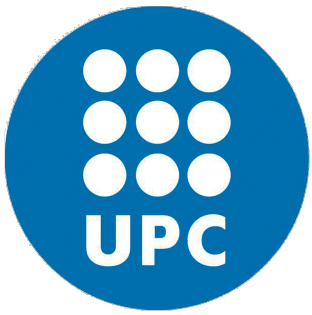
Edge Computing infrastructure for 5G networks: a placement optimization solution

By

ALEJANDRO SANTOYO-GONZÁLEZ

Ph.D. Advisor

CRISTINA CERVELLÓ-PASTOR



Department of Network Engineering
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Thesis submitted to Universitat Politècnica de Catalunya in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY IN NETWORK ENGINEERING.

BARCELONA, APRIL 2020

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Abstract

This thesis focuses on how to optimize the placement of the Edge Computing infrastructure for upcoming 5G networks. To this aim, the core contributions of this research are twofold: 1) a novel heuristic called **Hybrid Simulated Annealing** (HSA) to tackle the NP-hard nature of the problem and, 2) a framework called **EdgeON** providing a practical tool for real-life deployment optimization.

In more detail, Edge Computing has grown into a key solution to 5G latency, reliability and scalability requirements. By bringing computing, storage and networking resources to the edge of the network, delay-sensitive applications, location-aware systems and upcoming real-time services leverage the benefits of a reduced physical and logical path between the end-user and the data or service host.

Nevertheless, the edge node placement problem raises critical concerns regarding deployment and operational expenditures (i.e., mainly due to the number of nodes to be deployed), current backhaul network capabilities and non-technical placement limitations. Common approaches to the placement of edge nodes are based on: Mobile Edge Computing (MEC), where the processing capabilities are deployed at the Radio Access Network nodes and Facility Location Problem variations, where a simplistic cost function is used to determine where to optimally place the infrastructure. However, these methods typically lack the flexibility to be used for edge node placement under the strict technical requirements identified for 5G networks. They fail to place resources at the network edge for 5G ultra-dense networking environments in a network-aware manner.

This doctoral thesis focuses on rigorously defining the Edge Node Placement Problem (ENPP) for 5G use cases and proposes a novel framework called **EdgeON** aiming at reducing the overall expenses when deploying and operating an Edge Computing network, taking into account the usage and characteristics of the in-place backhaul network and the strict requirements of a

5G-Edge Computing ecosystem. The developed framework implements several placement and optimization strategies thoroughly assessing its suitability to solve the network-aware ENPP. The core of the framework is the in-house developed heuristic HSA, seeking to address the high complexity of the ENPP while avoiding the non-convergent behavior of other traditional heuristics (i.e., when applied to similar problems).

The findings of this work validate our approach to solve the network-aware ENPP, the effectiveness of the heuristic proposed and the overall applicability of **EdgeON**. Thorough performance evaluations were conducted on the core placement solutions implemented revealing the superiority of HSA when compared to widely used heuristics and common edge placement approaches (i.e., a MEC-based strategy). Furthermore, the practicality of **EdgeON** was tested through two main case studies placing services and virtual network functions over the previously optimally placed edge nodes.

Overall, our proposal is an easy-to-use, effective and fully extensible tool that can be used by operators seeking to optimize the placement of computing, storage and networking infrastructure at the users' vicinity. Therefore, our main contributions not only set strong foundations towards a cost-effective deployment and operation of an Edge Computing network, but directly impact the feasibility of upcoming 5G services/use cases and the extensive existing research regarding the placement of services and even network service chains at the edge.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis and research advisor Dr. Cristina Cervelló Pastor, who believed in me without even knowing me in person. This thesis would not have been possible without her guidance, knowledge and her unique hands-on support. I could not have had a better person as my supervisor.

I would like to extend my gratitude to Dr. Pradeeban Kathiravelu and Dr. Eduard Marín Fàbregas for the time and efforts dedicated to assessing this thesis. Their reviews and comments were incredibly helpful and very much appreciated.

A special thanks to Dr. Dimitrios P. Pezaros for hosting me as a visiting researcher at the Network Systems Research Laboratory (NETLAB), University of Glasgow. Thank you Dimitrios for your support and incredible professional vision. My experience there was truly remarkable.

I would also like to thank all my colleagues at the university lab, the members of the Network Engineering Department at UPC and of NETLAB at Glasgow and all others who somehow contributed to this thesis and to the research process. It has been an amazing experience!

My deepest and most special gratitude to my family. Without doubt, this is your thesis. Specially to Lili, my wife, my partner and friend, ours is THE life.

"*To my family, the pillars to my earth*"

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

BBU Baseband Unit

C-RAN Cloud RAN

CAPEX Capital Expenditures

CDN Content Delivery Network

CDN-PoP Content Delivery Network-PoP

CORD Central Office Re-architected as a Datacenter

COTS Common-Of-The-Shelf

DDoS Distributed Denial of Service

DSL Domain Specific Language

EA Evolutionary Algorithm

EC Edge Computing

eBPF Extended Berkeley Packet Filter

eMBB Enhanced Mobile Broadband

EN Edge Node

ENPP Edge Node Placement Problem

ETSI European Telecommunications Standard Institute

EWMA Exponentially Weighted Moving Averages

FC Fog Computing

FLP Facility Location Problem

FN Fog Node

FPR False Positive Ratio

FNR False Negative Ratio

GLPK GNU Linear Programming Kit

HetNet Heterogeneous Network

HSA Hybrid Simulated Annealing

IaaS Infrastructure as a Service

ILP Integer Linear Programming

IoT Internet of Things

IP Internet Protocol

ISP Internet Service Providers

ISP-PoP Internet Service Providers-PoP

KPI Key Performance Indicator

LRPON Long-Reach Passive Optical Network

LTE Long-Term Evolution

M2M Machine-to-Machine

MAN Metropolitan Area Network

MEC Mobile Edge Computing

MILP Mixed Integer Linear Programming

mIoT Mobile IoT

mMTC Massive Machine-type Communications

MO-ENPP Multi-Objective ENPP

NFV Network Function Virtualization

NOUP Near-Optimal UPF Placement

OPEX Operational Expenditures

OUP Optimal UPF Placement

PISA Protocol Independent Switch Architecture

PoP Point-of-Presence

PTA Pre-Optimized TG Area

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

ROI Return of Investment

RTT Round-Trip Time

SA Simulated Annealing

SCeNB Enhanced Small Cells

SDN Software-Defined Network

SO-ENPP Single-Objective ENPP

TAP Traffic Aggregation Point

TCP Transmission Control Protocol

TDP Traffic Demand Point

TG Traffic Generator

TSA Traditional Simulated Annealing

UPF User Plane Function

VNF Virtual Network Function

WAN Wide Area Network

CHAPTER 1

INTRODUCTION

In recent years, there has been a steep surge in the amount of digital data generated worldwide due to the rapid evolution of emerging technological paradigms such as the Internet of Things (IoT). According to Cisco, the number of interconnected mobile devices will reach over 11 billion by 2021, while the global information provider IHS Markit has stated that 125 billion smart devices will exist by 2030 [1][2]. Within a decade, from 2010 to 2020, the transmitted digital data has multiplied its value 200 times and this multiplying factor will be increased 1000 times more by 2030 [3]. As a consequence, computation-intensive applications and business models have been quickly evolving and increasing at an incredible pace, stretching to the limit the capabilities of the remote cloud communication and processing architecture.

5G networking has been envisioned to answer the requirements of the use cases and technological trends associated with such traffic growth (see Figure 1.1). Throughout the past 5 years, the advances in 5G standardization and implementation have encouraged the industry to invest and accelerate the introduction of solution proposals for 5G use cases (e.g., Virtual and Augmented Reality, Autonomous Driving, Real-time Manufacturing). Namely, both the industry and academia have been dedicating extensive resources to develop appropriate frameworks, testbeds and prototypes of a 5G network architecture. The goal being to place such architecture over a shared (yet sliced) underlying infrastructure and flexible marketplace where isolation

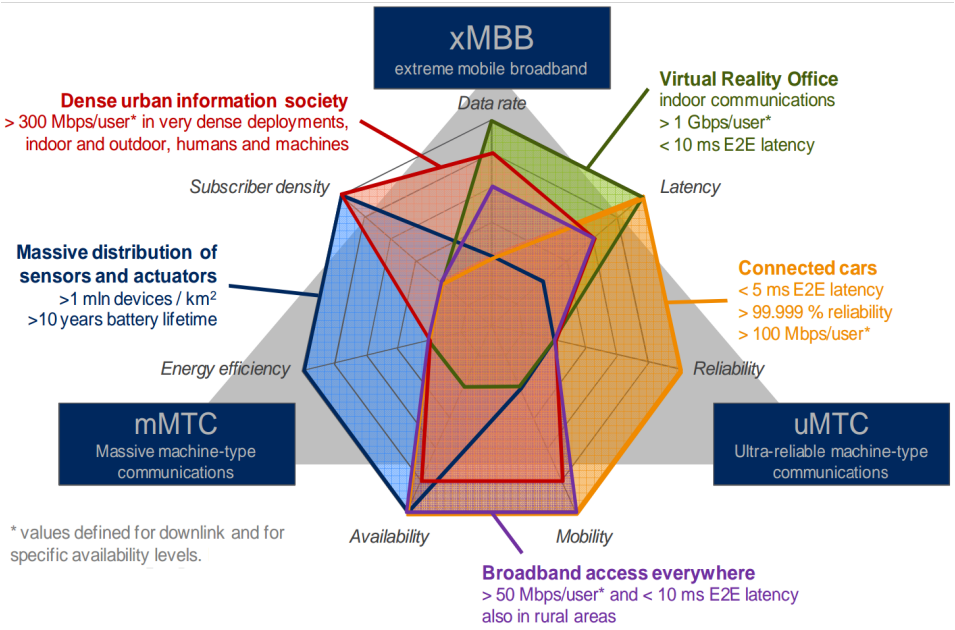


Figure 1.1: 5G use cases and requirements [5]. The 5G requirements are grouped into three main categories and five underlying subcategories in order to clarify use cases and scenarios.

is guaranteed throughout all operational layers, aiming at the efficient implementation of next generation services in the longer run. Overall, these efforts have targeted the following core purposes: top-level system flexibility, automation, self-awareness and cost-effective orchestration and operation [4].

