PLACEMENT OF 5G RAN SLICES IN MULTI-TIER OPEN RAN 5G NETWORKS WITH FLEXIBLE FUNCTIONAL SPLITS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EGEMEN SARIKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

PLACEMENT OF 5G RAN SLICES IN MULTI-TIER OPEN RAN 5G NETWORKS WITH FLEXIBLE FUNCTIONAL SPLITS

submitted by **EGEMEN SARIKAYA** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Prof. Dr. Ertan Onur
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. İbrahim Körpeoğlu
Computer Engineering, Bilkent University ______________

Prof. Dr. Ertan Onur
Computer Engineering, METU ______________

Assist. Prof. Dr. Pelin Angın
Computer Engineering, METU ______________

Date: 10.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Egemen Sarıkaya

Signature :

ABSTRACT

PLACEMENT OF 5G RAN SLICES IN MULTI-TIER OPEN RAN 5G NETWORKS WITH FLEXIBLE FUNCTIONAL SPLITS

Sarıkaya, Egemen
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Ertan Onur

September 2021, 55 pages

The network slicing concept has gained much attention with the development of software-defined network and network function virtualization technologies, enabling logically isolated networks for different purposes in the same network infrastructure. The virtualization of network functions enables functional split of radio access network functions, which can fulfill different 5G radio access network requirements. Functional split can be expressed as deciding the distribution of radio access network functionalities between the radio unit placed at the edge, the distributed unit placed close to the users, and the central unit placed centrally.

In this thesis, we considered the placement of virtualized radio access network functions in a multi-tier 5G radio access network architecture and formulated the problem as a Mixed-Integer Quadratically Constrained Programming, considering different functional split options for each network slice separately. Furthermore, a heuristic approach based on the best fit decreasing strategy, FlexBFDP, is proposed and compared with a state-of-the-art solution existing in the literature. Our experiments with different functional split options showed that flexible functional split utilizes physical network resources and satisfies different network slice requirements better than fixed functional split options. Moreover, our experiments comparing the solutions of the Gurobi solver, FlexBFDP, and the state-of-the-art algorithm showed that FlexBFDP outperforms the state-of-the-art algorithm. Even though two out of three Gurobi solutions outperform FlexBFDP in terms of optimization objective, FlexBFDP is preferable in terms of computation time as it takes less time to find a solution. Our results showed that Gurobi solutions with MIPGap 0.01 and 0.1 perform very similar to each other and achieves up to 21% more objective value compared to the FlexBFDP approach, while FlexBFDP outperforms the solution with MIPGap 1 by up to 26%, and the state-of-the-art by up to 140% when the number of network slice requests increases.

Keywords: VNF placement, baseband function placement, flexible functional split, RAN slicing

ÖZ

5G RADYO ERİŞİM AĞ DİLİMLERİNİN ESNEK FONKSİYONEL BÖLMELERLE ÇOK KATMANLI O-RAN 5G AĞLARINA YERLEŞTİRİLMESİ

Sarıkaya, Egemen
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Ertan Onur

Eylül 2021 , 55 sayfa

Ağ dilimleme kavramı, yazılım tanımlı ağ ve ağ işlevi sanallaştırma teknolojilerinin gelişmesiyle birlikte, aynı fiziksel ağ altyapısında farklı amaçlar için birbirinden mantıksal olarak yalıtılmış ağları mümkün kıldığı için son zamanlarda ilgi çekici bir konu haline gelmiştir. Ağ fonksiyonlarının sanallaştırılması, farklı 5G radyo erişim ağı gereksinimlerini karşılayan radyo erişim ağı fonksiyonlarının bölünüp dağıtık şekilde yerleştirilebilmesini sağlayarak daha esnek ağ yapılarına imkan tanımaktadır. Ağ fonksiyonlarının bölünmesi, uca yerleştirilmiş radyo birimi (RU), kullanıcılara yakın bölgelere yerleştirilen dağıtık birim (DU) ve diğerlerine kıyasla daha merkezi olarak yerleştirilen merkezi birim (CU) arasındaki radyo erişim ağı fonksiyonlarının dağılımına karar verilmesi olarak ifade edilebilir.

Bu çalışmada, çok katmanlı bir 5G radyo erişim ağı mimarisinde sanallaştırılmış radyo erişim ağı fonksiyonlarının yerleştirilmesini, her ağ dilimi için farklı fonksiyonel bölme seçeneklerini ayrı ayrı ele alarak, Karışık Tam Sayılı Kuadratik Kısıtlı Programlama (MIQCP) problemı olarak formülize ettik. Ayrıca, en uygun azaltma stratejisine dayanan bir sezgisel algoritma olan FlexBFDP algoritmasını tasarlayıp literatürdeki başka bir sevgisel algoritmayla kıyasladık. Farklı işlevsel bölme seçenekleriyle yaptığımız deneyler, her ağ dilimi için farklı bölme seçeneği kullanılan esnek işlevsel bölmenin, her ağ diliminin aynı fonksiyonel bölme seçeneğini kullandığı sabit fonksiyonel bölmeye kıyasla farklı ağ dilimi gereksinimlerini daha iyi karşıladığını gösterdi. Gurobi çözücüsünün çözümleri, FlexBFDP algoritması ve literatürde mevcut olan algoritmanın çözümlerini karşılaştıran deneylerimiz, FlexBFDP algoritmasının son teknoloji algoritmadan daha iyi performans gösterdiğini gösterdi. Optimizasyon hedefi açısından üç Gurobi çözümünden ikisi FlexBFDP algoritmasından daha iyi performans gösterse de, FlexBFDP algoritmasının bir sonuç bulma süresi Gurobi çözümlerine kıyasla çok daha az zaman aldığı için büyük ölçekli ağ yapılarında FlexBFDP algoritmasının kullanılması daha uygun olabilir. Sonuçlarımız, MIPGap 0.01 ve 0.1 parametrelerini kullanan Gurobi çözümlerinin birbirine çok benzer performans gösterdiğini ve FlexBFDP algoritmasına kıyasla $\%21$'e kadar daha iyi performans gösterdiğini, fakat ağ dilimi sayısı arttıkça FlexBFDP algoritmasının MIPGap 1 parametresi kullanılan Gurobi çözümünden $\%26$'ya kadar, literatürdeki algoritmadan ise $\%140$'a kadar daha iyi performans gösterdiğini gösterdi.

Anahtar Kelimeler: Sanallaştırılmış ağ fonksiyonu yerleştirmesi, tabanbant fonksiyon yerleştirmesi, esnek fonksiyonel ayrım, radyo erişim ağı dilimleme

To my family and close friends

ACKNOWLEDGMENTS

First of all, I want to thank Prof. Dr. Ertan Onur for his guidance and patience towards the work we done throughout my master education. Before the pandemic, I appreciate the time spent with my collegues at WINSLab, it was a productive and enjoyable journey.

Then, I would like to thank my family for their continuous support through my whole education, without their support I could not overcome the difficulties I faced.

Also this work was supported by Vodafone under the 5G and Beyond Joint Graduate Support Program regulated by BTK. Their financial support was crucial for me to focus on my thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

5G	Fifth Generation Mobile Network
BBU	Baseband Unit
BS	Base Station
C-RAN	Cloud Radio Access Network
CoMP	Coordinated Multipoint
CU	Central Unit
D-RAN	Distributed Radio Access Network
DU	Distributed Unit
eMBB	Enhanced Mobile Broadband
FlexBFDP	Flexible Best Fit Decreasing Placement
FlexNSP	Flexible Network Slice Placement
GOPS	Giga Operations Per Second
HLS	High Layer Split
ILP	Integer Linear Programming
InP	Infrastructure Provider
LLS	Low Layer Split
MAC	Medium Access Control
MIP	Mixed Integer Programming
MIQCP	Mixed Integer Quadratically Constrained Programming
mMTC	massive Machine Type Communications
NFV	Network Function Virtualization
O-CU	Open Central Unit
O-DU	Open Distributed Unit
O-RU	Open Radio Unit

PDCP	Packet Data Convergence Protocol
RAN	Radio Access Network
RLC	Radio Link Control
RRH	Remote Radio Head
RU	Radio Unit
SDN	Software-Defined Network
SOTA	State-of-the-Art
URLLC	Ultra-Reliable Low Latency Communications
VMP	Virtual Machine Placement
VNE	Virtual Network Embedding
VNF	Virtualized Network Function

CHAPTER 1

INTRODUCTION

In this chapter, the scope of the thesis is covered at first. Then, the motivation behind our work is mentioned, and our problem definition and the methodology we used are explained. After listing our contributions to the literature, the outline of the thesis is presented at the end.

1.1 Scope

This thesis focuses on optimizing the limited physical resources of the radio access network (RAN) part of the fifth-generation mobile networks (5G). The physical resources we considered are sites hosting virtual network functions and links connecting these sites. Sites have limited processing capacity, and links have limited bandwidth capacity with network delay, including transport and propagation delay. Open RAN [3] terminology adopted for virtual network functions of 5G radio access network. The core network and the network functions in the core network are out of scope.

1.2 Problem Definition

As mobile networks are getting more complex, dedicated machines providing specific network functions are not flexible enough to satisfy the diverse requirements of different 5G use cases. Dedicated proprietary infrastructures are replaced by general-purpose systems that can host various Virtual Network Functions (VNFs) thanks to the software-defined network (SDN) [4] and network function virtualization (NFV)

[5] technologies. These technologies enable us to design flexible network placements as sites with general-purpose machines can be scattered in different locations, and a virtual network function can be placed to any of these sites as long as the function's requirements are satisfied.

The virtualization of the network functions leads to the network slicing concept, one of the prominent paradigms to answer the diverse needs of 5G networks. One network slice may require low latency, while another may be latency tolerant but requires high bandwidth capacity. A network slice consists of a set of VNFs, and a traffic flow through these VNFs. In order to satisfy the needs of different network slices within a network infrastructure, the optimal placement of the VNFs of network slices onto the network infrastructure is one crucial challenge. The problem of placing VNFs onto the network infrastructure most efficiently and profitably considering different requirements of these functions is widely tackled in the literature, named the VNF placement problem [6–13].

VNF placement and Virtual Network Embedding (VNE) [14–17] are similar to each other. VNF placement problem concerns how to place VNFs to a particular physical infrastructure, whereas the VNE problem concerns the optimal mapping of virtual nodes and links satisfying specific requirements to substrate nodes and links.

New 5G services have a wide range of requirements [18], such as low latency, high throughput, and high connection density. 5G Radio access network (RAN) has to accommodate these diverse requirements. In previous mobile generation networks, Cloud-RAN (C-RAN) architecture decouples the base station (BS) functions into two entities, the baseband unit (BBU) and the remote radio head (RRH) [19]. RRHs are placed at the edge, and BBUs are centralized into BBU pools to achieve low cost and low energy consumption. However, C-RAN requires a tremendous amount of fronthaul capacity from aggregation sites to operate. In 5G, different functional splits are proposed [2] further to decouple the baseband unit into a distributed unit (DU), a centralized unit (CU), and RRU in C-RAN becomes the radio unit (RU). In the context of Open RAN, these elements are named O-RAN distributed unit (O-DU), O-RAN centralized unit (O-CU), and O-RAN radio unit (O-RU) [3]. Adopting Open RAN has several benefits. One of the essential benefits is interoperability between multi-vendor devices. The interoperability aspect prevents vendor lock-ins and provides flexibility while designing the network and replacing or upgrading the network with products from different vendors. The functional split architecture we considered, namely the dual split, splits the baseband functionalities with a high layer split (HLS) option and a lower layer split (LLS) option, which are illustrated in Figure 3.2.

In this thesis, we considered the optimal placement of network slices in multi-tier 5G radio access networks with a flexible functional split option for each network slice.

The placement of a network slice is formulated as a virtual network function placement problem named Flexible Network slice Placement (FlexNSP), considering a network slice is a set of virtualized baseband functions, O-RU, O-DU, and O-CU, chained under bandwidth capacity and latency requirements. To solve FlexNSP, we utilized Gurobi solver to find near-optimal solutions and proposed a heuristic approach based on the best fit decreasing strategy used in bin packing problems, Flexible Best Fit Decreasing Placement (FlexBFDP), and compare these solutions with a state-of-the-art solution from the literature.

1.3 Contributions

Our contributions in this thesis are as follows:

1. Placement of a 3-layer RAN slice (O-RU, O-DU, O-CU) and flexible dual split options are jointly considered. Most of the works in the literature tackle these problems separately.

2. We used different high layer split (HLS) and low layer split (LLS) options for each RAN slice as a variable, allowing the selection of optimum functional split for different slices. Each RAN slice has its own HLS and LLS option to find the optimal use of limited network resources to accommodate the slice requirements.

3. We consider eMBB, URLLC, and mMTC network slices, three main network slice types, and the impacts of admitting different types of network slices.

4. A network topology of multi-tier aggregation sites is considered to reflect real-life scenarios.

5. As can be seen in Chapter 5, we compared the flexible functional split with the fixed functional split options regarding the utilization of physical network resources and the admission ratio of RAN slice requests.

