radio Access Network

It is undeniable that the number of connected wireless devices accessing mobile networks is considered as one of the primary contributors to the ever increasing global mobile traffic. It is expected that the overall number of connections will increase from 8.8 billion in 2018 to 13.1 billion by 2023, with 10.6% of new 5G devices and 14.4% of Low-Power Wide-Area (LPWA) connections [1]. Figure 1.1 shows the connection growth of each mobile generation forecasted by Cisco in 2020. Besides, it is foreseen that, by 2023, 5G will generate a traffic of nearly seven fold increase over 2019, while the traffic of former generations is evolving without a significant increase. Figure 1.2 shows the global mobile traffic trends per connection between 2018 and 2023.

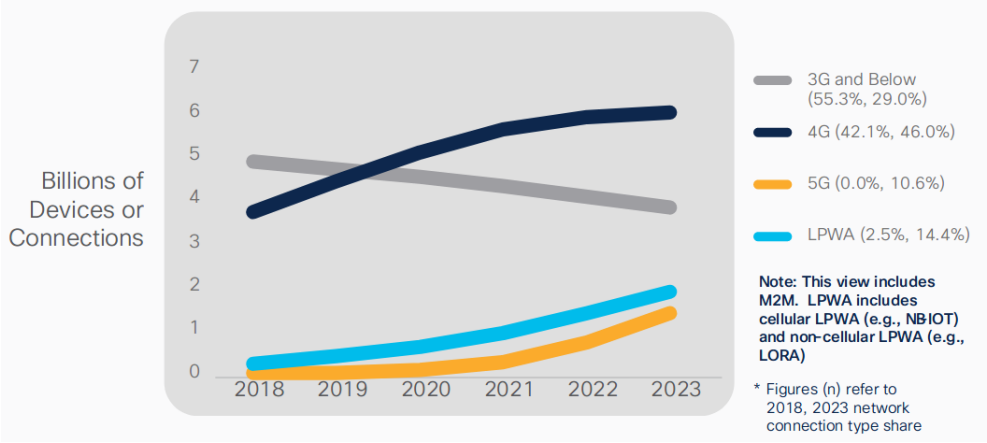


Figure 1.1: Global mobile device and connection growth [1]

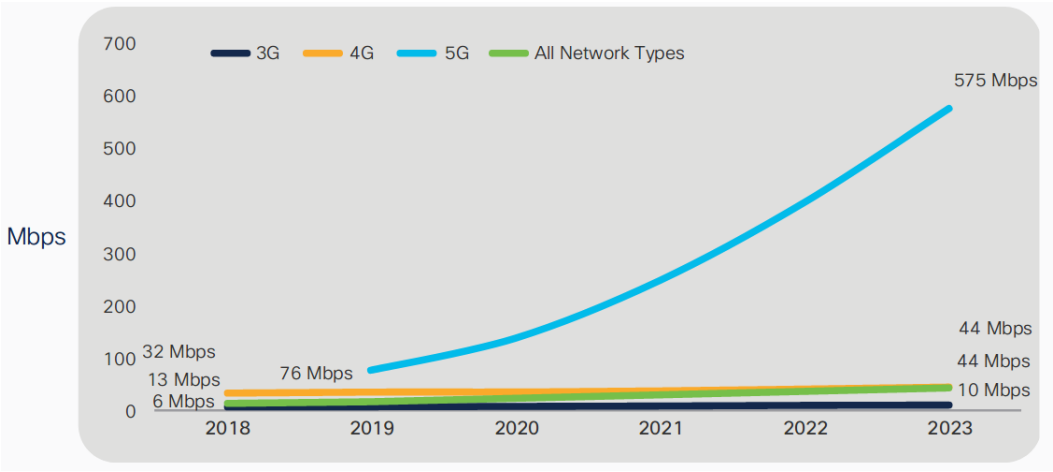


Figure 1.2: Global mobile traffic trends per connection [1]

Additionally, a major evolution of traffic characteristics is being noticed. Indeed, three key use-case categories are emerging with different throughput and latency requirements: enhanced Mobile Broadband (eMBB), ultra-Reliable Low Latency Communication (uRLLC) and massive Machine Type Communications (mMTC) [8]. eMBB is put forward for data-intensive applications and requires high data rates of several giga bits per second with moderate latency of few milliseconds. uRLLC supports ultra-reliable low latency communications in the order of 1 millisecond. mMTC supports smart cities and logistic applications with high connection density and energy efficiency [9]. Figure 1.3 summarizes the 5G requirements per use case, issued by the International Telecommunication Union (ITU) in 2015.

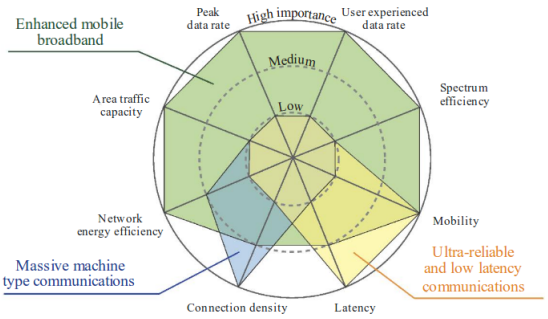


Figure 1.3: 5G capabilities for different use cases [10]

In this context, the network slicing concept [11] has emerged with the idea of enabling the network architecture to deliver on-demand end-to-end allocated resources (so called slice) as per service requirement. Within this perspective, multiple slices can be created on the same RAN infrastructure to convey services that have different requirements for latency, reliability and throughput.

1.2 Cloud RAN

Mobile Network Operators (MNOs) are expected to integrate major changes in their cellular communications beyond the new radio and wider spectrum. The objective is to build a flexible and cost efficient mobile network to cope with the exponential traffic growth and convey services for heterogeneous use case requirements.

1.2.1 Motivations

Currently, with up to 95% of network processes performed manually [1], MNOs may be outpaced by this rapid growth of data, struggling hence, to deploy a scalable network architecture supporting the slicing concept. As the underlying network infrastructure becomes more complex, it is foreseen that the OPerational EXpenditure (OPEX) will be two to three times higher than the CAPital EXpenditure (CAPEX) [1]. Thereby, automation is essential to efficiently operate and reduce the OPEX budget, while optimizing the return on investment as well as cutting time to market. This issue is especially relevant for the Radio Access Network (RAN), which is considered as the costliest and the most resource-demanding part of mobile networks [2].

Most of MNO budget is related to the building of RAN sites with almost 80% of CAPEX [2]. Figure 1.4 shows that more than 50% of RAN CAPEX budget are reserved to access site hardware and software with their power support and air conditioning equipments. Besides, an analysis of [2] shows that electricity and RAN operation & maintenance tasks account for over 34% of the total network OPEX.

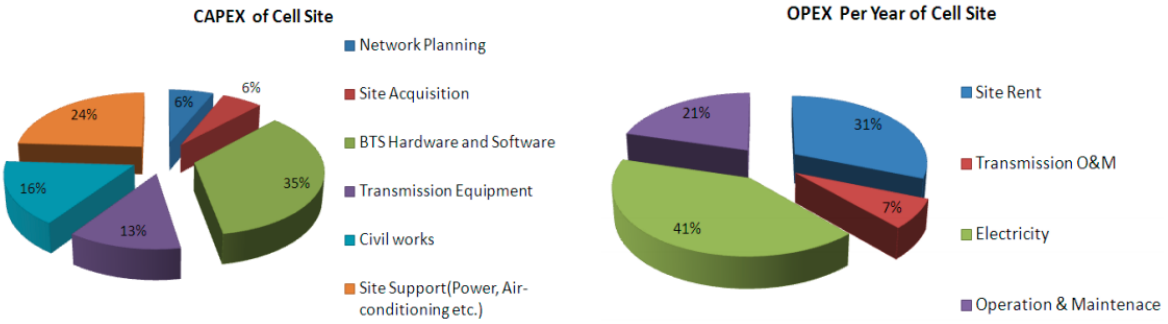


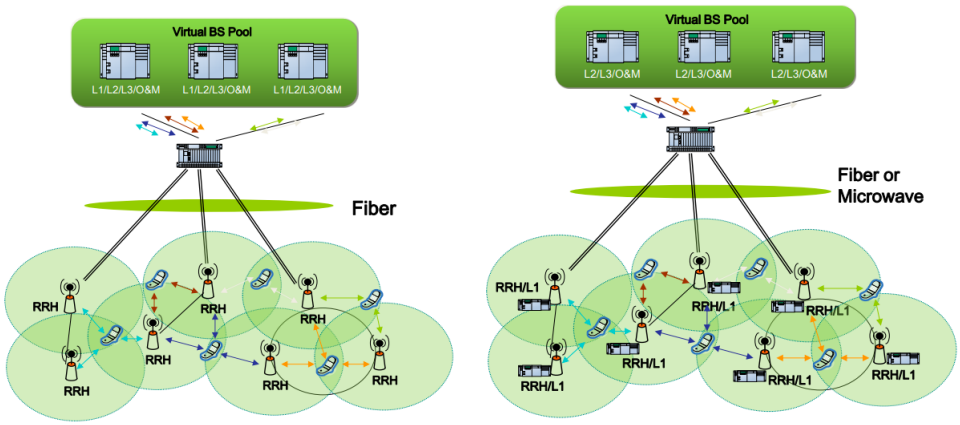
Figure 1.4: CAPEX and OPEX analysis of a traditional RAN network [2]

Therefore, it is crucial for MNOs to rethink the RAN architecture in order to lower the cost-per-bit investments, while making the RAN eco-friendly. To face the aforementioned challenges, the Cloud RAN (C-RAN) architecture has been proposed in 2011 [2], aiming at leveraging the Cloud infrastructure capabilities to operate efficiently the RAN resources.

1.2.2 Architecture

C-RAN fosters the virtualization of BaseBand signal processing Units (BBUs), which are traditionally located in a data unit near cell towers at the access sites. In doing so, BBUs are executed in a remote

Cloud site, leaving simple radio units known as Remote Radio Heads (RRHs) at the access site. RRHs and BBUs are connected via a fronthaul link. With this scope, C-RAN favors a flexible and cost efficient deployment thanks to the multiplexing gain of RAN processing resources [12]. The CMRI in [13] proves that C-RAN requires approximately 10 to 15% less CAPEX per square kilometer than traditional LTE networks. Besides, C-RAN enables advanced coordinated signal processing between co-located BBUs, which enhances the UE throughput and quality of experience [6]. Figure 1.5.(a) depicts the original C-RAN architecture with a full baseband centralization.



(a) Full baseband centralization

(b) Partial baseband centralization

Figure 1.5: C-RAN architecture [2]

Despite the aforementioned benefits, it is worth pointing out that baseband cloudification directly impacts the resource provisioning in the fronthaul link. Indeed, 5G fronthaul may require up to 157.3 Gbps of bandwidth with 10 µs - 250 µs of latency [14]. Such stringent constraints would make the dark fiber almost the only applicable fronthaul solution which cost increases the CAPEX, thus counteracting the original cost saving principle of C-RAN.

In order to relax these excessive fronthaul constraints without losing the benefits from baseband centralization, recent 5G contributors are proposing a hybrid C-RAN architecture that enables **flexible baseband function placement** between the **Cloud** and the **access** sites. Thanks to such an approach, the BBUs can be seen as a chain of virtual baseband that can be splitted at many conceivable points. In doing so, a partial baseband centralization is enabled, while leaving some functions at the access site. Accordingly, new interfaces between baseband functions are being identified with reference to the traditional Long Term Evolution (LTE) architecture. These interfaces enable a set of functional splits [15–17], wherein some baseband functions are kept in the access site, thus relaxing the bandwidth and latency requirements of the fronthaul link.

Figure 1.5.(b) depicts a partial baseband centralization in a C-RAN architecture. Wherein, the RRHs are equipped with computational resources to host the lower part of baseband functions, leaving the upper part at the Cloud site. However, the functional split concept may still reduce the advantages from baseband full centralization, while keeping some baseband functions at the access site [12].

1.2.3 Towards standardization of a disaggregated RAN

Next Generation Mobile Networks (NGMN) alliance highlighted in [18] the need for integrating the C-RAN architecture in next generation mobile networks. Furthermore, different requirements of functional splits are studied in both uplink and downlink directions. On the other hand, IEEE Next Generation Fronthaul Interface (NGFI) working group proposed in [15] an Ethernet-based Fronthaul C-RAN architecture, consisting of Remote Radio Units (RRUs) and Radio Cloud Center (RCCs), connected through link aggregators called Radio Aggregation Units (RAUs). Wherein, the RRU includes the antenna unit RRH along with its computational resources, while the RCC includes the cloudified BBU pool. Besides, several standardization activities in 3GPP foster the adoption of the RAN functional split by introducing the Next Generation 5G Radio Access Network (NG-RAN) architecture [17]. The primary goal of this design consists in enabling the deployment of different topologies tailored to the 5G use case requirements.

Interestingly, the Open RAN (O-RAN) initiative has been proposed by an alliance of mobile operators in 2016, promoting Openness and Intelligence for C-RAN [19]. The developed RAN software is based on 3GPP standards, which is essential to support MNOs in implementing new functionalities and services tailored to the 5G use cases. Besides, O-RAN aims at maximizing the use of common-off-the-shelf hardware, while minimizing proprietary hardware. Additionally, there is an ongoing work for delivering a Virtualized Infrastructure Manager (VIM) to enable both network slicing and functional split deployment [4]. However, the O-RAN initiative is still in its early stages.

1.2.4 5G Cloud RAN open source initiatives

5G RAN stakeholders are closely collaborating to introduce innovative technologies to their C-RAN infrastructures, bringing hence flexibility and agility. These technologies include Network Function Virtualization (NFV) and Software-Defined RAN (SD-RAN).

OpenAirInterface (OAI) software is an open source Software Defined Radio (SDR) implementation of radio access network, core network and user equipment of 3GPP cellular networks [20]. Specifically, OAI allows to implement, run and evaluate the 5G C-RAN architecture, in terms of fronthaul properties and processing software latency[21]. Furthermore, OAI community is working on proposing a multitenant architecture with disaggregated micro-service radio-processing [22].

TIP OpenRAN project group was initiated by the Telecom Infra Project (TIP) in 2017 [23]. The objective is to build 3GPP RAN solutions with virtualized and programmable RAN functions running on General Purpose Processing Platforms (GPPP). It is worth noting that other interesting open source initiatives such as free5GC [24], Open5GS [25] and OMEC (Open Evolved Mobile Core) [26], which where formed to deal with the 5G core network.

1.3 Cloud RAN challenges

The massive adoption of Cloud technology in mobile access networks has driven the operators and vendors to work together in order to make Radio Access Network (RAN) ecosystem more agile. In this context, the virtualization of network functions is the cornerstone of a successful Network Function Virtualization (NFV) environment. However, the stringent use case demands alongside the strict 5G RAN requirements, make C-RAN deployment more complex and prone to feasibility and performance

issues.

1.3.1 RAN virtualization and cloudification

The first step towards the implementation of C-RAN consists in virtualizing and cloudifing the baseband functions, which strongly depend on the underlying network infrastructure, configuration and topology.

The virtualization is performed by decoupling the software from hardware and making the network functions running on virtual machines [27] or a container-based technology namely Docker [28]. The hypervisor-based virtualization, i.e. using Virtual Machines (VMs), consists in running a complete guest operating system (OS). This results in a high computationally resource intensive operations, that may slow down the guest OS boot. On the other hand, the **containerization** technology permits the execution of the network functions as an application in an isolated system environment called container. Each container consumes directly the hardware resources, without requiring the deployment of a full OS, which makes it significantly lighter and swifter compared to hypervisor-based virtualization. Therefore, it is foreseen that the containerization is the most appropriate technology to support the C-RAN, due to its strict delay sensitive and resource scaling requirements [29].

Network function cloudification is performed by exploiting general purpose processors to enable on-demand deployment of network functions. The main challenges concern the feasibility of executing some baseband functions in the Cloud. As a matter of fact, some functions in the BBU physical layer such as the channel coding function are computational intensive. So, if not implemented on a dedicated hardware, this may result in a significant performance degradation in terms of latency [21]. Recent works such as in [30], investigate the use of parallel programming techniques in order to enable the cloudification of such baseband functions.

1.3.2 RAN disaggregation

Baseband functions have different properties depending on the processed data and the layer hosting them. Specifically, some Processing Functions (PF) perform at a user level, denoted by **User-centric Processing Functions** (UPF), i.e., once executed, they deal with one user at a time. Other PFs operate on a cell level, described as **Cell-centric Processing Functions** (CPF). A further challenge consists in determining whether to perform the functional split on cell basis or user basis. In the former approach, the BBU is splitted into a chain of CPFs aggregating, hence, UPFs of the same layer into a single CPF. In doing so, one single functional split per cell is possible at a time. In contrast, thanks to the latter approach, the UPFs of a single layer are decoupled to be deployed independently either in the access or Cloud sites, allowing, hence, the deployment of multiple user functional splits per cell at a time.

1.3.3 RAN slicing

RAN disaggregation has brought more flexibility to the RAN deployment but has to consider the UE use case requirements. On one hand, by keeping a high level of centralization, the majority of baseband functions run in the Cloud site which implies high latency and bandwidth requirements in the fronthaul [16, 17]. This scheme is broadly satisfied for use cases with tight latency requirement, however, increases the fronthaul congestion. On the other hand, a high level of decentralization puts more baseband functions in the access site. The latter scheme reduces the latency and bandwidth requirement in the fronthaul, which can be a good option to relax the congestion on the fronthaul link, while satisfying use

cases with adjustable latency requirement. Table 1.1 highlights the divergence in throughput and latency requirements for eMBB and uRLLC use cases. It is worth noting that the main performance requirement for a mMTC use case is the connectivity density.

Use case	Latency	Throughput	Connectivity density
eMBB	4ms one way delay	several Gbps	-
uRLLC	0.5ms one way delay	-	-
mMTC	-	-	up to 1 million devices per square km

Table 1.1: Use case requirements [9]

Eventually, the objective is to meet the UE requirements in terms of throughput and latency while considering multiple deployment designs encompassing heterogeneous RAN resources. In this context, RAN slicing [11] is proposed to allow the RAN infrastructure to deliver end-to-end allocated resources (so called slice) as per service requirement. With such a paradigm, multiple independent slices can be created on the same infrastructure to convey services that have different requirements for latency, reliability and throughput. In the context of C-RAN, end-to-end resources encompass radio, computational and link resources that should be allocated efficiently.

1.3.4 Energy-efficient C-RAN

There is a direct relationship between RAN operational cost reduction and energy benefit. Indeed, C-RAN leverages the Cloud infrastructure to allocate the computational resources on the fly, with respect to traffic load variation. Such a flexibility reduces the overall computational resource demand for network operation compared to the traditional RAN [2]. This fact makes C-RAN energy efficient "by design". Furthermore, Cloud infrastructures are characterized by an energy efficient indicator that expresses its Power Usage Effectiveness value (PUE) [31].

Moreover, thanks to C-RAN, cooperative radio processing between co-located BaseBand units enables the optimization of RRH power transmission in a predefined RRH cluster [32][33]. Other works opt for dynamic BBU-RRH assignment, while triggering the low consumption sleep mode for inactive cells [34].

However, in some cases, a fully centralized C-RAN architecture might become impractical due to the high amount of energy consumption in the transport network [35]. Henceforth, a partially centralized C-RAN scheme should be leveraged to reduce the energy consumption on end-to-end network resources [36].

1.3.5 C-RAN orchestration

MNOs are dealing with a multi-dimensional tradeoff between conflicting objectives. Indeed, by leaving some of the baseband functions at the access site, bandwidth needs are reduced and both latency and jitter are relaxed. However, such a strategy reduces the opportunities of coordinated signal processing and benefits from pooling the baseband functions. Therefore, MNOs should seek for the balance point between baseband function centralization and decentralization in order to jointly i) minimize the Fron-

thaul bandwidth, ii) reduce the computational resource demand with baseband cloudification, and iii) fulfilling the stringent requirements of 5G use cases.

Radio, computational and link resource allocation impacts directly the UE Quality of Service (QoS) and the deployment cost. This type of problem is often known as Multi Objective Combinatorial Optimization Problem (MOCOP), which can be expressed in an Integer Linear Problem (ILP). In general, this type of problem is not scalable [37] where the optimal solution leads to a high resolution complexity. Thus, there is a high need to design an efficient algorithm to solve it in a polynomial time.

Besides, the 5G context requires an up-to-date decision during each Transmission Time Interval (TTI) period. To deal with such a high real time decision making requirement, the adaptability of slice deployment needs to be adequately fast.

As the evolution of network management complexity progresses, the Self-Organizing Networks (SONs) [38] have been supported by 3GPP standardization for empowering the RAN with big data applications. Specifically, the use of machine learning techniques enables the development of self-aware, self-configuring, self-optimization, self-healing and self-protecting 5G systems, in what we call cognitive network management. By enabling end-to-end on the fly resource provisioning, RAN slices can be created, reconfigured and managed efficiently in a dynamic and scalable environment.

1.3.6 Business model transformation

5G brought radical technological improvements into the cellular networks. However, it is not expected that MNOs revenue-per-bit will cope with the cost-per-bit investments [1]. Therefore, MNOs are struggling to find a sustainable business model to monetize their offerings.

The concept of spectrum sharing has been introduced to enable the partitioning of licensed spectrum into slices that can be delivered as a service to virtual operators, also known as micro operators (μOs) [39]. This concept was first standardized by 3GPP in [40]. Wherein, the radio spectrum is shared besides equipments such as: the radio masts, transport infrastructure (fiber, cables, etc.) and BaseBand processing resources. Many works such as in [41–43] address the management of radio resource allocation with isolation and sharing capabilities.

Another interesting case is to invest in the entire mobile network infrastructure and deliver anything as a service (XaaS). This can be in the form of i) Infrastructure as a Service (IaaS), ii) Network as a Service (NaaS), or iii) Network Slices as a Service (NSaaS). As a matter of fact, providing IaaS helps operators to manage their network infrastructure, while leasing physical equipments. On the other hand, NaaS which is in our case RANaaS, is about delivering the RAN connectivity, while discarding operators from the infrastructure management complexity. Then, NSaaS is about offering a RANaaS with a customized resource provisioning scheme to fit specific use case requirements.

1.4 Thesis contributions

Hereafter, we summarize the significant contributions of this thesis.

- **State-of-the-art on C-RAN architecture**

  - **Deep analysis of 3GPP functional split options:** We provide an in-depth overview of each 3GPP functional split option [17] in terms of requirements, advantages and limitations.

We also give details about ongoing C-RAN initiatives and standardization efforts fostering C-RAN implementation.

– **Deep analysis of C-RAN resource provisioning strategies:** We study state-of-the-art C-RAN resource provisioning strategies. We can classify them into two main groups. The first one includes approaches aiming at reducing the RAN deployment cost and energy consumption by adopting the partial baseband placement strategy. The first group of approaches are denoted by RAN placement approaches. The second group comprises approaches aiming at fulfilling the UEs' QoS requirements, while reducing the RAN deployment cost and energy consumption. In the second approach, RAN slice allocation is performed by jointly allocating radio, link and computational resources. The second group of approaches are denoted by RAN slice allocation approaches. It is worth noting that, in the literature, there are additional optimization schemes addressing exclusively the radio resource allocation [41–52]. Their scope is limited, as RAN slicing also incorporates computational and link resources. Therefore, this group of approaches is not considered in this thesis as it does not respond to our objectives.

- **Implementation of a cost efficient C-RAN framework enabling on-demand deployment of RAN resources:** We propose and implement an experimental Agile C-RAN framework, denoted by AgilRAN. The latter is multi-sited which is in compliance with the NG-RAN 3GPP architecture [17]. We rely on Network Function Virtualization (NFV), and more specifically, on the container technology (e.g., LinuX Container LXC and Docker) to enable the virtualization of fine-grained baseband functions. We also refer to the latest advances of SD-RAN to monitor and control the RAN network state, while using the SDN FlexRAN controller [53]. AgilRAN enables a user centric split orchestration ensuring baseband function placement and their interconnection, while taking into account the temporal load variation of users and real-time network state.

- **Energy-efficient user-centric functional split solution optimizing the RAN deployment cost:**

  We propose a novel functional split orchestration scheme that aims at minimizing the RAN deployment cost. With a fine grained approach on user basis, we show that the proposed solution optimizes both processing and bandwidth resource usage, while minimizing the overall energy consumption compared to i) cell-centric, ii) fully distributed and iii) fully centralized C-RAN approaches. By enabling the selection of functional split for each user, link and computational requirements become more tunable, which is a key to build cost effective RAN deployment solutions. It is worth noting that the elaborated model is limited to one cell. Although this novelty has brought more flexibility to 5G C-RAN, the adopted approach does not consider the UE latency requirement, while performing the baseband function placement. Furthermore, the required amounts of computational and link resources for user-centric functional splits, depend on the user traffic load, i.e., the amount of allocated radio resource blocks. Hence, the baseband function placement can be further optimized when integrating the radio resource allocation in the split selection decision. At the end, by performing the joint radio, computational and link resource allocation at the user level, in a multi-sited C-RAN environment, the RAN deployment cost can be more tunable, while effectively considering the UE use-case requirements. This challenge expresses the RAN slice allocation, which will be addressed in the next contribution.

- **User-centric RAN slicing allocation in 5G C-RAN**
  - **Heuristic based approach:** We put forward a user-centric RAN slicing allocation scheme aiming at optimizing jointly radio, link and computational resources for each User Equipment (UE). Our scheme fulfills each UE QoS requirement while considering the underlying RAN infrastructure state. Our proposed heuristics are operating in a reasonable time within tens of milliseconds. However, the 5G RAN context requires an up-to-date decision within one Transmission Time Interval period, i.e., less than 1 ms [54]. Therefore, sophisticated algorithms may lead to decisions, that once taken, will be already obsolete and hence not applicable. Reactive models are highly recommended in these cases, where the allocation scheme is generated in real-time upon input data without performing an exhaustive calculation task.
  - **Deep Learning based approach:** We propose a Deep Learning based approach for User-centric RAN Slice Allocation scheme. The latter is able to decide in real-time, to jointly allocate the optimal RAN slice for each user equipment. Our proposal satisfies UE's requirements in terms of throughput and latency, while minimizing the infrastructure deployment cost.

1.5 Thesis outline

This thesis is organized as follows. In Chapter 2, we provide an in-depth overview of each 3GPP functional split option and the ongoing initiatives and standardization efforts dealing with C-RAN. Then, we discuss the different C-RAN resource provisioning strategies found in literature. In Chapter 3, we describe the proposed AgilRAN architecture and platform implementation. Wherein, the main components of our dynamic RAN Functional split orchestration solution are described. Chapter 4 details the energy-efficient user-centric functional split solution, optimizing the RAN deployment cost. The proposed scheme is based on the Swarm Particle Optimization approach. In Chapter 5, we present the joint approach of radio resource allocation and functional split selection to address the user-centric RAN slice allocation problem. The proposed scheme is based on the Swarm Particle Optimization approach and Dijkstra Algorithm. In Chapter 6, we set out to address the real-time challenge by proposing a Deep Learning based solution for RAN slice allocation. Finally, Chapter 7 concludes the thesis and presents our ongoing and future work in the area.

CHAPTER 2

C-RAN: FUNCTIONAL SPLIT AND RESOURCE PROVISIONING OVERVIEW

Contents

2.1 Introduction

C-RAN architectures are expected to play a crucial role in providing ultra-high data rate, extremely low latency, and nearly ubiquitous connectivity for 5G and beyond networks. Thanks to their high flexibility, the network capacity will be increased while improving energy efficiency and achieving high scalability. However, despite the attractive advantages of C-RAN, its deployment raises new challenges related to the design of fronthaul link and the optimization of resource provisioning.

In this Chapter, we first introduce the C-RAN Fronthaul, a key element motivating the Mobile Network Operators (MNOs) to rethink their RAN architecture, towards the functional split concept. Section 2.3 presents an in-depth analysis of each split option in terms of requirements, advantages and limitations. In Section 2.4, we give insights into the standardization efforts. Section 2.5 pins the problematic of the C-RAN resource provisioning. Next, Section 2.6 identifies the objectives that should be optimized to achieve an efficient provisioning approach. Afterwards, Section 2.7 describes the relevant C-RAN resource provisioning strategies proposed in the literature. Wherein, the related strategies are classified into two main groups. The first one performs baseband function placement in a disaggregated RAN, aiming at reducing the RAN deployment cost and energy consumption. The second group comprises approaches performing RAN slice allocation by jointly allocating radio, link and computational resources. The objective is to fulfill the UEs' QoS requirements, while reducing the RAN deployment cost and energy consumption.

2.2 Cloud RAN Fronthaul

The third generation of mobile networks introduced the term "fronthaul" to denote the connection link between the Remote Radio Head (RRH) and the Baseband Unit (BBU), both located at the access site. RRH is connected to the antenna and performs radio functions, while BBU performs the processing functions. Then, the fourth generation of mobile networks introduced the term of BBU pool by centralizing the BBUs in a strategic location to reduce the access site rental costs. Meanwhile, the C-RAN concept was proposed to virtualize and cloudify the BBU pool in order to enable on-demand computational resource deployment, achieving, hence, greater cost savings, cooperative inter-cell processing, among other advantages.

However, in both cases, the BBU centralization raises a major problem when dealing with the capacity demand on the fronthaul network. Indeed, the bandwidth demand is scaling up with the radio parameters. For example, with a configuration of a 100 MHz LTE using 8 downlink antennas and 256 Quadrature Amplitude Modulation (QAM), the bandwidth demand is up to 157.3 Gbps per RRH-BBU connection with only 250 µs of latency [14]. MNOs may find dark fiber almost the only applicable fronthaul solution, which counteracts the original cost saving principle of C-RAN.

In order to relax the stringent fronthaul constraints, while taking advantage of BBU centralization, both industry and academia are rethinking the RAN architecture. The aim is to enable the adoption of low expensive connection options such as Ethernet and wireless links, with new fronthaul solutions such as carrier Ethernet [55], which ensure a certain QoS to time critical data.

Recently, the fifth generation of mobile network introduced a disaggregated RAN model. The aim is to cope with use-cases requiring low latency by keeping time-critical baseband functions in the access site, while pooling non-time critical baseband functions in a centralized location. In this context, 3GPP

introduced in [17], the Next Generation 5G Radio Access Network (NG-RAN) architecture. Wherein, the new 5G eNodeB (so called gNB) is decoupled into: i) Central Unit (CU), ii) Distributed Unit (DU) and iii) antenna Radio Unit (RU). Accordingly, RU and DU are deployed at the access site, while CU is kept in the BBU pool. The expected distance between RU and DU varies in range of [1-20] km, those between DU and CU in range of [20-40] km and the backhaul connection can reach 300 km [56]. Consequently, the BBUs can be splitted at many conceivable points, enabling, hence, a partial baseband centralization, while leaving some functions at the access site. Figure 2.1 describes the RAN Architecture of the fourth generation (4G), full centralized C-RAN and 3GPP NG-RAN architectures, respectively.

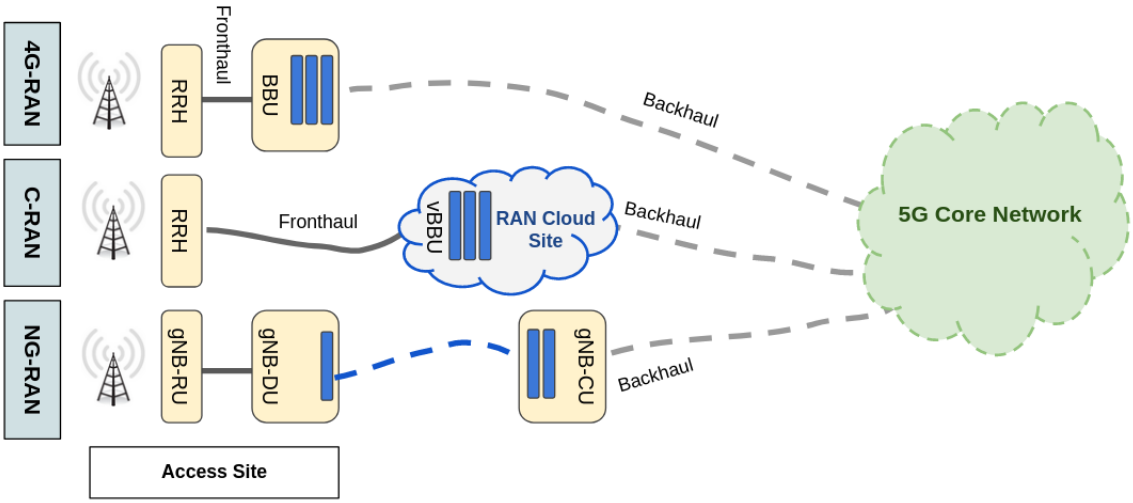


Figure 2.1: 4G, C-RAN and 3GPP NG-RAN architectures

2.3 Functional split requirement analysis

Several functional split options for a disaggregated 5G RAN have been proposed by 3GPP in release 14 [14] as shown in Figure 2.2. The LTE protocol stack is decomposed into a chain of processing functions, that can be splitted at many conceivable points. The latter are marked with dashed lines to separate functions performed in CU (at the top) and the ones performed in DU (at the bottom).

Such a disaggregation raises the question of which functions to put in CU and which functions to leave in the DU. In order to answer, a deep analysis of each functional split option is needed. In what follows, we provide an in-depth analysis of each split option in terms of requirements, advantages and limitations. We refer to 3GPP Specification [14], while the bandwidth requirement for each functional split option is calculated in downlink direction assuming a radio configuration with 100 MHz LTE using 8 downlink antennas and 256 Quadrature Amplitude Modulation (QAM).