Under these envisioned 5G networks, a user-centered ecosystem providing seamless integration between users and devices is to be achieved based on smart interconnection, artificial intelligence-based systems and automated self-aware orchestration and management. To this aim, the scenario classification devised by the International Telecommunications Union-Radiocommunication Sector (ITU-R), shows mission-critical services depending on strict delay constraints reaching less than 1 ms [3][6]. Real-time critical communications and traffic safety impose additional complexities as they require top-level reliability and availability while ensuring ultra-low latency. Meanwhile, emerging ultra-high bandwidth requirements joined to the evolution of service and traffic patterns, are leading to an unprecedented need for hyper-connectivity and ultra-reliable high performance.

Enhanced Mobile Broadband (eMBB) and Massive Machine-type Communications (mMTC)

will certainly push the limits of current networking platforms since around 1 million interconnected devices per squared kilometer are to be supported [2, 4], thus rising complex technical challenges regarding radio resource allocation, data transmission, routing/processing and Quality of Service (QoS) delivery. Smart Cities and e-Health deployments will pose strict data rate demands, while Autonomous Driving and Industry Monitoring will require nearly 100% reliability and millisecond-level latency [3]. In addition to severe QoS and Quality of Experience (QoE) needs, 5G is required to enforce high security and privacy for e-Banking, Security Monitoring, Traffic Safety and Mobile Health. Moreover, overall power consumption is to be reduced to ensure long-time battery life and *green* networking.

In this context, the remote datacenter model has become inefficient and unable to cope with the rising technical demands. By providing an end-to-end communication delay of around $60-100$ ms, current remote clouds are unable to guarantee the required 1 ms round-trip maximum latency and stable jitter for delay-sensitive and location-aware use cases [4, 5, 7]. Privacy and security concerns are additionally stretching the cloud capabilities. As the use of applications working over distributed platforms increases (e.g., blockchain-based systems, multimedia cache servers), centralized service models are being discarded in favor of decentralized, highly-resilient, close-to-the-user infrastructure. Scalability has additionally grown into a critical concern given the massive amounts of data to be processed. Deep data analysis mechanisms to accurately segment and generate maximum value from each customer are causing critical bottlenecks in the data transmission systems, while restraining the use of distributed and resource intensive deep/machine learning systems at higher scales.

The convergence of Edge Computing (EC), Network Function Virtualization (NFV), Software-Defined Networks (SDNs) and other enabling technologies will become the pillars to answer the aforementioned challenges and to implement next generation standalone 5G networks. Namely, EC has become a solid alternative to the traditional datacenter-based service scheme. By bringing computing, storage and networking resources to the users' vicinity, EC aims at reducing the physical and logical distance between hosts and end-users, while satisfying the requirements of distributed resource-intensive applications and delay-sensitive use cases through a geographically distributed set of small-sized Edge Nodes (ENs). Concretely, EC is able to effectively reduce

around 20% of the average response time and 90% of the north-south traffic when compared to a remote cloud service architecture, while significantly improving scalability [8–11]. However, a distributed set of nodes raises critical concerns regarding Capital Expenditures (CAPEX) and Operational Expenditures (OPEX), deployment strategies, QoS and QoE.

The cost of deploying an EN directly depends on two main factors: location-dependent and computation-dependent expenses [42]. The former accounts for the costs related to power and network connections installation, land (or space) acquisition, basic supplies (e.g., cost of the water and electricity). The latter refers to the required computing, storage and networking capacity to be allocated, software licenses, management expenses, staff salary.

On the other hand, the numerous tradeoffs involved make the EN network deployment challenging. At first glance, placing an EN is constrained by the underlying network capacity and the operators' Point-of-Presences (PoPs), Central Offices and other suitable sites, in order to ensure lower costs, maximum transmission efficiency, power usage reduction and high-performance interconnection. Moreover, a clear tradeoff results from the number of ENs and the allocated capacity, directly impacting the total expenses and the operators' ability to maximize the Return of Investment (ROI). Ensuring high-performance processing requires the utilization ratio to be preserved under a certain threshold to avoid QoS and QoE degradation due to capacity overload. However, if the capacity and demand allocation are not properly managed, both CAPEX and OPEX may rise significantly, due to underutilized or oversubscribed nodes.

In addition, close-to-the-user proximity to satisfy low latency requirements poses a challenge regarding the site selection. Placing the infrastructure at the Radio Access Network (RAN) nodes, following the Mobile Edge Computing (MEC) approach, is often seen as the solution. Nevertheless, this is commonly unfeasible since base stations are typically placed at remote locations with very limited physical equipment space (e.g., macro-cell towers located at the top of a remote hill, small-cells placed at street cabinets) [12]. Moreover, following a *continuous* placement approach (i.e., the territory is analyzed as a set of coordinates and all coordinate pairs are analyzed) is unfeasible as it increases the problem complexity and overall expenses. Nevertheless, a *discrete* strategy -i.e., where a list of potential sites is known beforehand- should carefully consider existing potential sites (i.e., Internet Service Providers-PoPs (ISP-PoPs), Content Delivery Network-

PoPs (CDN-PoPs), Central Offices) and any available unforeseen locations.

Under these circumstances, the EN placement strategy has become crucial. By optimizing the EN placement, the overall deployment and operation cost savings can be highly increased and the user requirements can be fully satisfied [27]. For 5G networks, ultra-dense networking will remarkably change the placement of mobile base stations, cache servers, datacenters and thousands of ENs are to be deployed within a city to satisfy 5G ultra-low-latency and reliability needs. Therefore, the economical feasibility of the 5G/EC ecosystem is tied to the optimization of the capacity planning and deployment strategies, i.e., the EN placement methods. However, most capacity planning studies assume that the service infrastructure has been already deployed focusing on the resource allocation and capacity problem, thus overlooking the need to optimize the location selection procedures [13, 14].

Extensive research has been found regarding problems closely related ot the EN site selection optimization: Facility Location Problems (FLPs), datacenter, base station and generic server placement (e.g., cache servers) [15–24]. Additionally, few articles were found targeting the edge server placement problem [13, 25–27]. Several limitations prohibit the use of these studies to effectively place an EN network under 5G constraints. FLP solutions, for instance, cannot be directly applied for the EC infrastructure deployment due to typical cost function simplicity, traditional convergence into a specific operational problem (e.g., Weber, coverage) and lack of non-technical restrictions analysis [21]. Datacenter and generic server placement strategies overlook the need for a shared and geographically distributed infrastructure where the member nodes must cooperatively solve offloaded tasks while maintaining minimum latency levels. In addition, the lack of flexibility forces these models to be discarded when applied to the ultra-dense networking demands of 5G networking [24, 28]. Base station placement is mostly done based on tessellation and clustering methods that may not be suitable for 5G traffic patterns and service trends under ultra-dense 5G networking [15, 16]. Finally, the edge server placement solutions found have not been tailored to 5G requirements, while covering a limited set of specific scenarios, overlooking the underlying network capacity constraints and over-simplifying the user demand distribution through traditional clustering approaches.

For the above reasons, our focus throughout this thesis will be to propose and solve the

optimization problem pursuing the cost-effective placement of ENs complying with the identified 5G requirements.

1.1 Research Problem and Objectives

From the context thoroughly detailed in the above section, rises a clear need to cost-effectively place the service infrastructure at the network edge to meet 5G requirements. Such problem is hereinafter referred to as: Edge Node Placement Problem (ENPP). As a direct consequence, this thesis seeks to answer the following research question: **Is it feasible to solve the ENPP, thus optimally placing the ENs in a given territory, while optimizing the overall deployment costs, ensuring both cost-effectiveness and 5G technical and non-technical requirement satisfaction?**

To address the identified problem, a set of main objectives and tasks were defined. The primary goals of the present thesis are summarized below:

1. To define a set of EN placement parameters merging the deployment principles of 5G, EC and its enabling technologies to effectively consider the tradeoffs involved in the ENPP.

2. To define and rigorously formalize the ENPP as a multi-objective optimization problem, considering the use cases and specific requirements of 5G environments and the characteristics of the in-place network and computing infrastructure.

3. To propose a novel ENPP solution for a reference EC architecture, aiming at overall cost minimization and balanced capacity allocation, while ensuring customer demand satisfaction.