6. Because of the exponential search space complexity of the optimization problem, we proposed a heuristic solution, namely FlexBFDP, that finds a solution much faster compared to mathematical optimization and compare our results with a state-of-the-art solution [20].

7. The novelty of our approach comes from jointly considering functional split options and their detailed requirements with the placement of the central unit, distributed unit, and radio unit of a network slice, including the routing between these units. In the literature, functional split and placement of network slices are considered separately; however, they are intertwined because determining which functional split is selected impacts the placement of functions directly.

1.4 Outline of the Thesis

The outline of the thesis is as follows. Chapter 2 presents the background information required for understanding this thesis and covers the related work in the literature. Our optimization framework, Placement of 5G RAN slices in Multi-tier O-RAN 5G networks with Flexible Functional Splits, is covered in Chapter 3. Our heuristic approach FlexBFDP discussed in Chapter 4. Then, in Chapter 5, the parameters are presented and explained, the experiment results are discussed. Lastly, Chapter 6 concludes the thesis, presents the final remarks, and discusses the future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Background

In this section, necessary background information is given. Evolution of the radio access network part of the mobile telecommunication system and the technologies that enable this evolution are discussed briefly. [21]

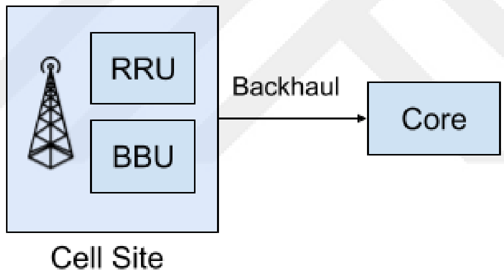

Figure 2.1: D-RAN architecture.

2.1.1 Distributed Radio Access Network (D-RAN)

In the early generations of mobile networks, distributed radio access network (D-RAN) architecture is adopted [22]. As Figure 2.1 shows, the baseband unit (BBU) and the remote radio unit (RRH) are located together at the base station in this architecture. The base station is connected directly to the core of the network. The connection between the base station and the core of the network is called backhaul.

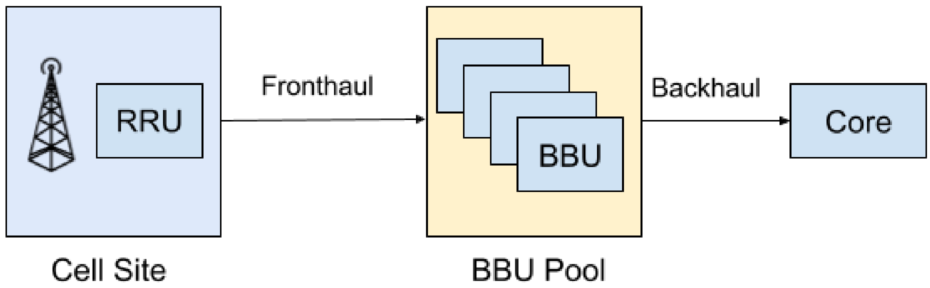

Figure 2.2: C-RAN architecture.

2.1.2 Cloud Radio Access Network (C-RAN)

C-RAN is proposed to overcome the weaknesses of the D-RAN architecture. BBUs are moved to a more central part of the network in C-RAN architecture and bundled together as BBU pools, which can be seen in Figure 2.2. Multiple RRHs can be connected to a BBU pool via the fronthaul network. Then, BBU pools are connected to the core of the network via the backhaul network.

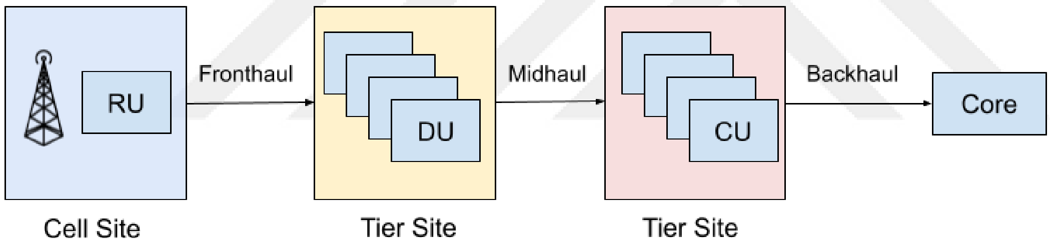

Figure 2.3: 5G New Radio Architecture with functional splitting.

2.1.3 5G Functional Splitting

The centralization of various baseband functionalities offers excellent benefits like coordinated multipoint (CoMP) [23]. However, because data is not processed at the edge of the network, the amount of data transferred in the fronthaul network is immense. The bandwidth requirement of C-RAN architecture is unsustainable when we consider the requirements of 5G networks [24] [25].

The functional split concept is proposed to flexibly design the network to satisfy the diverse requirements of 5G networks. The 5G New Radio architecture with functional splitting can be seen in Figure 2.3. The idea behind the functional split is that determining the distribution of the baseband functionalities (RF, PHY, MAC, RLC, PDCP, RRC) to three processing units named radio unit (RU), distributed unit (DU), and central unit (CU). Over time, different functional split options are proposed by different associations such as 3GPP [2], Small Cell Forum [1]. Single-split considering the functional split between DU and CU is prominent in the literature. However, double-split considering the functional split between RU and DU along with the functional split between DU and CU is a promising technique allowing more flexibility.

2.1.4 Fronthaul and Midhaul

Fronthaul and midhaul are relatively new terminology compared to backhaul because these terms are used with the recent network architectures C-RAN and functional splitting.

As can be seen in Figure 2.1, backhaul is the part of the network that resides between cell sites and the network core. In other words, backhaul is the connection between the access network consisting of cell sites and the core network. When network architectures became more modular, and the access network in these architectures became more scattered, the terms fronthaul and midhaul are the results of the need to name the connection between access network components.

In Figure 2.2, the access network consists of cell sites and BBU pools, and the connection between cell sites and BBU pools is named fronthaul network. Low latency and high bandwidth capacity are the characteristics of fronthaul because as functions are getting more fine-grained, their bandwidth requirements are increasing, and latency requirements are getting stringent [26].

Later, when functional split distributes the access network functionality even further in Figure 2.3, the connection between RU and DU is named fronthaul, and the connection between DU and CU is named midhaul. As we process the data closer to the core network, multiplexing gains reduce the bandwidth requirements. Therefore, midhaul bandwidth requirements are not as high as fronthaul bandwidth requirements but higher than backhaul [27]. The exact numbers will be discussed in chapter 5. The same behavior can be observed for latency requirements of midhaul compared to fronthaul, and it is not as stringent as fronthaul. In all of these cases, the connection that connects the access network to the core is always named backhaul. Since we only considered the access network part of the network, backhaul and its requirements are out of our scope.

2.1.5 Open RAN (O-RAN)

Open RAN (O-RAN) is a concept that enables and enforces interoperability between different vendor network components [28]. Similar to the functional split concept, network functionalities are distributed into three processing units named O-RAN Radio Unit (O-RU), O-RAN Distributed Unit (O-DU), O-RAN Centralized Unit (O-CU).

2.2 Related Work

Our work considers different aspects together as placement of virtual baseband network functions, flexible functional split of baseband functions, and routing between placed functions jointly. In this chapter, significant works about these aspects are classified and presented.

2.2.1 Virtual Network Function (VNF) Placement / Virtual Network Embedding (VNE)

VNF placement and VNE are similar concepts as VNF placement considers placing each virtualized network function onto the physical network satisfying the requirements between these functions. In contrast, VNE considers placing virtualized network functions along with virtualized links and connections onto the physical network in one piece. In other words, VNE can be seen as the mapping between the virtualized blueprint of the network and the physical network.

In [11], Guan et al. worked on a modified version of the VNE problem. They proposed an end-to-end network slice placement algorithm linking virtual resources to physical resources with the objective of minimizing physical network resource usage. In their model, each network slice is a virtual network instance that needs to be mapped to the physical network. They abstracted the infrastructure network as an undirected weighted graph.

The placement algorithm maps virtual network functions to physical resources, then selects the links connecting these functions. They considered three types of slices with different requirements, eMBB slices, mMTC slices, and uRLLC slices. Three different heuristic algorithms are proposed for these slice types separately.

In [12], Alleg et al. worked on the Placement and Chaining of Virtual Network Functions (PC-VNF) problem and formulated the problem as a Mixed-Integer Quadratically Constrained program (MIQCP). Their main difference from the literature is the relation between allocated resources and the processing delay of a VNF. A linear relation is assumed and considered in the mathematical formulation.

VNE problem with functional splitting is formulated as a Mixed Integer Quadratically Constrained Programming (MIQCP), and a MaxSAT problem to minimize the physical network resource usage minimization of network latency in [29]. They suggested different functional split options for different types of network slices. However, placing different types of network slices on the same network structure is not considered, which can affect the performance of individual network slices and the whole network. The topology they used is similar to the one we use, except they do not consider more resources available at more centralized entities. Scarce resource distribution on the edge of the network should be considered because it can affect the quality of results.

A similar problem to VNF placement and VNE is virtual machine placement (VMP). The underlying problem of VMP is the same with VNF placement and VNE problems, the optimal placement of virtual resources to limited physical network infrastructure. VMP considers the optimal placement of virtual machines compared to VNF that considers the optimal placement of virtual network functions that runs on virtual machines. Therefore, we should consider the work done about VMP related to our work.

Different objectives are considered while tackling the VMP problem. In [30], [31], [32], and [33]; the objective is minimizing the cost of placing a set of VMs considering their needs. However, crucial aspects are missing in some of these works like latency is not considered in [31].

Resource usage is another heavily considered objective among the work about VMP problem. The authors of [34], [35], [36], and [37] considered bandwidth usage, CPU usage, power consumption, and resource wastage as their objectives respectively. Reflecting these objectives to our work, despite the fact that our objective value is the financial gain from admitted network slice requests, our consideration of network slice requests' bandwidth and CPU consumption lead to minimal wastage of these resources while maximizing the profit.

VNF chain placement problem is another widely considered topic in the literature related to placement of virtual resources and network slices. It is a similar concept to our approach, the set of VNFs of a network slice request can be considered as a VNF chain and requirements can be reflected to this chain. Moens and Turck [13] was one of the early works about this problem. They proposed an ILP formulation that can be solved in a few seconds due to the low number of restriction in their network topology. Also in [38], the problem was modeled as ILP but it has limited applicability due to the nature of ILP taking too much time to compute a solution in large number of nodes. In [39], [40], and [41] proposed heuristic approaches utilizing First-Fit Decreasing, Variable Neighborhood Search, and load balancing methods respectively. Resiliency is considered, for single node failure and single link failure, in [42] with the objective of maximizing the link utilization.

In [43], Kazemifard et al. proposed an end-to-end VNF and Containerized Network Function (CNF) placement framework for O-RAN 5G networks. They mathematically modeled resource allocation and placement of network functions, considering a single path for network traffic to flow. Their objective is to minimize the end-to-end delay of the data plane. For traversing via multiple paths, they proposed a gradient-based solution to achieve near-optimal results. Their network infrastructure is a three-layered (local, regional, and core) data center architecture. However, they only considered one of the functional split options (options 2 - 7). Adding flexible, functional split to their architecture can yield better results.

In [44], Sriram et al. considered a constraint programming problem on C-RAN to minimize the energy consumption of the network. They adopted functional splitting with caching at the edge clouds to reduce fronthaul bandwidth consumption and content access delay. This idea is designed for content delivery networks rather than URLLC or mMTC network slices because low latency achieved by caching is not helpful for URLLC slices; end-to-end low latency must be achieved for mission-critical use cases. They classified the baseband processing functions as cell processing (CP) and user processing (UP) functions. They defined digital unit (DU) capacity in terms of the number of CP and UP functions. However, the assumption that two functions in the same group (CP or UP) consume the same amount of resources is not realistic.

2.2.2 5G Functional Split

In [45], Xiao et al. proposed a fine-grained functional split architecture alternative to standardized 3GPP functional split options [2] and present a quantitative model for the computational complexity of different functions. However, they do not consider the latency requirement between each Fine-grained unit (FU), which corresponds to the latency consideration of different functional splits.

In [46], a virtual network embedding (VNE) problem with different functional splits in a star-shaped topology is considered. This work has similar aspects to our work as their embedding model includes selecting functional splits flexibly. However, the authors select functional split options per substrate small cells, which takes away the flexibility of selecting a different functional split option for each request. Also, only lower layer splits are considered. Specifically, PHY-RF (option 8), PHY (option 7), and MAC (option 5) splits are considered. Considering both higher and lower layer splits increases flexibility further.

In [47], Coelho et al. illustrated the effects of different functional split options with different network sharing policies. They focused on end-to-end network slicing problems and interactions between slices, isolating each slice or some functions depending on network sharing policies.