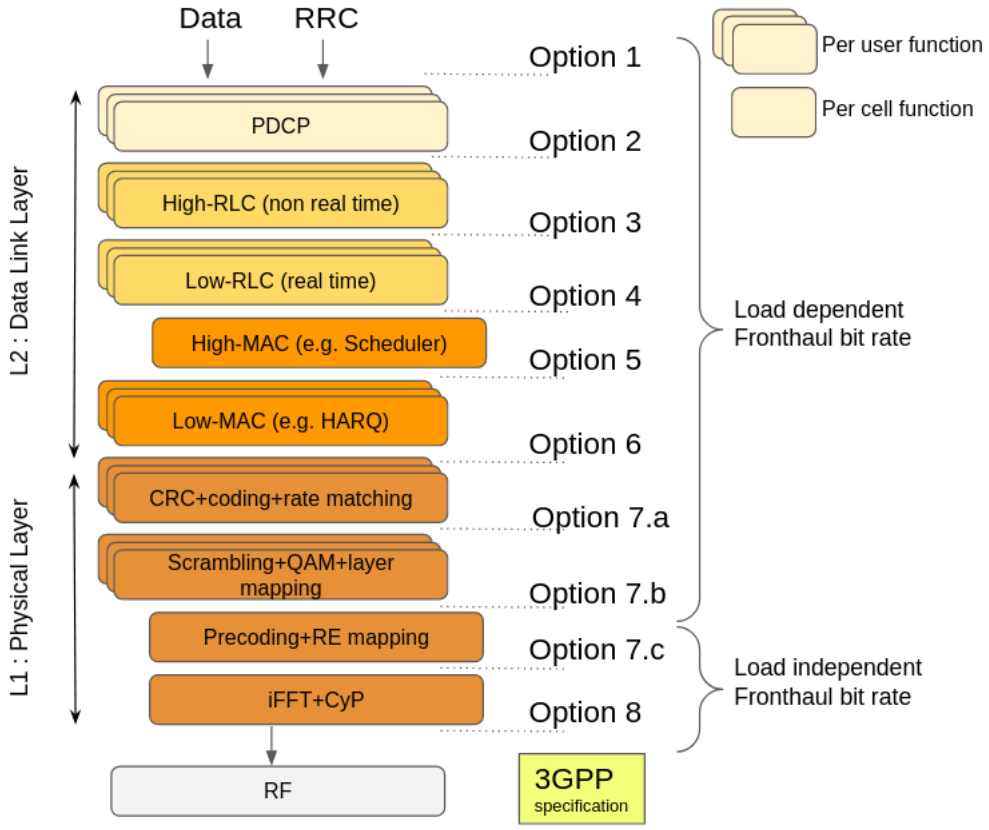


Figure 2.2: The LTE protocol stack with functional split options proposed by 3GPP

2.3.1 3GPP Option 1 : RRC/PDCP

3GPP Option 1 centralizes the Radio Resource Control (RRC) processing function. RRC constitutes the Control Plane (CP) of the LTE protocol stack operations such as system information, measurement configuration and User Equipment (UE) connection control, etc,. The User Plane (UP) operations are handled from the Packet Data Convergence Protocol (PDCP) function backwards to the Radio Frequency (RF) function. Therefore, option 1 also refers to the CP/UP split, where the fronthaul is expected to transport both RRC signaling and user plane traffic through radio bearers.

The latency requirement for this split option is in the order of 10 ms [57], comparing with 250 µs for a full centralized C-RAN. Besides, this interface generates a user load dependent bitrate on the fronthaul link achieving a bandwidth relaxation in the order of 97% compared to the traditional C-RAN [57].

With this split, RAN control functions are centralized, which is in compliance with the SD-RAN principle. Indeed, management decisions, such as Radio Resource Management (RRM), mobility and fast switching between Radio Access Technologies (RATs) can be further optimized leveraging the centralized view of RRC.

However, this design becomes less efficient in wide deployment due to the important overhead of RRC control messages that may limit the fronthaul bandwidth [58]. Besides, higher security and resiliency procedures should be addressed for the fronthaul. On the other hand, only few functions can

benefit from the shared processing power, since all UP functions are decentralized. Also, advanced techniques like inter-cell coordination cannot be supported since they require co-location of specific UP processing functions.

2.3.2 3GPP Option 2: PDCP/RLC

3GPP Option 2 centralizes the Packet Data Convergence Protocol (PDCP) function, while keeping the Radio Link Control (RLC) function in DU. Wherein, PDCP receives IP packets from higher layer to perform header compression and encryption operations. IP packets are, then, transported through the fronthaul link. It is worth noting that there is one instance of PDCP per user flow [59].

The latency requirement for this split option varies in the range of $[1.5, 10]$ ms [57]. Similarly to the split option 1, the fronthaul bandwidth is load dependent achieving a relaxation in the order of 97% compared to the traditional C-RAN [57]

With this split, PDCP flows can be transported to the RLC function of multiple remote DUs, making the split supporting the multi-connectivity, i.e., fast switching between Radio Access Technologies (RATs).

However, split option 2 requires a buffer at both sides of the fronthaul to put IP packets in order. This fact imposes additional processing burden adding extra latency to the fronthaul [16]. Besides, only few functions can benefit from the shared processing power with low benefits from coordinated cell processing.

2.3.3 3GPP Option 3: intra RLC

This split corresponds to the interface dividing RLC function into two sublayers: High-RLC and Low-RLC. High-RLC mainly performs the Automatic Repeat Request (ARQ) retransmissions, when IP packets are received out of sequence from PDCP. Low-RLC conducts the segmentation of PDCP PDUs following sizes indicated by the Medium Access Control (MAC). It is worth noting that there is one instance of RLC per user flow [59].

This split places High-RLC and PDCP in the same location. It helps, hence, to reduce, the transmission delay of re-establishment procedures.

Same as option 2, the latency requirement for this split option is in the range of $[1.5, 10]$ ms and the fronthaul bandwidth is load dependent with a relaxation in the order of 97% compared to the traditional C-RAN [57].

There is a handful of contributions related to this split option either in simulation or practical experiment [59], which leaves room for deeper studies. In this thesis, 3GPP Option 3 is not considered as computational and bandwidth requirements for both High-RLC and Low-RLC functions are still under investigation [59].

2.3.4 3GPP Option 4: RLC/MAC

The functional split option 4 centralizes the RLC function while leaving the Medium Access Control (MAC) function in the access site. The fronthaul link transports the RLC PDU to the MAC layer in downlink direction.

However, as described earlier, the Low-RLC function is tightly related to the MAC layer. In fact, the latter is sending frequent notification to specify the size of RLC PDUs to ensure a specific Quality of

Service (QoS) for each data flow. In the context of 5G, it is expected that subframes will be shorter. This fact requires more frequent decisions performed by the MAC scheduler [38], which makes the option 4 impractical.

The latency requirement for this split option is approximately 100 µs [57] which is very tight. Similarly to split option 1, the fronthaul bandwidth is load dependent achieving a relaxation in the order of 97% compared to the traditional C-RAN [57].

2.3.5 3GPP Option 5: intra MAC

This split corresponds to the interface decomposing the MAC function into two sublayers: High-MAC and Low-MAC. High-MAC encompasses the scheduler responsible for allocating the radio resources in frequency and time domain, which are called Resource Blocks (RBs). The scheduling is conducted via a controller and a random access control entity that operate at the cell level. This operation is performed repeatedly each Transmission Time Interval (TTI), corresponding to 1 ms for LTE. Eventually, the fronthaul link is transporting the multiplexed data flows and scheduling commands in downlink.

The driver behind this design is to centralize the MAC scheduler in order to enable efficient Coordinated Multi Point (CoMP) processing such as multi-cell Collaborative Scheduling (CS) and Joint Processing (JP).

According to [57], the latency requirement for this split option is in the order of hundreds of ms, which extremely relaxes the fronthaul bandwidth, but still depends on the realization and interaction of scheduling functions in the CU and DU [59]. Similarly to split option 1, the fronthaul bandwidth is load dependent achieving a relaxation in the order of 97% compared to the traditional C-RAN [57].

However, in the context of 5G with an even shorter TTI of 250 us [60], the performance of centralized MAC scheduler can be impacted by a non-ideal fronthaul latency, thus, limiting the performance of Coordinated Multi Point (CoMP) processing. In this thesis, 3GPP Option 5 is not considered as computational and bandwidth requirements for both High-MAC and Low-MAC functions are still under investigation [59].

2.3.6 3GPP Option 6: MAC-PHY

Split option 6 centralizes all the MAC sublayers, leaving the physical layer in DU. It is worth noting that Low-MAC performs the Hybrid ARQ (HARQ) process, which is a time critical function. Indeed, HARQ reports the scheduling operation feedback for each user periodically. In the LTE FDD setup, the HARQ mechanism imposes a feedback timing of 4 TTIs, which provides an upper bound for the total delay of both fronthaul link and BBU processing time. Besides, Low-MAC is responsible for building a transport block per UE based on the UE's context and its data buffer. It operates at the user level. Accordingly, the fronthaul link is carrying the transport blocks with an expected extra overhead from scheduling control and synchronization.

Being centralized, the HARQ process imposes a very tight fronthaul latency of 250 µs [14], which is impractical in case of sub-ideal fronthaul. This constraint is kept for the remaining splits 6, 7a, 7b, 7c and 8, as long as the HARQ process is centralized. The fronthaul bandwidth is load dependent with a small increase comparing to option 5 due to overhead, but keeping a relaxation in the order of 97% compared to the traditional C-RAN [57].

This split option enables efficient inter-cell scheduling leveraging the centralized view of MAC layer.

However, the fronthaul delay impacts the HARQ process which limits the ability of shorter subframes. Even though the entire network layer L3 and data link layer L2 are centralized, there is only around 20% of baseband processing resources taking benefit from the pooling gain of Cloud processing resources. The rest is located at the L1 physical layer.

2.3.7 3GPP Option 7a: High-PHY

According to [57], there are three functional split options for intra physical layer: option $7a$, $7b$ and $7c$. We propose to detail hereafter option $7a$ which centralizes the following functions: i) attachment of Cyclic Redundancy Check (CRC), ii) encoding and segmentation of transport blocks and iii) rate matching. It is worth noting that these functions operate on user basis. Using this split option, codewords are transmitted in downlink direction on the fronthaul link to be modulated in DU.

The encoding function is the most expensive in BBU LTE stack in terms of processing time [30]. Hence, a potential cloudification of this function may contribute to reduce the processing delay.

Similarly to option 6, the latency requirement for this split option is approximately 250 µs [57] which is very tight. The fronthaul bandwidth is load dependent achieving a relaxation in the order of 85% compared to the traditional C-RAN [57].

2.3.8 3GPP Option 7b: High PHY/Low PHY

Option $7b$ centralizes the following functions: i) codeword scrambling, ii) Quadrature Amplitude Modulation (QAM) and iii) layer mapping. It is worth noting that these functions operate on user basis. Therefore, option $7b$ transports subframe symbols with a variable bit rate on the fronthaul link. Thanks to split option $7b$, downlink CoMP coherent Joint Transmission (JT) can be supported without performance degradation.

Similarly to option 6, the latency requirement for this split option is approximately 250 µs [57] which is very tight. The fronthaul bandwidth is load dependent ensuring a relaxation in the order of 85% compared to the traditional C-RAN [57].

2.3.9 3GPP Option 7c: Low PHY

Option $7c$ divides the lower part of the physical layer into two sub-parts. Hence it is called the Low PHY split. This split option centralizes both precoding and resource element mapper functions. The former precodes the symbols on each layer for transmission on the antenna ports. The resource element mapper is a cell processing function responsible for converting the symbols into sub-carriers converting, hence, the fronthaul bit rate from variable to constant bit rate. The inverse Fast Fourier Transform (iFFT) and Cyclic Prefix (CyP) functions are left in DU side. iFFT is responsible for converting the sub-carriers from frequency domain into IQ symbols in the time domain, while CyP helps to distinguish the frames.

Accordingly, Option $7c$ transports the sub-carriers on the fronthaul link with a high and constant bit rate. In doing so, it achieves a relaxation in the order of 45% compared to the traditional C-RAN [57]. Similarly to option 6, the latency requirement for this split option is approximately 250 µs [57] which is very tight.

2.3.10 3GPP Option 8: PHY/RF

Split option 8 refers to the traditional full centralized C-RAN architecture, where the BBU LTE stack is fully centralized in CU. Specifically, this split interconnects the lower processing functions part of BBU, i.e., the physical layer, to the Radio Frequency (RF) function of RU. Hence, this interface is also referred to as PHY/RF interface. Accordingly, the generated IQ samples are radio waveforms encapsulated in a transport protocol such as: CPRI [61], CPRI over Ethernet [21] and compressed CPRI [62].

IQ samples are generated with a constant bit rate regardless the cell load. Indeed, the bandwidth demand for this option depends on the radio configuration like number of antenna, that may require up to 157.3 Gbps of bandwidth, which is non-affordable. Same as option 7, this split requires a latency of 250 µs [14].

In this configuration, all PFs are centralized, achieving the highest benefit from sharing the processing resources, while enabling BBU cooperation at many levels. However, the required fronthaul capacity is the highest. Besides, CPRI requires a very strict jitter that can limit the transmission over a packet switched network such as Ethernet. Indeed, this option requires high capacity fibers and real time communication on the fronthaul link.

Another interesting split option is proposed by the Small Cell Forum (SFC) in [16]. It further centralizes the Parallel to Serial conversion and CPRI encoding functions in case of using a fiber connection between DU and CU.

2.4 Functional split: Standardization effort

Figure 2.3 summarizes the requirements for fronthaul bandwidth and latency according to [57], when operating with a 100 MHz LTE using 8 downlink antennas and 256 Quadrature Amplitude Modulation (QAM). It is worth noting that the bandwidth requirement is static (i.e., load independent) for options $7c$ and 8, while it is variable (i.e., load dependent) for rest of options. Therefore, we show in Figure 2.3 the highest peak of bandwidth that can be achieved by the latter split options.

Today, there is a strong interest from research and telecom industry to leverage the disaggregated RAN design in order to provide a cost-effective transport network. Accordingly, the challenge is to build a RAN infrastructure that flexibly deploy the optimal split option in a dynamic fashion. Thereby, the partial C-RAN centralization solution can be offered as RANaaS. Hereafter, we give an overview of the standardization trends and industrial work towards a C-RAN architecture enabling flexible deployment of functional splits.

The IEEE 1914 Next Generation Fronthaul Interface (NGFI) Working Group [15] is working on standardizing packet based fronthaul transport networks. To do so, the functional split options are analyzed in terms of required data rate, latency, synchronization [63]. Then, possible deployment scenarios are proposed in compliance with the 3GPP specification [57]. Wherein, NGFI defines a two-level fronthaul: NGFI-I and NGFI-II. Accordingly, NGFI-I is the interface connecting RU and DU to deploy split options with stringent latency and high bandwidth requirement, while NGFI-II is the interface connecting DU and CU to deploy split options with low bit rate and relaxed latency requirement. Besides, IEEE 1914 aims at using the already existing Ethernet infrastructure as a fronthaul technology by first proposing to encapsulate the IQ data of low layer splits into Ethernet frames [64].

The ITU-T Technical Report on Transport network support of IMT-2020/5G [56] analyzes the 3GPP NG-RAN architecture [17] and their split option requirements in terms of bit rate, latency and synchro-

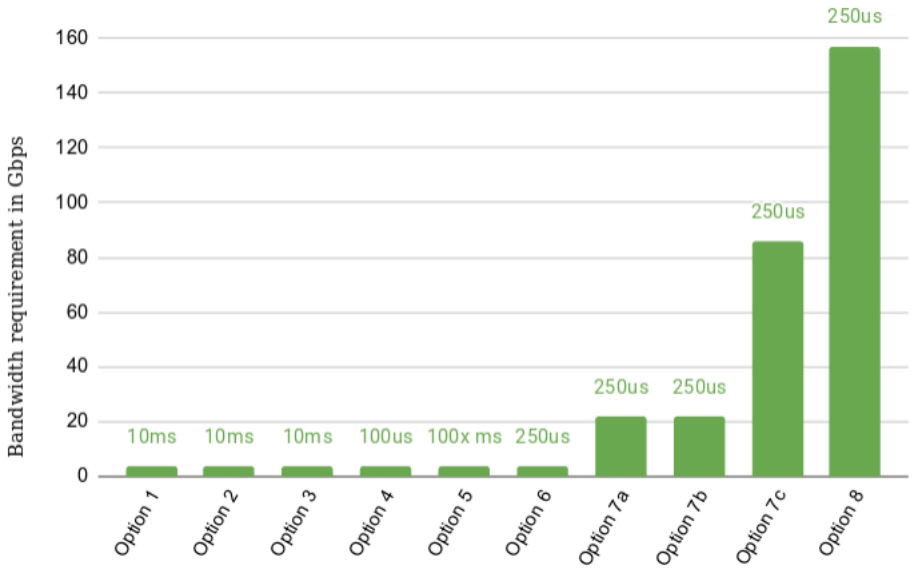


Figure 2.3: Fronthaul bandwidth and latency requirements for each functional split option

nization. They conclude that the fronthaul connection between DU and CU can support heterogeneous service requirements that can support 5G use cases like eMBB, uRLLC and mMTC. Therefore, in support with the 5G slicing vision, it is crucial to provide a fronthaul network with flexible functional split to transport services with heterogeneous QoS requirements.

Next Generation Mobile Networks (NGMN) alliance highlighted, in [65], the need for the functional decomposition of RAN to meet the diverse transport performance demands and align them with the requirements of next-generation service categories such as eMBB, uRLLC and mMTC.

In [16], the Small Cell Forum provides a very thorough study of different functional split options while analyzing their feasibility, key benefits and requirements in terms of bit rate and latency. Interestingly, they propose a theoretical model to compute the fronthaul link bit rate and the bandwidth requirements of each split option. In this thesis, we propose to rely on this calculation method to evaluate the impact of deployment of each split on the fronthaul traffic.

Mobile Central Office Re-architected as a Datacenter (M-CORD) [66], is a project aiming at creating an open reference datacenter implementation for 5G mobile wireless networks. The solution is cloud native built on SDN and NFV concepts to support various access technologies.

2.5 Cloud RAN resource provisioning challenge

One of the key success of C-RAN is its ability to offer a RANaaS solution with on-demand resource provisioning. Wherein, RAN resources encompass radio, computational and link resources, that should be allocated efficiently. The provisioning of radio resources consists in i) UE-gNB association, ii) partitioning the carrier bandwidth of each gNB among users, and iii) allocating the adequate amount of radio power on each resource block bandwidth. The objective is to fulfill the end-users' throughput demands, while minimizing the inter-cell interference level. Meanwhile, the functional split approach has been

standardized to add new deployment design capabilities, while performing flexible baseband function placement. To do so, the deployment of a functional split consists in allocating i) computational resources in CU, ii) computational resources in DU, and iii) link resources in the fronthaul link. A further challenge consists in meeting the multitude use-case's requirements while considering different design models in the physical infrastructure. This challenge expresses the RAN slice allocation problem, where an efficient provisioning scheme should address jointly i) radio, computational and link resources. In the context of 5G, an up-to-date allocation decision is required during each TTI period. To deal with such a high requirement, the joint RAN allocation and placement scheme should be fast enough to adapt to the dynamic context of 5G.

2.6 Cloud RAN resource provisioning criteria

Resource provisioning in C-RAN can be performed to achieve different objectives. In what follows, we focus on network performance metrics of optimization problems considered in the literature that can be classified into i) resource usage, ii) throughput maximization, iii) energy consumption, and iv) delay.

- **Resource usage:** A natural objective to achieve efficient resource provisioning in C-RAN is the maximization of the number of served UEs. Indeed, an efficient allocation approach with high acceptance rate results in maximizing the revenue [67][68][69].

  Another metric consists on minimizing the number of active RRHs. The idea is to exploit the sparsity of users in the network to identify and switch off inactive RRHs for minimizing the network power consumption [70][71][72].

  Most of related work, targeting end-to-end resource allocation in C-RAN with functional splits, put a cost value function that expresses a weighted sum of computation, link and eventually radio resource usage [12][73][74][75][76]. The idea behind this is to offer to MNOs the ability to tune the usage of different RAN resources by flexibly selecting the appropriate functional split as follows:

  $$cost_{value} = w_{BBU} \,.\, BBU \;+\; w_{link} \,.\, FH \;+\; w_{prb} \,.\, PRB$$

  In the cost value function, BBU corresponds to the usage of computational resources across RAN sites which is the ratio between the amount of allocated computational resources and the available computational capacity. Similarly, FH is the amount of allocated link resources on the fronthaul link. PRB expresses the spectrum efficiency, i.e., the utilization of Physical Resource Blocks. w_{BBU}, w_{link} and w_{prb} represent the associated weights. They are normalized such that their sum is equal to a unit value.

- **Throughput:** Maximizing the sum-rate is an important network performance metric that reflects the ability to satisfy users requirements in terms of throughput. To this end, an efficient spectrum allocation approach that optimizes user-gNB association and PRB allocation is needed [34][77][78][79]. As shown in the following sum-rate function, the final throughput is evaluated as the summation of served throughput R_{im} from $gNB\ m$, $\forall m \in gNBs$, to user i, $\forall i \in users$. Wherein, the sum-rate is usually maximized under the practical network constraints such as inter-cell interference and allocated radio power.

$$sum_{rate} = \sum_{i \in users} \sum_{m \in gNBs} R_{im}$$

- **Energy consumption:** In addition, energy consumption is one of the key objectives for C-RAN. Therefore, several works propose to minimize the overall transmit power [80] and BBU power consumption [81][82]. Interestingly, various works propose a linear model to translate the amount of allocated computational resources into a consumed power [83] [36].

- **Delay:** Several papers target the delay minimization caused either by BBU processing [30][84] or by fronthaul links [76] [85]. The end-to-end delay over the network is calculated as follows:

$$E2E_{delay} = \sum_{d \in DU} w_d \, delay_d + \sum_{c \in CU} w_c \, delay_c + \sum_{e \in FH} w_e \, delay_e$$

  where $delay_d$ expresses the delay of BBU function processing in DU d, $delay_c$ expresses the delay of BBU function processing in CU c and $delay_e$ expresses the delay caused by link e to transport data between d and c.

2.7 Cloud RAN resource provisioning approaches

Hereafter, we provide a taxonomy of the C-RAN resource provisioning optimization approaches found in literature. We propose to classify them into two main groups. The first one includes approaches aiming at reducing the RAN deployment cost and energy consumption by adopting the partial baseband placement strategy. The first group of approaches are denoted by RAN placement approaches. The second group comprises approaches aiming at fulfilling the UEs' QoS requirements, while reducing the RAN deployment cost and energy consumption. In the second approach, RAN slice allocation is performed by jointly allocating radio, link and computational resources. The second group of approaches are denoted by RAN slice allocation approaches. It is worth noting that, in the literature, there are additional optimization schemes addressing exclusively the radio resource allocation. Their scope is limited, as RAN slicing also incorporates computational and link resources. Therefore, this group of approaches is not considered in this thesis as it does not respond to our objectives.

2.7.1 RAN placement approaches

Hereafter, we detail the state of the art on ongoing research for the RAN placement optimization approaches in C-RAN:

Authors in [76], propose a graph based framework to reduce the resource allocation computational cost in both access and Cloud sites. This framework takes into account both traffic load in the fronthaul link and the delay requirement for each cell, which are contradictory goals. To this end, a genetic algorithm is proposed, in order to place optimally the BBU functions across RAN sites. However, this approach is based on the assumption that the computational, bit rate and delay requirements for each split are static and does not reflect the most basic properties of real RAN systems.

A network calculus approach is elaborated in [85] proposing a multi objective function to minimize the deployment cost of BBU functions, while considering the strict delay requirements of critical ser-

vices. Unfortunately, this work does not refer to a quantitative model for the computational requirement of each split.

In [86], authors propose a detailed Total Cost of Ownership (TCO) minimization model in a fiber based RAN with BBU splits, which takes into account quantitative models for computational and link resource requirements. However, this model is still considering the use of splits with a coarse grained decision, as it generates a split per cell for all attached users.

A uRLLC slice embedding with a functional split approach is proposed in [87]. A heuristic is elaborated to minimize the link resource requirement in the fronthaul connection. However, it is worth noting that the aforementioned work is addressing the RAN function placement problem from a cell-centric point of view. Wherein, a unified functional split is selected for all end-users in one gNB.

In [88], authors present a model for minimizing the overall energy consumption of the 5G infrastructure by flexibly tuning the functional split on the optical transport. A Long Short-Term Memory (LSTM) based neural network is proposed to predict functional split decision. Unfortunately, the split deployment is cell-centric.

In [89], authors propose an ILP model for optimal functional split selection in a 3-layer RAN architecture. The benefits of the 3-layer architecture is compared with the 2-layer architecture, showing that the optimal centralization degree depends on processing capacity, transport network capacity and fronthaul traffic latency. However, this model is still considering the use of splits with a coarse grained decision, as it generates a split per cell for all attached users.

In [90], authors propose a model for RAN virtual network function placement on a physical infrastructure, while minimizing the bandwidth on the aggregated fronthaul link. A heuristic is proposed to dynamically select the optimal split option, while taking into consideration the daily traffic profile, number and placement of CU and DU elements. Unfortunately, the split deployment is cell-centric.

In [91], a BBU function placement approach is proposed, while adopting a functional split approach. The aim is to minimize the RAN energy consumption by minimizing the active DUs, while keeping a low latency in the fronthaul link, which is a contradictory goal. A heuristic is proposed to evaluate the impact of dynamic resource management facilitated through Virtual Machine (VM) live migration. Unfortunately the split deployment is cell-centric.

Authors in [12], elaborate a teletraffic theory to analyze the gain of aggregating the fronthaul traffic of multiple cells with different split configurations. An objective function is elaborated for maximizing the energy and cost savings. To do that, the authors evaluate the allocated amount of computational and radio resources along with the generated data rate in the fronthaul. Interestingly, this work proves that the gain is function of the traffic profile and monitoring method. In other words, the user load and type of traffic deeply impacts the split gain. For this reason, we conclude that a fine-grained split approach per user basis will certainly achieve higher benefits. However, this approach is not evaluated in [12].

Differently from above works, [73] proposes a model with user split orchestration that aims at minimizing the system energy and bandwidth consumption in the fronthaul link. The elaborated model is based on quantitative models to calculate the computational and link requirements for each split. However, this work relies on unrealistic split model as the platform control function which includes the MAC scheduler is assumed to be a user centric processing which is not accurate. In this thesis, we stick to reliable analytical models [16][83] and we elaborate a practical scheme based on realistic properties of each baseband processing function. Moreover, authors in [73] assume that the radio resource allocation is fixed for each user. Whereas, in our work, we take into consideration the traffic load variation and analyze its impact during the split decision which is the key for an efficient user-centric approach.

In [74], the same authors elaborate an end-to-end delay model to analyze the impact of a user delay request on split decision, which impacts in turn the total cost and energy consumption. The model is evaluated for a single user having optimal network conditions, therefore, the split decision is still considered as per cell basis.

As a summary, the aforementioned methods propose different approaches and models for minimizing the energy and deployment cost of C-RAN, while supporting the BBU splitting. However, the split decision is mainly taken with coarse grained on cell basis. As for [73], even the authors formulate their approach on a user basis, the elaborated split model seems unpractical with unrealistic assumptions. In [74], unfortunately there is only one user considered. Consequently, this would be seen as a monolithic approach, unlike our proposal in which we opt for a user-centric functional split approach based on analytical models and a practical scheme reflecting the RAN real properties.

2.7.2 RAN slice allocation approaches

By enabling joint radio, link and computational resource provisioning, RAN slices can be created and managed in a dynamic fashion, ensuring the RAN-as-a-Service (RANaaS) vision. One of the major concerns is how to meet the multitude use-case's requirements while considering different designs in the physical infrastructure. In doing so, decisions on what amount of allocated radio resources and what network functions to place in DU or CU raise many challenges. Eventually, RAN slice allocation impacts directly the end-user QoS performances and the operation cost, which is essential to design an orchestration solution able to rise these challenges. The network slicing approach that jointly optimizes the radio resource allocation and the functional split selection has motivated many research works.

In [84], authors elaborate a joint functional split and BBU scheduling problem in order to minimize the overall processing delay of downlink frames. The problem is formulated as a constrained shortest-path problem and solved with a heuristic algorithm that iterates between solving the two sub-problems. However, in this work, the functional split selection is performed on cell-basis.

An architecture for slice orchestration is proposed in [92], while supporting the BBU functional split. The proposed network resource management scheme jointly address the cloud-computing offloading and bandwidth allocation in the transport network. However, this work does not integrate the radio resource allocation in the slice approach.

In [93], a RAN runtime framework for slice control and orchestration is proposed. Then, a detailed approach on radio resource slicing with different levels of isolation and sharing is described. Although the disaggregated deployment scheme is integrated in the framework design, there is no problem modeling for functional split selection.

A multi-tenant slicing scheme in Cloud-RAN is proposed in [94], taking into account tenant priority, BBU resources, transport network capacities and interference levels. However, this work considers only a full centralized deployment scheme.

In [95], authors formulated a problem of the functional split selection while considering the inter-cell interference level. A new heuristic is proposed to minimize jointly the inter-cell interference and the bandwidth utilization on the transport network. However, the functional split approach is performed at cell level.

Authors in [96] propose a framework for slice management with functional split selection. They address the problem of joint radio allocation and split selection to meet the different use-case requirements. However, authors consider a cell-centric approach. Therefore, the current study aims to fill the

aforementioned gaps.

In [35], authors elaborate a power consumption model for radio, network function placement and transport network, leveraging the functional split capabilities. In one hand, a fully centralized C-RAN architecture is needed to provide good energy performance. However, it is expected to encounter radio performance limitation in the distance between the access site and the cloud RAN. On the other hand, the fully Distributed RAN (D-RAN) architecture is still interesting to minimize the high power consumption of the transport network. Thus, a partial centralization approach can achieve an optimal trade-off, subject to the radio configuration and traffic load. However, this work does not investigate the full split option solutions.

Unlike most existing network slicing solutions, which aim to aggregate all users traffic belonging to the same cell, we put forward a user-centric slicing scheme which instantiates an end-to-end network resources for each user. Our proposal is tailored to different user quality-of-service requirements and to the diverse functional splits resource requests.

2.8 Summary

Table 2.1 presents a comprehensive survey of the aforementioned C-RAN resource allocation algorithms found in literature. A taxonomy of these strategies in terms of: i) objective function, ii) functional split approach, iii) radio resource allocation approach, iv) constraints, v) problem modeling, and vi) used algorithm, are highlighted. Note that the fronthaul connection link is denoted by "FH".

2.9 Conclusion

This Chapter provided an overview of the different C-RAN resource allocation strategies found in literature. First, we introduced the C-RAN Fronthaul as key element that motivates the MNOs to rethink the RAN architecture. Second, we presented an in-depth analysis of each split option in terms of requirements, advantages and limitations. Next, we gave an insight into the standardization efforts. Then, we detailed the problematic of the C-RAN resource provisioning. Afterwards, we summarized the discussed related C-RAN strategies, while outlining the main objective, the proposed model and applied algorithm. Considering all the above criteria, the problem of on-demand resource allocation in C-RAN becomes a very challenging task. Unfortunately, the majority of the surveyed works consist in adopting a cell-centric approach for functional split deployment. Wherein, a split option is deployed for all associated UEs. However, to achieve greater flexibility and better resource utilization, a user-centric approach should be more exploited.

In this thesis, we address the challenge of the efficient deploying of functional splits on user basis, while dealing with temporal load variation of users. Consequently, to the best of our knowledge, our study is the first attempt to present an optimized RAN slice allocation in C-RAN that jointly optimizes the radio, link and computational resource provisioning on user basis. We also take into consideration the Quality of Service (QoS) requirements of each user in terms of throughput and latency. Our solution is based on analytical models and a practical scheme reflecting the RAN real properties. We integrate our proposal in a novel RAN orchestration Framework design that will be described in the next Chapter.