4. To evaluate the developed strategy by comparing it to other heuristic and meta-heuristic implementations applied to the ENPP solution or closely related problems (e.g., FLPs).

5. To propose and evaluate two real-life scenarios where the placed ENs is effectively used to deploy and execute Virtual Network Functions (VNFs), i.e., a Distributed Denial of Service (DDoS) attack detection, 5G User Plane Functions (UPFs).

Furthermore, a set of core tasks is specified, directly linked to the main research problem and goals:

1. To perform a thorough and systematic review of the state-of-the-art and prior literature on 5G, EC and other relevant enabling technologies.

2. To study closely related research problems (e.g., FLPs, datacenter/server/base station placement problem) in order to determine the scope and core challenges of the ENPP and its underlying complexities when tailored to a demanding ecosystem such as 5G.

3. To develop a controlled simulated environment in order to evaluate the proposed solution against exact mathematical methods, seeking to accurately analyze the developed solution performance and capabilities to be applied in real-life scenarios.

4. To evaluate the proposed EN placement strategy by comparing it to other heuristic and meta-heuristic implementations applied to the ENPP solution, using realistic simulated scenarios.

5. To analyze the proposed VNFs executed over the EC network in order to evaluate its performance and core benefits for 5G use cases.

1.2 Methodology

The research carried out in this thesis was divided in several main working areas defined by the objectives presented in Section 1.1. The following subsections explain the structure of the methodology followed during this work.

Systematic Literature Review

Despite current efforts in EC development and standardization, there is still no formal definition of what is the "*edge*" or what edge nodes should be. As a result, there is a lack of research regarding where to cost-effectively place the physical service infrastructure at the users' vicinity (i.e., the ENs) for 5G use cases.

To answer this question, a study of the future 5G/EC ecosystem is performed in this document. Current research about the related 5G technologies is critically assessed, considering standardization, development efforts and available solutions. Additionally, closely related problems are identified and studied. Among them, high priority is given to FLPs, server and base station placement, low to high-density datacenter placement and edge server site selection.

ENPP placement parameters definition

The baseline to solve the ENPP is the definition of a comprehensive and effective set of placement parameters. Through achieving this goal, the solution proposed in this thesis is able to thoroughly evaluate each EN potential site and achieve cost optimization without affecting user demand satisfaction and overall service performance.

From the revised literature and the 5G requirements, certain parameters stand out (e.g., latency, bandwidth, site rentals). However, additional considerations such as non-technical restrictions and future service patterns are also analyzed. The objective was to propose a set of criteria to assess potential sites to return a location subset optimizing the deployment and operation cost of the EN network. To this aim, besides location-dependent and infrastructure-dependent costs, accurate predictions on future service trends and technological advances were considered, along with industry advances on 5G enabling technologies, telecommunication operators market strategies and emergent business opportunities.

On the other hand, service demand geo-distribution and hourly behavior are also directly linked to the EN placement. Nevertheless, its interrelation remains an open question in the 5G context due to the lack of operational data. Such continuously changing environment has been systematically and carefully reviewed, as a cost-effective placement solution should offer adequate flexibility levels to cope with such ecosystem dynamics

Overall, cost reduction, usability and applicability of any ENPP solving approach, depends on the scope and effectiveness of its evaluation and optimization criteria. For this reason, a list of parameters for the ENPP solution is one of the main outputs obtained with this work.

Network-agnostic ENPP solution

5G service performance for ultra-low latency scenarios will primarily depend on the offloading

of computation tasks to ENs. To efficiently handle the peak load and satisfy the requirements of remote program execution in real-life scenarios, any EN placement strategy must be aware of the underlying network capacity and current usage ratio. Such condition imposes additional concerns an complexities to the ENPP formulation and solution.

In order to effectively solve the ENPP in such heavily constrained scenarios, solid modeling, testing and analysis methods must be developed as foundations. To this aim, we firstly propose a latency-constrained network-agnostic solution to the ENPP based on an in-house developed heuristic. From this starting point, additional parameters are added to the model and the problem solution is extended through the proposal and evaluation of a fully extensible framework.

Network-aware ENPP solution

Based on two network-agnostic variants of the ENPP solved through in-house placement strategies, we propose a network-aware solution to the ENPP considering 5G technical requirements with special focus on: ultra-low latency, ultra-high reliability and ultra-dense networking. To ensure the practicality and applicability of our proposal, we present a framework (implementing several placement strategies in order to thoroughly assess our placement heuristic) for the EN placement in the devised scenarios.

As no standardized EC architecture has been defined yet, we theoretically formalize and solve the ENPP for a reference deployment architecture and EN definition. The research decision of solving the ENPP for a simplified reference architecture goes far beyond problem simplification. Given the lack of operational knowledge about 5G networking and the evolution of next generation service and traffic patterns, assuming a flexible but realistic reference architecture for the 5G/EC ecosystem, allowed us to effectively analyze the EN placement tradeoffs and propose a placement strategy tailored to the requirements of the 5G verticals. In addition, with the advent of revolutionary wireless RAN technologies, such as millimeter-wave communications, we envision geographical areas where the the 5G/EC environment would most likely be deployed over a convergent (i.e., mixing fixed and wireless technologies) backhaul network [6, 29].

Evaluation and optimization of the ENPP solution strategies

After proposing effective solving schemes for the ENPP variants analyzed, it was mandatory to

evaluate them and compare them with other existing placement methods. This allowed us to significantly improve and optimize our solutions while assuring the applicability of their results.

To this aim, several heuristics were implemented, tailored to the ENPP characteristics and following an homogeneous development approach, using a suitable testbed for evaluation purposes. Through a critical analysis of the testing process, the validation of the solutions was achieved while optimizing their underlying components and practicality to the point of proposing a generic flexible framework for the placement of ENs within next generation networks.

Finally, the EN placement strategy evaluation was extended by analyzing two case studies where VNFs were assumed to be deployed over the optimally placed edge infrastructure.

1.3 Resources

To carry out this research, the Department of Network Engineering (ENTEL) provided all the required resources and support, along with outstanding professional guidance and expertise. Additionally, access to training activities, knowledge exchange spaces and other relevant scientific opportunities was granted in order to increase the reach and scope of this research while obtaining adequate feedback.

As of specific materials and tools, Python was the programming language selected to implement all coding tasks, while the mathematical formulations were developed and solved using Pyomo [30, 31] and Gurobi/GNU Linear Programming Kit (GLPK) [32] as underlying solvers. In addition, part of the input data was gently supplied by Telegeography's GlobalComms from its proprietary database[1], about ISP-PoPs operating in Spain.

1.4 Contributions

Answering the core problem and research objectives pursued within this thesis and based on several articles published in recognized journals and conferences, our main contributions can be summarized as follows:

[1] https://www2.telegeography.com/globalcomms-database-service

1. A novel **set of EN placement parameters tailored to 5G use case requirements**. Mainly aiming at achieving ultra-low latency and ultra-high reliability, through a thorough analysis of location-dependent and capacity-related expenses and both technical and non-technical restrictions.

2. Rigorous **definition of the ENPP through mathematical models base on linear programming** targeting relevant variations of the problem, in order to offer maximum flexibility for its practical application in real-life scenarios.

3. An **in-house developed heuristic called Hybrid Simulated Annealing (HSA)**, allowing a flexible and cost-effective placement of an EN network under 5G service constraints and considering both technical and non-technical restrictions as well as current IT-capable locations (e.g., Central Offices, base stations).

4. A novel **framework proposal to extend the capabilities of the proposed heuristic** in order to enhance its usability and practicality by delivering a flexible and expandable platform for operators to adapt to their particular needs and use cases.

5. A **state-of-the-art edge-based DDoS detection system** deployed over the optimally placed ENs, ensuring high-performance processing and low overhead, thus complying with the VNF requirements for IoT devices over 5G networks.

1.5 Thesis outline

The outline of this thesis can be observed in Figure 1.2, where each chapter is described in terms of its contribution scope and its main results.

Chapter 1 presents the motivation for this thesis, the research problem to be addressed, the research objectives, methodology and used resources. Meanwhile, Chapter 2 presents the literature review, focusing on ENPP closely related problems and the most widely used algorithms used to solve placement problems. Additionally, Chapter 2 provides the theoretical background

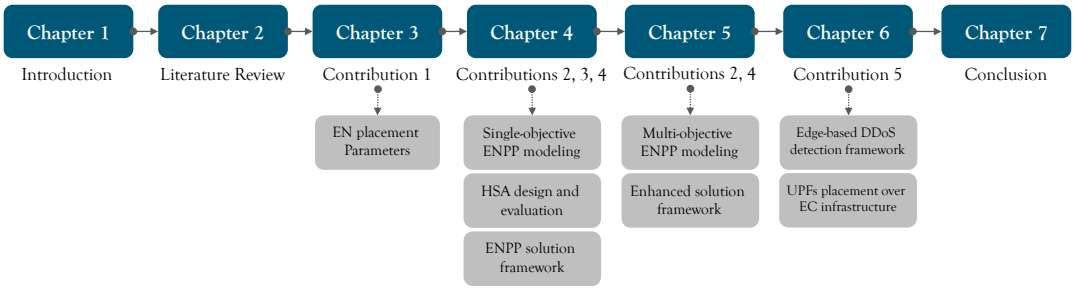


Figure 1.2: Thesis outline.

needed to accomplish the research goals presented in Chapter 1. Namely, an EC reference architecture is outlined and the definition of EN is presented.