In [48], Alba et al. proposed a dynamic functional split design where a RAN can change selected functional split option during run-time depending on the current traffic. Their work illustrates the details of the implementation of functional split and how the procedure of switching between two function split options can be realized. In their implementation, switching between two functional split options is illustrated.

In [49], Zhang et al. worked on maximizing the energy efficiency of a BBU pool in a C-RAN architecture. In their model, they dynamically turn on and off the BBU cards in the BBU pool. They formulate the problem as a version of the 2D bin-packing problem.

Yu et al. [50] considered the placement of DU and CU functions over optical aggregation networks. They proposed an ILP model for baseband function placement. Functional split options 2 and 7 are fixed as fronthaul and midhaul interfaces respectively, flexible function split options are not considered.

In [51], Yu et al. investigate the isolation-aware 5G RAN network slice mapping problem in WDM metro-aggregation networks. Their network structure consists of cell sites (CSs) at the edge, access central offices (COs), main COs, and core COs. They proposed a heuristic algorithm to minimize the number of used central offices and wavelength channels while satisfying the requirements of different types of slices (URLLC, eMBB). They claimed that they proposed a flexible functional split concept, but the flexible functional split is not considered in their work; fronthaul and midhaul interfaces are fixed to split options 2 and 7.

In [52], Harutyunyan et al. introduced a CU placement problem for mobile networks as an integer linear programming (ILP) problem while considering the impact of different functional split options. Since ILP is not scalable to vast networks, they proposed a heuristic algorithm to solve the mentioned problem. They considered PHY-RF split, PHY split, MAC split, and PDCP-RLC split, corresponding to options 8, 7, 5, and 2 in Figure 3.2. In their problem definition, they have virtual DUs, CUs, and links mapped to the substrate DUs, CUs, and links. Unlike this work, our problem definition allows a site to host different DUs and CUs simultaneously, leading to more flexible placements.

In [53], Zorello et al. an optimization framework considering functional split is proposed to minimize power usage of links and nodes. They choose split options 2 (PDCP-RLC) and 6 (MAC-PHY) and compute these options' bandwidth, process, and latency requirements.

[48], [47] and [53] consider the effects of functional split. [12], [11], [49], [45] and [51] work on the placement of network functions. [54], [46], [29], [50], [52], and [20] work on the placement of functions considering different functional splits, but none of them used functional split as a decision variable in their problem definition. To observe the effects of a selected functional split of a slice on other slices in the network, and optimally utilize the physical network infrastructure, flexible functional split per RAN slice should be considered. We considered different types of slices with a unique functional split option assigned to each slice coexisting in the same physical network structure.

Table 2.1: Comparison of the related work.

Work	Problem	Functional Split	Network Slice Types	Objective	Solution
[11]	VNE	Not considered	eMMB, mMTC, uRLLC	Minimize physical network resource usage	Heuristic
[12]	VNF Placement	Not considered	Not considered	Minimize resource consumption	MIQCP
[29]	VNE	Fixed split per slice type	eMBB, mMTC, uRLLC	Minimize network latency	MIQCP, MaxSAT
[13]	VNF Placement	Not considered	Not considered	Minimize resource consumption	ILP
[39]	VNF Placement	Not considered	Not considered	Minimize traffic and response time	ILP, first-fit decreasing (FFD) based heuristic
[40]	VNF Placement	Not considered	Not considered	Minimize required resource allocation	Variable neighborhood search based meta-heuristic
[43]	VNF Placement	Fixed split, options 2 and 7	Not considered	Minimize the end-to-end delay of the data plane	Heuristic named Gradient-based Minimum Delay (GBMD)
[45]	Fine-grained split of an end-to-end slice	Flexible functional split	Not considered	Minimize the number of used processing pools	MILP
[46]	VNE	Flexible functional split per small cell, options 5, 7, and 8	eMBB, mMTC, uRLLC	Minimize inter-cell interference	ILP, heuristic
[47]	Network slice provisioning	Functional split per network policy, options 1 to 8	eMBB, uRLLC	Minimize the network slices' deployment cost	MILP
[48]	Adaptive functional split during runtime	Dynamic functional split	Not considered	Minimize the traffic on the fronthaul	Adaptive functional split heuristic
[49]	2D bin-packing problem for BBU aggregation	Not considered	Not considered	Maximize the energy efficiency	ILP, heuristic
[50]	VNF Placement	Fixed split, options 2 and 7	eMBB, mMTC, uRLLC	Minimize power consumption	ILP
[51]	VNF Placement	Fixed split, options 2 and 7	eMBB, mMTC, uRLLC	Minimize the number of used central offices	Heuristic
[52]	CU placement	Fixed split, options 2, 5, 7, and 8	Not considered	Minimize the power consumption	ILP, heuristic
[53]	CU placement	Fixed split, options 2 and 6	Not considered	Minimize power usage of links and nodes	ILP
Our work	VNF placement	Flexible dual functional split per network slice, options 2, 3, 6, 7, and 8	eMBB, mMTC, uRLLC	Maximize financial gain from admitted network slice requests	MIQCP (FlexNSP), heuristic (FlexBFDP)

CHAPTER 3

PLACEMENT OF NETWORK SLICES WITH FLEXIBLE FUNCTIONAL SPLIT (FLEXNSP)

In this chapter, we explained our network structure and proposed Flexible Network Slice Placement problem (FlexNSP) as a Mixed Integer Quadratically Constrained Program (MIQCP).

Diverse requirements of 5G requires network slicing concept [55] to be utilized to meet different user groups' requirements within the same physical network infrastructure. To represent these different use cases of network slicing, 3GPP proposed three main network slice types [56]. URLLC slices require low latency and high reliability; eMBB slices require high spectral efficiency, high user experienced data rate, and mMTC requiring high connection density (for example, 1 million devices/km^2 in an ITU technical report [57]).

According to the requirements of different user groups of network slices, each network slice fits into one of the main slice types, while the requirements of slices of the same type can differ to some degree. That is, not all the eMBB slices are identical in terms of capacity and latency requirements. There may be more than one suitable functional split option for each slice, satisfying the slice requirements and fitting with the network infrastructure. In our model, we select the optimal functional split option of a slice from the perspective of Infrastructure Providers (InPs), meaning that the selected functional split option maximizes the network's total financial value while satisfying the needs of that network slice. From the perspective of Infrastructure Provider (InP), each of the network slice requests has a value demonstrating the financial gain InP acquires if the network slice is admitted to the network infrastructure. While assigning financial values to the network slice requests, analysis of financial

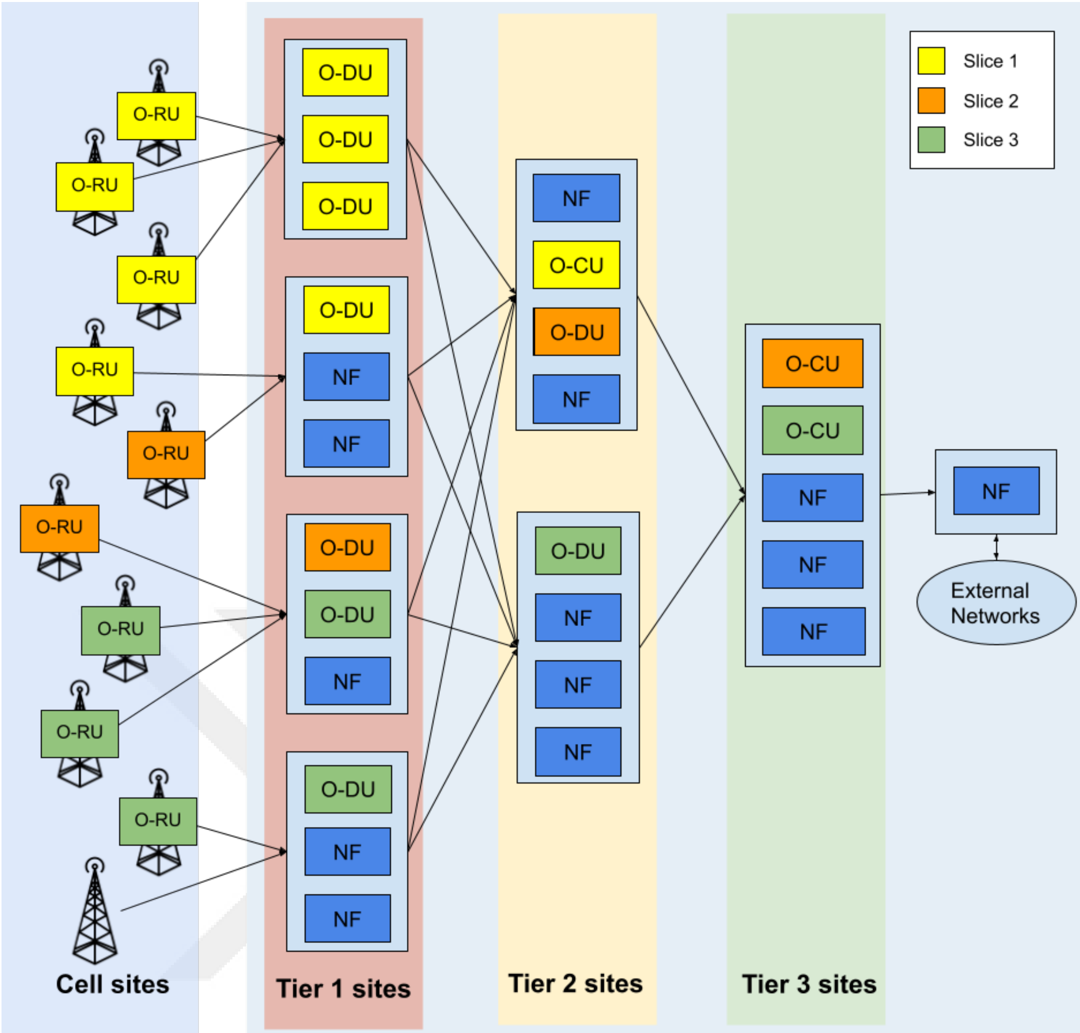

Figure 3.1: Example placement on a multi-tiered 5G RAN structure.

values considering individual network slice requests is out of our work's scope. We employ a simple strategy to assign values to network slice requests, a random value from a defined range of values depending on the type of the network slice.

3.1 Network Structure

This section presents the network structure considered in this thesis. The physical network structure consists of edge sites, tier sites, links that connecting these sites.

In [58], multi-tier 5G RAN illustrated, which can be seen in Figure 3.1. In this example structure, cell sites are connected to tier-1 sites, tier-1 sites are connected to tier-2 sites, and tier-2 sites are connected to a tier-3 site. Note that there can be more than three tiers in a multi-tier structure; the three-layered structure in Figure 3.1 is just an example for illustration purposes. From another perspective, the hierarchy can be considered as tiers in one or more data centers.

There are T site levels in the network such that tier 1 sites are the closest sites to the cell sites; tier T sites are the closest sites to the external networks. $\mathcal{N}$ is the set of sites in the network. $\mathcal{N}_t$ represents the set of tier t sites, There are also cell sites, $\mathcal{N}_c$, connected to $\mathcal{N}_1$ sites. It is convenient to have more tier sites closer to the edge because all cell sites hosting O-RU functions should connect to a physically close tier site to accommodate high bandwidth and low latency requirements of fronthaul communication. Therefore, central tier sites have fewer instances compared to the sites that close to the edge,

$$|\mathcal{N}_t| \geq |\mathcal{N}_{t+1}|, t = 1, ..., T-1.$$

Cell sites and tier sites have limited computational capacity in terms of giga operations per second (GOPS). $\mathcal{C}$ is the set of resource capacities of sites,

$$\mathcal{C} = \{C_1, ..., C_{|\mathcal{N}|}\},$$

where C_n represents computational capacity of a site $n \in \mathcal{N}$. High-capacity network resources at the edge are not feasible because deployment and operational costs are high due to the high number of instances at the edge; thus, centralized sites have more capacity than peripheral sites.

$\mathcal{L}$ is the set of links that connect any pair of sites. l_{n_1,n_2} represents a link that connects sites n_1 and n_2,

$$l_{n_1,n_2}, l \in \mathcal{L}_t, n_1 \in \mathcal{N}_t, n_2 \in \mathcal{N}_{t+1}.$$

$\mathcal{Q}$ and $\mathcal{E}$ are the set of capacities and average latencies of links in $\mathcal{L}$ respectively. q_l and e_l denote the capacity and latency of link $l \in \mathcal{L}$.

Table 3.1: List of parameters used in the formulation of FlexNSP.