Table 2.1: Comparison of C-RAN resource allocation optimization strategies

Ref.	Objective function	Functional split approach	Radio resource allocation approach	Constraints	Problem modeling	Algorithm
[76]	Joint computational cost & FH delay minimization	User-centric	-	Cell delay	Graph-clustering	Genetic algorithm
[85]	Joint power consumption & FH delay minimization	Cell-centric	-	Cell throughput DU, CU, FH capacities	Integer linear problem	Lagrangian Relaxation
[86]	Joint computational cost & FH bandwidth minimization	Cell-centric	-	DU, CU, FH capacities	Integer linear problem	IBM CPLEX Optimizer
[12]	Joint power consumption & FH bandwidth minimization	Cell-centric	Spectrum efficiency	Cell traffic load	Teletraffic approach	OPNET Simulator
[73]	Joint power consumption & FH bandwidth minimization	User-centric	-	DU, CU, FH capacities	Mixed integer Problem	IBM CPLEX Optimizer
[74]	Joint power consumption & FH bandwidth & E2E delay minimization	User-centric	-	DU, CU, FH capacities UE delay request	Mixed integer Problem	IBM CPLEX Optimizer
[84]	Joint scheduling & FH delay minimization	Cell-centric	Frame scheduling	deadline constraints	Shortest-path problem	Djikstra algorithm
[92]	Joint computational cost & FH bandwidth & E2E delay minimization	Service-centric	-	DU, CU, FH capacities Traffic load	Mixed integer linear problem	OPNET Simulator
[95]	Joint interference & FH bandwidth minimization	Cell-centric	PRB allocation	inter-cell interference level	Integer linear problem	Heuristic
[96]	Joint throughput maximization & FH delay minimization	Full centralized C-RAN / D-RAN	UE-gNB association PRB allocation	DU, CU, FH capacities UE delay request	Integer non-linear problem	Greedy heuristic

Ref.	Objective function	Functional split approach	Radio resource allocation approach	Constraints	Problem modeling	Algorithm
[35]	Power consumption minimization for radio, FH & function placement	Cell-centric	Radio power allocation	Radio & FH configuration traffic load	Integer linear problem	in-house static system simulator
[87]	Joint RAN function placement with FH bandwidth minimization	Cell-centric	-	DU, CU & FH capacities	Integer linear problem	Heuristic
[88]	Joint power consumption & FH delay minimization	Cell-centric	-	Cell throughput DU, CU, FH capacities	Integer linear problem	LSTM based neural network
[89]	Joint computational cost & FH delay minimization	Cell-centric	-	DU,CU, RU, FH capacities UE delay request	Integer linear problem	-
[90]	Joint computational cost & FH bandwidth minimization	Cell-centric	-	Daily traffic profile RU, DU, CU, FH capacities	Integer linear problem	Heuristic
[91]	Energy efficient BBU function placement	Cell-centric	-	FH delay	Integer linear problem	Heuristic

CHAPTER 3

AGILRAN: AGILE COST EFFECTIVE CLOUD RAN ARCHITECTURE

Contents

3.1 Introduction

In this chapter, we present our cost efficient C-RAN architecture design and implementation, enabling on-demand user-centric deployment of RAN resources. First, we describe our agile RAN architecture along with its building blocks. Second, we present our experimental C-RAN prototype, which makes use

of Open Air Interface (OAI) [20] and FlexRAN SDN controller [53]. Finally, we evaluated quantitatively and qualitatively the bandwidth and computational consumption for a subset of functional split options.

3.2 `AgilRAN` architecture overview

We propose a 5G Agile RAN architecture, denoted by `AgilRAN`, which is in compliance with the 3GPP NG-RAN architecture, described previously in [Chapter 2, p. 14]. Characterized by two-level sites of processing, `AgilRAN` enables the placement of baseband functions in a dynamic fashion, while considering the UE stringent requirements and RAN network state. It is straightforward to see that similar C-RAN architectures have been proposed in the literature, such as [19]. The latter has been designed by mobile operators to respond to their 5G vision for building a virtualized RAN on open hardware, with embedded Artificial Intelligence powered radio control to enable SDN-like based capabilities. However, in this work, we go a step further and adopt a "highly disaggregated" RAN model.

The main idea behind our design is to ensure a user level orchestration of baseband functions in a hierarchical Cloud infrastructure, while using lightweight virtualization techniques. In fact, the BBU is re-architected to be disaggregated into microservices. The latter corresponds to a decomposed baseband functions which are instantiated into containerized network functions to perform either cell or user processing tasks. In doing so, BBU microservices can easily interact with each other and scale separately which make them Cloud native.

3.2.1 Disaggregated C-RAN infrastructure

We leverage the disaggregated RAN deployment approach where the gNB BBU is splitted between a Distributed Unit (DU) and a Central Unit (CU) connected through a fronthaul network. The latter multiplexes the traffic of multiple gNBs to/from the cloud site, where CUs are pooled. Accordingly, the `AgilRAN` architecture, as shown in Figure 3.1, is characterized by two layers in which baseband functions can be instantiated. The lower layer, referred to as Access Site, consists of a number of computational resources Distributed Units (DUs), which serve a set of Radio Units (RUs). Note that a DU corresponds to a server which could be whether Commercial Off-The-Shelf (COTS) or equipped with accelerators, e.g., FPGA (Field Programmable Gate Array). Each DU is capable of performing partial or full BaseBand processing. The remaining subset of baseband functions can be deployed at the second level system, a.k.a. Cloud site which hosts the Central Units (CUs) of multiple gNBs.

3.2.2 Cloud-native RAN

We rely on Network Function Virtualization (NFV), and more specifically, on the container technology (e.g., LinuX Container LXC and Docker [28]) to enable the virtualization of fine-grained RAN network functions in both DU and CU. It is worth noting that a container-based virtual environment guarantees higher performances compared to virtual-machines based environments, as they run directly on the kernel, use less memory and make run-time execution more efficient. Being packaged in containers instead of virtual machines, PFs can be dynamically instantiated and destroyed within few microseconds. Indeed, according to our experiments, we have quantified the average deployment time of a container-based PF to 1.8 μs $\pm$ 0.2 μs.

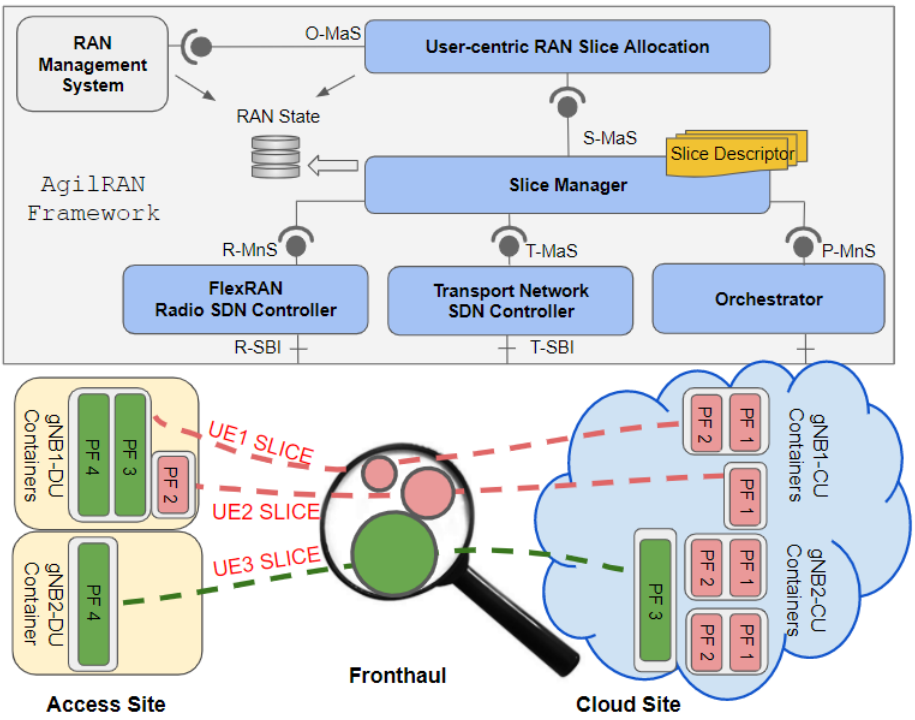


Figure 3.1: AgilRAN Framework

3.2.3 RAN function placement

As depicted in Figure 3.1, the Orchestrator module is responsible for dynamically instantiating the containers at the access and edge sites respectively, while connecting them with a fronthaul link. Therefore, the RAN slice decision's execution is performed by the Orchestrator module. Once the user leaves the cell, the Orchestrator triggers the destruction of containers and deletes the slice.

3.2.4 RAN function control

We make use of an SDN controller to configure the link bandwidth and latency in the transport network. The control is performed through a Transport SouthBound Interface (T-SBI), while the northbound API is insured via a Transport Management Service interface (T-MaS).

Besides, RAN radio resources are allocated and configured by means of an SDN controller for radio resource management, namely FlexRAN [53]. The latter provides an API for radio controlling over multiple gNBs through its Radio SouthBound Interface (R-SBI). The northbound API is insured via a Radio Management Service interface (R-MaS).

3.2.5 RAN slice allocation & orchestration

On top of this virtualized RAN architecture, we design and implement an algorithm ensuring the optimization of user-centric RAN slice allocation. As depicted in Figure 3.1, the optimization entity monitors the RAN state information including radio conditions (i.e., available spectrum, interference levels,

UE radio channel estimation), link state (i.e., available bandwidth) and server capacities (i.e., processing power) of all RAN sites. The aim is to elaborate an optimized RAN slice allocation decision that satisfies both UE throughput and latency demands while keeping a cost effective RAN deployment. Hence, based on the aforementioned parameters, an optimized RAN slice allocation is performed for each UE, by assigning the appropriate proportions of i) radio spectrum, ii) computational resources in the DU site, iii) computational resources in the CU site, and iv) bandwidth in the fronthaul network. This information is registered in the slice descriptor resource requirements and then triggers the Slice Manager entity through the Slice Management Service (S-MaS) interface.

The Slice Manager interacts with three resource management entities in order to deploy each user slice conforming to the slice descriptor specification. First, proportions of radio resources are allocated and configured by means of the Radio SDN controller. Second, the user processing resources are allocated both in DU and CU sites via the Processing Management Service interface (P-MaS). At this stage, baseband functions are instantiated into containerized network functions that can easily interact with each other and scale separately by mean of the Orchestrator. Third, the Slice Manager entity interconnects the gNB-CU and gNB-DU containers by programming the link bandwidth provisioning and latency control in the transport network. This is handled by the Transport SDN controller, leveraging its centralized and abstract network view.

For instance, and as we can see in Figure 3.1, our user-centric RAN Slice Allocation algorithm instantiates dynamically 3 user slices on top of the same physical infrastructure. In doing so, the algorithm decides to centralize PF_1 and PF_2, while keeping the below PFs at the access site for UE_1 generating an eMBB traffic. In this specific scenario, the Orchestrator will trigger the instantiation of 2 containers as shown in Figure 3.1. Note that PF_3 and PF_4 are common functions, requiring a common container for serving all users attached to gNB1. Then, only PF_1 is centralized in the cloud for UE_2 generating a high eMBB traffic. The aim is to reduce the data flow in the transport link. Meanwhile, only PF_4 is kept in the DU site for UE_3 generating a uRLLC traffic. Indeed, the functional split of PF_3-PF_4 interface requires a stringent transport delay which satisfies UE_3 latency requirement.

Our proposed RAN Slice Allocation algorithm exposes an Optimization Management Service interface (O-MaS), through which, the infrastructure provider, namely the MNO, can tune the RAN resource usage threshold, which affects the optimal split decision. In doing so, the RAN Management System component subscribes to the RAN state entity through an event driven interface. Thus, it can be notified if a resource usage amount exceeds a given threshold, for example, the amount of traffic in the transport network. When it is the case, the infrastructure provider decides to tune the optimization entity by penalizing the allocation in the transport network in order to reduce the link resource usage. Therefore, the adopted user-centric approach offers a high level agility and considerably optimizes the resource usage compared to a cell-centric approach.

We recall that our main objective is to maximize the total offered throughput for the users across the network, while minimizing the total deployment cost. To achieve our objective, we jointly optimize, for each user, i) the cell attachment and radio resource allocation, and ii) the functional split.

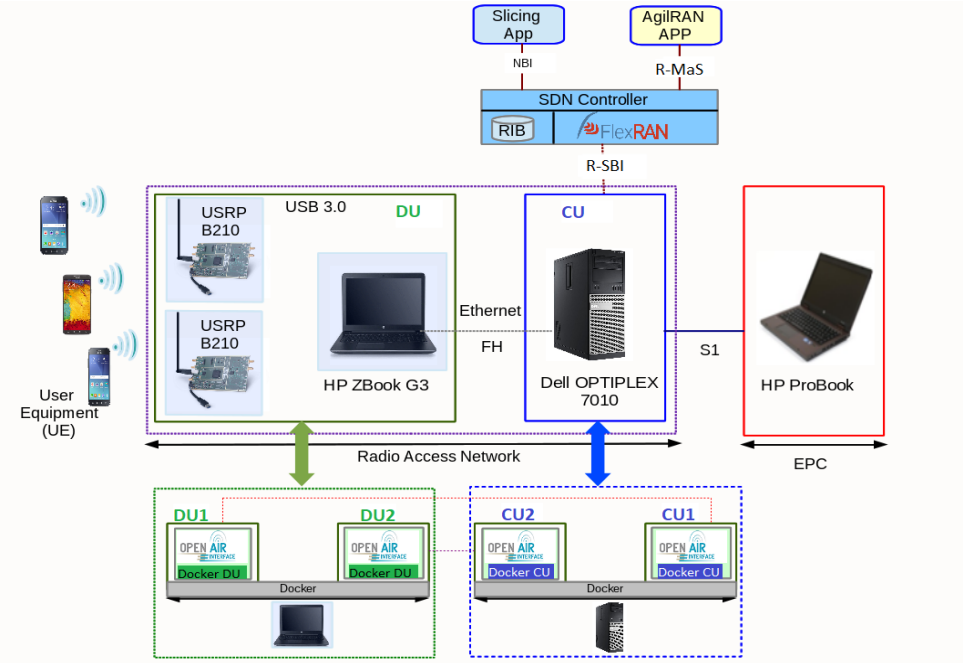


Figure 3.2: Testbed architecture

3.3 C-RAN prototype

3.3.1 Disaggregated C-RAN Infrastructure implementation

We validate the feasibility of our architecture in a C-RAN prototype based on the Open Air Interface (OAI) software [20] and FlexRAN SDN controller [53]. Note that the current version of our prototype supports up to 3 types of functional splits, referred to as LTE (option 1), IF4p5 (option 7c) and IF5 (option 8). Accordingly, these cell-centric configurations are considered for our performance evaluation.

Distributed Unit (DU): As illustrated in Figure 3.2, our prototype makes use of the OAI software and it is compliant with the 3GPP NG-RAN architecture, described previously in [Chapter 2, p. 14]. The DU consists of a "Ubuntu 14.04" laptop, equipped with a CPU Intel $i7 - 6500U$ 4-core (@2.50 GHz), a Random Access Memory (RAM) of 16 GB and 1 Gigabit Ethernet card.

Radio Unit (RU): The DU is connected to a Universal Software Radio Peripheral (USRP) B210 card [97] (hereinafter referred to as RU) via an USB 3.0 interface. By means of 2.4 GHz antennas co-located with the USRP card, the RU irradiates the 4G signal to the whole cell.

Central Unit (CU): The DU is in turn connected to CU through an Ethernet "category 5e" patch cable, supporting up to 1000 Mbps. The CU consists of a server with an Intel i7-3770 8-core (@3 GHz) CPU, a RAM of 16 GB and running with the same operating system as DU.

Evolved Packet Core (EPC): The CU is connected to a second laptop, running "Ubuntu 16.04", equipped with a CPU Intel $i5 - vPro$, 4-core (@2.5 GHz). The latter implements the functionalities of the Evolved Packet Core (EPC), according to the OAI software [20]. Our prototype is connected to 3 smartphones that act as Commercial Off-The-Shelf (COTS) users (UEs).

3.3.2 Container-based environment implementation

In order to enable the flexibility required by the proposed `AgilRAN` architecture, we virtualized the functions of the LTE protocol stack, by leveraging the Docker technology [28]. The latter is a tool that allows packing applications with all their dependencies in containers. They can share the kernel of the host operating system, while providing user space isolation. Such an isolation feature enables running multiple virtual DUs within the same host, bypassing the limit of the classic hardware-based implementation of OAI software. Moreover, DU and CU containers run directly on top of the kernel, letting us to match the strict performance requirements of the OAI software [20]. Indeed, we have quantified the average deployment time of a container-based PF to $1.8\ \mu s \pm 0.2\ \mu s$. Our prototype relies on 2 USRP cards, making it possible to instantiate a maximum of 2 DU containers at the same host, each connected to a different CU container.

3.3.3 RAN function placement

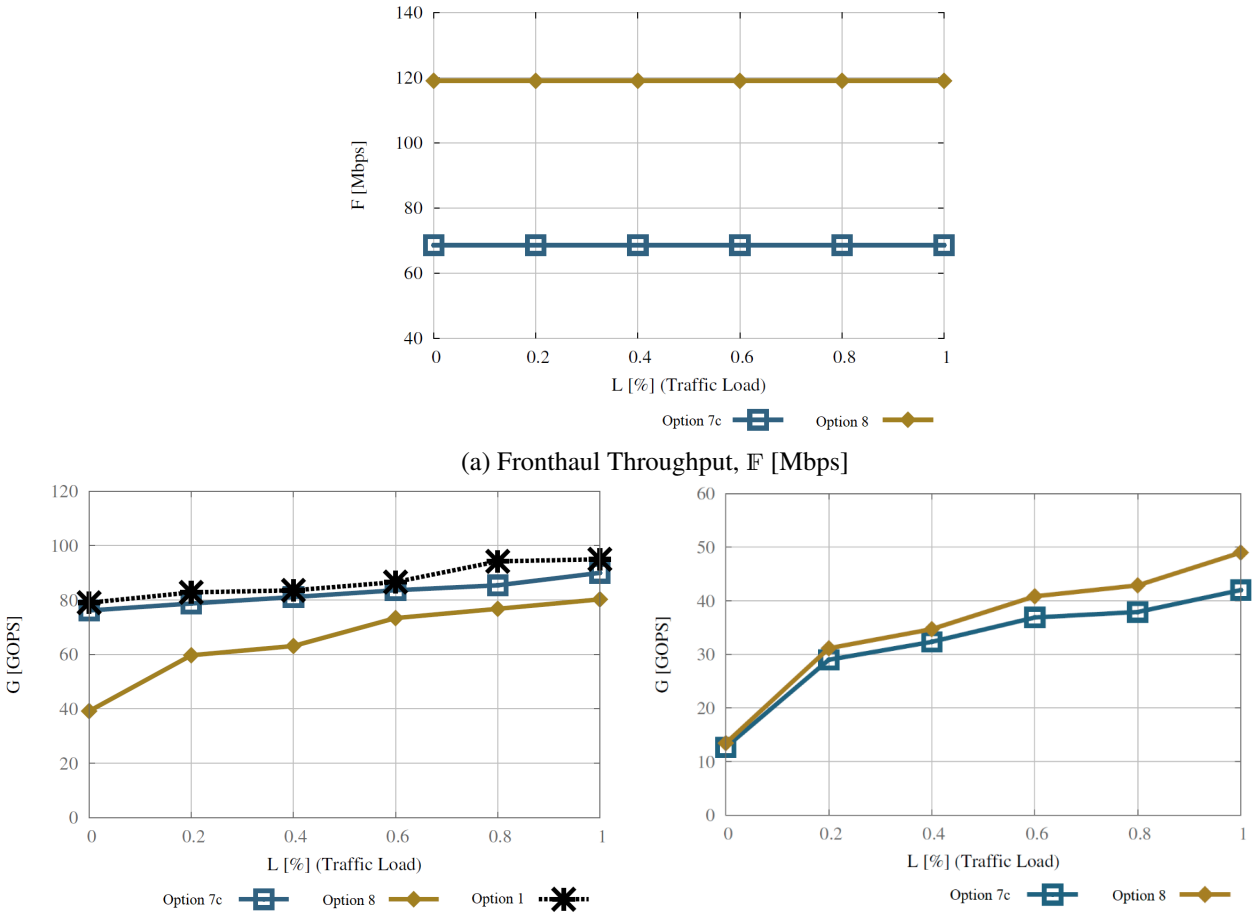
In order to implement the features of the `AgilRAN` architecture, i.e., enable on-demand functional split deployment, we store multiple images of OAI DU and CU at the DU and CU hosts, respectively, for each split configuration. As stated in [98], the Round Trip Time (RTT) of a flow transmission in a one-hop 1 Gbps ethernet-based fronthaul for 5 MHz bandwidth is 300 μs with compression and 550 μs without. Accordingly, the requirement delay of all deployed splits are respected with reference to [16]. According to the output of our RAN slice application, our prototype runs the appropriate instance of DU and CU images, connecting them via the appropriate fronthaul interface.

3.3.4 Implementation of radio control function

Note that our prototype leverages the SDN paradigm to enable remote allocation of the bandwidth resources in the CU host. In fact, as it can be seen from Figure 3.2, the CU node is connected to a specific SDN controller, named FlexRAN [53], through a Radio SouthBound Interface (R-SBI) using Google Protobuf [53]. By means of such an R-SBI interface, the FlexRAN Controller can easily interact with CU and hence, collect information about the RAN network state. Moreover, FlexRAN makes available a set of REST NorthBound Interfaces for Radio Management Service interface (R-MaS), which can be used to manage the RAN environment in an abstract way. From the execution point of view, we are constrained by some limitations of OAI software which does not support the dynamic configuration. In order to partially overcome such a limitation, we have deployed a script that shuts down and re-activate the OAI base station on-demand.

3.3.5 RAN slice allocation & orchestration implementation

On the top of FlexRAN, we have implemented 2 northbound applications, referred to as `AgilRAN APP` and `Slicing APP` respectively. The former implements the proposed RAN slice allocation optimization algorithm, which will be further detailed in Chapters 4, 5 and 6. The latter provides the Application Programming Interface (API) for configuring the radio bandwidth allocation process among different users in a centralized fashion.



(a) Fronthaul Throughput, $\mathbb{F}$ [Mbps]

(b) Computational Cost at DU , $\mathbb{G}$ [GOPS]

(c) Computational Cost at CU, $\mathbb{G}$ [GOPS]

Figure 3.3: Key Performance Indicators in our C-RAN Prototype

3.3.6 RAN resource consumption analysis for each functional split option

By means of experiments, we evaluate the impact of the different functional splits on the fronthaul traffic, computation cost and power consumption. We set a bandwidth of 5 MHz, while all the UEs are supposed to stream videos from a web-server for the whole duration of the experiments. We are interested in evaluating the impact of the allocated radio resources on the fronthaul traffic for each split configuration. To this end, we limit the upper bound of the available RBs at each TTI. This is made possible by using the SDN Slicing APP. Accordingly, we evaluate the KPI of our prototype by varying the upper limit of radio bandwidth utilization. We also define the following additional metrics.

- $\mathbb{F}$ refers the total fronthaul throughput measured in Mbps.
- $\mathbb{G}$ corresponds to the computational cost of DU or CU measured in GOPS.

Figure 3.3.(a) shows the average of one-way consumed bandwidth by the fronthaul interface for each functional split, by varying the aforementioned upper bandwidth limit. The fronthaul traffic is measured

by using the “nload" linux tool, that provides an average (over 300 ms) of the consumed bandwidth in a given network interface. As it can be seen, the fronthaul throughout is constant and traffic independent.

Figure 3.3.(b) and (c) show the impact of the users' traffic load on the computational cost G (i.e., CPU load) at DU and CU respectively, for each type of split. Note that the CPU load metric is made available by the “Docker stats" tool. From Figure 3.3.(b), it can be observed that the computation amount needed by Option 1 at DU is higher than the computation amount required by the Option 7c and Option 8 respectively. This is expected, since Option 7c and Option 8 assume to move a set of physical layer (PHY) functions from DU to CU. Interestingly, Figure 3.3.(b) shows that Option 8 at DU outperforms both Option 1 and Option 7c. Moreover, the Option 8 gain is higher in lower traffic load scenarios, while decreasing with higher traffic load. It is worth noting that different from the fronthaul bandwidth model, the computation model is load-dependent. The Option 8 requires an up to double amount of computation at DU when 100% of spectrum is used as compared to the scenario with no traffic.

Different from DU, the Option 8 requires more computation resources at CU than the Option 7c. This is expected since in the Option 8 case, more physical layer functions are moved to the Cloud. Note that Option 1 does not execute any functions at CU, therefore the cost impact of Option 1 at CU will be considered null in our case.

3.4 Conclusion

The massive adoption of Cloud technology, virtualization techniques and SDN in mobile access networks has driven the operators and vendors to work together in order to make the Radio Access Network (RAN) ecosystem more agile. In this respect, we put forward `AgilRAN`, a flexible RAN architecture which enables a user-centric on-demand RAN slice allocation, while considering the UE stringent requirements and RAN network state. Thanks to `AgilRAN`, baseband processing chain is virtualized and splitted in a fine-grained manner. The disaggregated basesband processing is then deployed while minimizing jointly the power consumption and fronthaul traffic. Besides, to assess the feasibility of our approach, we implement `AgilRAN` in an experimental C-RAN platform, based on Open Air Interface (OAI) and FlexRAN SDN controller. In the next Chapter, we will give insight into our proposed user-centric functional split orchestration scheme, which optimizes the allocation and placement of baseband functions.

CHAPTER 4

USER-CENTRIC FUNCTIONAL SPLIT ORCHESTRATION IN CLOUD RAN

Contents

4.1 Introduction

In this Chapter, we put forward a user-centric functional split orchestration solution aiming to optimize the placement of baseband functions, while dealing with temporal load variation of users. We propose,

in a first step, to address the aforementioned problem in the scope of one cell. Then, by enabling the selection of a functional split for each type of traffic, data rates in the fronthaul link and computational requirements in each site become more tunable, which is key to build cost effective RAN deployment solutions.

Our objective is to jointly minimize the fronthaul bandwidth and the computational resource consumption at both DU and CU sites. This leads to tackle a placement problem with contradictory goals. From one side, the more baseband functions are centralized, the more energy efficiency and hence, CAPEX and OPEX will be reduced. From the other side, high degree of centralization will increase the fronthaul traffic, thus, resulting in higher connection costs. Therefore, we model the user-centric functional split problem as an Integer Linear Problem (ILP), which minimizes the network deployment cost in terms of computational and link resource usage. Our aim is to ensure the best trade-off between baseband function centralization and fronthaul network consumption.

With high dense of user traffic, the resolution time becomes intractable. In order to operate in a polynomial time, we propose a heuristic based on Swarm Particle Optimization approach [99], denoted as `SPLIT-HPSO`. The algorithm consists in generating initially a set of potential solutions. Then, iteratively, the candidate solutions collaborate and evolve towards the best global solution. Our scheme is proved to be scalable, running within four Transmission Time Interval (TTI) units, which makes our solution operational. We expect that the optimization process is triggered periodically to optimize the deployment cost in a pro-active manner. Fourth, we validate this proposal using our experimental C-RAN prototype detailed in Chapter 3 to enable dynamic configuration of functional splits, according to the outputs of `SPLIT-HPSO`.

The remainder of this Chapter is organized as follows. In Section 4.2, we present the adopted set of functional split options. In Section 4.3, we describe our ILP formulation, which aims at minimizing jointly the bandwidth and computational resource allocation. In Section 4.4, we detail the proposed heuristic `SPLIT-HPSO`, while a description of our simulation and experimental environments and major results are presented in Section 4.5.

4.2 Functional split model

Hereafter, we detail the adopted functional split options depicted in Figure 4.1. We refer to the disaggregated 3GPP RAN model detailed previously in [Chapter 2, p. 15]. Note that Option 2, Option 3, Option 4 and Option 5, are not taken into account at this stage. This is due to the fact that the computational model for those aforementioned splits are still under investigation. Therefore, we consider the following splits: Option 1, Option 6, Option 7a, Option 7b, Option 7c and Option 8. Besides, it is worth mentioning, that an additional split is considered in our work (i.e., $split_6$), which centralizes the Parallel to Serial conversion and CPRI encoding function, in case of using a fiber connection between DU and CU. The latter split is detailed by the Small Cell Forum (SFC) in [16].

It is interesting to see that PF_1, PF_4, PF_5 and PF_6 are Cell-centric Processing Functions (CPF), while PF_1 and PF_2 are User-centric Processing Functions UPF. Consequently, $split_0$, $split_4$, $split_5$ and $split_6$ are cell-centric splits, while $split_1$, $split_2$ and $split_3$ are user-centric splits.

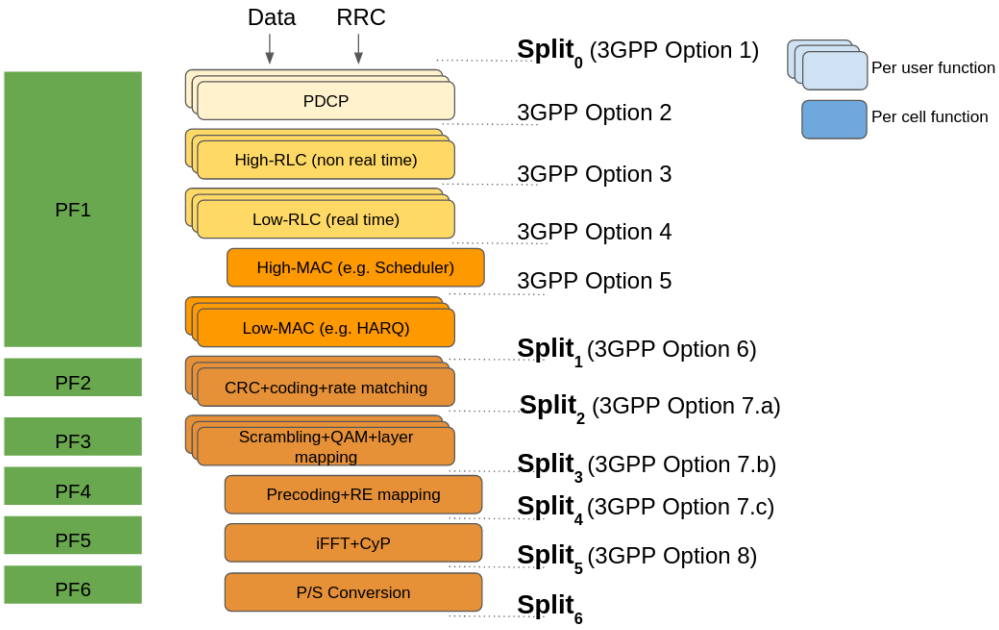


Figure 4.1: Adopted functional split options

4.3 Problem Formulation

In this Section, we formulate the user-centric functional split orchestration problem. First, we describe the computation model of each processing function and their corresponding fronthaul bandwidth requirements. Afterwards, we describe the power consumption model characterizing both DU and CU sites. Finally, we detail the problem formalization based on an ILP model.

4.3.1 Computational resource requirement Model

In order to quantitatively study the computational requirement for each split, we refer to the conducted analysis in [83] expressing the amount of computational resources, g_i, for each PF_i in Giga Operations Per Second (GOPS) as follows:

$$PF_1 : g_1 = G_1^{ref} . \frac{A}{A_{ref}}$$

$$PF_2 : g_2(M_i, L_i) = G_2^{ref} . \frac{B}{B_{ref}} . \frac{M_i}{M_{ref}} . \frac{A}{A_{ref}} . \frac{L_i}{L_{ref}}$$

$$PF_3 : g_3(L_i) = G_3^{ref} . \frac{B}{B_{ref}} . (\frac{A}{A_{ref}})^2 . \frac{L_i}{L_{ref}}$$

$$PF_4 : g_4(\{L_i\}) = G_4^{ref} . \frac{B}{B_{ref}} . \frac{A}{A_{ref}} . \sum_{i}^{Users} \frac{L_i}{L_{ref}}$$

$$PF_5 : g_5 = G_5^{ref} . \frac{B}{B_{ref}} . \frac{A}{A_{ref}}$$

$$PF_6 : g_6 = G_6^{ref} . \frac{B}{B_{ref}} . \frac{A}{A_{ref}}$$

It is worth noting that G_i^{ref} refers to the PF_i's GOPS value in the reference scenario. This constant is multiplied by scaling parameters which includes the carrier bandwidth (B), number of antennas (A), user traffic load (L) and modulation (M). As mentioned earlier, PF_5 and PF_6 are cell-centric for time-domain along with PF_1 that corresponds to a platform control processing. Hence, their computational requirement is load independent. In contrast, PF_2, PF_3 and PF_4 are processing in frequency domain so they take into account only frequency carriers having data signals, which make them load dependent.

4.3.2 Fronthaul bandwidth requirement Model

As for the bandwidth requirement model for each type of split, we refer to [16] that quantitatively estimates the bandwidth on the fronthaul link as follows:

$$
\begin{aligned}
&Split_0 : f_0(\{L_i\}) = \alpha_0. \sum_{i}^{Users} L_i \\
&Split_1 : f_1(L_i) = \alpha_1.L_i \\
&Split_2 : f_2(M_i, L_i) = \alpha_2(M_i).n_{RB}.L_i + \beta_1 \\
&Split_3 : f_3(L_i) = \alpha_3.A.n_{RB}.L_i + \beta_2.A \\
&Split_4 : f_4 = \alpha_4.A.n_{RB} \\
&Split_5 : f_5 = \alpha_5.A.n_s \\
&Split_6 : f_6 = \alpha_6.A.n_s
\end{aligned}
$$

where the coefficients α_i and β_i are constants for the model with reference to [16]. n_{RB} corresponds to the total number of Resource Blocks (RBs) and n_s refers to the sampling rate. We recall that $Split_0$ refers to the deployment scheme, where all PFs are fully decentralized at DU. In contrast, $Split_6$ corresponds to the conventional C-RAN, where all PFs are fully centralized in CU. $Split_1$ corresponds to the placement of PF_1 at CU, while keeping PF_i, $2 \leq i \leq 6$ at DU. When $Split_2$ is triggered, only PF_1 and PF_2 are placed in the Cloud. $Split_3$ and $Split_4$ refer to the PF_3-PF_4 and PF_4-PF_5 splits, respectively. Finally, $Split_5$ corresponds to the instantiation of PF_6 at DU, while moving PF_i, $1 \leq i \leq 5$ to CU. We recall that $Split_4$, $Split_5$ and $Split_6$ are generated from cell-centric processing functions. The latter form a sequence of functions in the physical layer, where user signals are multiplexed, generating a constant bit rate in the fronthaul link.