Chapter 3 introduces our first contribution by describing the proposed set of parameters to evaluate each EN site within the optimization process.

Chapter 4 presents two network-agnostic single-objective models using a Mixed Integer Linear Programming (MILP) mathematical formulation to optimize the EN placement. In this chapter, key performance elements and concepts (i.e., Traffic Generators (TGs)) are defined and tested. Furthermore, a novel placement strategy called HSA is presented, evaluated and extended through a framework called **EdgeON** aiming at providing operators with an extensible platform to solve the ENPP under custom conditions.

Chapter 5 significantly extends the results presented in Chapter 4 by providing a network-aware multi-objective MILP model to realistically formulate the ENPP. Additionally, the framework presented in Chapter 4 is significantly extended and the enhanced capabilities of the HSA heuristic are showcased, ensuring high usability and flexibility for the future use of the findings as operational tools.

Chapter 6 describes our contributions towards edge-based DDoS detection and UPF placement based on optimally placed ENs. Namely, a novel DDoS detection scheme is proposed based on cutting-edge high-performance packet processing technologies (i.e., Extended Berkeley Packet Filter (eBPF)) and SDN. Moreover, the placement of UPFs under 5G requirements and optimal EN locations is evaluated through a joint EN/UPF placement framework proposal.

Chapter 7 presents a summary of the core findings of this thesis and provides key directions for future work and open research questions.

The diagram in Figure 1.3 aims at helping the reader to follow how the chapters relate to each objective and contribution, as well as the papers each chapter is based on (i.e., the publications listed in Appendix A).

Chapter	Publications	Objectives					Contributions				
		1	2	3	4	5	1	2	3	4	5
3	A. Santoyo-González and C. Cervelló-Pastor, "**Edge Nodes Infrastructure Placement Parameters for 5G Networks**," in 2018 IEEE Conference on Standards for Communications and Networking (CSCN 2018), (Paris, France), p. 6, IEEE, 2018.	✓					✓				
4	A. Santoyo-González and C. Cervelló-Pastor, "**On the Optimal NFVI-PoP Placement for SDN-Enabled 5G Networks**," in Trends in Cyber-Physical Multi-Agent Systems. 15th PAAMS 2017, 2017. ●										
	A. Santoyo-González and C. Cervelló-Pastor, "**Latency-aware cost optimization of the service infrastructure placement in 5G networks**," Journal of Network and Computer Applications, vol. 114, pp. 29–37, 2018.		✓	✓	✓			✓	✓		
	A. Santoyo González and C. Cervelló Pastor, "**Edge Computing Node Placement in 5G Networks: A Latency and Reliability Constrained Framework**," in 2019 6th IEEE CSCloud/ 2019 5th IEEE EdgeCom, (Paris, France), pp. 183–189, IEEE, 2019.		✓	✓	✓			✓	✓	✓	
	A. Santoyo-González and C. Cervelló-Pastor, "**A Framework for Latency-constrained Edge Nodes Placement in 5G Networks**," in XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018), (Granada, Spain), 2018.				✓				✓	✓	
5	A. Santoyo-González and C. Cervelló-Pastor, "Network-aware Placement Optimization for Edge Computing Infrastructure under 5G," IEEE Access, 2020.		✓	✓	✓			✓	✓	✓	
6	A. Santoyo-González, C. Cervelló-Pastor and D. P. Pezaros, "**High-performance, platform-independent DDoS detection for IoT ecosystems**," 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 2019, pp. 69-75.					✓					✓
	A. Santoyo-González and C. Cervelló-Pastor, "**A Framework for Latency-constrained Edge Nodes Placement in 5G Networks**," in XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018), (Granada, Spain), 2018.				✓				✓	✓	
	I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "**A Framework for the Joint Placement of Edge Service Infrastructure and User Plane Functions for 5G**", Sensors, vol. 19, no. 18, p. 3975, 2019.				✓	✓				✓	

● Short paper to present the research problem

Figure 1.3: Diagram matching each research objective and its backing publication

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

To optimize the placement of the service infrastructure at the network edge, an initial analysis of related concepts/technologies and state-of-the-art literature must be performed. In the sections below, a rather comprehensive study of EC and its current implementations is presented, along with a thorough review of the latest findings and their limitations, regarding the placement of datacenters, base stations, cache and edge servers and an overview of the most used methods to solve placement problems.

2.1 Edge Computing

As stated in previous sections, EC brings resources to the edge of the network, where the "edge" can be defined as an arbitrary location along the path between the service request or data source and the service or data processing host [33, 34]. The general aim with EC is to reduce the physical and logical distance between the service path endpoints. The advantages of moving the cloud or more precisely, extending it to the edge, are indisputable in 5G ultra-low latency and real-time scenarios, just to mention a few.

On the other hand, VNFs are to be placed within the service infrastructure deployed by EC at the users' vicinity. Is at this point where EC and NFV converge to ensure 5G services feasibility

and performance. To detail the big picture: EC focuses on the placement of physical infrastructure or resources near the end users, while NFV mainly deals with the service deployment through the placement of VNFs over EC hardware.

Albeit the simplicity behind the EC definition, when revising existing literature it is easy to be overwhelmed by confusion and, in some cases, contradicting information. From [33], [34] and [35], EC could be seen as a paradigm including Fog Computing (FC), MEC and cloudlet computing as its implementations or even a separate technology coexisting with these technologies. The OpenFog consortium has stated that FC is a system-level architecture for services across networks and between devices that reside at the "*edge*", while EC is limited to place servers, applications or small clouds at the user premises [36, 37]. What is more, for the OpenFog Consortium EC runs primarily in isolated silos while FC has extensive peer-to-peer interconnection capabilities between nodes.

To clarify and ease the comprehension of these concepts, this research assumes the classification proposed in [34] and [35]. Consequently, EC is assumed to be defined by the following characteristics:

- **Node infrastructure:** microdatacenter-like infrastructure (i.e., Datacenter in a Box, hyper-convergent micro-datacenter) providing storage, computing and networking resources at the network edge.
- **Proximity:** deployed in the users' premises, commonly within one network hop from the traffic aggregation point (i.e., RAN node, Wi-Fi access point, etc.), although further placement at multiple hops is supported (depending on the use case requirements).
- **Access technology:** an edge node is assumed to be commonly connected to the traffic aggregation points through the backhaul mobile network or the Internet Service Providers (ISP) access network, using any physical interconnection technology or network architecture available. No limitation in this regard is enforced, thus opposing rigid access method configurations for edge technologies presented in some studies [35, 38].
- **Computation offloading model:** EC supports both isolated and cooperative task execution architectures. Therefore, a given user requesting a service or information from an

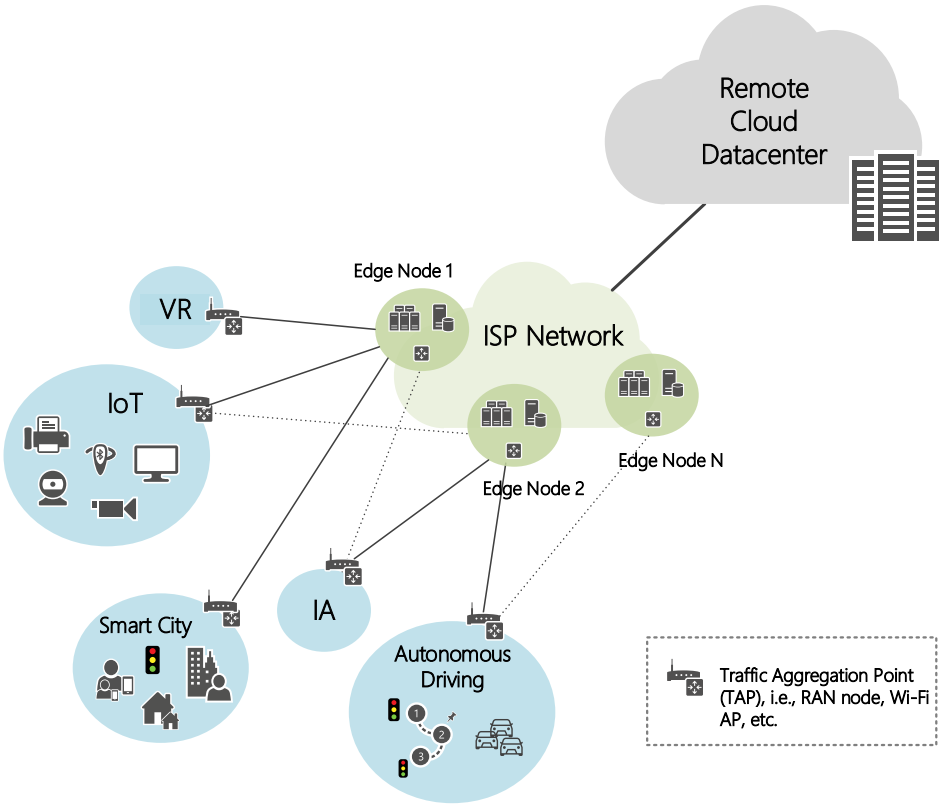


Figure 2.1: Edge Computing reference architecture.

arbitrary VNF or chain of VNFs can be served by one or more edge nodes.