Parameter	**Explanation**	**Unit**
T	The set of tiers in the multi-tier network	
$\mathcal{N}$	The set of all sites	
$\mathcal{N}_t$	The set of sites in tier t	
$\mathcal{N}_c$	The set of cell sites	
C_n	The capacity of a site $n \in \mathcal{N}$	GOPS
$\mathcal{L}$	The set of all links	
$l_{(n_1,n_2)} \in \mathcal{L}$	The link that connects site $n_1, n_2 \in \mathcal{N}$	
e_l	Link latency on link $l \in \mathcal{L}$	ms
q_l	Link capacity on link l	MHz
$\mathcal{P}$	The set of all paths consist of a subset of links in $\mathcal{L}$	
$p_{(n_1,n_2)} \in \mathcal{P}$	Denoted the path that connects n_1 and n_2	
$\mathcal{S}$	The set of RAN slices.	
R_s^e	Denotes the end-to-end latency requirement of a slice $s \in \mathcal{S}$.	ms
R_s^{CS}	Denotes the set of requested cell sites of a slice $s \in \mathcal{S}$.	
$A_{s,j}^{HLS/LLS}$	Indicated whether HLS/LLS option j is selected for slice s.	
$J^{HLS/LLS}$	The set of all HLS/LLS options	
$\mathcal{D}$	The set of server capacity demands depending on the function split options	
D_s^{DU}	Denotes the cost of a DU function of slice s.	GOPS
$H_s^{mid/front}$	Denotes the midhaul/fronthaul requirement of slice s	
$m_{s,n}^f$	The binary variable to indicate if VNF $f \in F_s$ (O-RU,O-DU or O-CU) is placed on server n or not.	
x_s	The binary variable to indicate if slice s is admitted or not.	
$k_s^{r,d}$	The binary variable to indicate if $RU_{s,r}$ is connected to $DU_{s,d}$.	
$y_{s,p}^{mid,d}$	The binary variable to indicate if path p is connecting DU $d \in F_s^{DU}$ and CU_s.	
$y_{s,p}^{front,r}$	The binary variable to indicate if path p is selected for RU $r \in F_s^{RU}$ to connect it to a DU of slice s	
F_s^f	The set of network functions $f \in$ {O-RU, O-DU, O-CU} of a slice s	
ϵ_s	Centralization factor of the slice s	
v_s	Potential financial value when the slice is admitted	

$\mathcal{P}$ is the set of all paths in the network between two tier sites. $p_{n_1,n_2} \in \mathcal{P}$ represents a path between servers n_1 and n_2 and consists of a set of links in $\mathcal{L}$.

For reference, all parameters and variables used can be seen in Table 3.1

3.1.1 Computational Cost Calculation

When placing network slices to the network infrastructure, the computational cost of O-RU must be satisfied by the cell site that it is placed. On top of that, the tier sites that host O-DU and O-CU of a slice must have sufficient resources to satisfy the computational costs of O-DU and O-CU.

To calculate the computational requirements of O-RU, O-DU, and O-CU of a network slice according to each functional split in terms of Giga operations per second (GOPS), a computational complexity estimation model is derived from [45],

$$D_s^F = \alpha_{s,F}(3A + A^2 + \frac{MCL}{3})\frac{\Gamma}{5},$$

where F denotes the network slice function O-RU, O-DU or O-CU that we are calculating the computational requirements, A denotes the number of antennas, Γ denotes the number of resource block required by the network slice, M denotes modulation bits, C denotes coding rates, L denotes the number of MIMO layers of each radio unit. $\alpha_{s,F}$ is a scaling factor of the network slice function F depending on the selected functional split option of slice s. In other words, α is the ratio of functionalities in F to all functionalities in the network slice.

When the scaling factor α of the model equals one, the model calculates the total computational requirement of the network slice. Scaling factor α distributes the computational cost to the network slice functions proportional to the baseband functionality (RF, PHY, MAC, RLC, and PCDP) processed in the network slice function. In higher functional split options like option 2, the scaling factor of O-CU is close to one, whereas lower layer split options like option 8 have a lower scaling factor value of O-CU and a higher scaling factor value of O-RU.

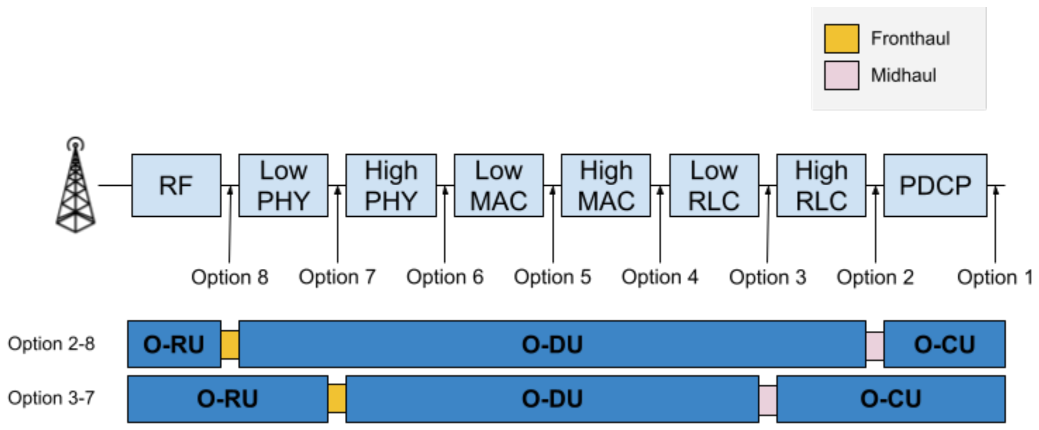

Figure 3.2: Functional split options proposed by 3GPP [2] and two example mapping of dual functional split options to O-RU/O-DU/O-CU.

3.1.2 Formulation of FlexNSP

As mentioned, FlexNSP is formulated as a Mixed Integer Quadratically Constraint Program (MIQCP). As we will describe the formulation, functional split option selection and routing decisions of each network slice request makes this problem's constraints too complex for it to be formulated as a Mixed Integer Program (MIP).

A RAN slice can be seen as a set of virtual RAN functions (O-RU, O-DU, O-CU) that together constitutes a complete radio access network. $\mathcal{S}$ is the set of RAN slice requests that can be placed in the network,

$$\mathcal{R} = \{R_1, ..., R_{|\mathcal{S}|}\}.$$

$\mathcal{R}$ is the set of RAN slice requirements that need to be satisfied if the slice is placed. R_s can be expanded as different types of requirements of a RAN slice. R_s^e denotes the latency requirement of RAN slice s. R_s^{mid} and R_s^{front} corresponds to the midhaul and fronthaul capacity requirements of slice s respectively,

$$R_s = \{R_s^e, R_s^{front}, R_s^{mid}\}, s \in \mathcal{S}.$$

The distribution of RAN functions between O-RU, O-DU and O-CU is vital to meet slices' requirements. In Figure 3.2, different function split options proposed by 3GPP [2] and some example mappings of baseband functionalities to O-RU/O-DU/O-CU can be seen. It is important to note that not all mapping possibilities (Options 2-8, 2-7, 2-6, 3-8, 3-7, 3-6 for our problem) are represented in Figure 3.2 to avoid cluttering.

Baseband functions on the left of the LLS option are assigned to O-RU, and functions on the right of the HLS option are assigned to O-CU. The functions between the LLS option and the HLS option are assigned to O-DU.

The placement of these functions on the mentioned multi-tier aggregation 5G O-RAN is a crucial aspect of network slicing, as placing functions closer to the edge provides benefits like low latency required by some specific types of slices. However, as functions are moved to the edge, some techniques that require centralization of the functions, such as Coordinated Multipoint (CoMP), cannot be utilized in the network [24].

Our objective is to maximize the profit of InP. Each slice has a financial value v_s that represents the financial benefit gained by admitting slice $s \in \mathcal{S}$. Our problem's objective is to maximize the total financial value while satisfying the requirements of admitted slices.

Considering the functional splits in Figure 3.2, we considered functional split options 2 and 3 as high layer split options; functional split options 6, 7 and 8 as low layer split options,

$$J^{HLS} = \{j_2, j_3\},$$

$$J^{LLS} = \{j_6, j_7, j_8\}.$$

J^{HLS} is the set of higher layer functional split options. J^{LLS} is the set of lower layer functional split options.

It is important to note that functional split option 8 is the split used in the C-RAN architecture. In option 8, all baseband processing is centralized and fronthaul requirements are very demanding. The reason for excluding options 1, 4, 5 is that these functional split options are not feasible to select in any scenario, bandwidth and latency requirements of these functional splits can be too demanding for the benefits they present [2].

$\mathcal{A}$ is the set of functional split variables,

$$\mathcal{A} = \{A_1, ..., A_{|\mathcal{S}|}\}.$$

A_s^{HLS} and A_s^{LLS} are the functional split option variables for slice $s \in \mathcal{S}$,

$$A_s^{HLS} = \{A_{s,j}^{HLS} | j \in J^{HLS}\},$$

$$A_s^{LLS} = \{A_{s,j}^{LLS} | j \in J^{LLS}\},$$

where $A_s^{HLS^T}$ and $A_s^{LLS^T}$ denote the transposes of A_s^{HLS}, A_s^{LLS} respectively. For a RAN slice s, the selected HLS and LLS options dictate latency and midhaul requirements.

The computational cost of O-RU, O-DU, and O-CU depends on which HLS and LLS options are used. As mentioned in subsection 3.1.1, $\mathcal{D}$ denotes the set of processing demands of O-RU, O-DU, and O-CU functions according to different functional splits. $\mathcal{D}_s^F$ is the computational cost of $F \in$ {O-RU, O-DU, O-CU} of slice s.

$\mathcal{H}^{mid}$ and $\mathcal{H}^{front}$ denote the set of midhaul and fronthaul capacity requirements according to different functional splits respectively,

$$\mathcal{H}^{mid} = \{H_j^{mid} | j \in J^{HLS}\},$$

$$\mathcal{H}^{front} = \{H_j^{front} | j \in J^{LLS}\}.$$

R_s^{mid} and R_s^{front} denote the midhaul and fronthaul capacity requirements of slice s,

$$R_s^{mid} = \mathcal{H}^{mid} A_s^{HLS^T},$$

$$R_s^{front} = \mathcal{H}^{front} A_s^{LLS^T}.$$

$\mathcal{G}^{mid}$ and $\mathcal{G}^{front}$ denote the sets of transport latency requirements, meaning maximum allowed latency of midhaul and fronthaul paths,

$$\mathcal{G}^{mid} = \{G_j^{mid} | j \in J^{HLS}\},$$

$$\mathcal{G}^{front} = \{G_j^{front} | j \in J^{LLS}\}.$$

G_s^{mid} and G_s^{front} denote the HLS and LLS latency requirement of slice s,

$$G_s^{mid} = \mathcal{G}^{mid} A_s^{HLS^T}, s \in \mathcal{S},$$

$$G_s^{front} = \mathcal{G}^{front} A_s^{LLS^T}, s \in \mathcal{S}.$$

In the light of these definitions, the mathematical formulation of FlexNSP is as follows:

$$\max_{x} \quad \sum_{s\in\mathcal{S}} x_s v_s \epsilon_s \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{j\in J^{HLS}} A^{HLS}_{s,j} = x_s, s \in \mathcal{S} \tag{3.1b}$$

$$\sum_{j\in J^{LLS}} A^{LLS}_{s,j} = x_s, s \in \mathcal{S} \tag{3.1c}$$

$$\sum_{n\in N} m^{CU}_{s,n} = x_s, s \in \mathcal{S} \tag{3.1d}$$

$$\sum_{n\in N} m^{DU,d}_{s,n} = x_s, s \in \mathcal{S}, d \in F^{DU}_s \tag{3.1e}$$

$$\sum_{p\in\mathcal{P}} y^{front,r}_{s,p} = x_s, s \in \mathcal{S}, r \in R^{CS}_s \tag{3.1f}$$

$$\sum_{p\in\mathcal{P}} y^{mid,d}_{s,p} = x_s, s \in \mathcal{S}, d \in F^{DU}_s \tag{3.1g}$$

$$y^{front,r}_{s,p_{n_1,n_2}} \leq m^{RU,r}_{s,n_1}, s \in \mathcal{S}, p \in \mathcal{P}, r \in R^{CS}_s \tag{3.1h}$$

$$y^{front,r}_{s,p_{n_1,n_2}} \leq m^{DU,r}_{s,n_2}, s \in \mathcal{S}, p \in \mathcal{P}, r \in R^{CS}_s \tag{3.1i}$$

$$y^{mid,d}_{s,p_{n_1,n_2}} \leq m^{DU,d}_{s,n_1}, s \in \mathcal{S}, p \in \mathcal{P}, d \in F^{DU}_s \tag{3.1j}$$

$$y^{mid,d}_{s,p_{n_1,n_2}} \leq m^{CU}_{s,n_2}, s \in \mathcal{S}, p \in \mathcal{P}, d \in F^{DU}_s \tag{3.1k}$$

$$\sum_{s\in\mathcal{S}} \sum_{d\in F^{DU}_s} m^{CU}_{s,n} D^{CU}_s + m^{DU,d}_{s,n} D^{DU}_s$$

$$\leq C_n, n \in \mathcal{N} \tag{3.1l}$$

$$\sum_{p\ni l, p\in\mathcal{P}} \sum_{s\in\mathcal{S}} y^{mid,r}_{s,p} R^{mid}_s + y^{front,r}_{s,p} R^{front}_s$$

$$\leq q_l, l \in \mathcal{L} \tag{3.1m}$$

$$\sum_{p\in P} \sum_{l\in p} y^{mid}_{s,p} e_l \leq G^{mid}_s, s \in \mathcal{S} \tag{3.1n}$$

$$\sum_{p\in P} \sum_{l\in p} y^{front}_{s,p} e_l \leq G^{front}_s, s \in \mathcal{S} \tag{3.1o}$$

$$\sum_{p\in P} \sum_{l\in p} y^{front}_{s,p} e_l + y^{mid}_{s,p} e_l \leq R^{e}_s, s \in \mathcal{S} \tag{3.1p}$$