4.3.3 Power consumption Model

Based on the reference model presented in [83], the calculated baseband complexity in GOPS can be multiplied by a technology-dependent factor P_f expressing the number of operations that can be performed per second and per Watt (W). This factor is equal to 40 GOPS/W for the reference case and default technology, i.e., 65 nm for General Purpose CMOS (Complementary Metal-Oxide-Semiconductor). Intuitively, we would place the processing functions in sites with less power cost. To do that, we characterize a deployment site by an energy efficient indicator that expresses its Power Usage Effectiveness value (PUE).

By denoting Pow_{in} as the input power for a given site and Pow_{out} as the output power after server processing, PUE is expressed as follows, $PUE = Pow_{in}/Pow_{out}$. Consequently, the smaller is the PUE values, the lower is the power consumption of IT resources, and hence the better is the site power.

4.3.4 User-centric functional split problem formulation

We consider a set of N users connected to one cell in the AgilRAN infrastructure. Each user generates a load L_i, which corresponds to an amount of RBs allocated in DL. We assume a set of K splits, as explained in Section 4.2. We characterize a DU by a computational capacity of C_{DU} GOPS, maximum of power consumption P_{DU} and Power Usage Effectiveness PUE_{DU}. We recall that one DU is dedicated to one cell. The DU is connected to CU, which is characterized by a computational capacity of C_{CU} GOPS, maximum of power consumption P_{CU} and Power Usage Effectiveness PUE_{CU}. The two sites are connected via a fronthaul link of capacity B Mbps. The amount of GOPS consumed at DU (respectively CU) for the split k of user $_i$ is denoted by D_i^k (respectively C_i^k). Besides, R_i^k corresponds to the fronthaul bandwidth generated by the split k of user i. Then, R_F denotes for the aggregated fronthaul traffic of all users.

Our aim is to minimize jointly i) the overall cost of baseband function placement defined as the sum of computational cost across sites and ii) the bandwidth consumption on fronthaul across virtual user traffic, which are contradictory objectives. To do so, a binary matrix X of optimal splits is generated where $x_i^k = 1$ when split k is selected for user i and 0 otherwise. In some cases, when a cell split is activated then all users should be affected to this split. Hence, we need to also model a set of cell splits K_{cell} as a subset of the total split options K. We define $y_j \ \forall j \in \{0, .., K_{cell}\}$ as a binary variable that takes the value 1 when a cell split is activated and 0 otherwise.

We model the problem of baseband function placement, while ensuring a trade off between computational and bandwidth consumption cost as an optimization problem. The latter is formulated as an ILP detailed hereafter:

$$\mathcal{LP}_0 : \text{Minimize} : \alpha.PUE_{DU}.\frac{P_D}{P_{DU}} + \beta.PUE_{CU}\frac{P_C}{P_{CU}} + \gamma.\frac{R_F}{B}$$

$$\text{subject to} : \sum_{k=0}^{K} x_i^k = 1 \ \forall i \in N \quad (1)$$

$$\sum_{j=0}^{K_{cell}} y_j \leq 1 \quad (2)$$

$$\sum_{i=1}^{N} x_i^j = N y_j \ \forall j \in K_{cell} \quad (3)$$

$$\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k R_i^k \leq B \quad (4)$$

$$\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k D_i^k \leq C_{DU} \quad (5)$$

$$\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k C_i^k \leq C_{CU} \quad (6)$$

$$x_i^k \in \{0,1\} \ \forall i \in N \ \forall k \in K \quad (7)$$

$$y_j \in \{0,1\} \ \forall j \in K_{cell} \quad (8)$$

$$\text{where : } P_D = (1/P_f).\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k D_i^k \quad (9)$$

$$P_C = (1/P_f).\sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k C_i^k \quad (10)$$

$$R_F = \sum_{i=1}^{N}\sum_{k=0}^{K} x_i^k R_i^k \quad (11)$$

Constraint (1) expresses that only one split can be selected for each UE_i. (2) denotes that at most one cell split can be possibly chosen. In (3), means all attached users should deploy the activated cell split. (4), (5) and (6) express the upper bound limit for the total generated rate in the fronthaul link, the DU and CU computational resource requirements, respectively. Afterwards, we calculate the total amount of consumed power in DU and CU, P_D and P_C respectively. Indeed, the total resource computational demand is divided by the Power factor P_f as shown in (9) and (10). The total generated rate on the fronthaul link is expressed as R_F in (11).

The objective function aims to find the trade off between the centralization level weighted by β and the CU PUE factor PUE_{CU} and between the decentralization level weighted by α and the DU PUE factor PUE_{DU}. It is worth noting that we take into account the traffic load on the fronthaul by calibrating the weighting factor γ. The latter can affect in its turn the computational and power requirement in both DU and CU. This is a contradictory goal that is optimized if we find the appropriate set of user splits.

4.4 Proposal: `SPLIT-HPSO` Algorithm

4.4.1 Particle Swarm Optimization Algorithm

In this Section, we solve the optimization problem of user-centric functional split formulated in the previous Section 4.3.4 using Particle Swarm Optimization Algorithm [99]. Indeed, our formulated problem is ILP, so it is nondeterministic polynomial hard at high scale number of users if the aim is to solve it directly with a general-purpose ILP solver [37]. Thus, we need to design an algorithm to solve it in a polynomial time by generating a near-optimal solution.

In this context, we make use of Particle Swarm Optimization Algorithm, which is a population-based stochastic approach. More specifically, once a set of random initial solutions are generated, there is a need of collaboration between them in order to share internal information and optimize the common objective function.

4.4.2 Particle Design

We propose hereafter an adaptive approach of the Particle Swarm Optimization Algorithm to solve our problem. The particles are candidate solutions that collaborate and evolve along the algorithm iterations in order to find the best solution optimizing the total deployment cost expressed in $\mathcal{LP}_0$. Each particle p; $p \in \{1, .., P\}$ is characterized by a position $\mathcal{SPLIT}_p$, a velocity $\mathcal{VL}_p$ and the local best visited position $\mathcal{SPLIT}_{Lbest,p}$. The first component (position) presents the candidate solution configuration. The second one (velocity) is the change vector that allows the particle to evolve to the next position. The third component (best local position) is to memorize the local best solution configuration made so far,

which is evaluated by its Cost, $C(\mathcal{SPLIT}_{Lbest,p})$, with reference to the objective function expressed in $\mathcal{LP}_0$. We define $\mathcal{SPLIT}_{Gbest}$ as the best solution configuration among all the best local solutions of particles $\mathcal{SPLIT}_{Lbest,p}$; $\forall\ p \in \{1, .., P\}$.

Algorithm 1: SPLIT−HPSO

Inputs: Users MCS with proportions of allocated RBs
Output: $\mathcal{S}_{opt}$ set of optimal user splits
Begin:
  for k in K_{cell} **do**
    if $C(k) < C(\mathcal{SPLIT}_{Gbest})$ **then**
      $\mathcal{SPLIT}_{Gbest} \leftarrow$ k
    end if
  end for
  for p = 1 to P **do**
    for i = 1 to N **do**
      $\mathcal{SPLIT}_{p,i} \leftarrow$ k ∈ random(K_{user})
      $\mathcal{VL}_{p,i} \leftarrow$ random($[-|K_{user}|, |K_{user}|]$)
    end for
  end for
repeat
  for p = 1 to P **do**
    if $C(\mathcal{SPLIT}_p) < C(\mathcal{SPLIT}_{Lbest,p})$ **then**
      $\mathcal{SPLIT}_{Lbest,p} \leftarrow \mathcal{SPLIT}_p$
    end if
    if $C(\mathcal{SPLIT}_{Lbest,p}) < C(\mathcal{SPLIT}_{Gbest})$ **then**
      $\mathcal{SPLIT}_{Gbest} \leftarrow \mathcal{SPLIT}_{Lbest,p}$
    end if
  end for
  for p = 1 to P **do**
    Update the velocity $\mathcal{VL}_p$ according to Algorithm 2
    for i = 1 to N **do**
      $\mathcal{SPLIT}_{p,i} \leftarrow \mathcal{SPLIT}_{p,i} + \mathcal{VL}_{p,i}$
    end for
  end for
until $iter = ITER_{MAX}$;
$\mathcal{S}_{opt} \leftarrow \mathcal{SPLIT}_{Gbest}$ is the optimal solution

When the algorithm is performed, each particle p iteratively collaborates with the others in order to define its new velocity component $\mathcal{VL}_p$. This process is formulated in equation (E1) where the new velocity $\mathcal{VL}_{new,p}$ is constructed based on the old velocity $\mathcal{VL}_{old,p}$ of previous iteration, $\mathcal{SPLIT}_p$, $\mathcal{SPLIT}_{Lbest,p}$ and $\mathcal{SPLIT}_{Gbest}$. In equation (E1), u_1 and u_2 are coefficients to improve the random nature of the evolution process. This is essential to ensure investigating all the search space before converging to the near-optimal configuration. Once the new velocity is determined, the new position $\mathcal{SPLIT}_p$ is updated according to equation (E2).

$$\begin{aligned} \mathcal{VL}_{new,p} =& \mathcal{VL}_{old,p} \cap [u_1 \otimes (\mathcal{SPLIT}_{Lbest,p} \ominus \mathcal{SPLIT}_p) \\ & + u_2 \otimes (\mathcal{SPLIT}_{Gbest} \ominus \mathcal{SPLIT}_p)] \qquad (E1) \\ \mathcal{SPLIT}_p =& \mathcal{SPLIT}_p \oplus \mathcal{VL}_{new,p} \qquad (E2) \end{aligned}$$

Considering the inherent characteristics of our problem, we divide the set of splits K into a subset of cell splits K_{cell} and a subset of user splits K_{user}. That is, $K = K_{cell} \cup K_{user}$ and $K_{cell} \cap K_{user} = \emptyset$. Specifically, in our proposed algorithm, a particle is designed as a matrix of $[N * K_{user}]$. Initially, each user is affected a random user split k in K_{user}, meaning that for particle p and user i, $\mathcal{SPLIT}_{p,i,k} = 1$ and $\mathcal{SPLIT}_{p,i,k'} = 0 \,; \forall\, k' \neq k$. The velocity of each particle is a vector of $[N]$ that calculates for each user, the number of transitions to meet the new user split. Assuming that a user is affected a user split $k = 3$. If the velocity component in particle p for user i is $\mathcal{VL}_{p,i} = -2$, then the new user split should be $\mathcal{SPLIT}_{p,i} = k + \mathcal{VL}_{p,i} = 1$. Based on this, we define the velocity space as $[-|K_{user}|, |K_{user}|]$ to limit the allowed split transitions.

4.4.3 Functional Split Orchestration based on Hybrid Particle Swarm Optimization SPLIT-HPSO

Our proposed algorithm works as follows. We first evaluate the cost $C(k)$ of each cell split $k \in K_{cell}$ and update the global best solution $\mathcal{SPLIT}_{Gbest}$ by choosing the cell split k with the lowest cost. In a second stage, we search for the best solution configuration among all possible user splits k in K_{user}. The final best solution is either a cell split from K_{cell} or a combination of user splits from K_{user}. The second stage is described as follows. Initially, each user i in particle p is assigned a random user split k and a velocity value $\mathcal{VL}_{p,i}$. Then, iteratively,

- Each particle p updates its local best solution $\mathcal{SPLITT}_{Lbest,p}$.
- The global best solution is updated accordingly.
- Each particle p updates its velocity $\mathcal{VL}_p$ according to Algorithm 2.
- Each particle p updates its new solution configuration $\mathcal{SPLIT}_p$ by considering the new calculated velocity.

The major steps of our proposed algorithm are described in Algorithm 1, where the procedure for implementing `SPLIT-HPSO` is giving.

4.4.4 Velocity update strategy

The velocity update for each particle in Algorithm 1 is roughly described and we propose to detail it in Algorithm 2. The velocity update is a complex step, where two phases are integrated: $exploitation$ and $exploration$. The first phase is expressed in $(\mathcal{SPLIT}_{Lbest,p} \ominus \mathcal{SPLIT}_p)$. Thanks to the latter, we determine the velocity update vector to move from current configuration $\mathcal{SPLIT}_p$ to the best local configuration $\mathcal{SPLIT}_{Lbest,p}$. The second phase is expressed in $(\mathcal{SPLIT}_{Gbest} \ominus \mathcal{SPLIT}_p)$. Similarly, we determine the velocity update vector to move from current configuration $\mathcal{SPLIT}_p$ to the best global configuration $\mathcal{SPLIT}_{Gbest}$. Note that these two phases are weighted by random values u_1 and u_2. Hence, one particle can move, in each iteration, towards its best local position with a probability u_1 or towards its global position with a probability u_2. The aim here is to not fall into a local optima. In

Algorithm 2: `Velocity Update`

Output: $\mathcal{VL}_{new,p}$ of particle p
Begin:
Generate u_1 and u_2 with $u_1 + u_2 < 1$
for $i = 1$ to N **do**
  if $u_1 > u_2$ **then**
    $\mathcal{VL}_{new,p,i} \leftarrow \mathcal{SPLIT}_{Lbest,p,i} - \mathcal{SPLIT}_{p,i}$
  else
    $\mathcal{VL}_{new,p,i} \leftarrow \mathcal{SPLIT}_{Gbest,i} - \mathcal{SPLIT}_{p,i}$
  end if
end for
Sort users descending with respect to user load.
for $i = 1$ to N **do**
  if $(\mathcal{VL}_{old,p,i} \neq \mathcal{VL}_{new,p,i})$ **then**
    Calculate $C_1 = C(\mathcal{SPLIT}_{p,i} + \mathcal{VL}_{new,p,i})$
    Calculate $C_2 = C(\mathcal{SPLIT}_{p,i} + \mathcal{VL}_{old,p,i})$
    if $(C_2 < C_1)$ **then**
      $\mathcal{VL}_{new,p,i} \leftarrow \mathcal{VL}_{old,p,i}$
    end if
  end if
end for

Algorithm 2, we propose a heuristic based velocity update that embeds a local optimizer to expedite the convergence. For each user, we evaluate the gain from keeping the actual user split and the gain from moving to the user split of the best particle. More specifically, if the cost of actual user split is lower than the cost proposed by the best solution, then such a split is kept. The new velocity component selects, for each user, either the actual split configuration or the user split of best particle, favoring the lowest deployment cost.

At the end, each candidate solution is evolved towards finding the best global configuration by applying the `SPLIT-HPSO` heuristic. In each iteration, we build a new and feasible configuration in a way that constraints are satisfied in each step. The runtime of the proposed approach is computed as $O(P\ N^2\ ITER_{MAX})$. Such an approach can further optimize the best solution and hence, fasten the algorithm convergence as will be shown in the simulation results.

4.5 Performance Evaluation

In this Section, we evaluate the performance of our `SPLIT-HPSO` proposal making use of both system-level simulations and an experimental platform based on OAI [20], presented in Chapter 3. In the following, we first define the performance metrics as well as the baselines used for performance comparison. Second, we detail the simulation environment and the generated results. To evaluate the efficiency of `SPLIT-HPSO`, we compare it with most prominent strategies in C-RAN. Third, we provide the details of our emulation environment. Finally, we describe the results of the experimental prototype.

4.5.1 Simulation Performances

4.5.1.1 Simulation Baselines and Performance metrics

To assess the effectiveness of our approach while increasing the number of end users, we compare it with a simplex algorithm, denoted by `optimal-split`, and different cell-centric configurations. It is worth noting that the simplex algorithm performs an exhaustive research to reach the optimal solution. To achieve its objective, it makes use of Branch-and-Cut algorithm after relaxing the integer variables of our ILP problem. Besides, 7 cell-centric configurations are considered for our performance evaluation: Distributed-RAN (D-RAN) corresponding to $Split_0$, Cloud-RAN (C-RAN) corresponding to $Split_6$, in addition to $Split_1$, $Split_2$, $Split_3$, $Split_4$ and $Split_5$. Note that these related strategies are detailed in [Chapter 2, p. 15].

Hereafter, we define the metrics used to gauge the performance of our proposal in the simulation environment.

- $\mathbb{C}$ corresponds to the total deployment Cost as defined in the objective function in Section 4.3.4: $\mathbb{C} = \alpha.PUE_{DU}.\frac{P_D}{P_{DU}} + \beta.PUE_{CU}\frac{P_C}{P_{CU}} + \gamma.\frac{R_F}{B}$.
- $\mathbb{P}$ corresponds to the percentage of deployed user splits and quantifies the rate of each type of split.
- $\mathbb{F}$ refers the total fronthaul throughput measured in Mbps.
- $\mathbb{T}$ measures the average computation time in milliseconds (ms) to solve one instance of the user-centric problem.

4.5.1.2 Simulation environment

We designed and implemented a Java-based discrete event simulator to evaluate `SPLIT-HPSO` performances, while varying the number of connected UEs. Besides, we integrated `SPLIT-HPSO` and the related splitting strategies: $Split_0$, $Split_1$, $Split_2$, $Split_3$, $Split_4$, $Split_5$ and $Split_6$. We compared the aforementioned strategies with respect to the defined performance metrics. Similarly to [83] and [31], we consider the following benchmark scenario: We consider one DU of capacity C_{DU} equals to 1060 GOPS and a PUE_{DU} equals to 2.3. CU has a capacity C_{CU} equals to 1060 GOPS and a PUE_{CU} equals to 1.5. Both DU and CU are connected via a fronthaul link of 1228.8 Mbps corresponding to the highest required bandwidth [16]. The network configuration is detailed in Table 4.1.

Furthermore, we consider that N static UEs, varying in the range of $[20; 100]$, are randomly placed within the coverage of one cell. For each UE, we calculate the MCS index I_{MCS}, the modulation order Q_m of UEs as stated in Table 4.2. During each algorithm execution, each UE generates a service demand, which consists of a) video traffic with a throughput varying in the range of $[2; 13]$ Mbps and b) web traffic varying in the range of $[0.032; 0.064]$ Mbps according to [100]. Note that UEs asking video traffic are considered as "high loaded" UEs. They are affected a higher amount of radio resources and their proportion may vary between 3% and 15%. In the same way, we define as "low loaded" UEs those asking for web traffic. We make use of a proportional fair scheduler to compute the number of RBs to be allocated for each UE in Downlink traffic. Moreover, we assume that the proportion of UEs asking for a video traffic R inside the cell may vary between 0% and 100%.

Table 4.1: Simulation parameters

Parameters	Values
G_1^{ref}, G_2^{ref},G_3^{ref},G_4^{ref}	200 GOPS, 20 GOPS, 10 GOPS
G_4^{ref}, G_5^{ref}, G_6^{ref}	30 GOPS, 80 GOPS, 720 GOPS
A_{ref}, A	1 antenna
Ref. Bandwidth (B_{ref})	20 MHz
Bandwidth (B)	20 MHz
Total number of RBs, n_{RB}	100 for data (PDSCH)
L_{ref}	1 (Full load)
(L)	variable in [0;1]
Transport blocks	1 TBS per sub-frame
Sampling rate	30.72 MHz
Headers per IP packet	PDCP(2 bytes), RLC(5 bytes), MAC(2 bytes)
DL FAPI overhead per UE	1.5 Mbps
Number of Res for PCFICH	$PCFICH_{REs}$ = 16
PHICH group	$PHICH_{REs}$ = 12
Aggregation level 4	$PDCCH_{REs}$ = 144

Table 4.2: Modulation Order

1-6	QPSK	2	0-9
7-9	16-QAM	4	10-16
10-15	64-QAM	6	17-28

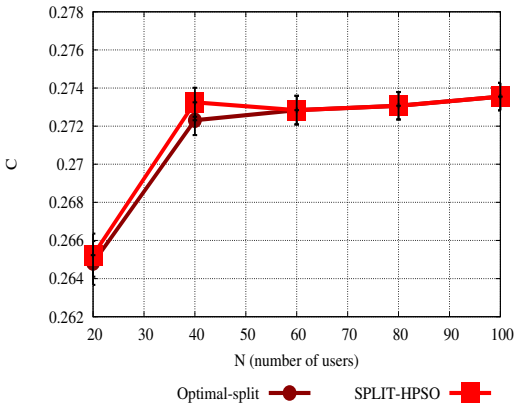
SPLIT−HPSO parameters are set to $ITER_{MAX}$ equals to 15 iterations and a population P of 10 particles. The coefficients u_1 and u_2 are uniformly distributed in $U(0, 0.2)$ and $U(0, 0.8)$, respectively. Note that we plot the average of 30 simulations with the confidence level set to 95%. Tiny confidence intervals are not shown in the following figures.

4.5.1.3 Simulation results

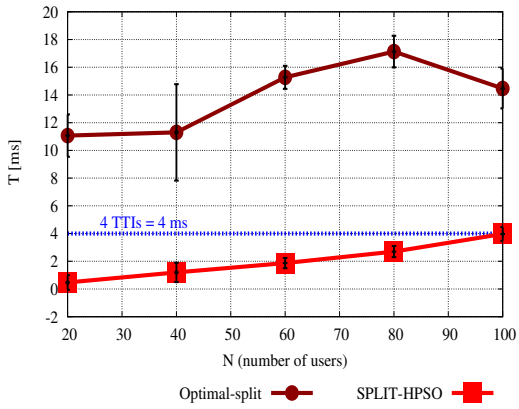
4.5.1.3.1 Convergence Analysis In what follows, we vary N in $[20; 100]$ with a rate of UEs generating a video traffic R = 50% in each iteration. We set the weights α, β and γ to the values 0.1, 0.1 and 0.8 respectively, which corresponds to a high deployment unit cost in the fronthaul link. We aim to evaluate the performance of SPLIT−HPSO in case of high density of UEs.

In Figure 4.2.(a), we compare SPLIT−HPSO to a simplex algorithm, which converges to the optimal solution. It is straightforward to see that our solution generates near optimal solutions when the number of UEs N is lower that 60. Within this range, the optimal solution provides a user split configuration that ensures a total cost, $\mathbb{C}$, lower than our proposed approach. Whereas, when N is higher than 60, our proposed approach achieves the same cost deployment of the optimal solution.

With regards to scalability, Figure 4.2.(b) illustrates the resolution time $\mathbb{T}$ of the different strategies versus the number of UEs N. Note that the Transmission Time Interval (TTI) in C-RAN is equal to 1



(a) Deployment Cost $\mathbb{C}$ (P = 10, $ITER_{MAX}$ = 15)



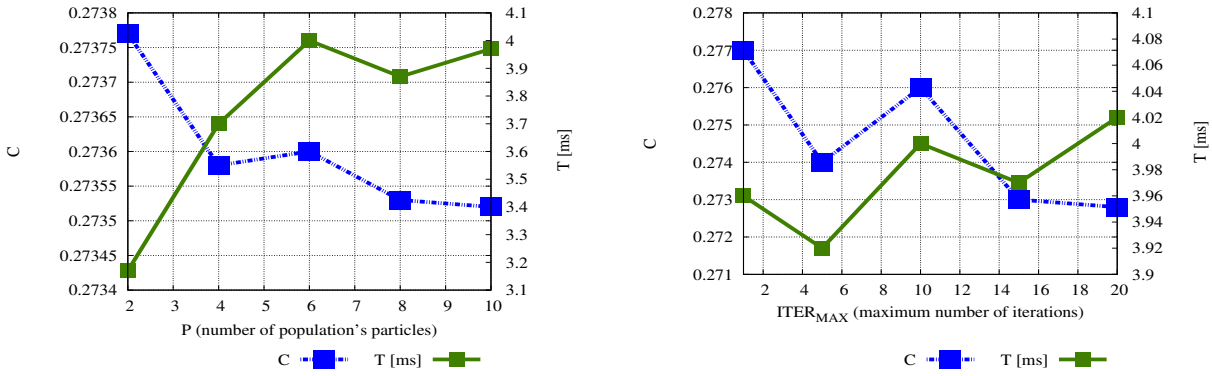
(b) Computation Time $\mathbb{T}$ (P = 10, $ITER_{MAX}$ = 15)

Figure 4.2: SPLIT−HPSO Convergence Analysis

millisecond according to [54]. It is straightforward to see that the non-scalable optimal solution takes a significantly longer time than SPLIT−HPSO to solve one instance of the optimization problem. Indeed, the optimal solution struggles to scale, as it takes values in $[14; 20]$ milliseconds to solve instances of N higher than 60 UEs. In contrast, SPLIT−HPSO can easily solve any size of instance (i.e., N in $[20, 100]$) in the range of $[0; 4]$ milliseconds. Eventually, by speeding up the computation time up to 5 orders of magnitude than Optimal-Split, SPLIT−HPSO is able to take an up-to-date decision and execute it after 4 TTI period. Unfortunately, Optimal-Split is not able to do so since its decision, once taken, will be already obsolete and hence not applicable.

Figure 4.3 evaluates the impact of the number of particles P and the number of iterations $ITER_{MAX}$ on the solution quality (i.e., the deployment cost). However, we should also take into account the resolution time. Indeed, our aim is to find the trade-off between the deployment cost and the resolution time. Figure 4.3.(a) assesses the efficiency of SPLIT−HPSO while varying the number of particles P. Indeed, for a fixed number of UEs (i.e., $N = 100$) and fixed number of iterations (i.e., $ITER_{MAX} = 15$), we can observe that the deployment cost $\mathbb{C}$ decreases when P increases. This proves that the size of P impacts the quality of the solution. However, it is straightforward to see that while increasing the number of particles, the computation time $\mathbb{T}$ increases. It is clear to see that when the number of particles P is higher than 8, the deployment cost $\mathbb{C}$ becomes stable. Such a behavior is predictable, as the solution quality is enhanced as soon as the number of particles is increased, which in turn, requires more computation time to solve the problem.

Figure 4.3.(b) assesses the convergence behavior of SPLIT−HPSO while varying the number of iterations $ITER_{MAX}$. Indeed, for a fixed number of UEs (i.e., $N = 100$) and a fixed number of particles (i.e., $P = 10$), we can observe that the deployment cost $\mathbb{C}$ decreases when $ITER_{MAX}$ increases. This proves that the solution quality is iteratively enhanced, however, impacting the increase of the computation time $\mathbb{T}$. It is interesting to see that starting from $ITER_{MAX} = 15$, the deployment cost is lightly decreased while the computation cost is highly increased.



(a) N = 100, $ITER_{MAX}$ = 15 (b) N = 100, P = 10

Figure 4.3: Trade off between the Deployment Cost $\mathbb{C}$ and Computation Time $\mathbb{T}$ for SPLIT-HPSO

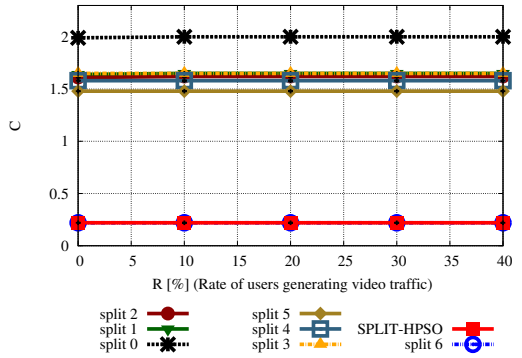
4.5.1.3.2 Scenario penalizing the power consumption We have performed extensive simulations in order to gauge the impact of the rate R of UEs generating video traffic on the split decision. Furthermore, we evaluated the impact of the weights α, β and γ on the total cost deployment $\mathbb{C}$ in order to analyze the trade off between link and power consumption. Baseline methods correspond to the different cell-centric configurations.

In what following, we assume that all the previously described parameters are kept static (i.e., P = 10 and $ITER_{MAX}$ = 15) during the simulation and only the rate R of UEs generating video traffic is varying. In this scenario, the number of UEs is fixed to 50 and R is varying in $[0\%; 100\%]$. The weights α, β and γ are set to 0.8, 0.1 and 0.1 respectively.

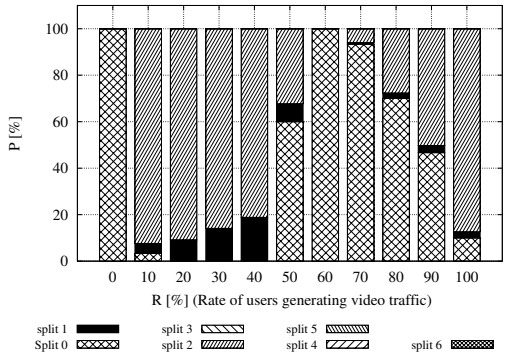
In order to emphasize the gap between our proposal and the related strategies, we evaluate, in Figure 4.4 (a), the deployment cost $\mathbb{C}$ while increasing the rate of video UEs R. We notice that SPLIT-HPSO achieves the same deployment cost as $split_6$. Besides, it reduces this cost by 85.14% compared with the second strategy corresponding to $split_5$. This can be explained by the fact that the increase of the weight α considerably impacts the deployment cost $\mathbb{C}$ in DU. Consequently, a fully centralized deployment will achieve the best performances.

4.5.1.3.3 Scenario penalizing the link consumption In the same way, we assume that all the previously described parameters are kept static (i.e., P = 10 and $ITER_{MAX}$ = 15, N = 50) during the simulation and only the rate R of UEs generating video traffic is varying in $[0\%; 100\%]$. We set the weights α, β and γ to 0.1, 0.1 and 0.8 respectively. We aim to analyze the split selection strategy of SPLIT-HPSO when the unit cost of the fronthaul link is high.

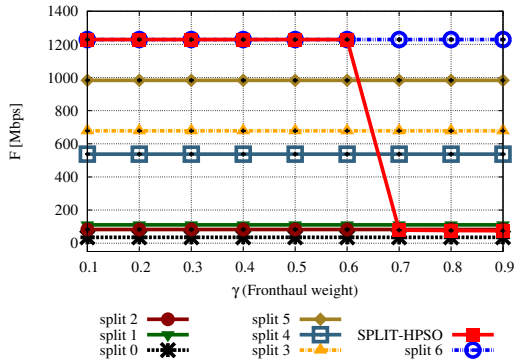
Figure 4.4.(b) depicts the percentage P of deployed user splits versus the rate R of UEs generating video traffic. It is straightforward to see that SPLIT-HPSO selects $split_0$ when there is no users requiring video traffic. In this configuration, the load served for the existing UEs is distributed almost equally. SPLIT-HPSO opts for $split_0$ to lower the costly traffic in the fronthaul link. Whereas, when R is in $[10\%; 40\%]$, we notice that $Split_2$ is predominantly selected to serve the UEs generating web traffic, while $split_1$ is selected to serve UEs generating video traffic. Note that, $split_0$ can be selected when R = 10%. Moreover, when the R exceeds 50%, the competition on radio resources intensifies and the radio



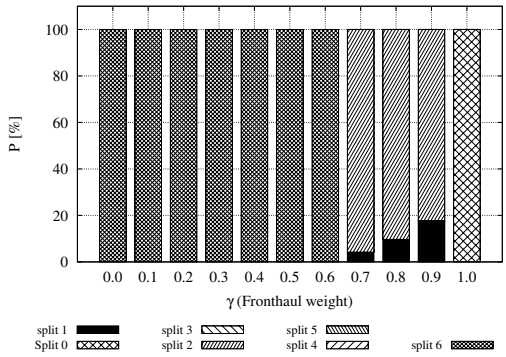
(a) Deployment Cost $\mathbb{C}$ ($\alpha = 0.8, \beta = 0.1$, $\gamma = 0.1$)



(b) Percentage of splits $\mathbb{P}$ [%] ($\alpha = 0.1, \beta = 0.1$, $\gamma = 0.8$)



(c) Fronthaul Throughput $\mathbb{F}$ [Mbps] ($\beta = 0.1$, $R = 50\%$)



(d) Percentage of splits $\mathbb{P}$ [%] ($\beta = 0.1$, $R = 50\%$)

Figure 4.4: SPLIT−HPSO Performance evaluation ($N = 50$)

resource becomes scarce. Hence, the served load for UEs generating video traffic is reduced and the load distribution becomes almost equal between all the cell's UEs. Eventually, the $split_0$ is frequently selected when the load distribution is equal among UEs.

Indeed, $split_0$ is a default solution to lower the costly link consumption. When the load distribution is not equal among UEs, the computational consumption of some user processing functions UPF will grow accordingly. Such fact will increase the DU deployment cost being weighted by α. Hence, $split_0$ is no more a cost effective solution as it deploys all the PFs for high loaded users at DU. To contract this side-effect, our algorithm finds a satisfactory trade off where it affects both $split_1$ and $split_2$ to its UEs as following. $Split_1$ is deployed priory for UEs with higher loads. $Split_2$ is deployed for UEs with low loads.

4.5.1.3.4 Trade off between Power and Link consumption

We assume that all the previously described parameters are kept static (i.e., $P = 10$ and $ITER_{MAX} = 15$, $N = 50$) during the simulation. R is fixed to 50. We aim to understand the trade off between power and fronthaul link consumption. To achieve our objective, we assume that the CU power consumption weight β is fixed as low as possible to the value 0.01 as data centers are natively efficient in power consumption. We assume that γ is increasing in the range of $[0, 1]$ while α is decreasing in the range of $[0, 1]$. We aim to generate the according split decision and analyze the solution in terms of deployment cost $\mathbb{C}$.

As depicted in Figure 4.4.(c), the link consumption $\mathbb{F}$ is stationary for cell splits. Figure 4.4.(d) shows that our solution adopts $split_6$ when the fronthaul weight is lower than 0.6 meaning the DU weight is higher than 0.3. Then, when the fronthaul consumption weight γ is higher than 0.6, the algorithm adopts mainly $split_1$ and $split_2$ until γ reaches 0.9 in order to minimize the traffic in the fronthaul. This explains the important decrease in the link consumption shown in Figure 4.4.(c). $Split_1$ is attributed to UEs generating video traffic while $split_2$ is attributed to UEs generating web traffic. When $\gamma = 1$, The fronthaul consumption is highly penalized which explains the adoption of $split_0$ is this case.