- **Architecture:** the communication and computing model from any arbitrary pair user-service host is considered hierarchical because a given task can be offloaded to an edge node and further offloaded to a remote datacenter, thus implying a tiered architecture from the data source to the actual processing host. However, the EC network can be considered to be flat, as a sole tier of edge nodes is assumed to be placed between the traffic aggregation points and the remote cloud infrastructure. Figure 2.1) depicts a simplified reference architecture for EC where each EN is assumed to converge storage, networking and computing resources.

The following sections provide a comprehensive overview of the most common EC related technologies and implementations.

2.1.1 Mobile/Multi-access Edge Computing

MEC was defined by the European European Telecommunications Standard Institute (ETSI) in 2014. Mainly, it was born as a platform to provide computing capabilities within the RAN. Therefore, when compared to FC, a key difference is that the "*edge*" is precisely defined as the RAN site, while in the case of fog nodes, the "*edge*" could be located anywhere on the user premises and additionally involves shared tasks by leased resources from the end-user devices [39–44]. In September 2016, the ETSI's Mobile Edge Computing group changed the name of this technology to Multi-access Edge Computing after realizing that the benefits of this paradigm reached beyond mobile networking, into Wi-Fi and fixed access technologies.

MEC core aim is to reduce network congestion and improve application performance by executing task processing closer to the user. Furthermore, it is designed to improve content and application delivery. Several use cases can profit from this technology: Augmented and Virtual Reality, which benefit from ultra-low latency communications; connected cars, which also thrive in high-bandwidth, low-latency, highly available settings; IoT applications that rely on high performance and smart utilization of network resources [39].

MEC nodes can be implemented both indoors and outdoors depending on the access technology. With respect to the outdoors, macro cells place computing and virtualization capabilities into radio network elements. For indoor deployments, such as Wi-Fi and 3G/4G access points, edge clouds can serve as gateways, running specific regional services. Examples of the latter are Machine-to-Machine (M2M) ecosystems where MEC services can monitor weather conditions and crowded areas (e.g., airports, where MEC applications can be used for passengers guidance).

2.1.2 Cloudlet Computing

Cloudlets are conceptually similar to MEC as they can be seen as small datacenters with Common-Of-The-Shelf (COTS) infrastructure located at the network edge (a cloudlet could be particularly defined as a Datacenter in a Box). The difference among them is that from a cloudlet point of view the "*edge*" is just the logical end-user proximity, and not a well stated frontier (i.e., the RAN node) as in MEC [35].

In general, cloudlets are said to have four main attributes: small, low-cost, maintenance-free

appliance design, based on standard cloud technology; powerful, well-connected, and secure; maintains only soft state (built for micro-services and containers); and located at the edge of the network, close to the intelligent devices it will communicate with.

For the purpose of this research, Cloudlets are the closest conceptual "*black box*" we are referring to when talking about an EN. The core difference is that an EN conceptually extends the cloudlet computing base idea by allowing the coexistence of proprietary and commodity hardware in a wide variety of service operation conditions and approaches while extending the execution scheme through a collaborative approach as proposed by FC.

2.1.3 Fog Computing

From [45], FC can be defined as a scenario where computing tasks are heavily decentralized and performed by end devices, ENs and the cloud in a cooperative way. In particular, and this is one of the main differences with other EC implementations, the assigned tasks can be not only executed by ENs or cloud servers, but using resources leased by the end devices.

In a more formal way, FC could be considered a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from remote cloud to "*things*". Basically, it supports multiple industry verticals and application domains, delivering intelligence and services to users and business. In addition, it enables services and applications to be distributed closer to the end devices, and anywhere along the path between cloud and end users (or "*things*", when referring to smart devices).

Overall, FC extends from the end devices, over the network edge, through the cloud, and across multiple protocol layers. [36–38, 45, 46].

2.1.4 Other related technologies

There are other ongoing initiatives such as Central Office Re-architected as a Datacenter (CORD). Although CORD could be placed under the EC technological umbrella, it is a complete open-source service platform combining commodity servers, white-box switches, and segregated access technologies to provide an extensible service delivery platform . Basically, its purpose is to redesign the CO concept into an edge-based platform capable of allowing residential, mobile, and

enterprise customers to configure and manage their service packages rather easily and in almost real time. Consequently CORD, does not fall into the scope of this research. However, CORD sites are considered as potential EN locations within our ENPP solution proposal presented in the following chapters.

Other EC closely related technologies/concepts are Mist Computing, Mobile ad hoc Cloud Computing, etc [33, 35]. Nevertheless, none of them brings relevant conceptual elements to the discussion of the EC infrastructure placement problem. Therefore, no further information about them is provided in the remaining of this document.

In conclusion, FC, MEC and Cloudlets have their similarities and differences but they all converge in a decentralized architecture of distributed IT capabilities. This pose the question of where to efficiently place the required computing resources.

2.1.5 Edge Node: the definition

Due to the remarkable conceptual proximity among EC implementations, a remaining open question is whether there can be a clear and well-accepted definition of the functional and conceptual base entity of EC: the Edge Node.

Generally speaking, an EN can be defined as "*the facility or infrastructure entity placed at the users' premises providing computing, storage and networking resources for service execution purposes*". However, such concept does not clarify the "*edge*" boundaries along the service path, nor provides further details about the EN specifications and operation models. What is certain is that a formal EN definition must represent the operational and functional nature of all EC implementations. This way, from a developer's perspective, a given service could be running either on a Fog Node (FN), a Cloudlet or a MEC server in an isolated or cooperative manner. While from a service provider perspective, such service would span across a set of ENs with different characteristics, capacities and functionalities.

The wide range of features and broad operation scope to be inherited -e.g., from FC, cloudlet computing and MEC- increase the complexity of a generic EN definition. Attempting a flexible, yet thorough, EN definition can only be accomplished considering the broadest deployment scenario where all EC implementations inter-operate along the user-to-cloud service path. In this

context, the following operating schemes may occur regarding the task execution model for an EC/Cloud ecosystem:

1. **Single-node processing**: ENs hosting the requested service(s) collect and individually pre-process the input request(s) from the end devices. If further processing is required, a request is sent to the upper layer (i.e., the remote cloud datacenter). Otherwise, the response is sent back to the end device.

2. **Multi-node processing**: multiple ENs collect and *cooperatively* pre-process the client request(s). If further processing is required, a request is sent to the cloud. Otherwise, the response is sent to the request source.

3. **EN-End device cooperative processing**: the end devices lease computing, storage and networking resources in order to participate in the collaboration scheme to process any given task/request. The service execution is carried out *cooperatively* among ENs and the end devices. The task offloading to the cloud is expected to occur with low probability.

These operation models are not exclusive and others may be already defined [33, 47]. From the first two operating schemes the EN definition is straightforward as the idea of an "*infrastructure entity*" is clearly defined as an arbitrary location between the Traffic Aggregation Points (TAPs) and the remote cloud. However, the last case is rather complex. In such scenario the EN logically comprises the infrastructure placed at the edge and the resources leased from the end devices to execute a given task, function or service chain.

Taking this into account and considering that there is still no consensus on what or where is the "*edge*", an EN in the context of this work is defined as follows:

> **Definition 1.** *Infrastructure entity bringing computing, storage and networking resources to the network edge. It ensures both isolated and cooperative execution capabilities for services and applications (in the past exclusively hosted in the remote cloud). Any Edge Node may comprise infrastructure in one or more physical locations according to the service and application executing scheme, although viewed as a single device from a management layer perspective."*

In this definition, the "*edge*" is assumed to be the logical service path excluding the remote cloud and geographically located at the users' vicinity (e.g., according to a given latency requirement).

Nevertheless, since this work is focused on the placement of the physical infrastructure entity defined above, an EN is hereinafter assumed to be the hardware infrastructure -i.e., an isolated silo with computing, storage and networking resources in a microdatacenter-like solution- to be placed in order to satisfy certain pre-defined service requirements.

2.2 Placement Problems

At first glance, the ENPP described in Chapter 1 can be seen as, for instance, a traditional FLP or server placement problem such as those found in [15–18][19]. Nevertheless, there are some important differences that stand out after a detailed analysis of these problem types in the 5G context, since the main goal in FLPs is to select the best facility locations (among a set of initial potential sites) to achieve costs minimization and customer demand satisfaction [20–23, 48, 49].

When revising the literature, not only FLPs can be linked to the ENPP. There are several studies available regarding service infrastructure placement for specific use cases: mobile base stations and cache-enabled nodes for Content Delivery Networks (CDNs) are two examples. In the following sections, these use cases are analyzed and the similarities and differences with the ENPP are pointed out.

2.2.1 Facility Location Problems

FLPs deal with the placement of a facility or set of facilities (often from a list of feasible locations) to best meet the use case constraints and requirements. These problems have been thoroughly studied due to their utility when planning the placement of public service facilities such as hospitals, fire fighter stations or commercial facilities such as warehouses. In [23], [20] and [50] comprehensive surveys about this topic are presented.