The objective expressed in (3.1a) indicates the maximization of the total value of admitted RAN slices. Centralization factor ϵ_s is a multiplier in the equation, increasing the slice value as more central function split options are selected. We want to locate our functions as central as possible because the centralization of a function enables centralization benefits such as Coordinated Multipoint (CoMP) and low operational expenditures (OpEx). Each RAN slice has a centralization factor ϵ_s correlated to the selected functional split of slice s to reflect the centralization gains to our objective. Lower functional split options have a lower centralization factor; higher functional split options have a high centralization factor to reflect that centralization of a slice increases the value of the slice by increasing efficiency and decreasing the cost of managing the slice.

x_s is a decision variable indicating whether slice s is admitted or not. $m_{s,n}^{CU}$ is a decision variable indicating if CU function of slice s is placed on site n. Similarly, $m_{s,n}^{DU,d}$ indicates if DU function d of slice s is placed on site n. Since a RAN slice consists of a single CU function and multiple DU functions, each DU function has a separate decision variable. However, $m_{s,n}^{RU,r}$ is not a decision variable because placements of RU functions depend on the desired location, which is given information with the slice request. In our problem, a functional split option is selected for each slice to accommodate the slice's requirements. Constraints (3.1b), (3.1c) ensure that each slice selects one HLS option and one LLS option if the slice is admitted. Functions of a slice should be placed in the network if and only if the slice is admitted. Constraint (3.1d) ensures CU function of slice s is placed if slice s is admitted. Constraints (3.1e) ensures a DU function of slice s is placed for each RU if slice s is admitted.

Note that the number of DU functions of slice s, $|F_s^{DU}|$, is equal to the number of RU functions of the same slice s, $|R_s^{CS}|$, because each RU function has to connect a DU function.

After placing the functions to tier sites, the connectivity between them must be ensured. From all possible paths in the network, appropriate ones should be assigned to connect slice functions. Constraints (3.1f) and (3.1g) ensure that the number of fronthaul and midhaul paths selected is equal to the number of RU_s and DU_s functions placed. A path p_{n_1,n_2} connects a lower tier server n_1 to a higher tier server n_2,

$$t_1 < t_2, t_1 \in T, t_2 \in T, n_1 \in \mathcal{N}_{t_1}, n_2 \in \mathcal{N}_{t_2}, p_{n_1,n_2} \in \mathcal{P}.$$

F_s is the set of functions that composes the slice s. $F_s^{RU} \in F_s$ is the set of RU functions in the slice s. Constraints (3.1h) and (3.1i) ensure that a path p_{n_1,n_2} can be selected as a fronthaul path of O-RU function $r \in R_s^{CS}$ if and only if O-RU function is placed at cell site n_1 and O-DU function is placed at tier site n_2. Constraints (3.1j) and (3.1k) ties the selected midhaul paths to the placement of DU functions and the CU function of slice s.

The capacity of physical network resources, tier sites, and links should satisfy the requirements of all placed slices and slice functions. Constraint (3.1l) ensures that total resource demand in a site $n \in \mathcal{N}$ cannot exceed the total capacity of the site C_n. Constraint (3.1m) ensures that the the capacity of the link l, q_l, is not exceeded.

Constraints (3.1n), (3.1o) ensure that selected midhaul and fronthaul path p for a slice s can satisfy the midhaul and fronthaul functional split latency requirement of slice s, G_s^{mid}, G_s^{front} respectively. Constraint (3.1p) dictates that the total one-way latency between the user and slice functions should be lower than the slice latency requirement.

CHAPTER 4

FLEXIBLE BEST FIT DECREASING PLACEMENT

Although mathematical optimization yields optimal results for a given network state, scalability is an issue because optimizing large network infrastructures results in too many variables, taking an unreasonable amount of time to find an optimal solution. We proposed a heuristic approach, Flexible Best Fit Decreasing Placement (FlexBFDP) to find placements with reasonable objective value in small amounts of time to overcome the scalability problem.

The intuition of the heuristic approach comes from the bin packing problem. The bin packing problem is fitting a set of items with weights into the minimum number of bins. We adopted the best fitting decreasing approach to our problem.

The heuristic algorithm is divided into three parts:

- Place-DU procedure described in Algorithm 1 tries to deploy O-RU and O-DU functions of a network slice request s. The procedure checks if O-RU functions can be deployed to corresponding cell sites, determines fronthaul paths and feasible tier sites for O-DU functions.

- Place-CU procedure described in Algorithm 2 tries to deploy the O-CU function of the network slice request s according to the results of Place-DU procedure 1. This procedure

- The main procedure of FlexBFDP described in Algorithm 3 uses Algorithm 1 and Algorithm 2 to select which network slice requests are admitted to the network infrastructure.

Algorithm 1 Placement of O-DU

Input: Network slice request s, HLS option a^{HLS}, LLS option a^{LLS}, fronthaul paths P_f, cell sites N_c, tier sites N_t

Output: Selected O-DU sites $N_{DU}^{selected}$, fronthaul paths $P_f^{selected}$

procedure PLACE-DU
  $P_f^{selected} \leftarrow \{\}$ ▷ Selected fronthaul paths
  $N_{DU}^{selected} \leftarrow \{\}$ ▷ Selected sites to deploy O-DU functions
  $N_{RU}^{s} \leftarrow$ The set of cell sites to host O-RU functions of network slice request s
  for each O-RU r of network slice request s **do**
    Check if O-RU requirements are satisfied by cell sites N_{RU}^{s}.
    $C_f \leftarrow \{\}$ ▷ Candidate fronthaul paths
    $C_{DU} \leftarrow \{\}$ ▷ Candidate O-DU sites
    for each fronthaul path $p \in P_f$ **do**
      ▷ p_{n_1,n_2} connects cell site $n_1 \in \mathcal{N}_{\rfloor}$ and tier site $n_2 \in \mathcal{N}_{\sqcup}$
      $(n_1, n_2) \leftarrow p$ ▷ Sites connected with p
      if There is no O-RU deployed at n_1 **then** ▷ Current fronthaul path p is not connected to a O-RU.
        Continue with next fronthaul path.
      end if
      if p can not satisfy the bandwidth and latency requirements of network slice s **then**
        Continue with next fronthaul path.
      end if
      if n_{du} can not satisfy O-DU computational requirements **then**
        Continue with next fronthaul path.
      end if
      Add p to $C^{fronthaul}$.
    end for
    Stable sort C^f by remaining bandwidth capacity
    Stable sort C^f by remaining computational capacity of n_{du}
    Selected fronthaul path for O-RU r, $p_f^r \leftarrow$ first element of C_f
    Update the bandwidth capacity of p_f^r
    Add $p_{n_1,n_2}^{f,r}$ to selected fronthaul paths $P_f^{selected}$.
    Add n_2 to $N_{DU}^{selected}$.
  end for
  return $N_{DU}^{selected}$, $P_f^{selected}$
end procedure

4.1 Heuristic Main Procedure

First, network slice requests are sorted by value in decreasing order. Then, network paths are classified as candidate fronthaul paths and candidate midhaul paths depending on the starting and ending site of the path. After sorting candidate paths by path latency, we loop through each network slice request s to admit the slice or not. For each network slice request s, combinations of high-layer split and low-layer split options are iterated starting from the most central option pairs, meaning we prioritize placing baseband network functions into O-CU, then O-DU, then O-RU. The Place-DU procedure returns a set of tier sites that can host O-DU functions of s and the set of selected fronthaul paths connected to these O-DU functions on one end. After determining potential O-DU tier sites and fronthaul paths, the Place-DU procedure tries to place the O-CU function according to these O-DU locations and fronthaul paths and determines which midhaul paths will be used. If there is a feasible tier site to host O-CU and midhaul paths to connect O-DUs to this O-CU, then the algorithm admits the network slice request s to the network infrastructure, updates the remaining computational capacity of used sites and links in the used fronthaul and midhaul paths.

4.2 Place-DU Procedure

Given a network slice request s, HLS and LLS options a^{HLS}, a^{LLS}, fronthaul paths P_f, cell sites N_c, and tier sites N_t, this procedure returns the selected sites for O-DU functions of s and selected fronthaul paths $N_{DU}^{selected}$, $P_f^{selected}$. Each network slice request has a set of cell site request that must be satisfied. The reason the set of cell sites is fixed is that we assumed that a network slice is serving a specific physical area that can be served with a limited number of cell sites. If cell sites can not satisfy the computational requirements of O-RUs, we will continue with the following network slice request. After placing O-RUs, fronthaul paths are iterated to select the feasible fronthaul paths for the current network slice request.

Algorithm 2 Placement of O-CU

Input: Network slice request s, HLS option a^{HLS}, LLS option a^{LLS}, selected tier sites for O-DU to deploy $N_{DU}^{selected}$, selected fronthaul paths $P_f^{selected}$

Output: Updating network slice request s as admitted or not, setting functional split options if admitted.

```
1:  procedure PLACE-CU
2:      Sort tier sites N_t by remaining computational capacity
3:      for each site n_CU^candidate ∈ N_t do
4:          if n can not satisfy O-CU computational requirements then
5:              Continue with next tier site.
6:          end if
7:          P_m^selected ← {}                                        ▷ Selected midhaul paths
8:          for each O-DU hosting tier site n_DU^selected ∈ N_DU^selected do   ▷ All sites in N_DU^selected must be connected to the selected O-CU site
9:              Filter midhaul paths connecting n_DU^selected and n_CU^candidate.
10:             Sort filtered midhaul paths by remaining bandwidth capacity.
11:             p_m^selected ← The first midhaul path in sorted filtered midhaul paths satisfying link requirement according to split options.
12:             if There is no p_f^selected path available then
13:                 ▷ The algorithm will try a new tier site for O-CU, selected midhaul paths will be invalidated.
14:                 Unselect all midhaul paths, update their link capacity usage accordingly.
15:                 Continue with next n_CU^candidate ∈ N_t (break for loop, continue from line 3)
16:             end if
17:             Update link capacity of p_f^selected
18:             Add p_f^selected to P_f^selected
19:             n_CU^selected ← n_CU^candidate                        ▷ Current n_CU can host the O-CU function
20:             Update computational capacity of n_CU^selected
21:             Update network slice request s as admitted with functional split options a^HLS, a^LLS.
22:         end for
23:     end for
24: end procedure
```

4.3 Place-CU Procedure

Given a network slice request s, an HLS option, an LLS option, and the outputs of Place-DU procedure as input to this procedure, the output of this procedure is the set of selected network slice requests $S^{selected}$ and their selected functional split options to calculate centralization factor and objective value. For each tier site that can satisfy the computational requirements of the O-CU function of network slice request s, we find candidate midhaul paths connecting selected O-DU sites $N_{DU}^{selected}$. Then, the path with the least remaining bandwidth capacity connecting O-DU and O-CU functions and satisfying the midhaul path requirements of s is selected and added to the set of selected midhaul paths $P_f^{selected}$. If no such path is found between an O-DU site and the O-CU candidate site, all selected midhaul paths are cleared, and we will try another tier site as a candidate O-CU site. After selecting midhaul paths and O-CU site, the algorithm concludes that network slice request s can be placed and proceeds to update related tier sites, links. Finally, the procedure returns if the given network slice request s is admitted or not to the network infrastructure.

Algorithm 3 FlexBFDP Algorithm

```
procedure FLEXBFDPLACEMENT(The set of Network Slice Requests S, HLS-LLS option pairs
    FS, Fronthaul paths P_f, Midhaul paths P_m, cell sites N_c, tier sites N_t)
    Sort network slice requests S by value in descending order.
    Sort fronthaul paths P_f by latency in descending order.
    Sort midhaul paths P_m by latency in descending order.
    for each network slice request s ∈ S do
        for each a^{HLS}, a^{LLS} pair ∈ FS do
            N_DU^{selected}, P_f^{selected} ← PLACE-DU(s, a^{HLS}, a^{LLS}, P_f)
            s^{selected} ← PLACE-CU(s, a^{HLS}, a^{LLS}, N_DU^{selected}, P_f^{selected})
        end for
    end for
    Calculate financial gain as objective value from the set of selected network slice requests
    S^{selected} and their selected functional split options.
    return Total objective value from network slice requests S.
end procedure
```

4.4 Search space of FlexNSP and time complexity of FlexBFDP

In this section, the search space of FlexNSP will be discussed, and we will derive the time complexity of FlexBFDP. As we can see in Table 3.1, $\mathcal{S}$ is the set of RAN slices. There are $|\mathcal{S}| \times (|N| + |N| \times |R| + |N|^2 + |N|^2 + 2 + 3)$ variables where R is the set of RUs in a network slice, and the terms in the parenthesis represents CU placement, DU placement, fronthaul paths, midhaul paths, high layer split, and low layer split respectively. Moreover, most of these variables are independent of each other, and our constraints are mostly quadratic, causing a vast search space for mathematical optimization. This is why the computation time of the Gurobi solver increases exponentially when the number of network slice requests $|\mathcal{S}|$ increases.