4.5.2 Experimental Evaluation

4.5.2.1 Experimental Baselines and Performance metrics

We validate the feasibility of our approach in a C-RAN prototype based on OAI. Note that the current version of our prototype supports up to 3 types of splits, referred to as LTE ($split_0$), IF4p5 ($split_4$) and IF5 ($split_5$). Accordingly, these cell-centric configurations are considered for our performance evaluation. We also define the following additional metric.

- $\mathbb{D}$ refers to the served throughput measured in Mbps.

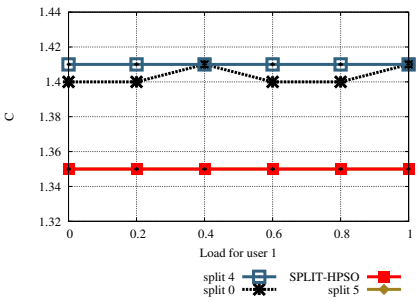
4.5.2.2 Experimental results

We expose here the results of a set of experiments conducted in our prototype to validate the feasibility of the proposed approach. Note that our prototype makes use of a 5 MHz carrier bandwidth, that is shared among 2 RUs, while using SISO antenna mode. In the baseline scenarios (i.e., $split_0$, $split_4$ and $split_5$), we assume that a static split is deployed for all the UEs. Moreover, in a scenario with 1 RU, only 1 split can be selected by our solution `SPLIT-HPSO`, while 2 different splits can be selected in a scenario with 2 RUs. In what follows, 3 static smartphone UEs, are statically located in the proximity of the RUs, while we assume the UE load varies as follows:

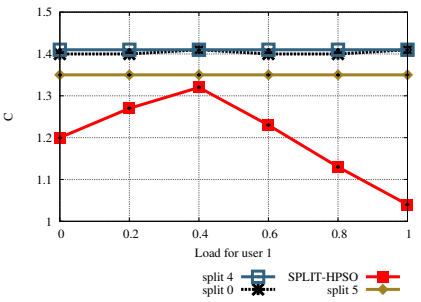
1. Load for UE 1 proportionally increases, getting 0%, 20%, 40%, 60%, 80%, 100% of the available RBs.

2. UE 2 and UE 3 shares the remaining RBs with $2/3$ for UE 2 and $1/3$ for UE 3.

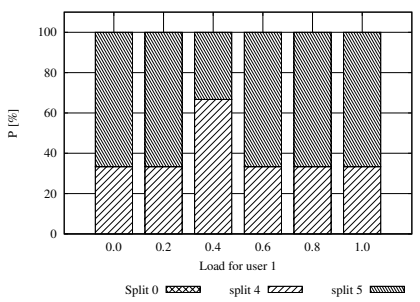
4.5.2.2.1 Scenario penalizing the power consumption We set the weights α, β and γ to 0.6, 0.1 and 0.3, respectively. Figure 4.5.(a) shows the deployment cost $\mathbb{C}$ for each scheme when only one RU is deployed. In this scenario, our solution opts for $split_5$ to lower the deployment cost $\mathbb{C}$. Indeed, the high value of α makes the deployment at DU a costly solution. Figure 4.5.(b) shows that our solution can further lower the deployment cost $\mathbb{C}$ in a scenario with 2 RUs. Indeed, it can be observed from Figure 4.5.(c) that our solution chooses mainly $split_4$ and $split_5$ to migrate more functions to CU. The $split_0$ is excluded here as it is evaluated as a costly solution. Figure 4.5.(d) shows the average throughput in downlink (DL) of all UEs. Note that UE throughput has been measured by the Android tool “Simple System Monitor" running at the UE smartphone. Looking at the baseline scenarios, we observe that $split_4$ offers a better throughput performance. Consequently, when `AgilRAN` opts for $split_5$ (i.e. when



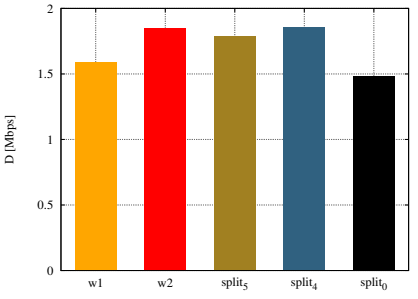
($\alpha = 0.6, \beta = 0.1\ , \gamma = 0.3$)
(a) Deployment Cost $\mathbb{C}$ (1 RU)



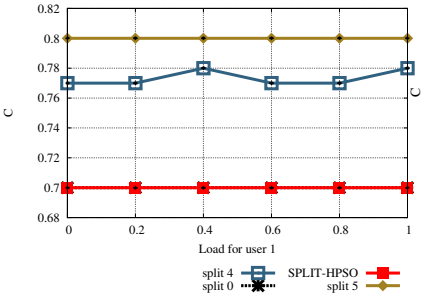
($\alpha = 0.6, \beta = 0.1\ , \gamma = 0.3$)
(b) Deployment Cost $\mathbb{C}$ (2 RUs)



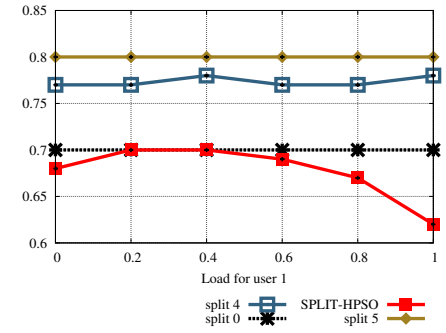
($\alpha = 0.6, \beta = 0.1\ , \gamma = 0.3$)
(c) Percentage of splits $\mathbb{P}$ [%] (2 RUs)



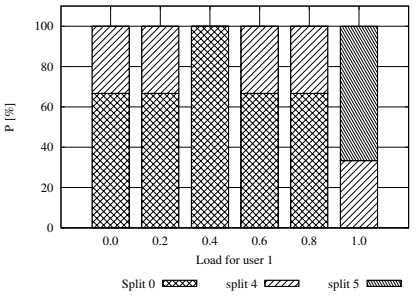
($\alpha = 0.6, \beta = 0.1\ , \gamma = 0.3$)
(d) DL UE Throughput $\mathbb{D}$ [Mbps]



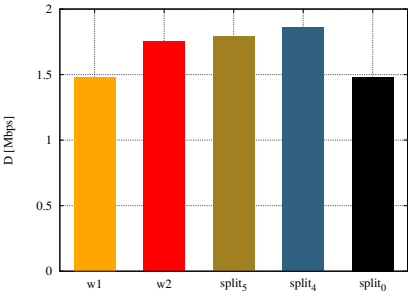
($\alpha = 0.3, \beta = 0.1\ , \gamma = 0.6$)
(e) Deployment Cost $\mathbb{C}$ (1 RU)



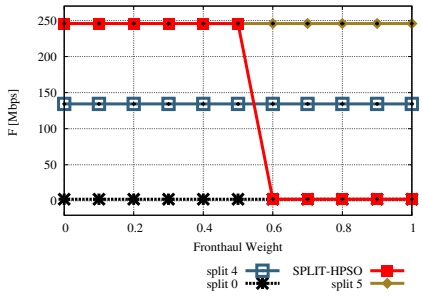
($\alpha = 0.3, \beta = 0.1\ , \gamma = 0.6$)
(f) Deployment Cost $\mathbb{C}$ (2 RUs)



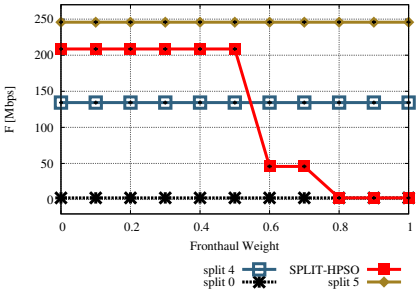
($\alpha = 0.3, \beta = 0.1\ , \gamma = 0.6$)
(g) Percentage of splits $\mathbb{P}$ [%] (2 RUs)



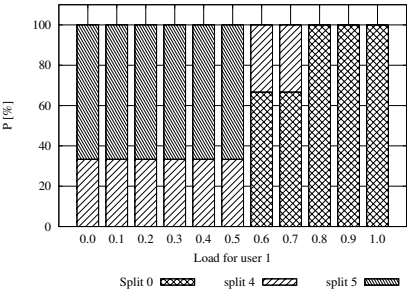
($\alpha = 0.3, \beta = 0.1\ , \gamma = 0.6$)
(h) DL UE Throughput $\mathbb{D}$ [Mbps]



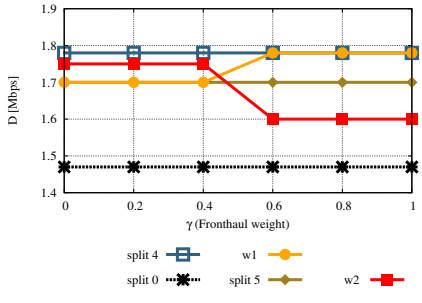
(i) Fronthaul Throughput $\mathbb{F}$ [Mbps] (1 RU)



(j) Fronthaul Throughput $\mathbb{F}$ [Mbps] (2 RUs)



(k) Percentage of splits $\mathbb{P}$ [%] (2 RUs)



(l) DL UE Throughput $\mathbb{D}$ [Mbps]

Figure 4.5: SPLIT-HPSO experimental evaluation

1 RU is deployed), the UE throughput is lower than the optimal baseline. However, when AgilRAN opts for $split_4$ for 2 UEs and $split_5$ for 1 UE, the served throughput is increased.

4.5.2.2.2 Scenario penalizing the link consumption We set the weights α, β and γ to 0.3, 0.1 and 0.6, respectively.

As it can be seen from Figure 4.5.(e), our solution opts for $split_0$ to lower the deployment cost $\mathbb{C}$. In fact, when the fronthaul link weight γ is high, the link consumption becomes a costly solution, which favors the decentralized scheme. From Figure 4.5.(f), it can be observed that when 2 RUs are deployed, $\mathbb{C}$ is load variable for SPLIT-HPSO which employs heterogeneous split decision. As shown in Figure 4.5.(g), our algorithm chooses mainly splits 4 and 0 to decrease the link consumption. It can be observed from Figure 4.5.(h), that the average UE throughput in AgilRAN is lower than the baseline scenarios, when 1 RU is used (w1). However, the performance of AgilRAN is significantly improved in a scenario with 2 RUs (w2), thanks to the reduced interference level compared to the scenario with only 1 RU.

4.5.2.2.3 Trade off between Power and Link consumption In this experiment, we assume that the load of each UE is fixed as follows: 0.8 for UE 1, 0.13 for UE 2 and 0.06 for UE 3, respectively. We assume that β is fixed as low as possible to the value 0.01, as data centers are natively efficient in power consumption. Accordingly, we assume that γ increases in the range $[0, 1]$, while α decreases in the same range.

As depicted in Figure 4.5.(i) and (j), the link consumption is constant for cell splits 4 and 5 as they are load independent and with a small variations for $split_0$, that is not visible here due to the large-scale. As shown in Figure 4.5.(i), AgilRAN adopts for $split_5$ till γ reaches 0.5, in case of 1 RU. Then, it adopts a full $split_0$ decision when γ is higher than 0.5. Figure 4.5.(k) shows that with 2 RUs, AgilRAN adopts both split 5 (2/3 of the time) and $split_4$ (1/3 of the time) till the γ reaches 0.5. Then, it gradually adopts $split_0$, till the γ reaches 0.8. Starting from this value, a full $split_0$ decision is made. Finally, Figure 4.5.(l) shows the average of UE throughput. As it can be observed, in a scenario with 1 RU (w1), the UE throughput increases with higher values of γ, while the opposite behavior is observed when 2 RUs are employed (w2). This is explained by the nature of the split decision, which favors $split_0$ for higher values of γ.

We recall that the goal of the aforementioned experiments is to validate the feasibility to implement the dynamic features of AgilRAN architecture in a real prototype, that in our first implementation does not take into account the radio resource allocation process. Therefore, the UE throughput is not optimized. We will take into account the radio allocation process in future works.

4.6 Conclusion

In this Chapter, we put forward our heuristic based approach, denoted as SPLIT-HPSO, to optimize the orchestration of user-centric functional splits, while considering both the requirements of its RAN resources and the capabilities of the Cloud infrastructure. Based on Particle Swarm Optimization, SPLIT-HPSO is scalable and achieves optimized user-centric split solution in a satisfactory time. Based on extensive simulations, we have shown that SPLIT-HPSO achieves good performances in terms of total deployment cost and resolution time. Besides, to assess the feasibility of our approach, we eval-

uated it on our C-RAN platform AgilRAN. Obtained results have proven that our solution ensures a fine-grained link and computational resource allocation while achieving a low deployment cost. In the next Chapter, we will give insight into our proposed user-centric RAN slice allocation scheme, which jointly optimizes radio, link and computational resource allocation in 5G Cloud RAN.

CHAPTER 5

HEURISTIC BASED USER-CENTRIC RAN SLICE ALLOCATION SCHEME IN CLOUD RAN

Contents

5.1 Introduction

In this Chapter, we put forward a user-centric RAN slicing scheme that provides suitable proportions of radio, link and computational resources for each User Equipment (UE). Our scheme fulfills each UE QoS requirement while considering the underlying RAN infrastructure state. Furthermore, in this Chapter, we propose to address the aforementioned problem in the scope of a multi-sited RAN infrastructure.

First, we model the user-centric RAN slice allocation problem as an Integer Linear Problem (ILP) with multi-objective function. In one hand, we aim to maximize the overall served throughput of users across the network through radio resource allocation. We make use of the regression linear method to approximate the final served user throughput. In the other hand, we aim to minimize the network deployment cost, while tuning the computational and link resource usage.

Considering that the above optimization problem is NP-Hard, the resolution time becomes intractable in case of high density of user traffic. In order to operate in a polynomial time, we propose a low-cost and efficient heuristic algorithm based on the Particle Swarm Optimization approach [99]. The algorithm consists in creating initially a set of potential allocation solutions. Then, iteratively, the candidate solutions collaborate and evolve towards a best global allocation solution.

The performance of `E2E-USA` is evaluated throughout extensive simulations using 3GPP eMBB and uRLLC traffic scenarios [8]. Obtained results show the effectiveness of our proposal in terms of scalability, QoS satisfaction and RAN deployment cost. We highlight the exploration and exploitation dilemma during the solution generation which is ruled by the ε-greedy approach. We expect that the optimization process is triggered periodically to optimize the user RAN slice allocation in a pro-active manner.

The organization of this Chapter is as follows. In Section 5.2, we enumerate the adopted set of functional split options. We detail, in Section 5.3, the user-centric RAN slice allocation model. In Section 5.4, we describe our proposed heuristic, while a description of our simulation and major results are presented in Section 5.5.

5.2 Functional split model

Hereafter, we detail the adopted functional split options depicted in Figure 5.1. We refer to the disaggregated 3GPP RAN model detailed previously in [Chapter 2, p. 15]. Note that Option 3 and Option 5 are not taken into account at this stage. This is due to the fact that, the computational model for those aforementioned splits are still under investigation. Therefore, we consider the following splits: Option 1, Option 2, Option 4, Option 6, Option 7a, Option 7b, Option 7c and Option 8. Also, note that an additional split is considered in our work (i.e., $split_6$), which centralizes the Parallel to Serial conversion and CPRI encoding function, in case of using a fiber connection between DU and CU. The latter split is detailed by the Small Cell Forum (SFC) in [16].

It is interesting to see that PF_3, PF_6, PF_7 and PF_8 are Cell-centric Processing Functions (CPF), while PF_1, PF_2, PF_3 and PF_4 are User-centric Processing Functions UPF. Consequently, $split_6$, $split_7$ and $split_8$ are cell-centric splits, while $split_0$, $split_1$, $split_2$, $split_3$, $split_4$ and $split_5$ are user-centric splits.

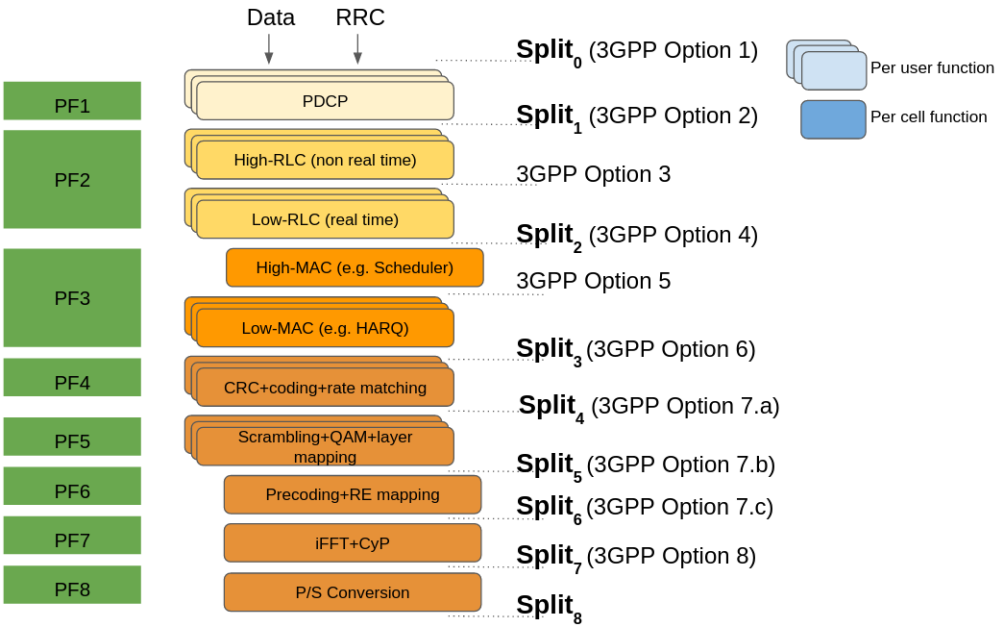


Figure 5.1: Adopted functional split options

5.3 Problem Formulation

5.3.1 Functional Split Model

In this Section, we formulate the user-centric RAN slice allocation problem. First, we describe the computation model of each processing function and their corresponding fronthaul bandwidth requirements. Afterwards, we detail the problem formalization based on an ILP model.

5.3.2 Computational resource requirement Model

In order to quantitatively study the computational resource requirement for each split in each RAN site, we refer to the conducted analysis of [83] and [101] expressing the amount of computational resources in Giga Operations Per Second (GOPS) consumed by PF. In what follows, we denote by g_k the computational requirement model of each processing function PF_k in DL direction:

$$PF_1 : g_1(L_i) = G_1^{ref} . \frac{A}{A_{ref}} . \boldsymbol{L_i} \quad (E1)$$

$$PF_2 : g_2(L_i) = G_2^{ref} . \frac{A}{A_{ref}} . \boldsymbol{L_i} \quad (E2)$$

$$PF_3 : g_3 = G_3^{ref} . \frac{A}{A_{ref}} \quad (E3)$$

$$PF_4 : g_4(Qm_i, L_i) = G_4^{ref} . \frac{W}{W_{ref}} . \frac{A}{A_{ref}} . \frac{\boldsymbol{Qm_i}}{Qm_{ref}} . \boldsymbol{L_i} \quad (E4)$$

$$PF_5 : g_5(L_i) = G_5^{ref} . \frac{W}{W_{ref}} . (\frac{A}{A_{ref}})^2 . \boldsymbol{L_i} \quad (E5)$$

$$PF_6 : g_6(\{L_i\}) = G_6^{ref} . \frac{W}{W_{ref}} . \frac{A}{A_{ref}} . \sum_{i}^{UEs} \boldsymbol{L_i} \quad (E6)$$

$$PF_7 : g_7 = G_7^{ref} . \frac{W}{W_{ref}} . \frac{A}{A_{ref}} \quad (E7)$$

$$PF_8 : g_8 = G_8^{ref} . \frac{W}{W_{ref}} . \frac{A}{A_{ref}} \quad (E8)$$

where G_k^{ref} refers to the PF_k's GOPS value in the reference scenario [83]. W is the carrier bandwidth, A is the number of antennas; L is the proportion of allocated Resource Blocks (RB) for UE_i and Qm is the QAM modulation. It is worth noting that, PF_7 and PF_8 are cell-centric for time-domain while PF_3 corresponds to the platform control processing. Hence, their computational requirement is load independent. In contrast, PF_1, PF_2, PF_4 and PF_5 perform in frequency domain, i.e., take into account only frequency carriers having data signals which make them load dependent.

5.3.3 Fronthaul bandwidth requirement Model

Assuming the model proposed in [16], we quantitatively study the bandwidth requirement for each functional split in the transport network. Accordingly, the generated traffic of each split interface in DL is estimated as follows:

$$Split_0 : f_0(L_i, Qm_i) = c_0(\boldsymbol{Qm_i}).A.B.\boldsymbol{L_i} \quad (E9)$$

$$Split_1 : f_1(L_i, Qm_i) = c_1(\boldsymbol{Qm_i}).A.B.\boldsymbol{L_i} \quad (E10)$$

$$Split_2 : f_2(L_i, Qm_i) = c_2(\boldsymbol{Qm_i}).A.B.\boldsymbol{L_i} \quad (E11)$$

$$Split_3 : f_3(L_i, Qm_i) = c_3(\boldsymbol{Qm_i}).A.B.\boldsymbol{L_i} + c_4 \quad (E12)$$

$$Split_4 : f_4(L_i, Qm_i) = A.B.(c_5 + c_6.A).\boldsymbol{L_i}.\boldsymbol{Qm_i} + c_7 \quad (E13)$$

$$Split_5 : f_5(L_i) = A.B.(c_8 + c_9.A).\boldsymbol{L_i} + c_{10}.A \quad (E14)$$

$$Split_6 : f_6 = c_{11}.A.B \quad (E15)$$

$$Split_7 : f_7 = c_{12}.A.n_s \quad (E16)$$

$$Split_8 : f_8 = c_{13}.A.n_s \quad (E17)$$

where coefficients $c_j, \forall j \in \{1, 13\}$, are constants for the model [16]. B corresponds to the number of RBs and n_s refers to the sampling rate. It is straightforward to note that when the centralization level of PFs increases, the computational requirement in the Cloud site increases accordingly which rises the amount of the circulating data flow in the transport link.

5.3.4 User-centric RAN slice allocation Problem

In this Section, we consider a multi-cell RAN system with M gNBs. Each gNB is characterized by a Distributed Unit (DU) located near the antenna unit and a Central Unit (CU) located at the Cloud site. The computational capacity of one DU,(respectively one CU) is denoted by C_{MAX}^D, (respectively C_{MAX}^C) Giga Operation Per Second (GOPS). We assume that a set of K functional splits can be deployed for N UEs. Then, we consider the amount of GOPS consumed by UE_i in DU (respectively in CU) of gNB_m when split k is deployed, is denoted by C_{imk}^D (respectively C_{imk}^C). By aggregating all the computational requirements, we define C_m^D, respectively C_m^C, as the total amount of GOPS consumed

at DU, respectively the CU, of gNB_m. The connection between both DUs and the CUs locations is maintained via an aggregated transport link with a capacity of R_{MAX} Mbps. Wherein, R_{imk} corresponds to the amount of data flow generated for UE_i attached to gNB_m with split k. We define also R as the aggregated link bandwidth generated by all UEs in M gNBs. Formally, C^D, C^C and R are variables expressed as linear functions of UE loads L. We recall that UE load $ld_{im}, \forall i \in N, \forall m \in M$ corresponds to the fraction of allocated RB for UE_i in gNB_m.

Our aim is to find the appropriate split k for each UE_i in gNB_m that minimizes the total deployment cost. Therefore, we define x_{im}^k as the binary variable, which is equal to 1 when split k is selected for UE_i in gNB_m and 0 otherwise. Then, we assume that the total available split options K can be divided into 3 subsets: K_c, K_{u_1} and K_{u_2}. K_c is the set of cell splits, namely splits $\{8, 7, 6\}$ according to Section 5.2. K_{u_1} is the first set of user splits, namely $\{0, 1, 2\}$ according to Section 5.2. Finally K_{u_2} is the second set of user splits, namely $\{3, 4, 5\}$, according to Section 5.2. Let y_m^k be the binary variable, $\forall k \in \{0, .., K\}$ and $\forall m \in \{1, .., M\}$, that takes value 1 if the split k is activated for any UE in gNB_m and 0 otherwise. We also define the binary variable u_1^m (u_2^m respectively) that takes value 1 if a user split in subset $\{0, 1, 2\}$, ($\{3, 4, 5\}$ respectively) is activated in gNB_m and 0 otherwise. We model the attachment of UE_i to gNB_m with a binary variable t_{im}. The latter is equal to 1 when UE_i is attached to gNB_m and 0 otherwise.

In what follows, we propose our model for RAN deployment cost minimization by optimizing the user split selection $x^\star$. We make use of the Big-M modeling [37] to linear the different constraints.

$$\mathcal{LP}_1 : \text{Min } \alpha \sum_{m=1}^{M} \frac{C_m^D}{C_{MAX}^D} + \beta \sum_{m=1}^{M} \frac{C_m^C}{C_{MAX}^C} + \gamma . \frac{R}{R_{MAX}}$$

$$\text{s.t. : } \quad t_{im} = \sum_{k=0}^{K} x_{im}^k \ ,\forall i \in N \ ,\forall m \in M \tag{1}$$

$$x_{im}^k . \nu_k \leq \nu_i \ ,\forall i \in N \ ,\forall m \in M \ ,\forall k \in K \tag{2}$$

$$\sum_{i=1}^{N} x_{im}^k \leq M_1 y_m^k \ ,\forall m \in M \ ,\forall k \in K \tag{3}$$

$$y_m^k \leq \sum_{i=1}^{N} x_{im}^k \ ,\forall m \in M \ ,\forall k \in K \tag{4}$$

$$\sum_{i=1}^{N} x_{im}^{k'} \leq \sum_{i=1}^{N} t_{im} + M_1 (1 - y_m^{k'}) \ ,\forall m \in M \ ,\forall k' \in K_c \tag{5}$$

$$\sum_{i=1}^{N} x_{im}^{k'} \geq \sum_{i=1}^{N} t_{im} - M_1 (1 - y_m^{k'}) \ ,\forall m \in M \ ,\forall k' \in K_c \tag{6}$$

$$\sum_{k=1}^{K_{u_1}} y_m^k \leq |K_{u_1}| u_1^m \ ,\forall m \in M \tag{7}$$

$$u_1^m \leq \sum_{k=1}^{K_{u_1}} y_m^k \ ,\forall m \in M \tag{8}$$

$$\sum_{k=1}^{K_{u_2}} y_m^k \leq |K_{u_2}| u_2^m \ ,\forall m \in M \tag{9}$$

$$u_2^m \leq \sum_{k=1}^{K_{u_2}} y_m^k \ ,\forall m \in M \tag{10}$$

$$\sum_{k=1}^{K_c} y_m^k + u_1^m + u_2^m \leq 1 \ ,\forall m \in M \tag{11}$$

$$R = \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{k=0}^{K} x_{im}^k R_{imk} \leq R_{MAX} \tag{12}$$

$$C_m^D = \sum_{i=1}^{N} \sum_{k=0}^{K} x_{im}^k C_{imk}^D \leq C_{MAX}^D \ ,\forall m \in M \tag{13}$$

$$C_m^C = \sum_{i=1}^{N} \sum_{k=0}^{K} x_{im}^k C_{imk}^C \leq C_{MAX}^C \ ,\forall m \in M \tag{14}$$

$$x_{im}^k \in \{0,1\} \ ,\forall i \in N \ ,\forall m \in M \ ,\forall k \in K \tag{15}$$

$$y_m^k \in \{0,1\} \ ,\forall m \in M \ ,\forall k \in K \tag{16}$$

$$u_1^m, u_2^m \in \{0,1\} \ ,\forall m \in M \tag{17}$$

The objective function in $\mathcal{LP}_1$ expresses the ability to tune the computational and link resource usage to minimize the RAN deployment cost while considering the infrastructure capacity and UE latency constraints. This can be done leveraging the user functional split that helps to find a trade-off between the centralization and decentralization levels of baseband functions. The first level of $\mathcal{LP}_1$ expresses the computational resource usage across DUs, weighted by α. The second level is expressed as the computational resource usage across CUs, weighted by β. In addition, third level of the objective function in $\mathcal{LP}_1$ expresses also tune the traffic in the aggregated fronthaul by calibrating the weighting factor γ.

Constraint (1) expresses that each attached UE_i in gNB_m can be assigned only one split. Constraint (2) denotes that the selected split k for UE_i in gNB_m should satisfy the latency required by UE_i. Constraint (3) activates the binary variable y_m^k when at least one user split k is activated in gNB_m. (4) expresses that when split k is deactivated for gNB_m, then no UE is assigned split k. (5) and (6) denote that the activation of one cell split k' in gNB_m, results in assigning split k' for all attached UEs. Constraints (7) and (8) activate the variable u_1^m when at least one split k in subset K_{u_1} is activated in gNB_m. Constraints (9) and (10) activate the variable u_2^m when at least one split k in subset K_{u_2} is activated in gNB_m. (11) denotes that for a given gNB_m, we may activate i) either one cell split k' in K_c or ii) a combination of user splits in K_{u_1} or iii) a combination of user splits in K_{u_2}. In (12), the total generated rate in the aggregated transport link should not exceed the link capacity R_{MAX}. (13) and (14) express that the total allocated computational resources in DU, respectively CU, of gNB_m must not exceed the total computational capacity C_{MAX}^D, respectively C_{MAX}^C.

5.3.5 Radio Resource Allocation Problem

In what follows, we consider N UEs statically located in a system of M gNBs with a frequency reuse factor of 1, i.e., the same set B of RBs are reused by each cell, which may induce interference on RB

level. Each UE_i , $\forall i \in N$, generates a flow of throughput λ_i and latency ν_i.

Considering the DL transmission direction, we calculate the Signal to Interference plus Noise Ratio (SINR) experienced by UE_i from gNB_m, $\forall i \in N, \forall m \in M$, expressed as following:

$$SINR_{im} = \frac{\overline{P_{gNB}} \cdot h_{im}}{I_{im} + \sigma^2}, \quad I_{im} = \sum_{m' \neq m} \overline{P_{gNB}} \cdot h_{im'} (E18)$$

where h_{im} denotes for the channel gain between each gNB_m and UE_i, I_{im} stands for the interfering power received by the UE_i from other gNBs $m' \neq m$. σ^2 is defined as the noise power, while we assume that every gNB transmits a static amount of power, denoted by $\overline{P_{gNB}}$. Based on the SINR average estimation, we calculate the Channel Quality Indicator (CQI), the Modulation and Coding Scheme (MCS) and the Transport Block Size Index (I_{TBS}) between UE_i and gNB_m, $\forall i \in N, \forall m \in M$.

In order to proceed to radio resource allocation, we rely on following binary variables t, w and L. We recall that t_{im} is a binary variable, which is equal to 1 if UE_i is attached to gNB_m and 0 otherwise. w_{imb} is equal to 1 if the RB b of gNB_m is allocated to UE_i and 0 otherwise. L_{im} expresses the radio load of UE_i in gNB_m corresponding to the fraction of total allocated RBs to UE_i in gNB_m: $L_{im} = \frac{\sum_{b=1}^{B} w_{imb}}{B}$. Once allocated, a resource block is affected an amount of radio power $\overline{p_{RB}}$.

Our aim is to find an attached gNB with the appropriate set of RBs in order to fulfill the throughput requirement λ_i of each UE_i. By means of the linear regression method [102], we propose an approximation of the served Transport Block Size, denoted by $\widetilde{TBS}$. With reference to the table 7.1.7.2.1-1 in [103], we calculate the linear approximation $\widetilde{TBS}$ according to each TBSI value as following:

$$\widetilde{TBS}(TBSI) = TBS^L(TBSI).B.\boldsymbol{L_{im}} + TBS^o(TBSI) \quad (E19)$$

where $B.\boldsymbol{L_{im}}$ is the supposed number of allocated RBs for UE_i in gNB_m. $TTBS^L(TBSI).B.\boldsymbol{L_{im}}$ is called the response that depends from user load and $TBS^o(TBSI)$ is called the predictor which is independent from user load. Consequently, we express the approximation of the final served throughput $\widetilde{r}_{im}$ for UE_i in gNB_m, $\forall i \in N, \forall m \in M$ as function of the linear approximation of the Transport Block Size $\widetilde{TBS}_{im}$ with a multiplication factor c_{34} for the conversion from bytes to bits per second:

$$\widetilde{r}_{im} = c_{14}.\widetilde{TBS}_{im} \quad (E20)$$

We define a second objective function $\mathcal{LP}_2$ that aims at maximizing the overall served user throughput across the network. This is achieved by finding for each UE_i, i) the best attached gNB m, $t^{\star}_{im}$ and ii) the best set of RBs $w^{\star}_{imb}$, while keeping a low interference level:

$$\mathcal{LP}_2 : \text{Max} \quad \sum_{i=1}^{N} \frac{\sum_{m=1}^{M} \widetilde{r}_{im}}{\lambda_i}$$

$$\text{s.t. :} \quad \sum_{m=1}^{M} t_{im} \leq 1, \forall i \in N \tag{18}$$

$$\sum_{b=1}^{B} w_{imb} \geq t_{im}, \forall m \in M, \forall i \in N \tag{19}$$

$$M_2 t_{im} \geq \sum_{b=1}^{B} w_{imb}, \forall i \in N, \forall m \in M \tag{20}$$

$$\sum_{m=1}^{M} \widetilde{r}_{im} \leq \lambda_i, \forall i \in N \tag{21}$$

$$\sum_{i=1}^{N} w_{imb} \leq 1, \forall m \in M, \forall b \in B \tag{22}$$

$$\sum_{m' \neq m} \sum_{i' \neq i} . \overline{p_{RB}} . h_{i'm'} . w_{i'm'b} \leq I^{MAX} + M_3 . (1 - w_{imb}), \forall i \in N, \forall m \in M, \forall b \in B \tag{23}$$

$$t_{im}, w_{imb} \in \{0, 1\}, \forall i \in N, \forall m \in M, \forall b \in B \tag{24}$$

Constraint (18) expresses that each UE should be attached at most to only one gNB. (19) specifies that UE_i can get more than one RB when it is attached to gNB_m. In (20), the total amount of allocated RBs to UE_i in gNB_m is constrained by the upper bound limit $M_2 = B$. In (21), the final served throughput for UE_i should be less than what is required with λ_i. In (22), each RB to only one UE. (23) expresses the interference constraint for each allocated RB, where I^{MAX} refers to the interference threshold and M_3 is a Big-M constant to tolerate interference on unallocated RBs.