Traditional FLP formulations and solutions cannot be directly applied to the ENPP because of the following reasons:

- *Non-convergence into a particular problem type:* As seen in [21], FLPs are mostly formulated following the guidelines of a specific operational research family: Weber, median, covering, constrained, uncapacitated, location–allocation, location-routing, dynamic, competitive, network and undesirable location problems. Therefore, although multi-criteria FLPs have been already tackled, they all converge into a particular problem type such as coverage or Weber, while the ENPP is mostly a mixture among a variety of such problems. In the particular case of interest in this research, the ENPP converges the capacitated, networked and constrained FLPs with coverage restrictions, dynamic placement requirements, and even a certain location-routing focus.
- *Cost function complexity:* The cost function on the ENPP is far more complex to obtain than those of traditional FLPs, given the number of contradicting and dynamic trade-offs involved. First, reducing the costs forces a reduction in the number of nodes to deploy, but this entails a conflict with the strict requirements in place. In addition, CAPEX and OPEX are linked to the node capacity, which has a negative relationship with the minimization of the nodes number. Meanwhile, implicit non-specific requirements such as deployment flexibility, pose additional challenges to be considered. Overall, the ENPP can be considered as a multi-objective optimization problem in nature, going far beyond the revisited FLP formulations.
- *Non-technical restrictions:* common FLP formulations do not deal with non-technical restrictions since they start from a given list of pre-selected feasible sites. Nevertheless, the need to include non-technical restrictions on the site selection is mandatory for the ENPP mostly due to the high density of nodes to be deployed over a relatively small area in comparison to facility location density in FLPs, where the length of the sites set is significantly smaller.

What is more, a core difference between FLPs and the ENPP is that the former commonly considers user demands and transportation distances or costs as main elements in order to carry out the optimization process. However, for the ENPP the optimal node placement is tightly coupled to the 5G requirements, mobile/fixed network traffic model, location-dependent rentals,

costs of interconnecting data sources and service hosts and the technical and non-technical restrictions inherent to any placement location.

Considering [21], [50] and the aforementioned elements, the ENPP can be classified as a Multi-criteria Multi-attribute FLP under the specific 5G operational environment. This classification attempts to overcome current modeling limitations overlooking the complexities derived from: traffic dynamics and variations expected in future 5G networks, convergence of several operators in a presumably unique sliced/virtualized infrastructure, challenging cost-effective resource sizing and broad range of interconnecting technologies with a direct impact on the CAPEX/OPEX and user demand satisfaction.

2.2.2 Datacenter placement

Excellent background knowledge to solve the ENPP can be obtained from the guidelines to place small, large and mid-range datacenters. However, the actual key steps followed by companies like Google, Facebook and Amazon to place their datacenters remain confidential.

To the best of our knowledge, [24] and [28] are two of the few publicly available papers referred to the datacenter placement optimization. On the former, the authors formulate the problem as a linear programming model seeking to minimize the costs of the entire datacenters network over a given geographical area (a set of potential locations was given as input) in order to satisfy known demands. As a particularity of their method, they assume as inputs the maximum number of servers to deploy and the user per server ratio. As solving strategy, the authors proposed a set of heuristics: Simulated Annealing and Optimized Simulated Annealing, in combination with linear programming, with promising results in cost savings.

This research allowed us to thoroughly comprehend most of the key physical aspects of service infrastructure placement such as energy consumption, build and land costs. However, its limitations when directly applied to our particular scenario include the exponential number of nodes to deploy in a small to medium-sized area (when compared to the article baseline number of datacenters), communication restrictions between users and services as result of the latency constraints forcing the formulation of a "*coverage*" problem and the absence of a proper demand characterization in order to achieve real-life optimization.

On [28], the main goal is to obtain a placement baseline for all the components of a fog network based on micro datacenters and a Long-Reach Passive Optical Network (LRPON). In this case, the limited scope of the formulation is a first restriction for the ENPP solution, as the interconnecting methods under 5G are expected to be significantly diverse, while in this case the whole formulation depends on the particularities of the LRPON and its components.

2.2.3 Base Station placement

Mobile networks have been around for quite a long time and thus, mobile network planning and specifically base station placement have been extensively addressed [15, 16, 51–53]. Although for base station site selection, coverage, capacity and costs are the main concerns, exhaustive research have been carried out in order to characterize user traffic patterns, demand geo-distribution and other topics closely related to the ENPP.

A budget constrained method is presented in [16], where authors model the Heterogeneous Network (HetNet) small cell deployment problem seeking to contain the overall costs to a given limit value, while considering other requirements (transmission power limitation and rate requirements of users). Authors in [15] propose a novel method for mobile network planning considering a scenario based on HetNets, which is envisioned to be part of the 5G reality. The presented idea is to firstly deploy a set of macro-cells whose coverage area is estimated based on a simplification of the underlying demand modeled as Traffic Demand Points (TDPs). This partition is done by a tessellation of the geographical area ensuring a fair distribution of the demand and complete TDPs coverage. Secondly, by checking the workload data on the existing base stations, the planning algorithm is able to propose the addition of new low-power cells in order to return to the demand distribution status-quo and guarantee customers QoS.

In [52] the main target is to find the optimal locations to deploy temporal base stations to cope with the special characteristics of emergency situations. A very simplified scenario was used, in which the interest territory is divided into squared areas with a fixed arbitrary demand value. The strategy followed was to greedily place base stations until no further areas remain uncovered and then adjust the base station initial positions using an evolutionary algorithm to maximize their capacity usage and thus lower the overall costs. The approach followed in this

paper considers a HetNet scenario where high-power and low-power base stations coexist to cover existing demand.

In both [15] and [52], the customer demand assumed as input is over simplified. As a result, they lack the flexibility to deal with highly dynamic scenarios as those found under future 5G, where the demand continuously changes hindering the placement optimization process. In contrast, Zhou et al. dive into the relation between base station deployment locations and the traffic characteristics in cellular networks [51]. They present a deep mathematical study of the relation between the base station locations and a large database of collected data from operational mobile networks. The core finding is that the homogeneous Poison Point Process (PPP) can only be accurately used over small areas to accurately model traffic density, while inhomogeneous PPP can be used for this purpose regardless of the area size. The data collected about peak-hour traffic densities, was then used to propose a useful framework for the base station operation optimization. This research can be used as baseline for more efficient base station placement algorithms and for the ENPP solution.

2.2.4 Generic Server placement

Further investigation in the field of mobile networking has attempted to optimize the location selection of the remaining network components. An example can be found in [53], where new metrics are proposed for the placement of the Serving Gateways and Packet Data Network Gateways. In summary, the proposal adds new metrics such as the end-to-end connection and service/application types to the process of selecting the most suitable data anchor gateway for a given host-to-host communication.

Under the MEC paradigm umbrella, Enhanced Small Cellss (SCeNBs) and other concepts and platforms such as the proposed in [54–56], significantly differ in their deployment location considerations. While some solutions (Small Cell Clouds and Mobile Cell Clouds) assume to place the computation capacities within the RAN sites, others maintain the approach of a further away location of the resources at centralized datacenters but introducing new components and inter-working procedures to ensure better performance.

2.2.5 Edge Server placement

Very few articles are available about EC infrastructure placement, most likely due to the youth of the related technologies and the lack of operational deployments. Furthermore, the papers found throughout this research mainly cover quite specific scenarios. Thus, a broad view of the problem with a more general solution method remains an open question.

Yannuzzi et al. analyze the placement of fog nodes in the specific context of a city like Barcelona [57]. The pursued goal is to cope with the requirements of smart cities by deploying FNs to satisfy broadly distributed use case scenarios such as event-based video and traffic management. The general architecture and the components of the FN are explained, although the placement strategy just suggests the use of available street cabinets in order to somehow reduce costs. Furthermore, the QoS-aware placement of FC nodes is solved in [58] based on the "*k-means*" algorithm (i.e., as detailed in [59]) to find the best network gateways to place the fog nodes such that the overall latency is minimized. The core limitations of [58] include the rigid uncapacitated formulation and the assumption that each node transmit data to only one fog node, thus reducing the applicability of the solution to real-life scenarios.

IoT is another subject closely related to the ENPP, since it has become a core paradigm driving 5G networking development [60]. The article in [18] is based on the premise that gateways for an IoT network are far more expensive than IoT smart devices, and as a result their placement optimization can help minimize CAPEX. In addition, the OPEX is reduced by minimizing the number of required gateways while satisfying predefined QoS demands. The problem is formulated as Integer Linear Programming (ILP) and the placement is based on the selection of feasible locations among the set of Voronoi vertices and facility locations.

The study in [27] presents a framework for the placement of edge servers. A novel approach is used to discover and evaluate unforeseen suitable sites by analyzing user behavior and by assuming that users are close enough to edge service facilities in real-life scenarios. To simplify the problem and find a feasible solution, the set of users is also assumed to be somehow clustered and edge sites are proposed as near to the optimal locations for each cluster as possible. Capacity provisioning is addressed through an ILP formulation aiming at cost minimization. User demand variation is taken into consideration in the framework design by preparing the system to cope

with the worst case scenarios, given a baseline data about demand patterns and user distribution.

The paper in [27] presents some limitations that should be addressed for the context specified in this thesis. First, user clustering should be done based on the use cases and typical demands of 5G networking, including both fixed and mobile baseline data, along with an analysis of demand geo-distribution characterization. A main issue here is the ultra-high density of TAPs given the presumable deployment of ultra-dense small cells over any area size, coupled with a tight interlacing between different demand requirements (due to the broad mixture of use case scenarios). Secondly, capacity provisioning should account for a more comprehensive set of metrics including but not limited to: number of users, application-based requirements, high reliability and availability margins, ultra-high bandwidth requirements. Additionally, the placement strategy should consider cost-related issues to reach its optimal solution, for instance, location-dependent costs and energy consumption.