The time complexity of FlexNSP is $|\mathcal{S}| \times (|R||N|^2 + |D||N|^3)$, which is $O(|D||N|^3)$. The first part in the parentheses comes from the Place-CU algorithm and the second part comes from the Place-DU algorithm. Even though the worst-case time complexity is cubic, filtering unfeasible paths and tier sites during execution makes the average case much more efficient.

4.5 The logic behind FlexBFDP

The main structure of the FlexBFDP algorithm is as follows: First, network slice requests are sorted by their values gained if the request is admitted to the network infrastructure. The reasoning behind this is that we want to prioritize high-value network slice requests to achieve maximum total gain from our limited resources in the network infrastructure. Then, fronthaul and midhaul paths are sorted by latency in descending order. In this context, all possible paths starting from edge sites are fronthaul paths, and all possible paths connecting different tier sites are midhaul paths. The descending order of latency ensures that we use the maximum allowed latency path to allow more suitable choices after processing the current network slice request. For example, more latency tolerant network slices will not use a site close to the edge instead of a more central site. This ordering decision also benefits the centralization of the network slice as much as possible because the paths connecting more central sites have more latency, as we expected.

After ordering network slice requests and paths, the algorithm iterates over each network slice request and tries to admit it to the network infrastructure with the best functional split options possible. By saying the best functional split options, it is from the perspective of the infrastructure provider, trying to admit the slice with the least amount of resources available to maximize the total financial gain from the set of network slice requests.

The algorithm places the virtual network functions of the network slice using the procedures we described before; the Place-DU procedure places the O-DU function, and the Place-CU places the O-CU function of the network slice.

After iterating over all network slice requests, total financial gain is computed according to selected requests and their selected functional split options.

CHAPTER 5

RESULTS AND DISCUSSIONS

In this chapter, we discussed the solutions to our Mixed-Integer Quadratically Constrained Programming (MIQCP) problem. We solved the MIQCP problem defined in equations (3.1a), ..., (3.1p), with different optimization parameters, using Gurobi solver, which is a commercial mathematical optimization solver [59]. For conducting the experiments, we used a computer with Intel® Core™ i5-8250U Processor and 12 GB RAM. After discussing Gurobi solutions, we discussed the performance of our heuristic solution FlexBFDP and compared it with a state-of-the-art heuristic solution [20] and Gurobi solutions. Different functional split requirements derived from [1] can be seen in Table 5.1.

Table 5.1: Functional split requirements of different split options. [1]

Functional Split Option	**Bandwidth Requirement**	**Latency Requirement**	**Centralization Factor**
Option 2	4.016 Gb/s	10 ms	1
Option 3	4 Gb/s	6 ms	1.1
Option 6	4 Gb/s	8 ms	1.1
Option 7	38 Gb/s	5 ms	1.2
Option 8	157.3 Gb/s	3 ms	1.3

5.1 Experimental Settings

In this part, the experiment variables and assumptions are discussed. Each experiment is conducted 30 times to have a reasonable error bar. The error bars in all our results represent the 95% confidence interval.

Table 5.2: Radio parameters.

Parameter	**Value**
Number of Antennas	4
Number of MIMO layers	2
Modulation	64-QAM
Coding	5/6

The experiment topology has a similar structure to Fig. 3.1. Processing requirements of O-RU, O-DU, and O-CU are calculated using the model mentioned in subsection 3.1.1, with the radio parameters listed in Table 5.2. In each experiment, experiment values are set between mentioned ranges randomly. In tier 3, there is one site with [1600, 2000] GOPS capacity available for VNFs. In tier 2, there are five sites, each containing [800, 1000] GOPS capacity available. In tier 1, 25 sites with [400, 500] GOPS capacity are located. The capacity of each link between tier 2 sites and tier 3 sites is [500, 1500] Gbps, between tier 1 sites and tier 2 sites is [200, 300] Gbps.

Table 5.3: Network slice parameters.

Slice Type	**Number of O-RUs**	**Number of RBs per O-RU**	**Maximum latency tolerance**	**Admittance value**
eMBB	[2, 5]	50	[4, 10]	[3, 5]
uRLLC	[1, 10]	25	[1, 2]	[5, 7]
mMTC	[6, 10]	5	[10, 30]	[2, 4]

We consider three types of network slices with different requirements which can be seen in Table 5.3

5.2 Results

In this section, results of our experiments are discussed.

5.2.1 Impact of MIPGap

First, we conduct experiments with different MIPGap values to select which MIPGap value is suitable for our following experiments. In our results, mathematical solutions with different MIPGap values are presented. MIPGap is the Mixed Integer Program (MIP) gap, representing the gap between the upper and lower objective bound. This MIPGap concept is used in mathematical solvers like Gurobi because finding exactly optimal solutions for complex problems like FlexNSP could take years, or there may not be a provable optimal solution for the solver to find. When we use lower MIPGap values, the solution we found is closer to the optimal solution. However, using too low MIPGap values increases the time consumption of the optimization process to unreasonable levels, whereas using too high MIPGap values generates poor sub-optimal results.

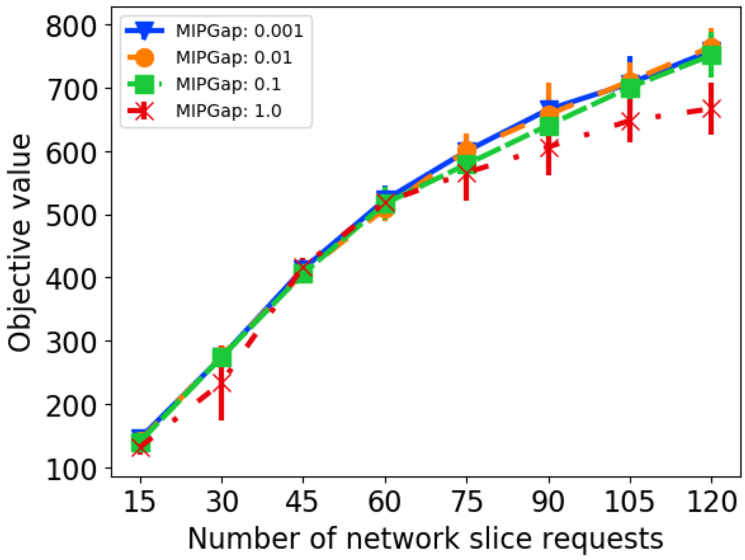

Figure 5.1: Performances of flexible functional split for different MIPGap values.

Fig. 5.1 illustrates the effects of different MIPGap values on the quality of the solution Gurobi finds. As expected, better results are achieved with smaller MIPGap values. It is important to mention that it took ~ 50000 seconds (more than 13 hours) to find the solution at 0.001 MIPGap value when there are 120 RAN slice requests, although the achieved objective value is not significantly higher than 0.01 MIPGap at any point of the experiments, which took ~ 900 seconds to compute. For our other experiments, we chose the 0.01 MIPGap value because it is a good compromise between the quality of the solution and the time consumption of optimization.

For high number of network slice requests like 200, Gurobi solver can not find a solution with lower MIPGap values like 0.001 and 0.01. Even though we waited for days for it to compute, the gap value got stuck at some thresholds and does not progress the solution further. This is the main reason we come up with a heuristic solution to be able to scale our solution to larger networks, which we will discuss the results later.

5.2.2 Impact of the Flexible Functional Split Options

In Fig. 5.2 and 5.3, we compared flexible functional split per RAN slice with fixed functional splits. The objective value represents the value of our problem's objective function that can be seen in (3.1a).

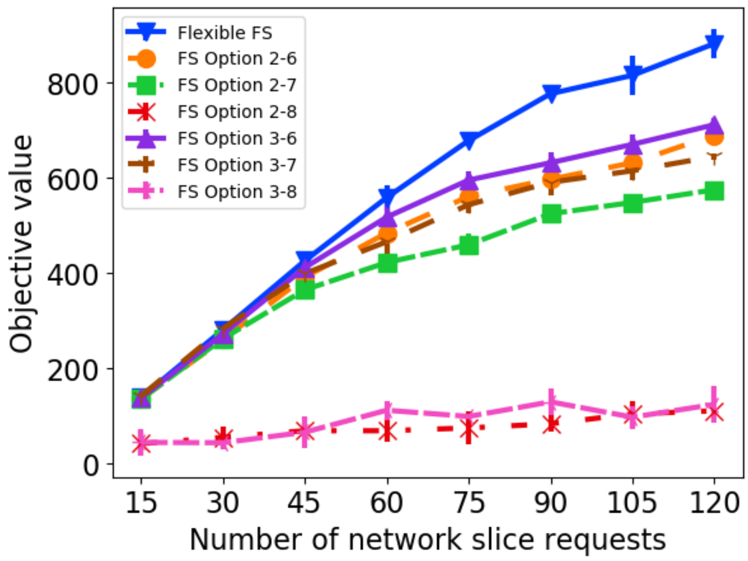

Figure 5.2: Determined optimization objective performances of different functional split options.

The first experiment can be seen in Fig. 5.2 compares the achieved objective value when different number of network slice requests placed in the same physical network infrastructure. For low number of network slice requests, we can see two groups of similar results. When number of network slice requests equals to 15 and 30; flexible functional split, option 2-6, option 2-7, option 3-6, and option 3-7 performs similar to each other as they achieve the objective value ~ 150 and ~ 260 at 15 and 30 network slice requests respectively. On the other hand, options 2-8 and 3-8 are performing significantly poorly compared to other options. Although low layer split option 8 provides the most centralization, only RF functionality is provided by O-RU, bandwidth requirement of low layer split option 8 is much higher than other low layer split options, and requires the lowest latency among all split options. Its important to note that C-RAN utilizes option 8, which is the reason C-RAN architecture demands very high bandwidth capacity in the fronthaul network.

When the number of network slice requests greater than or equal to 45, flexible functional split starts to outperform other options. As the number of network slice requests increases, the gap between flexible functional split and its closest competitor option 3-6 keeps increasing significantly. When the number of network slice requests reach to 120, flexible functional split outperforms option 3-6 by 23%. From these results, we can see that flexible functional splitting achieves the highest objective value compared to other function split options because of the flexibility provided by selecting unique function split options per network slice request. Diversifying the selection of functional split increases the possible placement options and enables achieving higher objective values.

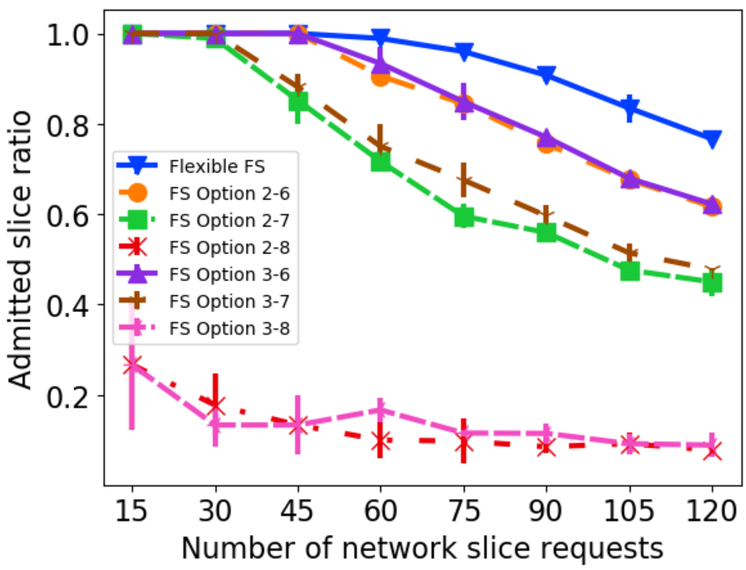

Figure 5.3: The ratio of admitted RAN slice requests to all RAN slice requests.

After analyzing objective value, the ratio of admitted network slice requests to all network slice requests can be seen in Fig. 5.3. Since our physical network resources are fixed in all cases, increasing the number of candidate network slices lowers the admission ratio for all functional split options.