To summarize, our user-centric slice allocation problem $\mathcal{LP}_3$ can be formulated as follows:

$$\mathcal{LP}_3 : \text{Max } \theta \mathcal{LP}_2 - \mu \mathcal{LP}_1$$
$$\text{s.t : } (1) - (24)$$

The objective is to find the trade-off between the total served user throughput expressed in $\mathcal{LP}_2$ weighted by θ and the total RAN deployment cost expressed in $\mathcal{LP}_1$ weighted by μ.

5.4 Proposal: `E2E-USA`: On-Demand RAN slice allocation approach

In this Section, we resolve the user-centric RAN slicing problem $\mathcal{LP}_3$, formulated in Section 5.3. The problem is classified as a nondeterministic polynomial hard problem [37]. This type of problem requires exhaustive search in the solution space in order to converge to optimal solutions. Hence, general purpose ILP solver [37] struggles to converge in case of high-scale of UE number. For this reason, there is a need for designing an heuristic to solve the formulated user-centric RAN slicing problem in a reasonable time with a near-optimal solution.

In this context, we propose an adaptive approach of the Particle Swarm Optimization (PSO) Algorithm [99], called `E2E-USA`, to solve our problem expressed in $\mathcal{LP}_3$. Our PSO based User-centric RAN Slice Allocation `E2E-USA` proceeds as follows: during the **`initialization`** stage, an initial set of feasible solutions is generated by affecting for each UE: i) attached gNB, proportions of RBs and ii) split selection. Then to solve the problem, our proposal proceeds iteratively on two folds. First, a better

radio allocation (i.e., $UE - gNB$ attachment and radio resource allocation) is explored. Second, UFSS algorithm is performed to find the optimal **split selection** for the already generated radio configuration in the first phase of current iteration.

5.4.1 Particle Swarm Optimization

Radio, computational and transport resource allocation impacts directly the end-user quality of service and the deployment cost. This type of problem is often known as Multi Objective Combinatorial Optimization Problem (MOCOP) that can be solved using algorithms based on the decomposition strategy in what we call Multi objective Evolutionary Algorithms based on Decomposition (MOEA/D). In doing so, the problem is decomposed into a set of single-objective subproblems using the weighted sum approach or others like the Tchebycheff approach, normal boundary intersection, etc. When solved, MOCOPs are generally nondeterministic polynomial complete or nondeterministic polynomial hard [37]. Thus, we need to design an algorithm to solve it in a polynomial time by generating a near-optimal solution.

In this context, Particle Swarm Optimization (PSO) approach is proposed as a population-based stochastic optimization algorithm inspired from birds foraging behavior. More specifically, PSO algorithm is characterized by an initial set of candidate solutions that collaborate to find the global optimum of the optimization problem. In practice and thanks to its inherent characteristics (i.e., fast computing speed and the parallel processing), swarm optimization algorithm for combinatorial optimization problem or what we call Set-based Particle Swarm Optimization S-PSO has been successfully applied in solving many problems like scheduling problem, vehicular routing problem, flow-shop scheduling problem, etc.

In order to solve our problem, we propose a set-based discrete particle swarm optimization based on decomposition by combining both approaches of MOEA/D and S-PSO.

5.4.2 Initialization Stage

Each particle p, $p \in \{1, .., P\}$, is characterized by a position $\mathcal{S}^p$, a velocity $\mathcal{V}^p$ and a best local position $\mathcal{S}^{L,p}$. The first attribute (i.e., position) corresponds to a candidate slice allocation solution. The second attribute (i.e., velocity) expresses the change vector that allows the particle to evolve to a next position. The third attribute (i.e., best local position) memorizes the best achieved local solution. Candidate solutions are evaluated through the utility function $\mathcal{U}_{\mathcal{F}}$ expressed in $\mathcal{LP}_3$. We also denote by $\mathcal{S}^G$, the best achieved solution among all best local solutions, $\mathcal{S}^{L,p}$, $\forall\ p \in \{1, .., P\}$.

In what follows, we design the position $\mathcal{S}^p$ of particle p as a 3-D matrix of $[N \times M \times (B+1)]$. The entry $\mathcal{S}^p_{imb}$ is a binary variable that takes the value 1 if UE_i is allocated the RB_b in gNB_m. Furthermore, we affect split k to UE_i in gNB_m. Formally, $\mathcal{S}^p_{im,(B+1)} = k$. The velocity component V^p of particle p is expressed as a 2-D matrix of $[N \times M]$. The entry $\mathcal{V}^p_{im}$ expresses the number of RBs to be added or removed in the next iteration for UE_i in gNB_m. Formally, $\mathcal{V}^p_{im}$ is in $[-B^{MAX}_{im}, +B^{MAX}_{im}]$, where B^{MAX}_{im} is the upper bound limit for UE_i allocation in gNB_m to satisfy his throughput λ_i.

Initially, each UE_i, $i \in \{1, .., N\}$ is attached to a random gNB_m, $m \in \{1, .., M\}$, with a random number n_{RB} of RBs. Meanwhile, we ensure that constraints (18-23) are satisfied. Constraint (18) leads to $\mathcal{S}^p_{im'b} = 0, \forall\ m' \neq m$. Constraint (19), (20) and (21) express that, $\mathcal{S}^p_{imb}$ can be positive n_{RB} times, where n_{RB} is in $[-B^{MAX}_{im}, +B^{MAX}_{im}]$. We privilege RBs suffering less interference level. Constraint (22) implies that $\mathcal{S}^p_{i'mb} = 0, \forall\ i' \neq i$. Finally UE_i is assigned a random split k and a random velocity $\mathcal{V}^p_{im}$ in

$[-B_{im}^{MAX}, +B_{im}^{MAX}]$.

5.4.3 RAN Slice Allocation based on Particle Swarm Optimization

Iteratively, each particle p evolves towards a new position $\mathcal{S}_p$ after updating its velocity $\mathcal{V}^p$ as stated in equation ($E21$). The velocity update process is formulated in equation ($E22$), where the new velocity is constructed based on the current velocity $\mathcal{V}^p$, current position $\mathcal{S}^p$, $\mathcal{S}^{L,p}$ and $\mathcal{S}^G$. Wherein, we integrate the coefficient ε to improve the random nature of the evolution process. More specifically, we define the first action, i.e., following the best local particle $\mathcal{S}^{L,p}$ with probability ε and a second action i.e., following the best global particle $\mathcal{S}^G$ with probability 1-ε.

$$\mathcal{S}^p = \mathcal{S}^p \oplus \mathcal{V}^p \quad (E21)$$

$$\mathcal{V}^p = \mathcal{V}^p \cap [\varepsilon \otimes (\mathcal{S}^{L,p} \ominus \mathcal{S}^p) + (1-\varepsilon) \otimes (\mathcal{S}^G \ominus \mathcal{S}^p)] \quad (E22)$$

More specifically, we adopt the ε-greedy method to alternate between following i) the best local particle $\mathcal{S}^{L,p}$ with probability ε and ii) the best global particle $\mathcal{S}^G$ with probability 1-ε. In doing so, the challenge is to find the balance between using local knowledge (exploitation) and investigating other options by following the global knowledge (exploration). We believe that this is essential to insure the investigation of the entire search space before converging to the near-optimal solution. `E2E-USA` is summarized in Algorithm 3.

5.4.3.1 Radio Resource Optimization

In this phase, we evaluate the utility function $\mathcal{U}_{\mathcal{F}}(\mathcal{S}^p)$ of each particle p and update $S^{L,p}$ and $\mathcal{S}^G$ accordingly. Afterwards, each particle p updates its velocity $\mathcal{V}^p$ and its new position $\mathcal{S}^p$. Then, our algorithm User Functional Split Selection, denoted by `UFSS` is performed to calculate the optimal split selection for each UE in particle p based on the newly generated radio configuration.

5.4.3.2 User-centric functional Split Selection based on Shortest Path Algorithm `UFSS`

In each iteration, once the $UE - gNB$ attachment and RB allocation is updated for particle p, our `UFSS` algorithm is executed to define the optimal split configuration. In what follows, we formulate the functional split selection optimal strategy as a shortest path problem. More specifically, we model all split possibilities as a Directed Acyclic Graph (DAG), G, with almost $N \times K$ nodes. Each node (m, i, k) is either a user split k for UE_i in gNB_m, or a cell split k for all UEs attached to gNB_m. Then, we consider only split node (m, i, k) which satisfy UE latency requirements, which receives links from other nodes with weights expressing the deployment cost from selecting the node (m, i, k). The weight of each ongoing link to node (m, i, k) is defined as:

$$\alpha \frac{C^D_{imk}}{C^D_{MAX}} + \beta \frac{C^C_{imk}}{C^E_{MAX}} + \gamma \frac{R_{imk}}{R_{MAX}} \quad (E23)$$

The graph G without link weights is depicted in Figure 5.2, for $M = 2$ $gNBs$, $N = 3$ UEs and $K = 3$ splits. Note that there are two extra nodes: s and f. S is a **s**tarting point, that is connected to the all split possibilities of first UE_i which are nodes (m,i,k) ; $k \in \{1, .., 3\}$. And all split possibilities of last $UE_{i''}$ are connected to node f, where f is a **f**inish point for the directed graph G whith all the ongoing

Algorithm 3: E2E-USA

Inputs: I^{MAX}, C^{D}_{MAX}, C^{C}_{MAX}, R_{MAX}, P, E_{MAX}, ε
λ_i, ν_i, $\forall\, i \in \{1, .., N\}$, $\forall\, m \in \{1, .., M\}$
$\alpha, \beta, \gamma, \theta, \mu, \nu_k, g_k, f_k, \forall\, k \in \{0, .., K\}$
Output: $\mathcal{S}^G$ with the best utility function from $\mathcal{LP}_3$
Begin
for $p = 1$ to P **do**
  for $i = 1$ to N **do**
    $m \leftarrow$ random (M)
    $n_{RB} \leftarrow$ random $([-B^{MAX}_{im}, +B^{MAX}_{im}])$
    $\mathcal{S}^p_{imb} \leftarrow 1$; n_{RB} times ; { *priory to RBs with less interference level*}
    $\mathcal{S}^p_{im,B+1} \leftarrow$ random (K)
    $\mathcal{V}^p_{im} \leftarrow$ random$([-B^{MAX}_{im}, +B^{MAX}_{im}])$
  end for
end for
while $iter < E_{MAX}$ **do**
  for $p = 1$ to P **do**
    if $\mathcal{U}_{\mathcal{F}}(\mathcal{S}^p) > \mathcal{U}_{\mathcal{F}}(\mathcal{S}^{L,p})$ **then**
      $\mathcal{S}^{L,p} \leftarrow \mathcal{S}^p$
    end if
    if $\mathcal{U}_{\mathcal{F}}(\mathcal{S}^p) > \mathcal{U}_{\mathcal{F}}(\mathcal{S}^G)$ **then**
      $\mathcal{S}^G \leftarrow \mathcal{U}_{\mathcal{F}}(\mathcal{S}^p)$
    end if
  end for
  for $p = 1$ to P **do**
    $r \leftarrow$ random $([0, 1])$
    for $i = 1$ to N **do**
      for $m = 1$ to M **do**
        if $r < \varepsilon$ **then**
          $\hat{\mathcal{V}}^p_{im} \leftarrow \sum_{b=1}^{B} \mathcal{S}^{L,p}_{imb} - \sum_{b=1}^{B} \mathcal{S}^p_{imb}$
        else
          $\hat{\mathcal{V}}^p_{im} \leftarrow \sum_{b=1}^{B} \mathcal{S}^{G}_{imb} - \sum_{b=1}^{B} \mathcal{S}^p_{imb}$
        end if
        if $\widetilde{r}_{im}(\hat{\mathcal{V}}^p_{im}) > \widetilde{r}_{im}(\mathcal{V}^p_{im})$ **then**
          $\mathcal{V}^p_{im} \leftarrow \hat{\mathcal{V}}^p_{im}$
        end if
      end for
    end for
  end for
  for $p = 1$ to P **do**
    for $i = 1$ to N **do**
      for $m = 1$ to M **do**
        $n_{RB} \leftarrow \sum_{b=1}^{B} \mathcal{S}^p_{imb} + \mathcal{V}^p_{im}$
        $\mathcal{S}^p_{imb} \leftarrow 1$; n_{RB} times ; { *priory to RBs with less interference level*}
      end for
    end for
    Run UFSS as described in 5.4.3.2
  end for

link weighted by zero. In Figure 5.2, both UE_i and $UE_{i'}$ are attached to gNB_m and $UE_{i''}$ is attached to $gNB_{m'}$. Each UE can be affected only two user splits, i.e., 1 and 2 and one cell split, i.e., 3.

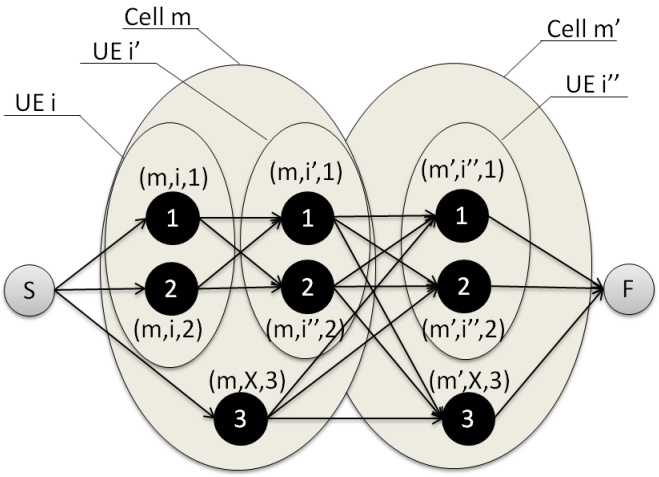


Figure 5.2: In graph G, a path from s to f corresponds to a functional split selection strategy, where the path cost is equal to the total deployment cost. Node (m, i, k) denotes for selecting a user split k for UE_i in gNB_m, while node (m, X, k) expresses the selection of cell split k all UEs in gNB_m

In doing so, a path P from s to f in graph G, corresponds to a selection strategy of functional splits for UEs that are already attached to different gNBs with a given radio load. It is worth noting that, the sum of links' costs traversed by the path P is equal to the deployment cost expressed in $\mathcal{LP}_1$. A path $P^\star$ of minimum cost corresponds to the optimal functional split decision that minimizes the overall deployment cost. The problem of calculating the optimal functional split selection is equivalent to finding a min-cost path in a DAG. The latter is resolved through the Dijkstra algorithm in $O(|E| + |V|log|V|)$time, where$|E|$ and $|V|$ are the number of edges and vertices. In our graph G, there exist $O(NK)$ nodes and $O(NK^2)$ links. So, finding the min-cost path takes $O(NK^2+NKlog(NK))$. At the end, the entire `E2E-USA` algorithm with the `UFSS` approach runs in $O(E_{MAX}PMB(NK)^2logNK))$ time.

5.5 Performance Evaluation

In this Section, we gauge the performance of our proposal `E2E-USA` based on extensive simulations. First, we describe the simulation environment setup and detail the various performance metrics. Then, we analyze the obtained results and discuss the effectiveness of our proposal compared with: i) commercial standard solvers such as IBM's ILOG CPLEX solver, ii) full Centralized deployment approach (i.e., C-RAN), iii) full Decentralized deployment approach (i.e., D-RAN), and iv) Cell-centric Split deployment approach denoted by `E2E-CSA`. Note that the interference mitigation is inherently implemented in the C-RAN approach, while we assume that this mechanism is adopted for the D-RAN case. We set the number of split options K to 9, where $Split_6$, $Split_7$ and $Split_8$ are cell-centric, while $Split_0$, $Split_1$, $Split_2$, $Split_3$, $Split_4$ and $Split_5$ are user-centric. To the best of our knowledge, there is no simulator

Table 5.1: Simulation parameters

Number of $gNBs$	$M = 7$
Number of UEs	$N = 100$
Inter-cell distance	$50\ m$
Number of RBs	$B = 100$
Spectrum Bandwidth	$W = 20\ MHz$
Antenna mode	$A = 1$, SISO
Average RB power	$\overline{P_{RB}} = 10\ mW$
Average cell power	$\overline{P_{gNB}} = 1\ Watt$
Transmit power gain	$Gtx = 8\ dBi$
Shadowing coefficient	$\Omega = 5\ dB$
Thermal Noise	$-174\ dBm/Hz$
$SINR^{MAX}$	10 dB
Path loss model (PL)	$148.1 + 37.6\ log(D)$, D in Km
Fading coefficient	$\rho = U(0, 1)$
Channel gain	$h = 10^{-PL/20} \cdot \sqrt{Gtx \cdot \Omega} \cdot \rho$
c_{14}	10^{-3}
I^{MAX}	$\frac{h * \overline{P_{RB}}}{SINR^{MAX}} - \sigma^2$
R_{MAX}	$3686,4$ Mbps [16]
C^D_{MAX}, C^C_{MAX}	960 GOPS [83]
$\theta, \mu, \alpha, \beta, \gamma$	0.5 , 0.5 , 0.33 , 0.33 , 0.33
uRLLC UEs	40% of total UE number

for RAN slice orchestration with user functional split selection deployment so far. In the following, we show the results of our JAVA-based simulator.

5.5.1 Simulation setup

We simulate our Cloud-RAN infrastructure with respect to our model described in Section 5.3. We consider N UEs uniformly distributed in an OFDMA based cellular network. UEs may generate a traffic in $[0, 1]$ Mbps with a latency in $\{1, 2, 3, 4\}$ ms for eMBB UEs and a latency in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ ms for uRLLC UEs [104]. UEs positions are randomly generated for each execution and remain fixed during their whole stay in the network. It follows that we calculate the CQI, MCS and $TBSI$ between each UE and gNB in order to approximate the linear function of generated TBS $\widetilde{TBS}$ between each UE and gNB. Table 5.1 reports the simulation parameters that have been used for our simulations [34]. Our results correspond to the average of 30 simulations with a confidence level set to 95%.

5.5.2 Performance metrics

We rely on following metrics to gauge the performance of our proposal `E2E-USA` compared with baseline strategies.

- $\mathbb{U}_{\mathbb{F}}$ is the Utility Function in LP_3 expressing the trade off between the served throughput and the deployment cost.

- $\mathbb{C}_{\mathbb{T}}$ is the average Convergence Time for user-centric Allocation in ms.
- $\mathbb{T}_{\mathbb{S}}$ is the Throughput Satisfaction rate expressing the ratio between the overall served and requested throughputs.
- $\mathbb{C}_{\mathbb{D}}$ is the Cost of Deployment expressing the computational and link resource usage as defined in LP_1, which is expressed as the weighted sum of the resource usage in i) the DU sites weighted by α, ii) the CU sites weighted by β, and iii) the transport link weighted by γ.
- $\mathbb{L}_{\mathbb{T}}$ is the Latency penalty of Total users expressed as $\sum_i \frac{\nu_k - \nu_i}{\nu_i}, \forall k \in K, \forall i \in N$ where, ν_k is the latency of split k in the transport network while ν_i is the required latency from UE_i.
- $\mathbb{S}$ corresponds to the percentage of Splits generated by our proposal.

5.5.3 Simulation results

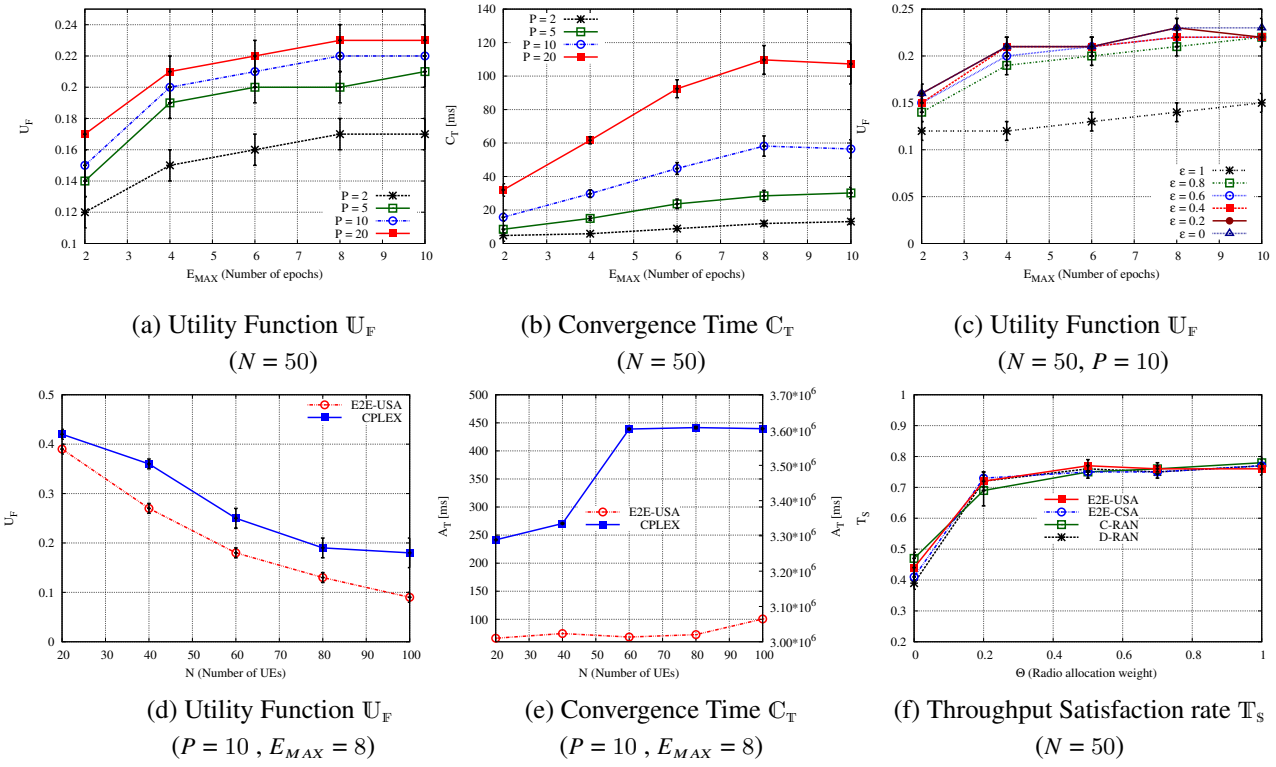
5.5.3.1 Convergence Analysis

In what follows, we aim to evaluate the impact of the number of particles P and the number of epochs E_{MAX} on the solution quality (i.e., the utility function $\mathbb{U}_{\mathbb{F}}$ and the convergence time $\mathbb{C}_{\mathbb{T}}$). Figure 5.3.(a) assesses the performance of `E2E-USA` with different swarm population size P, while varying the number of epochs E_{MAX}. Indeed, for a fixed number of UEs (i.e., $N = 50$), we can observe that the utility function $\mathbb{U}_{\mathbb{F}}$ of each swarm population is increasing when E_{MAX} increases. Besides, it is straightforward to see that the size of P impacts the quality of the solution. In particular, the curves corresponding to $P = 10$ and $P = 20$ have close values that outperform both $P = 5$ and $P = 2$. Then, it is interesting to see that $\mathbb{U}_{\mathbb{F}}$ keeps stable starting from $E_{MAX} = 8$.

In Figure 5.3.(b), we study the impact of the swarm population size P on the convergence time $\mathbb{C}_{\mathbb{T}}$. It is clear to see that when the number of particles P increases, the convergence time $\mathbb{C}_{\mathbb{T}}$ increases as well. Such a behavior is predictable, as the solution quality is enhanced as soon as P is increased, which in turn, requires more computation time to solve the problem. In particular, the curve corresponding to $P = 20$ costs much more computational time than the curves corresponding to $P = 10$, $P = 5$ and $P = 2$. In what follows, we fix P to 10 and E_{MAX} to 8.

Figure 5.3.(c) assesses the convergence behavior of `E2E-USA` with different values of ε while varying $ITER_{MAX}$. Indeed, for a fixed number of UEs (i.e., $N = 50$) and a fixed number of particles (i.e., $P = 10$), we can observe that $\mathbb{U}_{\mathbb{F}}$ increases when $ITER_{MAX}$ increases. We recall that ε is the probability of a particle to follow the local best position according to (E22). As depicted, when $\varepsilon = 1$, i.e., particles only follow their best local positions, the algorithm struggles to find an optimal solution. Meanwhile, the solution quality is enhanced when ε is less than 0.8. This proves that particles need to collaborate with each other to fasten the convergence process. It is interesting to see that, when $ITER_{MAX}$ is lower than 8 epochs, the curves corresponding to $\varepsilon = 0.2$ outperforms the one corresponding to $\varepsilon = 0$. This can be explained that `E2E-USA` rather favors a tradeoff between exploitation (ε) and exploration (1-ε) to achieve better results. In what follows, we fix the balance point of exploitation-exploration, ε to 0.2.

In what follows, we vary N in $[20; 100]$ with a rate of uRLLC UEs equal to 40% in each iteration. We set P and E_{MAX} to 10 and 8 respectively. We aim to evaluate the performance of `E2E-USA` in case of high density of UEs. In Figure 5.3.(d), we compare `E2E-USA` to the `CPLEX` solver, which converges to the optimal solution. It is straightforward to see that our solution generates near optimal solutions



(a) Utility Function $\mathbb{U}_{\mathrm{F}}$ ($N = 50$)

(b) Convergence Time $\mathbb{C}_{\mathrm{T}}$ ($N = 50$)

(c) Utility Function $\mathbb{U}_{\mathrm{F}}$ ($N = 50$, $P = 10$)

(d) Utility Function $\mathbb{U}_{\mathrm{F}}$ ($P = 10$, $E_{MAX} = 8$)

(e) Convergence Time $\mathbb{C}_{\mathrm{T}}$ ($P = 10$, $E_{MAX} = 8$)

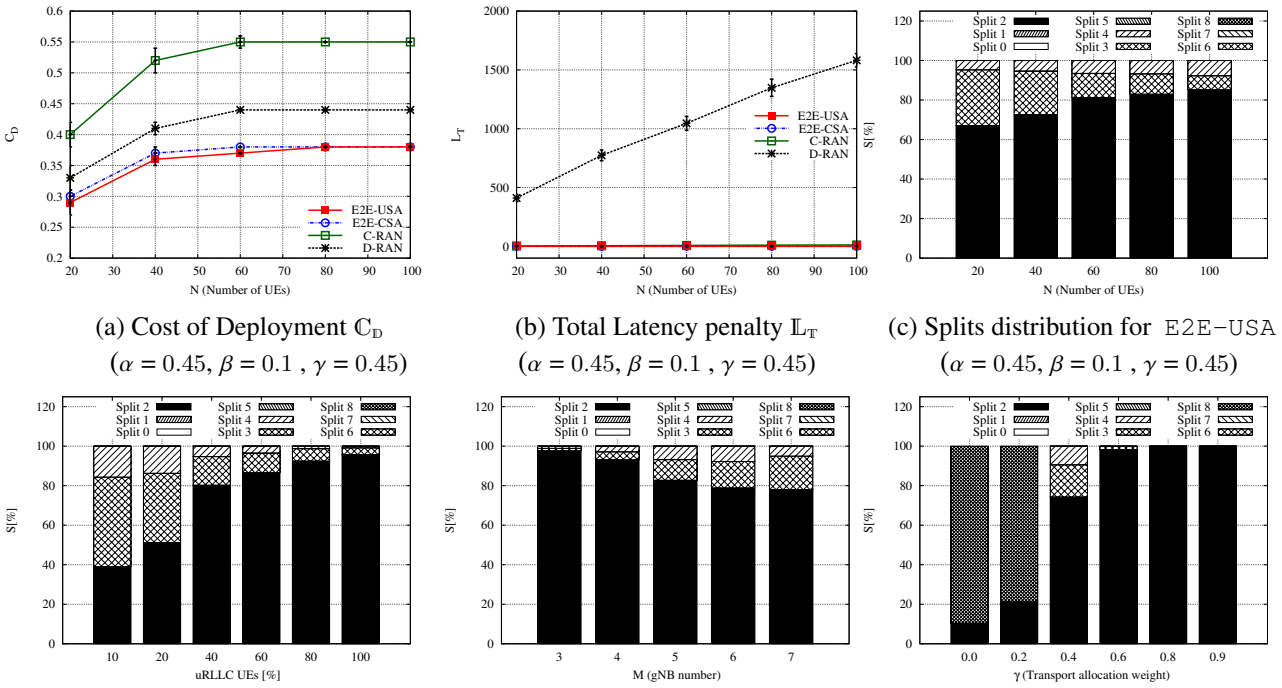
(f) Throughput Satisfaction rate $\mathbb{T}_{\mathrm{S}}$ ($N = 50$)

Figure 5.3: Convergence evaluation

when the number of UEs N is equal to 20. Whereas, when N is higher than 20, our proposed approach achieves a lower utility function with a gap of 28%.

With regards to scalability, Figure 5.3.(e) illustrates the average resolution time $\mathbb{A}_{\mathbb{T}}$ of the different strategies versus the number of UEs N. Note that the Transmission Time Interval (TTI) in C-RAN is equal to 1 millisecond according to [54]. It is straightforward to see that the non-scalable optimal solution takes a significantly longer time than `SPLIT-HPSO` to solve one instance of the optimization problem. Indeed, the optimal solution struggles to scale, as it takes several minutes to solve instances of N. In contrast, `E2E-USA` can easily solve any size of instance (i.e., N in $[20, 100]$) in the range of $[66; 100]$ milliseconds. Eventually, `E2E-USA` is able to take an up-to-date decision and execute it after 100 TTI period. Unfortunately, Optimal-Split is not able to do so since its decision, once taken, will be already obsolete and hence not applicable.

Figure 5.3.(f) illustrates $\mathbb{T}_S$, with respect to the radio allocation weight (θ). Wherein, for a fixed number of UEs (i.e., $N = 50$), we assume that θ is increasing in the range of $[0, 1]$ while μ is decreasing in the range of $[0, 1]$. As depicted, the throughput satisfaction is almost enhanced while θ is increasing. Furthermore, $\mathbb{T}_S$ reaches his maximum value at 0.77. This is explained by the fact that, when the throughput demand is high, radio resources become scarce which makes the selection of the appropriate set of resource blocks extremely challenging. Note that `E2E-USA` achieves nearly the same performance as baseline scenarios D-RAN, C-RAN and `E2E-CSA`.



(a) Cost of Deployment $\mathbb{C}_{\mathrm{D}}$ ($\alpha = 0.45$, $\beta = 0.1$, $\gamma = 0.45$)

(b) Total Latency penalty $\mathbb{L}_{\mathrm{T}}$ ($\alpha = 0.45$, $\beta = 0.1$, $\gamma = 0.45$)

(c) Splits distribution for `E2E-USA` ($\alpha = 0.45$, $\beta = 0.1$, $\gamma = 0.45$)

(d) Splits distribution for `E2E-USA` ($N = 50$)

(e) Splits distribution for `E2E-USA` ($N = 50$, uRLLC $UEs = 40\%$)

(f) Splits distribution for `E2E-USA` ($N = 50$, $\beta = 0.1$)

Figure 5.4: Performance evaluation

5.5.3.2 Performance Analysis

Hereafter, we fix θ to 0.5 while μ is equal to 0.5. We also fix the RCC computational consumption weight β to the value 0.1 as Cloud data centers are natively efficient in power consumption. We assume that α and γ are both equal to 0.45 to emphasize the tradeoff issue between minimizing DU computational cost weighted by α and optimizing the link resource usage weighted by γ.

In Figure 5.4.(a), we illustrate $\mathbb{C}_D$ with respect to the number of UEs. It is straightforward to see that, our approach `E2E-USA` further optimizes the computational and link resource usage cost comparing to baseline approaches. Indeed, our proposal is user-centric, hence, it adopts a fine-grained approach to optimize the resource allocation. It is worth pointing out that C-RAN and D-RAN achieves higher cost of deployment. As a matter of fact, the C-RAN approach allocate constantly the full transport link bandwidth, while, the D-RAN approach utilizes all the computational resources in DU sites.

Figure 5.4.(b) illustrates the penalty $\mathbb{L}_T$ as a function of the UEs' number. As we can see, both approaches `E2E-USA` and `E2E-CSA` approaches keep a zero penalty which means that all UEs are constantly served with splits satisfying their latency requirement. However, D-RAN approach causes high penalty because all UEs are served with $Split_0$ that implies a latency in the order of 10 ms which obviously violate the latency requirements of both eMBB and uRLLC UEs. C-RAN also implies a latency penalty in the range of $[1, 5]$ for some uRLLC UEs requiring a latency less than 0.2 ms.