Authors in [13] present two core problems: 1) the minimization of the number of access points co-located with an arbitrary edge server to guarantee customer demand satisfaction and, 2) the efficient task assignment to the edge servers. To solve the proposed problems, the authors divide the Wireless Metropolitan Area Network (MAN) into clusters, where the cluster heads are co-located with edge servers and all cluster members offload the tasks to its corresponding cluster head. Graph theory is then used to transform the problem into the minimum dominating set problem and a solution based on a greedy and Simulated Annealing (SA) algorithms is developed to find the near-optimal solutions. When compared to the ENPP described in Chapter 1, the research in [13] is limited by the translation of the delay constraint to simplistic Euclidean distance. Furthermore, as stated above, the clustering approach used lacks the flexibility required to deal with the EN placement under 5G requirements.

From [61], the placement of MEC servers can be studied. This paper addresses the MEC geo-clustering problem where the main goal is to optimize the MEC server placement (while balancing the overall workload) for the MEC clustered service areas. A graph-based algorithm is presented to enhance the partition of the pre-defined service areas, looking to improve the task offloading mechanisms. The limitations on this work include limiting the MEC server ability for collaborative task execution, due to the clustering approach used on the formulation and its

inability to consider 5G requirements (e.g., latency, reliability).

Similarly, a framework to solve the edge server placement within a geographical topology is showcased in [26]. As in [61], this work uses a clustering approach in order to simplify the overall problem complexity, thus incurring in the above mentioned limitations. Nonetheless, the authors consider end-user service demands, CAPEX/OPEX on edge infrastructure operators and end-user mobility patterns within the service area. However, assuming that areas with high density of Wi-Fi access points are more likely to have a user-managed edge server, this article does not take into consideration certain realities: a) the user willingness to operate and maintain the edge infrastructure, b) the edge service provider service cost evolution that could lead to the full externalization of the user needs (e.g., as with the Infrastructure as a Service (IaaS) service model), c) the real savings for the users in owning an edge server instead of leasing the resources. Finally, only the existing base stations are considered in this study as potential edge server locations.

In [25] the edge server placement problem is tackled for mobile edge computing environments in future smart cities. The novelty of this study lies in the multi-objective optimization model, aiming at both delay minimization and overall workload balance. This work assumes to know in advance the number of edge servers to be placed and uses the distance to estimate the network delay, thus limiting the applicability of the results to 5G ultra-dense networking and delay sensitive use cases.

The NFV middlebox placement is optimized in [62], aiming at ensuring optimal network performance based on the efficient route of service flows and the effective placement of the processing middleboxes. This article is heavily limited when considered for the ENPP, since it assumes that every path for any arbitrary pair of endpoints is known beforehand, thus restricting the model's ability to model the realistic interactions between service requests and current network capacities. The authors in [63] aim at optimizing the number of nodes in a fog network, with a strong focus on the optimization and allocation of the wireless parameters. This study does not consider the backhaul network capacities nor the possibility to place the edge infrastructure at both existing IT-capable sites and RAN locations. Additionally, the authors addressed a single objective ENPP variant for a scenario-specific hierarchical fog network.

What is more, none of the available studies found on edge server placement addressed the network-aware ENPP, where the placement of the ENs is made considering the underlying network capacity and utilization ratio. Since a massive surge in data processing and bandwidth usage is envisioned under 5G networks, a network-aware strategy becomes mandatory to satisfy the required Key Performance Indicators (KPIs) and avoid performance degradation in the long run.

2.3 Placement optimization methods

Most of the placement problems mentioned above are considered part of the NP-hard problem set [48, 49, 64–66]. The ENPP to be solved within this research, being a Multi-criteria Multi-attribute FLP, basically combining several FLPs problem characteristics, could be deduced to be NP-hard.

In summary, the ENPP implies the analysis of all possible EN-TG combinations and network paths in order to find the minimum cost solution. What is more, the latency constraints and the need to satisfy all TG demands in a capacity-dependent cost model, imply that the combinations cannot be splited to reduce computation time. Furthermore, a simplified variant of the ENPP (i.e., a network-agnostic formulation) has been already proven to be NP-hard in [25]. The need to consider the underlying network topology and capacity for the ENs deployment, significantly increases the number of feasible solutions, adding an extra layer of complexity in terms of execution time and computing resources.

On the other hand, there is still no working knowledge and operational data regarding 5G user behavior, future traffic patterns and service trends in an EC, NFV, 5G ecosystem. Therefore, predicting the number of ENs for a given service area is a nearly impossible task. What is certain, is that ultra-dense networking and 5G stringent requirements will push the amount of ENs to thousands in just a city. Although a MILP formulation makes a variant of the ENPP solvable by exact methods (see Chapter 4 and Chapter 5), for the 5G scalability requirements and a network-aware ENPP model the problem difficulty increases abruptly. Taking this into consideration, heuristic or meta-heuristic methods have to be used as placement solutions.

Since several heuristic-based strategies have been developed solving various placement

problems [20–24, 27, 49, 50], the most used are briefly described in the following sections, along with the main issues regarding their application to the ENPP solution.

2.3.1 Simulated Annealing

SA has been effectively used to solve FLPs [49, 67, 68]. Overall, SA has been used to solve FLPs based on its flexibility to solve combinatorial problems when compared to other solutions such as the Lagrangian method and Branch and Bound algorithms. In addition, SA has been already tested and compared to other heuristics when solving FLPs, showing excellent results in both performance and solution quality when compared to best known or heuristic-generated values [24, 49, 67, 69, 70].

Overall, SA offers a flexible strategy to cope with the ENPP without incurring in complex implementations. However, due to its simplicity, its basic procedure could not meet the practical requirements of our particular context. Thus, further improvements and analysis steps are mandatory, involving multi-objective environment consideration in search for Pareto-optimal fronts.

2.3.2 Tabu Search

Tabu Search is a meta-heuristic that guides a local heuristic search procedure. One of its main components is the use of adaptive memory, which creates a more flexible search behavior allowing the algorithm to search the solution space beyond local optimality by relaxing "*tabu constraints*" and visiting unexplored solutions. To achieve such behavior, Tabu Search implements the following procedures: aspiration, diversification and intensification [71]. Since local choices are guided by information collected during the search, Tabu Search contrasts with memoryless designs such as SA, that heavily rely on random or semi-random processes implementing the sampling steps.

The main challenge when adapting the Tabu Search to solve the ENPP is the solution generation and the adaptive memory/tabu lists usage. Since the number of nodes is quite large, the convergence of Tabu Search based on upfront feasibility calculations poses a complex challenge.

2.3.3 Evolutionary Algorithms

Evolutionary Algorithms (EAs) consist of several heuristics commonly employed to solve optimization tasks by imitating natural evolution [71]. EAs typically use different levels of abstraction, working on whole populations of possible solutions for a given task. The core idea is to apply combinatorial processes such as crossover and mutation to the initial search space (initial population), in order to find a near-optimal final solution through "*evolution*".

A main benefit of EAs is their ability to cope with multi-criteria and multi-objective problems in a fairly non-complex implementation [72]. When applied to the ENPP solution however, the problem representation becomes a critical concern. Coding the EN architecture variation, while meeting underlying requirements in a EA manner is far from trivial and could lead to non-correctness or slow convergence of the solutions.

2.3.4 Lagrangian Relaxation

Lagrangian Relaxation is basically a method of decomposition: the constraint set of the problems is separated into two groups, namely the "easy" or "bad" constraints and the "hard" or "good" constraints. The main idea is to relax the problem by removing the hard constraints and putting them into the objective function, assigned with weights (the Lagrangian multiplier) [73, 74]. Each weight represents a penalty which is added to a solution that does not satisfy the particular constraint.

In summary, the interest of the Lagrangian relaxation is that, in some cases, the optimal solution of the relaxed problem actually gives the optimal solution of the initial problem. When compared to SA and Tabu Search, Lagrangian Relaxation can be assumed to be more rigid, offering less adaptability to complex problem environments and intractable restriction sets. Namely, the restriction reduction under the ENPP is not feasible considering the number of parameters involved and use cases to satisfy. Furthermore, the multi-objective nature of the problem limits or even prohibits the constraint set splitting process.

2.4 Open Issues

Although extensive research can be found regarding topics and problems closely related to the ENPP, certain limitations restrict their use to formalize and solve the ENPP under 5G constraints [13, 25–27, 57, 58, 61, 62, 75]. These limitations can be grouped into the following categories:

1. **Limited formulation scope:** the vast majority of the placement problem models are unable to represent the underlying complexity of a 5G/EC ecosystem due to, for instance: a) unrealistic cost model overlooking the main costs affecting the EN placement (see Section 4.2), b) mathematical models not tailored to 5G requirements, namely, ultra-low latency, ultra-dense networking and ultra-high reliability, c) over-simplified delay constraints, commonly based on Euclidean distance, d) single objective formulation minimizing the cost, the number of nodes or balancing the workload, thus unable to comprehensively model and optimize the EN placement.

2. **Network-agnostic placement:** most edge server and other placement problem solutions avoid considering the underlying network within their solving methods due to the significant complexity this adds to the problem. Consequently, the applicability of the proposed solutions is not guaranteed on real-life scenarios, where the existing network capacities and capabilities must be considered to ensure use case demands satisfaction and flexible management/orchestration under 5G networking.