When the number of network slice requests 15 and 30, we see two groups of results similar to our previous experiment. Flexible functional split, options 2-6, 2-7, 3-6 and 3-7 can admit all network slice requests to the network infrastructure. However, option 2-8 and 3-8 can admit $\sim 30\%$ of the requests for 15 total requests and $\sim 20\%$ for total 30 requests. We are seeing the high bandwidth and latency requirements of option 8 causes options 2-8 and 3-8 to admit much less requests than other options. When the number of network slice requests reaches to 45, flexible functional split, option 2-6, and option 3-6 still admits all of the requests where option 2-7 and 3-7 can only admit $\sim 90\%$ of the requests. After exceeding the 45 limit for number of requests, flexible functional split clearly outperforms other options and option 3-6 and option 2-6 performs similarly behind flexible functional split. Finally, flexible functional split admits 30% more requests compared to option 3-6 and 2-6. Overall, flexible functional split has the highest admission ratio because flexible functional split can satisfy the diverse RAN slice request requirements. Although it is beneficial to have a high admission ratio, we do not try to achieve the highest admission ratio since each request has its own value; that is, admitting a highly valued request over low valued multiple requests can be beneficial for maximum profit. Even though high admission rate is not our first priority, flexible functional split admits the most number of requests compared to other fixed functional split options.

5.2.3 Impact of Network Slice Types

In all the experiments prior to this point, the network slice requests have mixed network slice types. Each type of network slice has its own requirements and financial value when admitted. We conducted two experiments comparing different network slice types with Gurobi solver to see the effects of different network slice types. In Fig. 5.4 and 5.5, effects of different network slice types are compared. In these two experiments, network slice types are evenly distributed in the mixed case, while all

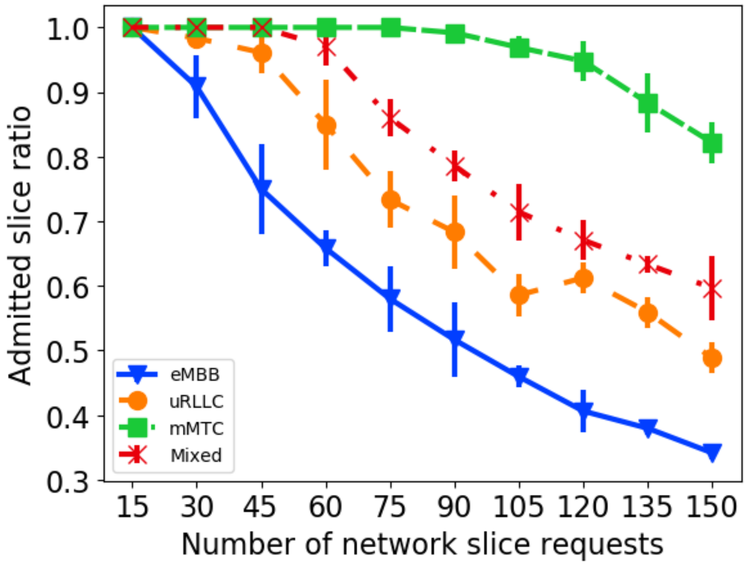

Figure 5.4: Admission rate of network slices of different network slice types

network slices have the same type in the other cases.

Fig. 5.4 indicates that when all network slices are mMTC slices, we achieve the highest admission rate despite requiring the highest number of O-RUs because each O-RU requires significantly fewer resource blocks compared to other types of network slices. Requiring fewer number of resource blocks allows edge sites to host more O-RU function of different network slices, increasing the admission ratio. While mMTC has the highest admission rate, eMBB slices have the lowest admission rate because of the higher resource block requirements than other types. Mixed scenario and uRLLC scenario have admission rates less than mMTC and more than eMBB. It is expected to have an average admission rate when we looked at mixed scenario because mMTC slices increases the ratio while eMBB slices decreases the ratio, balancing each other to an average value. When the number of network slice requests reaches 150, mMTC admission rate is 33% more than mixed scenario, while mixed scenario has 24% more admission rate compared to uRLLC slices. At last, the admission rate of uRLLC slices is 44% higher than eMBB slices.

When we move on to the experiments that measures objective value with different number of network slice requests represented in Fig. 5.5, a different order among slice types from admission rate experiment can be observed. Although mMTC had the

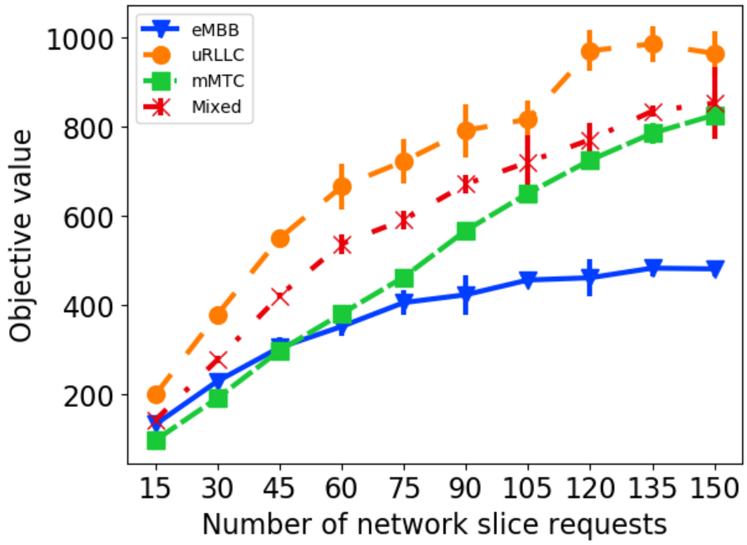

Figure 5.5: Determined optimization objective performances of different network slice types

highest admission rate among all slice types in the previous experiments, that does not translate directly to the objective value experiments as uRLLC and mixed scenarios achieves higher objective values in all cases. The reason behind uRLLC slices has the highest objective value at all numbers of network slice requests is uRLCC slices have higher financial value compared to other types of network slices, which can be seen in Table 5.3. uRLLC slices generally are mission-critical slices and high priority slices that cannot be afforded not to be admitted, therefore their admittance value is higher than other slice types. mMTC slices have low requirements but each individual slice has lower value compared to uRLLC and eMBB slices because generally low priority use cases use mMTC like collecting sensor data, low processing smart devices use these slices to send statistics about their status. eMBB still has the lowest objective value even each slice has higher value than a mMTC slice because value is not high enough to compensate such low admittance rates.

5.2.4 Comparison with FlexBFDP and SOTA

The state-of-the-art (SOTA) implementation is trying to place the functions of a network slice request in the same site. The reasoning behind this is avoiding the latency and bandwidth requirements between different network functions. If it cannot place a network slice request to the same site, then the algorithm tries to find a feasible functional split between O-RU and O-DU functions. SOTA considers a single functional split between O-RU and O-DU whereas we considered double split where the LLS is between O-RU and O-DU, and the HLS is between O-DU and O-CU in FlexBFDP.

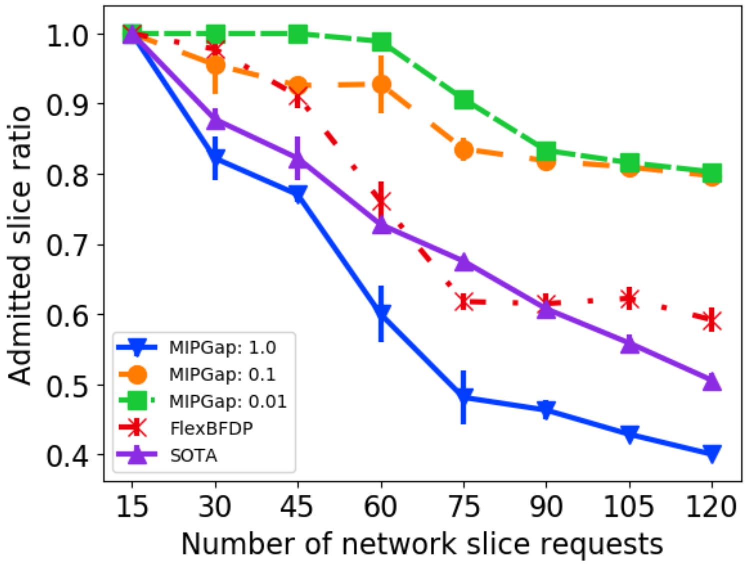

Figure 5.6: Admitted slice ratio achieved with different number of network slice requests

In the first place, their approach seems to be beneficial as placing the virtual network functions in the same site reduces the latency and bandwidth requirements between functions. However, as can be seen in Figure 5.6, placing together takes away the flexibility of distributing O-RU, O-DU and O-CU to different sites and results in a lower number of admitted network slice request compared to our approach. In SOTA, especially when the number of network slice requests are high, the tier sites closer to the edge occupied with the early network slice requests due to the tendency to place functions in the same site. This behaviour causes the latter slice requests to be not admitted because only central sites remain available after placing former slice requests closer to the edge, most likely unable to meet the latency requirements of these requests. When the number of network slice requests is 120, Gurobi solver with MIPGap values 0.01 and 0.1 has 33% more admission rate than FlexBFDP. However, the admission rate of FlexBFDP is 20% higher than state-of-the-art algorithm and

50% higher than the Gurobi solution with MIPGap value 1.

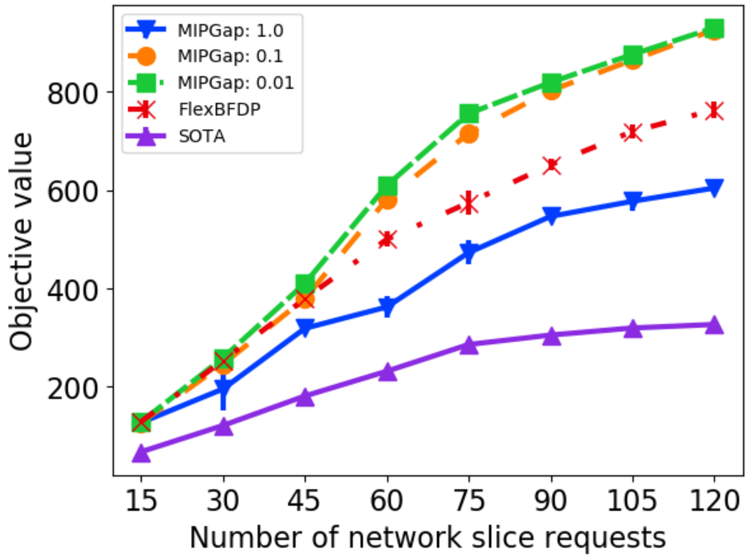

Figure 5.7: Objective value achieved with different number of network slice requests

As we mentioned, our objective value factors in the centralization of the resources. More we centralize the virtual resources, maintenance and operation costs are decreasing. The comparison between Gurobi solutions found with MIPGaps 0.01, 0.1, 1 compared with our heuristic approach FlexBFDP and SOTA can be seen in Figure 5.7. When the number of network slice requests is between 15 and 45, we can see that Gurobi solutions with MIPGaps 0.01, 0.1, and FlexBFDP perform similar to each other while the solution with MIPGap 1 and SOTA follows these results, respectively. After the number of requests exceeds 45, the gap between FlexBFDP and the solutions with MIPGap 0.01 and 0.1 widens with increasing requests. Finally, the number of requests reaches 120, and solutions with MIPGap 0.01 and 0.1 perform very similarly to each other and achieve 21% more objective value than the FlexBFDP approach, while FlexBFDP outperforms the solution with MIPGap 1 by 26%, and SOTA by 140%. The reason behind the poor performance of SOTA is that SOTA tries to place the functions of a network slice request at the same site. The selected site must be close to the edge to place all functionality to this site because communication between RF functionality and PHY functionality requires low latency and high bandwidth, as can be seen in split option 8's requirements at Table 5.1. Also, edge resources are more scarce and precious than the resources more central, and SOTA uses edge resources for the requests that are processed early, causing the denial of other requests with stringent requirements. As we can see, Gurobi solver with MIPGap 0.01 and 0.1 has achieved higher objective values than FlexBFDP, but FlexBFDP manages to outperform Gurobi solver with MIPGap 1 and SOTA approach.

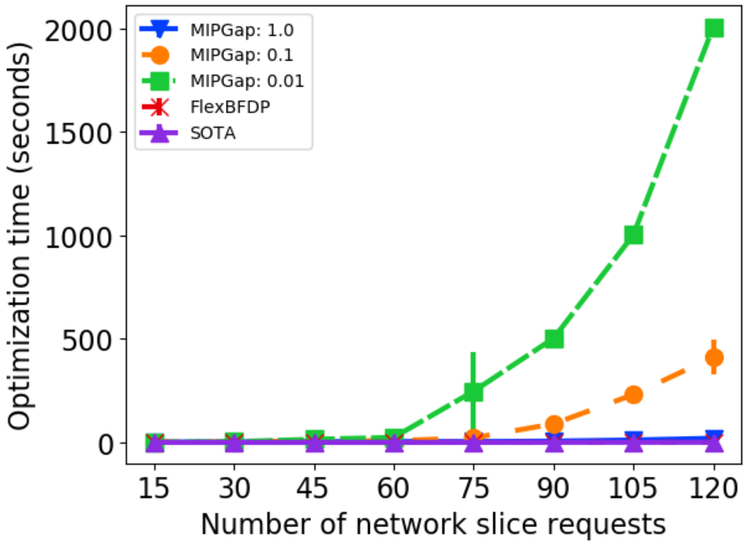

Figure 5.8: Computation time of Gurobi solver and heuristic approaches with different number of network slice requests

In Fig. 5.8, the time passed to find the solution with different approaches versus the number of network slice requests is represented. When the number of requests is between 15 and 60, all approaches find the solution in ~ 1 second. However, starting from 60 requests, the time consumption of the Gurobi solver with MIPGap 0.01 starts to increase exponentially. Also, MIPGap 0.1 starts to take significant time compared to heuristic solutions. As we can see in the figure, as the number of requests reaches 120, the solution with MIPGap 0.01 takes ~ 2000 seconds, significantly more than the solution with MIPGap 0.1, which takes around 410 seconds. The advantage of FlexBFDP over MILP solutions that Gurobi generated is that it takes significantly less time to compute. Gurobi takes close to 2000 seconds to compute when the number of network slice requests is approaching 120, whereas FlexBFDP finds a solution in ~ 1 second. Across all the values of the number of requests, FlexBFDP, SOTA, and the solution with MIPGAP 1 takes ~ 1 second. When we considered these three fastest approaches, our heuristic FlexBFDP achieves higher objective values than the other two approaches while not taking too much time to compute.