Figure 5.4.(c) depicts the splits distribution for `E2E-USA` while increasing the UEs' number. It is straightforward to see that, our approach favors $Split_2$, $Split_3$ and $Split_4$. $Split_0$ and $Split_1$ are

excluded since they induce a latency of 10 ms which does not satisfy neither eMBB nor uRLLC flows. Furthermore, $Split_5$, $Split_6$, $Split_7$ and $Split_8$ are not selected because they generate a high traffic in the transport network which impacts the deployment cost. Instead, our approach achieves a trade off between DU computational usage and link resource usage by adopting a partial centralization scheme of PF functions. Specifically, $Split_2$ increases proportionally with the UEs density while $Split_4$ and $Split_3$ decrease. This can be explained by the fact that, when N increases, the number of uRLLC UEs requiring a strict latency that can only be served by $Split_2$, increases accordingly. Assuming our model, this fact constraints $gNBs$ to deploy $Split_2$ for the remaining attached UEs. With reference to Section 5.3, $Split_2$ leads to a high computational deployment cost comparing to the other feasible splits. To counteract this side-effect, `E2E-USA` selects $Split_3$ and $Split_4$ in other $gNBs$ to centralize more functions in the Cloud.

Figure 5.4.(d) assesses the split selection strategy of `E2E-USA` with different percentage of uRLLC UEs. Indeed, for a fixed number of UEs (i.e., $N = 50$) and a fixed number of $gNBs$ (i.e., $M = 7$), we can observe that the adoption of $Split_2$ increases at the expense of $Split_3$ and $Split_4$. This emphasis the fact that UEs with stringent latency requirement less that 0.2 ms, restraint $gNBs$ to deploy only $Split_2$ excluding necessary other split options even for other attached UEs.

Figure 5.4.(e) illustrates the impact of the gNB number (i.e., M) on the split selection strategy $\mathbb{S}$. For a fixed number of UEs (i.e., $N = 50$), a uRLLC UEs percentage fixed to 40% and M increasing in the range of $[3, 7]$, the deployment of $Split_2$ decreases while the adoption for $Split_3$ and $Split_4$ increases. The reason behind this is that, `E2E-USA` is not anymore constrained to deploy $Split_2$ in some $gNBs$. Instead, `E2E-USA` finds a greater flexibility to deploy other splits to achieve the tradeoff between DU computational usage and link resource usage.

In Figure 5.4.(f), we study the tradeoff between the DU computational cost minimization, which is weighted by α and the link resource usage optimization, which is weighted by γ. Therefore, we assume that γ is increasing in the range of $[0, 1]$ while α is decreasing in the range of $[0, 1]$. As depicted in Figure 5.4.(f), our solution adopts $split_2$ and $split_8$ when γ is lower than 0.4 (i.e., α is higher than 0.6). Then, when γ is equal to 0.4, the algorithm adopts mainly $split_2$, $split_3$ and $split_4$ until γ reaches 0.6. Afterwards, $split_2$ is constantly deployed. The reason behind this behavior is that `E2E-USA` adopts splits with minimum DU computational cost when α is high (namely $split_8$) while $split_2$ is served for some uRLLC UEs. When γ is high, `E2E-USA` favors splits with minimum traffic flow in the transport link (namely $split_2$). It is interesting to see that when γ is equal to 0.4 and α is fixed to 0.6, the tradeoff is achieved by deploying simultaneously $split_2$, $split_3$ and $split_4$.

5.6 Conclusion

5G-RAN stakeholders aim to build a RANaaS concept with an innovative RAN infrastructure to respond to new 5G applications requirements. In this context, the slicing concept is introduced in order to handle the heterogeneity of new use-cases. Despite the great advances achieved by RAN functional split standardization, there is still a coarse grained approach in the deployment process. In this Chapter, we propose a User-centric RAN Slice Allocation approach `E2E-USA`. Wherein, each user is assigned a proportion of radio and a split option. At the end, multiple user RAN slices are created and managed on top of the physical infrastructure tailored to users' requirements. Our contribution is twofold. First, we elaborated user-centric RAN slice allocation problem as an Integer Linear Problem (ILP) with multi-objective function. Second, we propose a heuristic based on Particle Swarm Optimization that jointly

optimizes radio, link and computational resource allocation. Based on Particle Swarm Optimization, E2E−USA is scalable and achieves optimized user-centric RAN slice allocation solution in a satisfactory time. Based on extensive simulations, we have shown that E2E−USA achieves good performances in terms of total throughput satisfaction and deployment cost. In the next Chapter, we propose to operate the RAN slice allocation in real time, by proposing a Deep Learning based user-centric RAN slice allocation scheme.

CHAPTER 6

DEEP LEARNING BASED USER-CENTRIC RAN SLICE ALLOCATION IN CLOUD RAN

Contents

6.1 Introduction

The proposed heuristic in Chapter 5 generates near optimal solutions in an acceptable resolution time. Even short, such a convergence time needs to be further minimized in order to deal with 5G RAN time

constraint with less than 1 ms for a Transmission Time Interval [54], depending on each use-case. In this perspective, a real-time resolution should be performed ensuring, hence, an up-to-date decision.

To deal with such a high complexity, we put forward a real-time **D**eep **L**earning approach for optimized **U**ser-centric **S**lice **A**llocation (`DL-USA`) in 5G RAN. Our proposal proceeds in four stages: i) input generation for training, ii) dataset preprocessing, iii) training, and iv) testing. During the first phase, we solve multiple instances of the ILP problem, described previously in [Chapter 5, p. 57], in an offline manner using a powerful solver such as IBM CPLEX [105]. The aim is to generate various input parameters along with their corresponding optimized output decisions. During the second phase, data is filtered and datasets are constructed. Thirdly, we train our `DL-USA` solution based on a bidirectional Long-Short-Term-Memory (biLSTM) model [106]. The main objective is to construct a predictive allocation model of RAN user slices. Finally, `DL-USA` is tested using a new dataset ensuring the effectiveness of our proposed solution. Once trained, the model can be invoked in an online manner to generate real-time RAN slice decision based on collected RAN input parameters of end-users and the physical infrastructure.

The rest of this Chapter is organized as follows. In Section 6.2, we outline the proposed `DL-USA` solution, while a description of our simulator and evaluation results are detailed in Section 6.3.

6.2 Proposal: `DL-USA`: Deep Learning solution for RAN slice allocation

In this Section, we give insights into our proposed solution to ensure real-time user-centric RAN slice allocation expressed in the problem $\mathcal{LP}_3$. We leverage the machine learning technique using a Deep Neural Network (DNN) model [107]. It is straightforward to see that the problem $\mathcal{LP}_3$ [Chapter 5, p. 57] is classified as a nondeterministic polynomial hard problem [37], where the solution corresponds to the triplet: $UE - gNB$ association, RB allocation and split selection for each UE in the network. An exhaustive search will lead to check all possible combinations. For example, for N UEs, M gNBs, B RBs and K split options, optimal solutions will approximately calculate $N^{M\times B\times K}$ combinations, which is practically infeasible in case of high-scale of UE number. Additionally, proposed heuristics may help to reduce the resolution time. However, the 5G RAN context requires an up-to-date decision within a Transmission Time Interval period less than 1 ms [54]. Therefore, sophisticated algorithms may lead to decisions, that once taken, will be already obsolete and hence not applicable. Reactive models are highly recommended in this case, where the allocation scheme is generated in real-time upon input data without performing an exhaustive calculation task.

6.2.1 Deep Learning approach

Recently, Artificial Intelligence (AI) and Machine Learning (ML) techniques are investigated to enable efficient RAN resource allocation in a dynamic and scalable environment. The Self-Organizing Networks (SONs) [38] has been supported by 3GPP standardization for empowering the RAN with big data applications. Indeed, the emerging ML solutions are proven to be efficient in speeding up the optimization process as well as in finding heuristic solutions in an iterative manner [108]. With focus on the conventional supervised ML technique, the idea is to analyze a huge amount of <radio configuration, allocation decision> pairs for example. Then, a heuristic (trained model) is inferred to map new radio configurations on near optimized allocation decision. Currently, the ML technique is intensively adopted in RAN. However, related works are either limited to radio resource allocation [109–112] or addressing

the placement problem of virtual network functions, without considering the functional split selection problem [113–115]. Interestingly, the work in [88] adopts a Deep Learning approach with a Long Short-Term Memory (LSTM) model for addressing the functional split selection model. In this chapter, we go a step further and propose a Deep Learning based scheme for optimized User-centric Slice Allocation (DL−USA) in 5G RAN.

6.2.2 DL−USA overview

Our new scheme, named Deep Learning based User Slice Allocation (DL−USA), is put forward to optimize the RAN user slice allocation decision. Specifically, our approach performs in four stages: i) problem data input generation, ii) data pre-processing, iii) training phase and iv) testing phase. During the first stage, we solve multiple instances of the $\mathcal{LP}_3$ problem in an offline manner while using the B&C algorithm of IBM CPLEX solver. The aim is to generate various input parameters of problem $\mathcal{LP}_3$ along with their corresponding optimized output decisions. During the second stage, we construct datasets (i.e., sequences), each contains relevant input and output data of one execution instance. During the third stage, we train our DL−USA solution based on these datasets to construct an efficient allocation model of user slices. Finally, DL−USA is tested on new datasets which enables to verify the accuracy of our proposed solution. Once trained, the model can be invoked in online manner like a simple call function to express the slice decision based on received input parameters.

6.2.3 Data Generation Phase for training

We solve $\mathcal{LP}_3$ optimization problem for RAN User Slice Allocation using the Branch-and-Cut algorithm B&C [105] of IBM CPLEX Solver.

6.2.4 Dataset Pre-Processing

During the pre-processing phase, data is filtered, normalized and organized into sequences in order to be processed by the deep neural network. It is straightforward to note that the inputs of our ILP in $\mathcal{LP}_3$ contain different features with different value ranges. For seek of simplicity, we assume that the infrastructure parameters are constant: I_{MAX}, B, M, C^D_{MAX}, C^C_{MAX}, R_{MAX}, α, β, γ, θ, μ, ν_k, g_k, f_k, $\forall\ k \in \{0, .., K\}$. Their set-up will be given in Section 6.3. We vary the UE model input parameters for: i) required data rate λ_i, $\forall\ i \in \{1, .., N\}$, ii) required latency ν_i, $\forall\ i \in \{1, .., N\}$ and iii) CQI_{im}, $\forall\ m \in \{1, .., M\}$, $\forall\ i \in \{1, .., N\}$. Then, we conducted a detailed statistical analysis, to monitor feature-wise values for minimum, maximum and average. We have dropped out the features with duplicate information or constants values. Finally, this analysis resulted in $[2 + M]$ unique features in the input dataset.

Afterwards, we construct the input dataset I which is a vector of dimension N_S, corresponding to the number of execution time of algorithm B&C. Indeed, each element of I contains the input data of one execution unit of algorithm B&C. Formally, one input sequence corresponds to a matrix of $N \times [2 + M]$, where N is the number of UEs and $[2 + M]$ is the number of features. The target output dataset O_T is a vector of N_S sequences, each of which contains the corresponding output data of one execution unit of algorithm B&C. Formally, one output sequence is a vector of N elements, each of which contains the triplet: i) attached gNB identifier, ii) amount of allocated RBs (radio load) and iii) user split identifier.

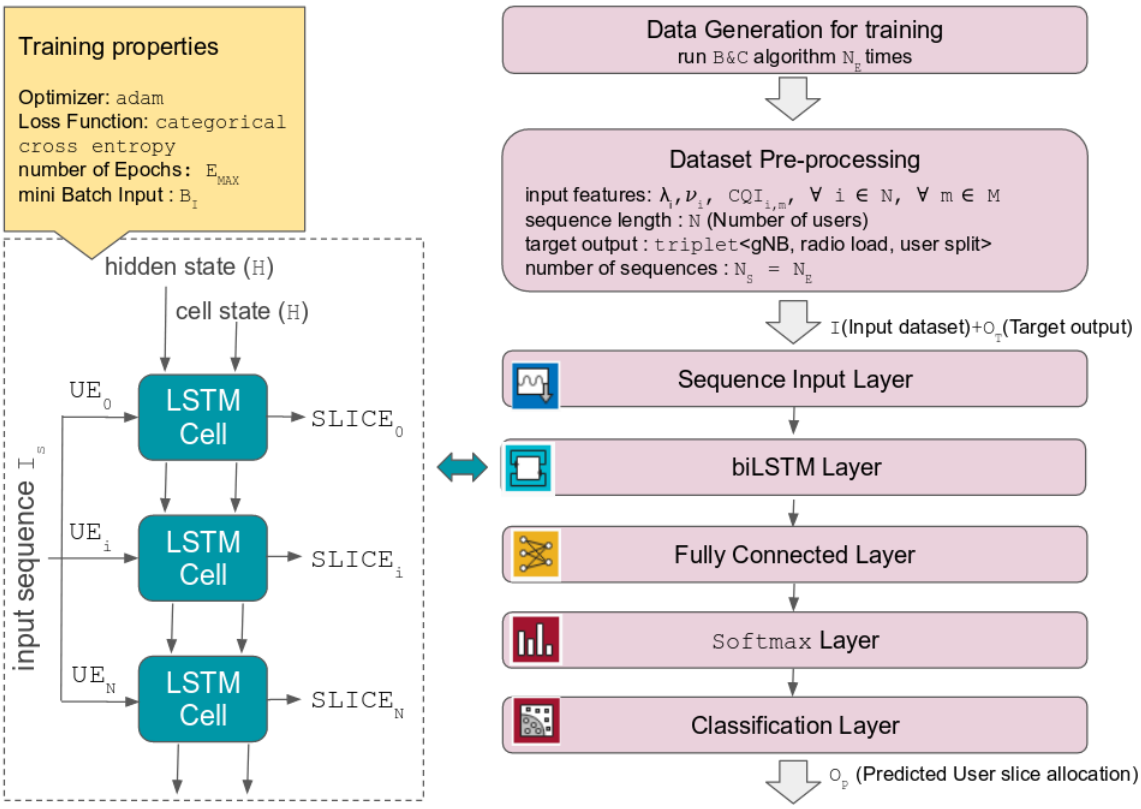


Figure 6.1: DL-USA

Finally, we characterize the number of categories N_C, the number of triplets, i.e., RAN slices, that could be generated by B&C.

6.2.5 Deep Neural Network model

We make use of a supervised machine learning technique with Deep Neural Networks [107] to perform an efficient RAN Slice Allocation. More specifically, our solution DL-USA makes use of the Long Short-Term Memory (LSTM) model [106] which converts the conventional neurons into memory cells with gates. Such a new structure facilitate the information storage and sharing between memory cells within the same layer. In doing so, LSTM outperforms the conventional neural networks when dealing with complex classification problems and prediction. Besides, unlike the classical neural networks, LSTM is capable of handling a variable-length sequence input which make it natively adapted to the dynamic 5G context.

Figure 6.1 illustrates the workflow of the proposed DL-USA scheme. **First**, the sequence input layer introduces each sequence data to the DNN network. **Second**, the bidirectional LSTM (biLSTM) layer processes the UE inputs sequentially and generate the according initial RAN slices. More specifically, the biLSTM layer consists of two sub-layers, each with N chained memory cells, both in different direction. For sake of simplicity, Figure 6.1 presents only one sub-layer. Wherein, each LSTM cell i performs the following tasks: i) process the UE_i problem input and calculate the initial RAN slice i, ii) transmit

the latter to the next LSTM cell as a hidden state, and iii) update the cell state. The aforementioned operations are insured with weighted gates of "sigmoid" activation functions which control the data flow, while the state activation functions "tanh" are implemented to facilitate the information storage. Note that both cell and hidden states are configured with the same number of hidden units H. **Third**, the output of the second LSTM sublayer is fed into a fully connected layer which has a number of neurons equals to the number of categories N_C. **Forth**, the Softmax layer calculates the probability distribution over the categories and the highest probability is selected for the output generation. At the end, the classification layer computes the cross entropy loss between the generated and B&C target outputs.

6.2.6 Learning phase

In order to generate the allocation model, the network is trained based on the previously generated input and target output dataset. The objective is to create a reliable model able to regenerate an approximation O_P for the target output O_T with an acceptable error rate. To do so, the entire dataset is divided into mini batches, each containing B_I sequences. Then, during the training phase, the gates' weights becomes variables subject to an optimization problem that aims at minimizing a penalty function. We make use of the cross-entropy operation as a loss function. Accordingly, it computes the loss between the generated category O_{Pij} and the target value O_{ij} across all users among the sequences.

The categorical cross-entropy loss function is expressed in $\mathcal{LP}_4$ as follows:

$$\mathcal{LP}_4 : \text{Minimize } -\frac{1}{B_I \times N} \sum_{i=1}^{N} \sum_{j=1}^{B_I} (O_{Tij}\ log(O_{Pij}) + (1 - O_{Tij})\ log(1 - O_{Pij})$$

We make use of Adam optimizer [116] as a stochastic gradient-decent algorithm iteratively to tune the aforementioned parameters, starting from the final classification layer back to the first initial sequence input layer. The main idea is to calculate the loss function in order to decrease the weight values with higher error rates in every layer and vice versa. By back-propagating the loss into the network, and finding out what loss every unit is responsible for, we can decrease the total loss of the model. Note that the entire dataset is processed within a single epoch. Therefore, we define E_{MAX} the necessary total number of epochs to make the network converge to an efficient allocation model.

6.2.7 Testing Phase

In general, DNN based approaches adopt an inductive reasoning which is fundamentally different from logic-based algorithm [117]. Indeed, the former constructs a model estimating predictions with a certain probability, while the latter make exact deduction. Therefore, the performance of a DNN based approach should be statistically measured on a sparsely distributed data. The objective is to accurately measure the proposed DL−USA performance in an operation context.

During the testing phase, we generate new datasets making use of the B&C algorithm, which constitute 25% of the size of training data. Then, performance is measured statistically to minimize the importance of individual error-inducing inputs.

Table 6.1: Network parameters

Number of $gNBs$	$M = 7$
Number of UEs	$N = 100$
Inter-cell distance	$50\ m$
Number of RBs	$B = 100$
Spectrum Bandwidth	$W = 20\ MHz$
Antenna mode	$A = 1$, SISO
Average RB power	$\overline{P_{RB}} = 10\ mW$
Average cell power	$\overline{P_{tx}} = 1\ Watt$
Transmit power gain	$Gtx = 8\ dBi$
Shadowing coefficient	$\Omega = 5\ dB$
Thermal Noise	$-174\ dBm/Hz$
$SINR_{MAX}$	10 dB
Path loss model (PL)	$148.1 + 37.6\ log(D)$, D in Km
Fading coefficient	$\rho = U(0, 1)$
Channel gain	$h = 10^{-PL/20} \cdot \sqrt{Gtx \cdot \Omega} \cdot \rho$
I_{MAX}	$\frac{h * \overline{P_{RB}}}{SINR_{MAX}} - \sigma^2$
R_{MAX}	3686, 4 Mbps [16]
C^D_{MAX}, C^C_{MAX}	960 $GOPS$ per gNB [83]
$\theta, \mu, \alpha, \beta, \gamma$	0.5 , 0.5 , 0.33 , 0.33 , 0.33 [Chapter 5, p. 67]
uRLLC UEs	40% of total UEs

6.3 Performance Evaluation

In this Section, we gauge the performance of our proposal DL−USA based on extensive simulations. In first order, we describe the simulation environment setup and the performance metrics. Then, we analyze the generated results and discuss the effectiveness of our proposal compared to the B&C algorithm. To the best of our knowledge, there is no RAN simulator enabling the user functional split deployment so far. In the following, we show the results of our implemented JAVA-based simulator.

6.3.1 Simulation setup

We simulate the Cloud RAN infrastructure with respect to our model described in [Chapter 5, p. 57]. We consider N UEs uniformly distributed in an OFDMA based cellular network. UEs generate a traffic in $[0, 1]$ Mbps with a latency in $\{1, 2, 3, 4\}$ ms for eMBB UEs and $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ ms for uRLLC UEs [104]. UEs positions are randomly generated for each execution and remain fixed during their whole stay in the network. It follows that we calculate the CQI, MCS and $TBSI$ between each UE and gNB in order to approximate the linear function of generated TBS $\widetilde{TBS}$ between each UE and gNB. Table 6.1 reports the simulation parameters used in our simulations. Besides, we train our model DL−USA based on 400 sequences (number of execution loops in B&C algorithm). Our performance analysis corresponds to the average of 100 simulations with a confidence level set to 95%.

6.3.2 Performance metrics

We use the following metrics to gauge the performance of our proposal DL−USA compared with B&C-based scheme.

- $\mathbb{A}_P$ is the Accuracy Percentage measured by LP_4 between the output of the DL−USA trained model and the B&C algorithm.
- $\mathbb{U}_\mathbb{F}$ is the Utility Function in LP_3 expressing the trade off between the served throughput and the deployment cost.
- $\mathbb{T}_S$ expresses the Throughput Satisfaction rate corresponding to the ratio between the overall served and requested throughputs.
- $\mathbb{C}_D$ expresses the Cost of Deployment expressing the computational and link resource usage as defined in LP_1, which is expressed as the weighted sum of the resource usage in i) DU site weighted by α, ii) CU site weighted by β, and iii) the fronthaul link weighted by γ.
- $\mathbb{L}_T$ denotes the Latency penalty of Total users expressed as $\sum_i \frac{l_k - l_i}{l_i}, \forall k \in K, \forall i \in N$ where l_k is the latency of the user split k while l_i is the required latency of UE_i. See [Chapter 5, p. 57].
- $\mathbb{T}_R$ is the Average Time of Resolution in ms.
- $\mathbb{S}$ corresponds to the percentage of Splits generated by our proposal DL−USA.

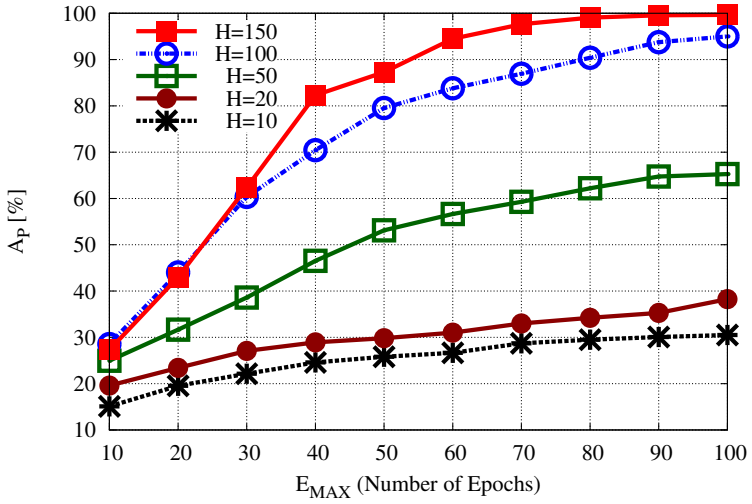
6.3.3 Simulation results

6.3.3.1 Convergence Analysis

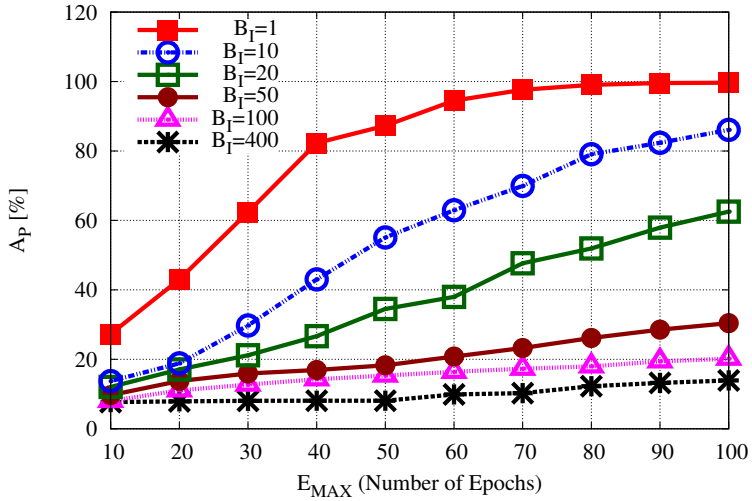
In what follows, we evaluate the impact of the number of hidden units H and the number of epochs E_{MAX} on the solution quality (i.e., accuracy percentage $\mathbb{A}_\mathbb{P}$). Figure 6.2.(a) assesses the convergence properties of DNN−USA for different number of hidden units H, while varying the number of epochs E_{MAX}. We recall that H is the size of both cell and hidden states of the LSTM cell. As depicted, we can observe that the accuracy percentage $\mathbb{A}_\mathbb{P}$ of each DNN network is increasing when E_{MAX} increases. Besides, it is straightforward to see that the size of H impacts the quality of the solution. In particular, the curve corresponding to $H = 150$ outperforms DNN network with $H < 150$. Then, it is worth noting that $\mathbb{A}_\mathbb{P}$ becomes stationary, for DNN network with $H = 150$, starting from $E_{MAX} = 80$.

In Figure 6.2.(b), we study the impact of the number of mini-bach B_I and the number of epochs E_{MAX} on the solution quality $\mathbb{A}_\mathbb{P}$. We recall that B_I is the number of sequences used to compute the loss function and trigger one training step. B_I is also considered as the frequency update of the DNN parameters. It is clear to see that $\mathbb{A}_\mathbb{P}$ increases efficiently when the number of mini-bach B_I is equal to 1. When B_I is higher than 50, the DNN network struggles to converge. This can be explained by the fact that our DL−USA requires higher frequency update of the DNN parameters with a fine-grained tuning (i.e., after processing each input data).

In what follows, we fix E_{MAX}, H and B_I to 80, and 150 and 1, respectively. It is worth noting that the achieved results are conformed to the bayesian optimization method [97], which make use of the Gaussian processes to tune the DL−USA hyper parameters.



(a) H tuning ($B_I = 1$)

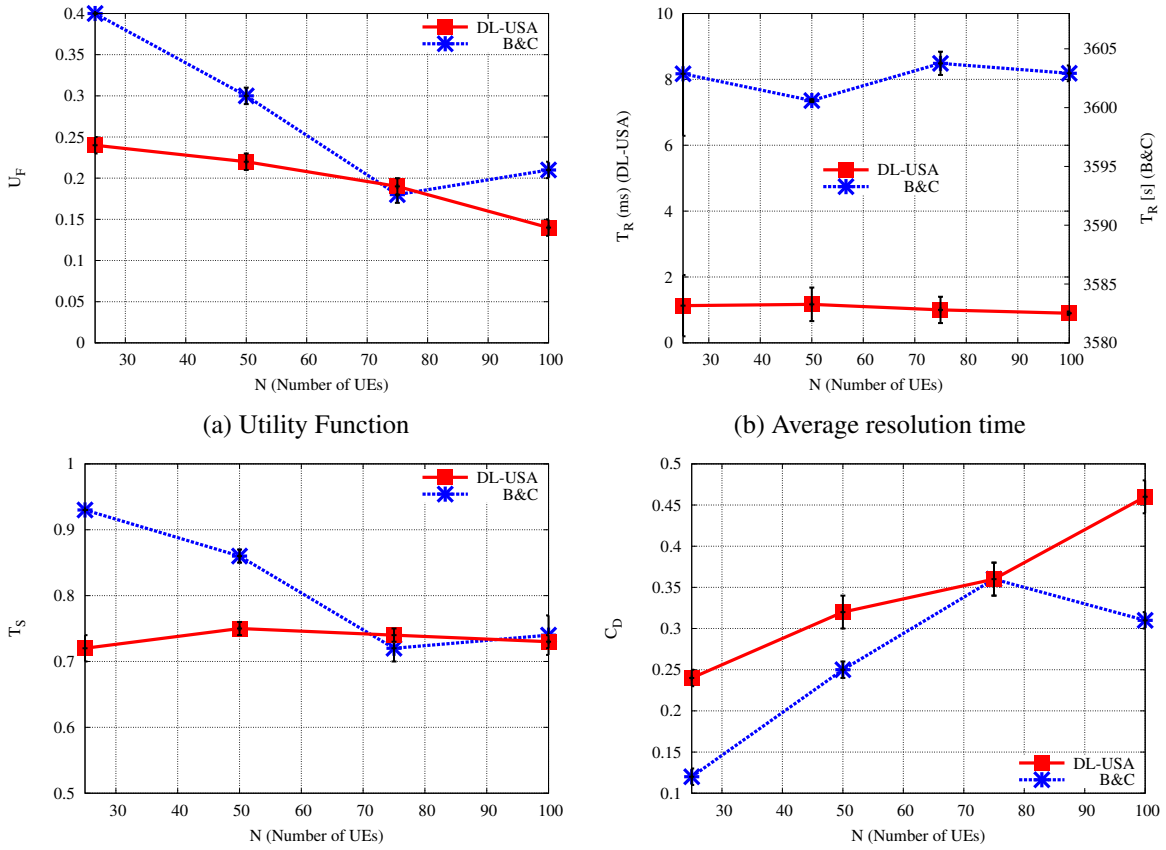


(b) B_I tuning ($H = 150$)

Figure 6.2: Convergence Analysis

6.3.3.2 Performance Analysis

Hereafter, we propose to analyze the scalability performance of our approach DL−USA. We vary N in $[25, 100]$ with a rate of uRLLC UEs equals to 40% in each iteration. We set E_{MAX}, H and B_I to 80, 150 and 1, respectively. We aim to evaluate the performance of DL−USA in case of high density of UEs. In Figure 6.3.(a), we compare DL−USA to the B&C based scheme. It is straightforward to see that our proposal generates near optimal solutions when the number of UEs N is lower than 75. Within this range, B&C provides a RAN slice allocation with a utility function $\mathbb{U}_{\mathbb{F}}$, greater than our proposed approach. Whereas, when N is equal to 75, our proposed approach achieves the same utility function of



(a) Utility Function

(b) Average resolution time

(c) Throughput satisfaction rate

(d) Cost of deployment

Figure 6.3: Performance Evaluation

B&C based solution.

Figure 6.3.(b) illustrates the impact of the number of UEs (i.e., N) on the resolution time $\mathbb{T}_{\mathbb{R}}$. As we can see, $\mathbb{T}_{\mathbb{R}}$ scales when N increases for B&C solution. The reason behind this is that B&C tries to solve iteratively the LP_3 problem with its related constraints for all N UEs. Therefore, the non-scalable solution takes a significantly longer time than DL−USA to solve one instance of the optimization problem. The B&C based solution struggles to operate in real time, as it takes values in $[3600, 3604]$ seconds to solve instances of N in $[25, 100]$. In contrast, DL−USA can easily solve any size of instance (i.e., N in $[25, 100]$) in the range of $[1, 2]$ milliseconds. It achieves, hence, to speed up the computation time of 36×10^5 magnitude, DL−USA is able to take an up-to-date decision. Unfortunately, the B&C based solution is not able to do so since its decision, once taken, will be already obsolete and hence not applicable.

Figure 6.3.(c) illustrates $\mathbb{T}_S$, with respect to the number of UEs (i.e., N). As depicted, the throughput satisfaction is decreasing almost linearly for B&C solution, while the UE density is increasing. Indeed, when the throughput demand grows, the radio resources become scarce which makes the selection of the appropriate set of resource blocks extremely challenging. Note that DL−USA'$\mathbb{T}_S$ becomes stationary with nearly the same performance as B&C when N is higher than 75.

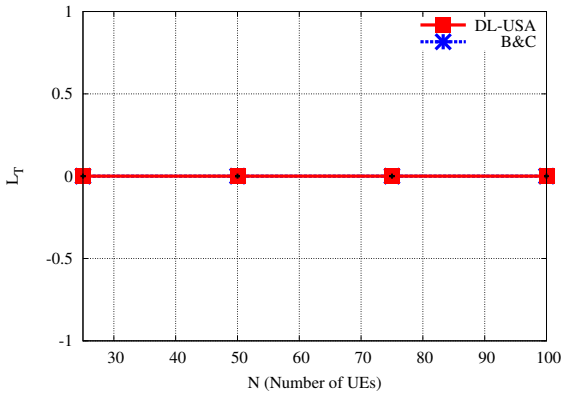


Figure 6.4: Latency penalty

In Figure 6.3.(d), we illustrate $\mathbb{C}_D$ with respect to the number of UEs. It is straightforward to see that, both B&C and DL−USA generate an increasing $\mathbb{C}_D$ when N in $[25, 75]$ while DL−USA generates a higher cost of deployment $\mathbb{C}_D$ for $N = 100$. Hence, DL−USA is able to afford an acceptable amount of cost of deployment while generating a scable allocation decision for link and computational resources.

Figure 6.4 illustrates the penalty $\mathbb{L}_T$ in accordance of the UEs' density. As we can see, our approach DL−USA along with the baseline solution keep a zero penalty which implies that all UEs are constantly served with splits satisfying their latency demands.

6.4 Conclusion

In this Chapter, we proposed a novel Deep Learning User-centric approach for RAN Slice Allocation prediction, DL−USA, that jointly optimizes link, computational and radio resource provisioning. DL−USA jointly meets the overall user requirement in terms of served throughput and latency, while minimizing the RAN deployment cost. The performance of DL−USA is evaluated throughout extensive simulations. Obtained results highlight the effectiveness of our proposal in terms of throughput satisfaction rate and total deployment cost, while responding within the 5G real time constraint.

CHAPTER 7

CONCLUSION

Contents

In this Chapter, we summarize our proposals outlined in this thesis and give insights into the major obtained results. Then, we discuss the possible future research directions that can be considered as extension to our works. Based on the discussed 5G challenges, we propose to further improve our proposals on three steps following i) Short-term, ii) medium-term and iii) long-term perspectives. Finally, we detail the list of publications that have been achieved during this thesis in Section 7.3.

7.1 Summary of contributions

In this thesis, we addressed the C-RAN resource provisioning problem within the context of 5G. The fundamental challenge is how to achieve optimal allocation scheme that fulfills the increased user demand generated by heterogeneous type of services, while minimizing the operational cost of resource deployment. To do so, we propose a fine-grained resource allocation scheme, on user basis, in order to achieve higher benefits. First, we propose a new disaggregated RAN architecture in compliance with the 3GPP specification [14], enabling on-demand deployment of radio, computational and link resources, denoted by `AgilRAN`. We propose to integrate the C-RAN concept and techniques such as Network Function Virtualization, Software Defined Network and RAN functional splits to provide the required flexibility. Second, we formulate the problem of user-centric RAN slice allocation as an ILP problem and solve it based on strategies using heuristics and machine learning techniques. In the first stage, we put forward

a novel user-centric functional split orchestration optimization scheme, so called SPLIT−HPSO. The latter is based on the Swarm Particle Optimization approach [99], aiming at optimizing the functional split selection for each user, while taking into consideration the load variation. Then, in the second stage, we put forward a user-centric RAN slice allocation optimization scheme, so called E2E−USA. The latter is based on Swarm Particle Optimization and Dijkstra approaches to deal with the joint radio, computation and link resource allocation problem. Finally, we propose to make use of the machine learning approach with deep learning to achieve real time performance. Wherein, we make use of the bidirectional Long-Short-Term-Memory (biLSTM) model [106] to propose Deep Learning approach for optimized User-centric Slice Allocation (DL−USA) in 5G RAN. Hereafter, we will summarize our main contributions.

The first contribution is a survey on C-RAN resource provisioning strategies, that includes the partial centralization scheme (i.e., the RAN functional split). We propose to classify the latter strategies into two essential groups: i) RAN placement approaches and ii) RAN slice allocation approaches. The first group of research works aim at minimizing the RAN operational cost by searching for the optimal functional split decision. The second group integrates the radio resource allocation to achieve optimal functional split decision. The latter approach takes into consideration the UEs QoS requirements, while minimizing the operational cost of resource deployment.

The second contribution consists in proposing a cost efficient C-RAN architecture enabling on-demand deployment of RAN resources, while dealing with temporal load variation of users. This Agile C-RAN architecture, denoted by AgilRAN is multi-sited which is in compliance with 3GPP NG-RAN architecture [Chapter 2, p. 14]. AgilRAN is managed by a user-centric split orchestration framework which performs baseband function placement and their interconnection taking into account real-time network state. The main idea behind our design is to ensure a user level orchestration of baseband functions in a hierarchical Cloud infrastructure, while using lightweight virtualization techniques. Characterized by two-level sites of processing, AgilRAN enables the placement of baseband processing network functions traditionally attached to the radio, into the Cloud, while considering their stringent requirements in terms of latency and bandwidth.

The third contribution consists in elaborating a novel user-centric functional split allocation scheme that aims at minimizing the RAN deployment cost, while considering the requirements of its baseband functions and the capabilities of the Cloud infrastructure. Since the problem is NP-hard and in order to deal with its computational hardness, we propounded a new scalable heuristic based on Swarm Particle Optimization approach [99], denoted as SPLIT−HPSO. Our scheme is proved to be scalable running within four Transmission Time Interval (TTI) units which makes our solution operational. We expect that the optimization process is triggered periodically to optimize the deployment cost in a pro-active manner. We further quantify the user split gain as function of the traffic load with reference to a quantitative model and compare it to the baseline cell splits. We have shown that SPLIT−HPSO achieves good performances in terms of total deployment cost and resolution time. Our proposal is evaluated in large scale high density of users. In addition, we validate the proposed solution within our experimental C-RAN prototype, which makes use of OAI [20] and FlexRAN Controller [53]. However, the adopted approach is limited to one cell and does not consider the UE latency requirement, while performing the baseband function placement. Besides, the required amounts of computational and link resources for user-centric functional splits depend on the user traffic load. Hence, the baseband function placement can be further optimized when integrating the radio resource allocation in the split selection decision. This challenge expresses the RAN slice allocation, which will be addressed in the next contribution.

The fourth contribution consists in creating RAN user slices on top of the proposed architecture with joint radio, computational and link resource allocation on user basis. The aim is to integrate the radio resource allocation in the RAN deployment scheme. Therefore, we go a step further and propose a RAN slicing approach with joint optimal radio resource allocation and user split selection on user basis. Considering that the optimization problem is NP-Hard, we propose a low-cost and efficient heuristic algorithm for RAN user-centric slice allocation, so called `E2E-USA`. We make use of the regression linear method to approximate the final served user throughput. Then, our new scheme is based on the Particle Swarm Optimization and Dijkstra approaches to achieve the required trade off. We take into consideration the UE quality-of-service requirements in terms of required throughput and latency, while tuning efficiently the underlying RAN resource usage, leveraging the functional split. We have shown that `E2E-USA` is scalable and achieves optimized user-centric slice allocation solution in a satisfactory time. Based on extensive simulations, `E2E-USA` achieves good performances in terms of joint total throughput satisfaction and deployment cost.

The fifth contribution deals with the real time aspect of the C-RAN allocation procedure. Indeed, the 5G RAN context requires an up-to-date decision within a Transmission Time Interval period less than 1 ms [54]. Therefore, sophisticated algorithms may lead to decisions, that once taken, will be already obsolete and hence not applicable. To this end, we propose a new scheme, named Deep Learning based User Slice Allocation (`DL-USA`), to optimize the RAN user slice allocation decision taking into consideration the real time 5G context. We make use of the bidirectional Long-Short-Term-Memory (biLSTM) model [106] to construct an efficient allocation model. Once trained, the model can be invoked in an online manner to generate real-time RAN slice decision based on collected RAN input parameters of end-users and the physical infrastructure. Hence, the allocation scheme is generated in real-time upon input data without performing an exhaustive calculation task.

7.2 Future work

7.2.1 Short-term perspectives

As a short-term planned work, our objective is to integrate the rest of functional split options (i.e., from Option 2 to Option 7b, detailed in [Chapter 2, p. 15], in our `AgilRAN` Framework. Then, we aim to test and validate our proposals by means of experimentation, with a large set of implemented functional splits. Besides, we aim to extend our proposed architecture into a 3 layer RAN architecture, where the BBU protocol stack can be disaggregated into 3 layers: Radio Unit (RU), Distributed Unit (DU) and Central Unit (CU). Wherein, each element is able to host the baseband functions. The advantage of such an architecture is to deploy two functional splits simultaneously as follows. The intra-PHY splits are deployed on the RU-DU connection (F2 interface), while high split option levels are deployed on the DU-CU connection (F1 interface). In one hand, the intra-PHY splits impose high requirements in terms of bandwidth and latency, while the RU-DU connection implements a dark fiber on a short distance. Eventually, this configuration seems suitable and cost effective. In the other hand, high split option levels that corresponds to splits intra-L2 are relaxing the throughput and latency requirements on the fronthaul link. Then, deploying them on the DU-CU connection is an adequate solution, since a non ideal solution such as Ethernet links are operational and cost effective.

7.2.2 Medium-term perspectives

We aim to target the energy efficiency objective in the elaborated models of RAN slice allocation (i.e., Chapter 5 and Chapter 6). Besides, performance evaluations in Chapter 5 and Chapter 6 have been performed through extensive simulations. However, despite the quality of obtained results depicting the efficiency of our algorithms, we believe that testbed experiments are essential to validate these proposals in real 5G environment. Currently, we have implemented the C-RAN resource orchestration framework `AgilRAN` to enable on-demand deployment of RAN functions. Our next step is to enable the on-demand radio slice deployment on user basis within our SDN radio controller FlexRAN. Hence, as future medium-term works, we aim to extend our `AgilRAN` implementation to enable RAN resource deployment and validate our propositions with proper timing requirements, while also working on other interesting algorithms for BBUs collaborative radio processing.

7.2.3 Long-term perspectives

The fronthaul link is perceived as the bottleneck issue of C-RAN and MNOs put a lot of effort to rethink the RAN architecture in order to relax the high requirements on this connection. With the recent evolution of 5G RAN standards bringing the functional splits and slicing concepts, it is expected that the fronthaul link will be transformed into a multi-service network [56]. Wherein, heterogeneous 5G use case traffics will be routed through a common transport network. Then, it is essential from a management perceptive to provide isolation that limits the interaction between traffic from heterogeneous 5G services. At the end, routing in the 5G transport network is another challenge to consider.

7.3 Publications

This section summarizes the publications that have resulted during this thesis:

- **Journals**
  - Salma Matoussi, Ilhem Fajjari, Salvatore Costanzo, Nadjib Aitsaadi and Rami Langar, “5G RAN: Functional Split Orchestration Optimization”, in IEEE Journal on Selected Areas in Communications (J-SAC), vol. 38, no. 7, pp. 1448-1463, July 2020.
- **Conference papers**
  - Salma Matoussi, Ilhem Fajjari, Nadjib Aitsaadi and Rami Langar, “Deep Learning Based User Slice Allocation in 5G Radio Access Networks”, in IEEE Local Computer Networks (LCN), Sydney, Australia, November 2020.
  - Salma Matoussi, Ilhem Fajjari, Nadjib Aitsaadi and Rami Langar, “User Slicing Scheme with Functional Split in 5G Cloud-RAN”, in IEEE Wireless Communications and Networking Conference (WCNC), Seoul, South Korea, April 2020.
  - Salma Matoussi, Ilhem Fajjari, Nadjib Aitsaadi, Rami Langar and Salvatore Costanzo, “Joint Functional Split and Resource Allocation in 5G Cloud-RAN”, in IEEE International Conference on Communications (ICC), Shanghai, China, May 2019.

  - Salma Matoussi, Ilhem Fajjari, Salvatore Costanzo, Nadjib Aitsaadi and Rami Langar, "A User Centric Virtual Network Function Orchestration For Agile 5G Cloud-RAN", in IEEE International Conference on Communications (ICC), Kansas City, MO, USA, May 2018.

- **Technical reports**

  - Salvatore Costanzo, Salma Matoussi et Rami Langar, "Implémentation d'un Réseau d'Accès Radio en tant que Service (RANaaS)", projet ELASTIC, Délivrable D3.2 - Logiciel et Prototype LIP6, Décembre 2017.
  - Salma Matoussi et Rami Langar, "Algorithmes d'orchestration de l'architecture virtualisée", projet ELASTIC, Délivrable D2.3a - Rapport sur les algorithmes d'urbanisation et allocation des ressources, Mars 2017.

- **Posters**

  - Salma Matoussi, Salvatore Costanzo and Rami Langar, "SDN-based virtual RAN", poster in RESCOM 2016, Guidel-Plages en Bretagne, June 2016.
  - Salma Matoussi, Salvatore Costanzo and Rami Langar, "Bringing SDN to C-RAN: implementation challenges in 5G testbeds", poster in ONOS Build 2016, Paris, November 2016.

LIST OF FIGURES

LIST OF TABLES

REFERENCES

[1] Cisco, "Cisco Annual Internet Report (2018-2023), White Paper," 2020. [online] Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] China Mobile Research Institute, "C-RAN: The Road towards Green RAN, White Paper," 2011.

[3] A. Checko, H. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks - A Technology Overview," *IEEE Commun. Surv. Tutor.*, pp. 405–426, 2015.

[4] L. Gavrilovska, V. Rakovic, and D. Denkovski, "From Cloud RAN to Open RAN," *Wireless Personal Communications*, 2020.

[5] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," *J. Netw. Syst. Manag.*, p. 567–619, 2015.

[6] Next Generation Mobile Network (NGMN) Alliance, "CoMP Evaluation and Enhancement," 2015. [online] Available: https://www.ngmn.org/wp-content/uploads/NGMN_RANEV_D3_CoMP_Evaluation_and_Enhancement_v2.0.pdf.

[7] M. Hossain, A. Mahin, T. Debnath, F. Mosharrof, and K. Islam, "Recent research in Cloud Radio Access Network (C-RAN) for 5G cellular systems - A survey," *J. Netw. Comput.*, p. 31–48, 2019.

[8] 3GPP, "TS 28.530 version 15.3.0 (Release 15), 5G ; Management and orchestration; Concepts, use cases and requirements," 2020. [online] Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128530/15.03.00_60/ts_128530v150300p.pdf.

[9] International Telecommunication Union (ITU), "Report ITU-R M.2410-0 – Minimum requirements related to technical performance for IMT-2020 radio interface(s)," 2017. [online] Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf.

[10] International Telecommunication Union (ITU), "MT Vision M.2083 – Framework and overall objectives of the future development of IMT for 2020 and beyond," 2015. [online] Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf.

[11] Next Generation Mobile Network (NGMN) Alliance, "NGMN Network Slicing "Description of Network Slicing Concept," 2016. [online] Available: https://www.ngmn.org/publications/description-of-network-slicing-concept.html.

[12] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, "Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings," *Journal of Communications and Networks*, pp. 162–172, 2016.

[13] V. Suryaprakash, P. Rost, and G. Fettweis, "Are heterogeneous cloud-based radio access networks cost effective?," *IEEE Journal on Selected Areas in Communications*, pp. 2239–2251, 2015.

[14] 3GPP, "TR 38.801 (Release 14), Study on new radio access technology: Radio access architecture and interfaces," 2017.

[15] IEEE Next Generation Fronthaul Interface (NGFI) Working Group, "1914.1 - Standard for Packet-based Fronthaul Transport Networks," 2019. [online] Available: https://standards.ieee.org/standard/1914_1-2019.html.

[16] Small Cell Forum, "Small cell virtualization: Functional splits and use cases," 2016. [online] Available: https://scf.io/en/documents/159_-_Small_cell_virtualization_functional_splits_and_use_cases.php.

[17] 3GPP, "TS 38.401 V.15.5.0 (Release 15), NG-RAN; Architecture description," 2019. [online] Available: https://www.etsi.org/deliver/etsi_ts/138400_138499/138401/15.05.00_60/ts_138401v150500p.pdf.

[18] Next Generation Mobile Network (NGMN) Alliance, "Further study on critical C-RAN technologies," 2015. [online] Available: https://www.ngmn.org/wp-content/uploads/NGMN_RANEV_D2_Further_Study_on_Critical_C-RAN_Technologes_v1.0.pdf.

[19] Open Radio Access Network (O-RAN) Alliance 2018. [online] Available: https://www.o-ran.org/.

[20] "OpenAirInterface Simulator/Emulator," [online] Available: http://www.openairinterface.org/.

[21] N. Nikaein, E. Schiller, R. Favraud, R. Knopp, I. Alyafawi, and T. Braun, "Towards a cloud-native radio access network," in *Advances in mobile cloud computing and big data in the 5G era*, pp. 171–202, Springer, 2017.

[22] "Towards building Cloud-Native Radio Access Network using OpenAirInterface," [online] Available: https://www.openairinterface.org/?page_id=1808.

[23] "TIP OpenRAN: Toward Disaggregated Mobile Networking," [online] Available: https://cdn.brandfolder.io/D8DI15S7/as/qc19tk-54bsw-305pae/TIP_OpenRAN_-Heavy_Reading_May_2020-_White_Paper.pdf.

[24] "free5GC open-source project," [online] Available: https://www.free5gc.org/.

[25] "Open5GS open-source project," [online] Available: https://github.com/open5gs/open5gs.

[26] "OMEC (Open Evolved Mobile Core) open-source project," [online] Available: https://www.fiercetelecom.com/telecom/t-mobile-poland-boots-up-fixed-mobile-service-using-onf-s-open-source-epc.

[27] B. Dzogovic, B. Feng, T. Van Do, *et al.*, "Building virtualized 5G networks using open source software," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018.

[28] H.-C. Chang, B.-J. Qiu, C.-H. Chiu, J.-C. Chen, F. J. Lin, D. De La Bastida, and B.-S. P. Lin, "Performance evaluation of Open5GCore over KVM and Docker by using Open5GMTC," in *NOMS- IEEE/IFIP Network Operations and Management Symposium*, 2018.

[29] L. Gavrilovska, V. Rakovic, and D. Denkovski, "Aspects of resource scaling in 5G-MEC: Technologies and opportunities," in *IEEE Globecom Workshops (GC Wkshps)*, 2018.

[30] V. Q. Rodriguez and F. Guillemin, "Towards the deployment of a fully centralized Cloud-RAN architecture," in *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017.

[31] "Data Centers Efficiency," [online] Available: https://www.google.com/about/datacenters/efficiency/internal/.

[32] X. Wang, S. Thota, M. Tornatore, H. S. Chung, H. H. Lee, S. Park, and B. Mukherjee, "Energy-Efficient Virtual Base Station Formation in Optical-Access-Enabled Cloud-RAN," *IEEE Journal on Selected Areas in Communications*, pp. 1130–1139, 2016.

[33] D. Pompili, A. Hajisami, and T. X. Tran, "Elastic resource utilization framework for high capacity and energy efficiency in cloud RAN," *IEEE Communications Magazine*, pp. 26–32, 2016.

[34] M. Y. Lyazidi, N. Aitsaadi, and R. Langar, "A dynamic resource allocation framework in LTE downlink for Cloud-Radio Access Network," *Computer Networks*, pp. 101–111, 2018.

[35] M. Fiorani, S. Tombaz, J. Martensson, B. Skubic, L. Wosinska, and P. Monti, "Modeling energy performance of C-RAN with optical transport in 5G network scenarios," *IEEE/OSA Journal of Optical Communications and Networking*, pp. B21–B34, 2016.

[36] D. Sabella, A. Domenico, E. Katranaras, M. Imran, M. Di Girolamo, U. Salim, M. Lalam, K. Samdanis, and A. Mäder, "Energy Efficiency Benefits of RAN-as-a-Service Concept for a Cloud-Based 5G Mobile Network Infrastructure," *IEEE Access*, pp. 1586 – 1597, 2014.

[37] A. Mignotte and O. Peyran, "Reducing the Complexity of ILP Formulations for Synthesis," in *isss*, pp. 58–64, 1997.

[38] A. Tukmanov, M. A. Lema, I. Mings, M. Condoluci, T. Mahmoodi, Z. Al-Daher, and M. Dohler, "Fronthauling for 5G and beyond," *Access, Fronthaul and Backhaul Networks for 5G & Beyond*, p. 139, 2017.

[39] P. Camps-Aragó, S. Delaere, and P. Ballon, "5G Business Models: Evolving Mobile Network Operator Roles in New Ecosystems," in *IEEE CTTE-FITCE: Smart Cities Information and Communication Technology*, 2019.

[40] 3GPP, "3GPP TS 23.251 (Release 11), Network Sharing; Architecture and functional description," 2009.

[41] S. Khatibi, L. Caeiro, L. S. Ferreira, L. M. Correia, and N. Nikaein, "Modelling and implementation of virtual radio resources management for 5G Cloud RAN," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–16, 2017.

[42] M. Morcos, T. Chahed, L. Chen, J. Elias, and F. Martignon, "A two-level auction for resource allocation in multi-tenant C-RAN," *Computer Networks*, pp. 240–252, 2018.

[43] Y. Zhu, H. Yu, R. A. Berry, and C. Liu, "Cross-network prioritized sharing: an added value MVNO's perspective," in *IEEE INFOCOM - Conference on Computer Communications*, 2019.

[44] C. Gao, G. Ozcan, J. Tang, M. C. Gursoy, and W. Zhang, "R-cloud: A cloud Framework for enabling Radio-as-a-Service over a Wireless Substrate," in *IEEE International Conference on Network Protocols (ICNP)*, 2016.

[45] Y. L. Lee, J. Loo, and T. C. Chuah, "A new Network Slicing Framework for multi-tenant Heterogeneous Cloud Radio Access Networks," in *IEEE International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, 2016.

[46] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Pérez, "Network Slicing Games: enabling Customization in multi-Tenant Mobile Networks," *IEEE/ACM Transactions on Networking*, pp. 662–675, 2019.

[47] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: a Software-defined RAN Architecture via Virtualization," *ACM SIGCOMM computer communication review*, pp. 549–550, 2013.

[48] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined Radio Access Network," in *Proceedings of ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013.

[49] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "SoftMobile: Control Evolution for Future Heterogeneous Mobile Networks," *IEEE Wireless Communications*, pp. 70–78, 2014.

[50] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "Openradio: a Programmable Wireless Dataplane," in *Proceedings of the first workshop on Hot topics in software defined networks*, 2012.

[51] W. Wu, L. E. Li, A. Panda, and S. Shenker, "PRAN: Programmable Radio Access Networks," in *Proceedings of ACM Workshop on Hot topics in Networks*, 2014.

[52] T. LeAnh, N. H. Tran, D. T. Ngo, and C. S. Hong, "Resource Allocation for Virtualized Wireless Networks with Backhaul constraints," *IEEE Communications Letters*, pp. 148–151, 2016.

[53] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A flexible and programmable platform for software-defined radio access networks," in *Proceedings of the International on Conference on emerging Networking EXperiments and Technologies*, 2016.

[54] 3GPP, "TR 38.804 V1.0.0 (Release 14), Study on New Radio Access Technology; Radio Interface Protocol Aspects ," 2017. [online] Available: https://www.etsi.org/deliver/etsi_ts/138400_138499/138401/15.05.00_60/ts_138401v150500p.pdf.

[55] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee, "5G Fronthaul–Latency and Jitter Studies of CPRI over Ethernet," *Journal of Optical Communications and Networking*, pp. 172–182, 2017.

[56] "Transport network support of IMT-2020/5G," ITU-T, Geneva, Switzerland, Rep. GSTR-TN5G, 2018.

[57] 3GPP, "TR 38.801 V2.0.0 (Release 14), Technical Specification Group Radio Access Network; Study on New Radio Access Technology; Radio Access Architecture and Interfaces," 2017.

[58] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5G radio access network architecture based on flexible functional control/user plane splits," in *IEEE European Conference on Networks and Communications (EuCNC)*, 2017.

[59] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Communications Surveys & Tutorials*, pp. 146–172, 2018.

[60] P. E. Mogensen, K. Pajukoski, E. Tiirola, E. Lähetkangas, J. Vihriälä, S. Vesterinen, M. Laitila, G. Berardinelli, G. W. O. Da Costa, L. G. U. Garcia, F. M. L. Tavares, and A. F. Cattoni, "5G Small Cell Optimized Radio Design," Globecom. IEEE Conference and Exhibition, 2013.

[61] CPRI Consortium, "CPRI Specification V7.0 Common Public Radio Interface (CPRI) ," 2015.

[62] C. Chia-Yu, S. Ruggero, N. Navid, S. Thrasyvoulos, and B. Christian, "Impact of packetization and functional split on c-ran fronthaul performance," in *Proc. of IEEE ICC*, 2016.

[63] IEEE Standard P1914.1/D1.1, "Draft Standard for Packet-Based Fronthaul Transport Networks," 2018.

[64] IEEE Standard P1914.3/D3.2, "Draft Standard for Radio Over Ethernet Encapsulations and Mappings," 2018.

[65] Next Generation Mobile Network (NGMN) Alliance , "5G End-To-End Architecture Framework v.3.0.8," 2019. [online] Available: https://www.ngmn.org/publications/5g-end-to-end-architecture-framework-v3-0-8.html.

[66] Mobile CORD 2018. [online] Available: https://wiki.opencord.org/display/CORD/Mobile+CORD.

[67] A. Douik, H. Dahrouj, T. Y. Al-Naffouri, and M. Alouini, "Coordinated Scheduling for the Downlink of Cloud Radio Access Networks," in *IEEE International Conference on Communications (ICC)*, 2015.

[68] M. Ali, Q. Rabbani, M. Naeem, S. Qaisar, and F. Qamar, "Joint user Association, Power Allocation, and Throughput Maximization in 5G H-CRAN Networks," *IEEE Transactions on Vehicular Technology*, pp. 9254–9262, 2017.

[69] S.-H. Wu, H.-L. Chao, C.-H. Ko, S.-R. Mo, C.-F. Liang, and C.-C. Cheng, "Green Spectrum Sharing in a Cloud-based Cognitive Radio Access Network," in *IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013.

[70] Y. Shi, J. Zhang, and K. B. Letaief, "Group Sparse Beamforming for Green Cloud-RAN," *IEEE Transactions on Wireless Communications*, pp. 2809–2823, 2014.

[71] S. Ali, A. Ahmad, and A. Khan, "Energy-efficient Resource Allocation and RRH Association in multitier 5G H-CRANs," *Transactions on Emerging Telecommunications Technologies*, p. e3521, 2019.

[72] K. Zhang, W. Tan, G. Xu, C. Yin, W. Liu, and C. Li, "Joint RRH Activation and Robust Coordinated Beamforming for massive MIMO Heterogeneous Cloud Radio Access Networks," *IEEE Access*, pp. 40506–40518, 2018.

[73] X. Wang, A. Alabbasi, and C. Cavdar, "Interplay of Energy and Bandwidth Consumption in CRAN with Optimal Function Split," in *IEEE International Conference on Communications (ICC)*, 2017.

[74] A. Alabbasi and C. Cavdar, "Delay-aware Green Hybrid CRAN," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2017.

[75] M. Tohidi, H. Bakhshi, and S. Parsaeefard, "Flexible function splitting and resource allocation in c-ran for delay critical applications," *IEEE Access*, pp. 26150–26161, 2020.

[76] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based Framework for Flexible BaseBand Function Splitting and Placement in C-RAN," in *IEEE International Conference on Communications (ICC)*, 2015.

[77] M. Awais, A. Ahmed, M. Naeem, M. Iqbal, W. Ejaz, A. Anpalagan, and H. S. Kim, "Efficient joint User Association and Resource Allocation for Cloud Radio Access Networks," *IEEE Access*, pp. 1439–1448, 2017.

[78] W. Xia, J. Zhang, T. Q. Quek, S. Jin, and H. Zhu, "Joint Optimization of Fronthaul Compression and Bandwidth Allocation in Heterogeneous CRAN," in *IEEE Global Communications Conference*, 2017.

[79] M. Baghani, S. Parsaeefard, and T. Le-Ngoc, "Multi-objective Resource Allocation in Density-aware Design of C-RAN in 5G," *IEEE Access*, pp. 45177–45190, 2018.

[80] Y. Shi, J. Zhang, and K. B. Letaief, "Optimal Stochastic Coordinated Beamforming for Wireless Cooperative Networks with CSI Uncertainty," *IEEE Transactions on Signal Processing*, pp. 960–973, 2014.

[81] M. Qian, W. Hardjawana, J. Shi, and B. Vucetic, "Baseband Processing Units Virtualization for Cloud Radio Access Networks," *IEEE Wireless Communications Letters*, pp. 189–192, 2015.

[82] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, "Robust Layered Transmission and Compression for Distributed Uplink Reception in Cloud Radio Access Networks," *IEEE Transactions on vehicular technology*, pp. 204–216, 2013.

[83] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative Analysis of Split Base Station Processing and Determination of Advantageous Architectures for LTE," *Bell Labs Technical Journal*, pp. 105–128, 2013.

[84] I. Koutsopoulos, "Optimal Functional Split Selection and Scheduling Policies in 5G Radio Access Networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017.

[85] A. Tzanakaki, M. Anastasopoulos, D. Simeonidou, I. Berberana, D. Syrivelis, T. Korakis, P. Flegkas, D. C. Mur, I. Demirkol, J. Gutiérrez, *et al.*, "5G Infrastructures Supporting end-user and Operational Services: The 5G-XHaul Architectural Perspective," in *IEEE International Conference on Communications Workshops (ICC)*, 2016.

[86] X. Wang, L. Wang, S. E. Elayoubi, A. Conte, B. Mukherjee, and C. Cavdar, "Centralize or distribute? a techno-economic study to design a low-cost cloud radio access network," in *IEEE International Conference on Communications (ICC)*, 2017.

[87] B. M. Khorsandi, F. Tonini, E. Amato, and C. Raffaelli, "Dedicated Path Protection for Reliable Network Slice Embedding based on Functional Splitting," in *IEEE International Conference on Transparent Optical Networks (ICTON)*, 2019.

[88] A. Pelekanou, M. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Provisioning of 5G Services Employing Machine Learning Techniques," in *IEEE International Conference on Optical Network Design and Modeling (ONDM)*, 2018.

[89] H. Yu, F. Musumeci, J. Zhang, Y. Xiao, M. Tornatore, and Y. Ji, "DU/CU Placement for C-RAN over Optical Metro-Aggregation Networks," in *Optical Network Design and Modeling*, Springer International Publishing, 2020.

[90] B. M. Khorsandi, D. Colle, W. Tavernier, and C. Raffaelli, "Adaptive Function Chaining for Efficient Design of 5G Xhaul," in *Optical Network Design and Modeling*, Springer International Publishing, 2020.

[91] N. Gkatzios, M. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Dynamic Softwarised RAN Function Placement in Optical Data Centre Networks," in *Optical Network Design and Modeling*, Springer International Publishing, 2020.

[92] C. Song, M. Zhang, Y. Zhan, D. Wang, L. Guan, W. Liu, L. Zhang, and S. Xu, "Hierarchical Edge Cloud enabling Network Slicing for 5G Optical Fronthaul," *Journal of Optical Communications and Networking*, pp. B60–B70, 2019.

[93] C.-Y. Chang and N. Nikaein, "RAN runtime Slicing System for Flexible and Dynamic Service Execution Environment," *IEEE Access*, pp. 34018–34042, 2018.

[94] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic Network Slicing for multitenant Heterogeneous Cloud Radio Access Networks," *IEEE Transactions on Wireless Communications*, pp. 2146–2161, 2018.

[95] D. Harutyunyan and R. Riggio, "Flexible Functional Split in 5G Networks," in *IEEE International Conference on Network and Service Management (CNSM)*, 2017.

[96] G. Tseliou, F. Adelantado, and C. Verikoukis, "Netslic: Base Station Agnostic Framework for Network Slicing," *IEEE Transactions on Vehicular Technology*, pp. 3820–3832, 2019.

[97] "USRP B200/B210 Specification Sheet," [online] Available: https://www.ettus.com/product/details/UB200-KIT.

[98] C.-Y. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "FlexCRAN: a Flexible Functional Split Framework over Ethernet Fronthaul in Cloud-RAN," in *IEEE International Conference on Communications (ICC)*, 2017.

[99] R. Eberhart and J. Kennedy, "A new Optimizer using Particle Swarm Theory," in *MHS'95. IEEE Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995.

[100] B. Rouzbehani, L. M. Correia, and L. Caeiro, "A Modified Proportional Fair Radio Resource Management Scheme in Virtual RANs," in *IEEE European Conference on Networks and Communications (EuCNC)*, 2017.

[101] D. Szczesny, A. Showk, S. Hessel, A. Bilgic, U. Hildebrand, and V. Frascolla, "Performance Analysis of LTE Protocol Processing on an ARM based Mobile Platform," in *IEEE International Symposium on System-on-Chip*, 2009.

[102] R. Ewing and K. Park, *Basic Quantitative Research Methods for Urban Planners*. Routledge, 2020.

[103] 3GPP, "Physical layer procedures , 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)," 2016.

[104] 3GPP, "TS 28.533 V.15.1.0 (Release 15), 5G; Management and orchestration; Architecture framework," 2019.

[105] J. E. Mitchell, "Branch-and-Cut Algorithms for Combinatorial Optimization Problems," *Handbook of applied optimization*, pp. 65–77, 2002.

[106] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *Journal of machine learning research*, pp. 115–143, 2002.

[107] F. E. Curtis and K. Scheinberg, "Optimization Methods for Supervised Machine Learning: From linear models to Deep Learning," in *Leading Developments from INFORMS Communities*, pp. 89–114, 2017.

[108] W. Ejaz, S. K. Sharma, S. Saadat, M. Naeem, A. Anpalagan, and N. Chughtai, "A comprehensive survey on resource allocation for CRAN in 5G and beyond Networks," *Journal of Network and Computer Applications*, p. 102638, 2020.

[109] I. G. Ben Yahia, J. Bendriss, A. Samba, and P. Dooze, "CogNitive 5G networks: Comprehensive Operator Use Cases with Machine Learning for Management Operations," in *Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017.

[110] L. Le, D. Sinh, B. P. Lin, and L. Tung, "Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management," in *IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018.

[111] A. K. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, and N. M. F. Qureshi, "An Optimal multitier Resource Allocation of Cloud RAN in 5G using Machine Learning," *Transactions on Emerging Telecommunications Technologies*, 2019.

[112] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Machine Learning Based Resource Orchestration for 5G Network Slices," in *IEEE Global Communications Conference (GLOBECOM)*, 2019.

[113] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks," *IEEE Internet of Things Journal*, pp. 7011–7024, 2019.

[114] S. Troia, R. Alvizu, and G. Maier, "Reinforcement Learning for Service Function Chain Reconfiguration in NFV-SDN Metro-Core Optical Networks," *IEEE Access*, pp. 167944–167957, 2019.

[115] X. Chen, Z. Li, Y. Zhang, R. Long, H. Yu, X. Du, and M. Guizani, "Reinforcement learning–based QoS/QoE-aware Service Function Chaining in Software-driven 5G Slices," *Transactions on Emerging Telecommunications Technologies*, p. e3477, 2018.

[116] D. P. Kingma and J. Ba, "Adam: A method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[117] Z. Li, X. Ma, C. Xu, C. Cao, J. Xu, and J. Lü, "Boosting Operational DNN testing Efficiency Through Conditioning," in *Proceedings of ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019.