5.2.5 General Discussion

We can see from all our experiments that mathematical optimization with different MIPGap parameters performs differently from each other significantly in terms of achieved objective value and computation time. The main learning point of our mathematical optimization experiments is that there is a clear trade-off between performance and time consumption, and we need to select the most suitable parameters according to our needs. After that, we compared different functional split options and their performance compared to a flexible functional split option selection for each network slice in the network infrastructure. The takeaway from these experiments is that flexibility provided by flexible functional splitting enables us to admit more network slice requests to our physical network infrastructure and increases the effective use of limited resources and profit off of these network slice requests. Then, to scale our solution to more extensive networks, we proposed a heuristic solution FlexBFDP and compared it to a state-of-the-art solution in the literature and Gurobi solutions discussed in early experiments. These experiments show the trade-off between time consumption and objective value by comparing FlexBFDP and Gurobi solutions with MIPGap 0.01, 0.1, and 1. Also, our heuristic outperforms the state-of-the-art solution because of the joint consideration of centralization and flexible placement of functions utilizing best fit decreasing strategy.

CHAPTER 6

CONCLUSIONS

6.1 Conclusion

New 5G New Radio architecture is required to support the diverse requirements of 5G use cases like massive mobile broadband in urban and rural areas, ultra-reliable low latency communication for mission-critical real-time applications, and massive machine-type communication in automated factories and sensors scattered on broad areas.

This thesis defined a network slicing placement problem on multi-tier 5G O-RAN network infrastructure. The network slice placement problem is mathematically modeled as an optimization problem maximizing the network's total financial value by admitting different types of network slices described as virtualized baseband function chains. The flexible functional dual-split concept is employed for each network slice to utilize network resources optimally. The resulting mathematical model is solved with the Gurobi solver, a mathematical optimization solver. Optimization results of flexible functional dual-split are compared with fixed functional split results. The results compare flexible functional split, and fixed functional split show that flexible functional splits enable infrastructure providers to utilize their limited resources better and satisfy the needs of more mobile network operators to increase both sides' profits compared to fixed functional split options.

A heuristic approach, FlexBFDP, is proposed to scale the framework to wider networks where Gurobi can not find an optimal solution in a feasible time. FlexBFDP is compared with Gurobi results and a state-of-the-art solution derived from [20]. The results showed that FlexBFDP outperforms the state-of-the-art solution in terms of total financial value gained. While Gurobi results with MIPGap 0.01 yield better results than FlexBFDP in terms of financial value, the heuristic approach finds the solution much faster than mathematical solver as expected. The trade-off between achieving better objective value and better time utilization becomes meaningful for more complex network infrastructures and large sets of network slice requests, where time consumption of Gurobi becomes unfeasible.

6.2 Future Work

In future work, we plan to improve the performance of the heuristic approach FlexBDFP to close the gap between mathematical optimization solutions. Although it is impossible to surpass the results from Gurobi solutions with low MIPGap values because they consider every possible solution, it is possible to close the performance gap between FlexBFDP and Gurobi solutions as much as possible, utilizing the advantage of much faster computation of the solution. Also, different types of network topologies will be covered to widen the coverage of our framework. Different topologies may require different solutions depending on the structure, which may lead to utilizing different heuristics to maximize performance on each topology. Furthermore, we will compare our approach with more state-of-the-art implementations to see the advantages and disadvantages of our approach on different types of networks and improve our approach by adapting the cumulative knowledge acquired in this domain. Finally, we plan to use a discrete-event network simulation environment to get results in more realistic network scenarios to see the effectiveness and shortcomings of our approach.

REFERENCES

[1] "Small cell virtualization functional splits and use cases," Tech. Rep. SCF159.07.02, Small Cell Forum, January 2016.

[2] "Technical specification group radio access network; study on new radio access technology, radio access architecture and interfaces (release 14)," Tech. Rep. 3GPP TR 38.801, 3GPP, April 2017.

[3] O. R. Alliance, "O-ran: towards an open and smart ran," *White Paper*, 2018.

[4] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[5] N. F. V. NFV, "Etsi gs nfv 001 v1. 1.1 (2013-10)," 2013.

[6] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vnf placement at the network edge," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 693–701, 2018.

[7] F. Carpio, S. Dhahri, and A. Jukan, "Vnf placement with replication for loac balancing in nfv networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[8] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong, "Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 172–185, 2017.

[9] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: Vnf placement algorithms for a dynamic amp; realistic edge cloud environment," in *Proceedings of the IEEE Global Communications Conference*, pp. 1–6, 2017.

[10] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and qos-aware path allocation and vnf placement for service

function chaining," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 374–388, 2018.

[11] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," *IEEE Access*, vol. 6, pp. 19691–19701, 2018.

[12] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware vnf placement and chaining based on a flexible resource allocation approach," in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–7, 2017.

[13] H. Moens and F. D. Turck, "Vnf-p: A model for efficient placement of virtualized network functions," in *10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 418–423, 2014.

[14] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE INFOCOM 2009*, pp. 783–791, 2009.

[15] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. De Meer, "Energy efficient virtual network embedding," *IEEE Communications Letters*, vol. 16, no. 5, pp. 756–759, 2012.

[16] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[17] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, "Virtual network embedding for evolving networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5, 2010.

[18] D. Jiang and G. Liu, "An overview of 5G requirements," *5G Mobile Communications*, pp. 3–26, 2017.

[19] M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, "A comprehensive survey of ran architectures toward 5G mobile communication system," *IEEE Access*, vol. 7, pp. 70371–70421, 2019.

[20] A. De Domenico, Y.-F. Liu, and W. Yu, "Optimal virtual network function deployment for 5G network slicing in a hybrid cloud infrastructure," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7942–7956, 2020.

[21] S. K. Singh, R. Singh, and B. Kumbhani, "The evolution of radio access network towards open-ran: Challenges and opportunities," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–6, 2020.

[22] S. Perrin, "Evolving to an open c-ran architecture for 5g," *Fujitsu Heavy reading White Paper*, p. 2017, 2017.

[23] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Communications Magazine*, vol. 49, no. 2, pp. 102–111, 2011.

[24] 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*, vol. 21, no. 1, pp. 146–172, 2018.

[25] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks—a technology overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[26] A. Pizzinat, P. Chanclou, F. Saliou, and T. Diallo, "Things you should know about fronthaul," *Journal of Lightwave Technology*, vol. 33, no. 5, pp. 1077–1083, 2015.

[27] T. Pfeiffer, "Next generation mobile fronthaul and midhaul architectures," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B38–B45, 2015.

[28] O. Alliance, "O-RAN use cases and deployment scenarios," 2020.

[29] J. Yusupov, A. Ksentini, G. Marchetto, and R. Sisto, "Multi-objective function splitting and placement of network slices in 5g mobile networks," in *Proceedings of the IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6, IEEE, 2018.

[30] C. C. T. Mark, D. Niyato, and T. Chen-Khong, “Evolutionary optimal virtual machine placement and demand forecaster for cloud computing,” in *IEEE International Conference on Advanced Information Networking and Applications*, pp. 348–355, 2011.

[31] X. Meng, V. Pappas, and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement,” in *Proceedings of the IEEE INFOCOM*, pp. 1–9, 2010.

[32] H. Nguyen Van, F. Dang Tran, and J.-M. Menaud, “Autonomic virtual resource management for service hosting platforms,” in *Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pp. 1–8, 2009.

[33] V. Cardellini, E. Casalicchio, F. Lo Presti, and L. Silvestri, “Sla-aware resource management for application service providers in the cloud,” in *Proceedings of the First International Symposium on Network Cloud Computing and Applications*, pp. 20–27, 2011.

[34] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.-M. Chao, and J. Li, “Virtual machine consolidated placement based on multi-objective biogeography-based optimization,” *Future Generation Computer Systems*, vol. 54, pp. 95–122, 2016.

[35] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, “A multi-objective ant colony system algorithm for virtual machine placement in cloud computing,” *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230–1242, 2013.

[36] X. Li, Z. Qian, S. Lu, and J. Wu, “Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center,” *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1222–1235, 2013.

[37] D. S. Dias and L. H. M. K. Costa, “Online traffic-aware virtual machine placement in data center networks,” in *Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 1–8, 2012.

[38] Q. Sun, P. Lu, W. Lu, and Z. Zhu, “Forecast-assisted nfv service chain deployment based on affiliation-aware vnf placement,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2016.

[39] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Computer Communications*, vol. 102, pp. 1–16, 2017.

[40] M. C. Luizelli, W. L. da Costa Cordeiro, L. S. Buriol, and L. P. Gaspary, "A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining," *Computer Communications*, vol. 102, pp. 67–77, 2017.

[41] A. Leivadeas, M. Falkner, I. Lambadaris, and G. Kesidis, "Optimal virtualized network function allocation for an sdn enabled cloud," *Computer Standards & Interfaces*, vol. 54, pp. 266–278, 2017.

[42] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, "Virtual network function placement for resilient service chain provisioning," in *Proceedings of the 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 245–252, IEEE, 2016.

[43] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for open ran (o-ran) 5G networks," *Computer Networks*, vol. 188, p. 107809, 2021.

[44] A. Sriram, M. Masoudi, A. Alabbasi, and C. Cavdar, "Joint functional splitting and content placement for green hybrid cran," in *Proceedings of the IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–7, IEEE, 2019.

[45] Y. Xiao, J. Zhang, and Y. Ji, "Can fine-grained functional split benefit to the converged optical-wireless access networks in 5G and beyond?," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1774–1787, 2020.

[46] D. Harutyunyan and R. Riggio, "Flex5G: Flexible functional split in 5G networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, 2018.

[47] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "On the impact of novel function mappings, sharing policies, and split settings in network slice

design," in *Proceedings of the 16th International Conference on Network and Service Management (CNSM)*, pp. 1–9, 2020.

[48] A. M. Alba, J. H. G. Velásquez, and W. Kellerer, "An adaptive functional split in 5g networks," in *Proceedings of the IEEE INFOCOM 2019*, pp. 410–416, 2019.

[49] J. Zhang, Y. Ji, X. Xu, H. Li, Y. Zhao, and J. Zhang, "Energy efficient baseband unit aggregation in cloud radio and optical access networks," *Journal of Optical Communications and Networking*, vol. 8, no. 11, pp. 893–901, 2016.

[50] 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 *International IFIP Conference on Optical Network Design and Modeling*, pp. 82–93, Springer, 2019.

[51] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, and Y. Ji, "Isolation-aware 5g ran slice mapping over wdm metro-aggregation networks," *Journal of Lightwave Technology*, vol. 38, no. 6, pp. 1125–1137, 2020.

[52] D. Harutyunyan, R. Riggio, S. Kuklinski, and T. Ahmed, "Cu placement over a reconfigurable wireless fronthaul in 5g networks with functional splits," *International Journal of Network Management*, vol. 30, no. 1, p. e2086, 2020.

[53] L. M. M. Zorello, S. Troia, M. Quagliotti, and G. Maier, "Power-aware optimization of baseband-function placement in cloud radio access networks," in *Proceedings of the International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–6, IEEE, 2020.

[54] D. Harutyunyan and R. Riggio, "Flexible functional split in 5G networks," in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–9, 2017.

[55] N. Alliance, "Description of network slicing concept," *NGMN 5G P*, vol. 1, no. 1, 2016.

[56] 3GPP", "Study on management and orchestration of network slicing for next generation network," tech. rep., 2016.

[57] "Minimum requirements related to technical performance for imt-2020 radio interface(s)," Tech. Rep. ITU-R M.2410-0, Radiocommunication Sector of International Telecommunication Union, November 2017.

[58] Next Generation Mobile Networks Alliance, "NGMN Overview on 5G RAN Functional Decomposition," p. 47, 2018.

[59] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020.