3. **Unrealistic assumptions:** in order to make a certain ENPP variant solvable, several studies assume that some critical data is known beforehand (e.g., the number of ENs or edge servers to be deployed) or arbitrarily selected (e.g., the network path interconnecting a given edge server and an end user/device). As a consequence, most solutions lack the flexibility to be adapted to complex placement scenarios such as those envisioned in next generation networks.

4. **Heterogeneity and rigidness:** no extensible platform or framework has been proposed, to the best of our knowledge, to solve any ENPP related problem. Therefore, the solutions proposed in the available literature cannot be extended or adapted to new scenarios and

use cases without incurring in significant development and modeling efforts. Simple tasks such as inserting new requirements or parameters into the placement strategy result in mid to long-term software modifications.

All limitations considered, the need for a flexible, network-aware, realistic ENPP model and solution strategy, tailored to the strict requirements and use cases of 5G networks is certain. Moreover, the proposed placement solution must be thoroughly evaluated considering the evolution in 5G implementations and standardization.

CHAPTER 3

ENPP PLACEMENT PARAMETERS

This chapter is based on:

- A. Santoyo-González and C. Cervelló-Pastor, "Edge Nodes Infrastructure Placement Parameters for 5G Networks," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN 2018)*, (Paris, France), p. 6, IEEE, 2018.

This chapter presents a set of placement parameters for the EN site selection, tailored to the ENPP solution. Additionally, core placement guidelines to be take into consideration are detailed at the end of the chapter.

3.1 Placement parameters

When solving the ENPP, a thorough study of convergent technologies, scalability issues, top-level and low-level architectures and inter-component synergistic are significant aspects to be considered.

When proposing a set of metrics to assess potential EN sites, certain extrapolation can be made for the ENPP from the sets of metrics presented in recent research [24, 76]. However,

Table 3.1: ENPP placement parameter categories

Parameter Category	Parameter
Demand	Latency (i.e., RTT) from an EN location to the served TAP. Throughput requirement imposed by the served TAP due to the aggregated requests of the underlying users.
Location-dependent costs	Costs directly linked to the site itself (i.e., Power line layout, Energy source, Network line layout, Interconnection capabilities, Land acquisition, Build costs).
Site capabilities/restrictions	Site deployment capabilities (e.g., IP-capable equipment) and non-technical restrictions (e.g., environmental, political and social limitations).
Reliability	Site-dependent characteristics (natural disaster exposure, site physical security) and TAP reliability requirements in terms of coverage redundancy.
Energy Consumption	Energy consumption based on the per path power analysis when interconnecting any EN-TAP pair.
Service Area Type	Area classification in terms of TAP demand and density (e.g., urban, rural).

although the placement of ENs may inherit some of these parameters, certain modifications and additions are mandatory.

Latency, for instance, should not account for the delay caused by the wireless access layer of mobile networks, as such value does not depend on the location of the service hosts (see Section 3.1.1). Similarly, latency constraints should be fixed in a per service or use case fashion, along with fault tolerance and availability, allowing certain locations to be more suitable for an arbitrary set of use cases than others. Staff expenses should consider the ultra-high automation levels expected in 5G and the multi-operator infrastructure management, along with recent and future advances in management and orchestration platforms, in order to accurately determine the related costs. In addition, land acquisition costs and other capital expenses should include VNF hosting capabilities on available nearby locations or in-use IT-capable sites such as CDN-PoP, ISP-PoP and Central Offices.

On the following subsections the proposed set of parameters for the EN site selection is presented. A summary of these findings can be consulted on Table 3.1.

3.1.1 Traffic Demands

When selecting a location to place an EN the traffic demand to be satisfied is crucial. In this research we consider traffic demands to be mainly formed by **latency** and **bandwidth** requirements, since a core goal of our solution is to answer the 5G requirements for ultra-low latency and ultra-high bandwidth scenarios.

Latency has been widely studied in the context of mobile networks and 5G use cases [7, 77]. However, under the ENPP, latency control entails certain particularities and complexities that must be addressed. The first challenge is to define the delay values that can be reduced through the EN placement optimization. Figure 3.1 showcases the various deployment scenarios and delays involved when considering a communication channel from a mobile user to a service hosted in an EN, with the top level depicting a "traditional" service path in this context. Following the work in [78], the total unidirectional transmission time of a 5G system depends on:

- L_{radio}: the radio layer packet delay, it occurs between the base station and user equipment (it includes the Transmission Time Interval which must be less than 1 ms under 5G, propagation delay, signal processing time at the receiver, and re-transmission time due to packet errors).
- $L_{Fronthaul}$: the delay between the base station front-end and the centralized Baseband Unit (BBU), if applicable.
- $L_{Backhaul}$: also called backhaul delay, it is the time taken to traverse the core network entities and gateways.
- L_{Core}: core network processing time.
- $L_{Transport}$: communication delay between the core network and the cloud/edge service host.

For EC, the latency optimization is to be carried out from the TAPs. Therefore, the EN site selection optimization can improve the RAN-to-EN delay (calculated as $L_{Fronthaul} + L_{Backhaul} + L_{Transport}$) for mobile networking and the TAP-to-EN delay for other network architectures[1]. As a

[1]From this point onwards, TAPs are assumed to include RAN nodes

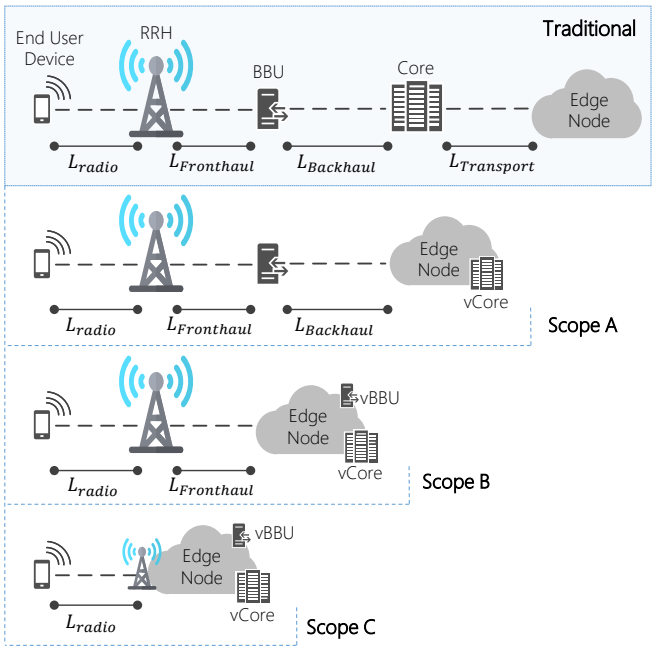


Figure 3.1: EN deployment scenarios for latency optimization.

result, considering the evolution of the mobile network core towards 5G presented in [79], namely a fully merged NFV/SDN architecture, three main scenarios could be expected (see Fig. 3.1):

- **Scope A:** The service hosts or the virtualized mobile core components are deployed in a distributed manner at the EN set. Within this scope, from a functional point of view, the User-EN communication could even occur without involving any core network entities, thus mostly excluding current L_{Core} delays. This way, when selecting a site to deploy an EN and considering a management and orchestration framework able to efficiently route traffic to the nearest core component through SDN-based mechanisms, $L_{Transport}$ becomes negligible. Consequently, the delay suitable for optimization through the EN placement strategy accounts for the sum of $L_{Fronthaul}$ and $L_{Backhaul}$. This means that only those EN locations where $(L_{Fronthaul} + L_{Backhaul}) \leq L_{max}$ can be selected as EN sites (where L_{max} is the maximum delay allowed between any User-EN pair including the related processing delays).
- **Scope B:** The edge infrastructure comprises the service hosts, core network components and the BBUs presumably as VNFs. Therefore, both $L_{Backhaul}$ and $L_{Transport}$ are mini-

mized and only $L_{Fronthaul}$ can be optimized through an efficient and accurate EN location selection strategy. Similarly to Scope A, the service path in Scope B could exclude any mobile core entities from being involved (as the User-EN communication may not necessarily involve the network core), and the core processing delays could be avoided or reduced. Additionally, RAN processing delays under this scenario may be minimized through the hosting of BBUs within the edge infrastructure.

- **Scope C:** Each RAN site is allocated computing, storage and networking capacities (it is upgraded to EN). As a pure co-location strategy is followed, no optimization is achieved by solving the ENPP (i.e., more cost-effective potential sites, such as Central Offices, are not considered).

At first glance, **Scope C** offers the best deployment solution as it maximizes the latency reduction. However, this deployment scheme is not feasible due to scalability and cost related issues. As the number of TAPs will significantly grow in 5G networks due to ultra-dense networking, the CAPEX and OPEX for upgrading each aggregation site to EN make this approach unfeasible. Furthermore, non-mobile service requests would not benefit from such placing strategy and thus 5G use case demands could not be entirely satisfied. In the case of **Scope A**, the limitation comes from not optimizing $L_{Backhaul}$ which is critical in order to achieve round-trip latency values under 1 ms. Moreover, the scenario in **Scope A** does not follow current 5G deployment advances and trends where the core components and the virtual BBUs coexist under the virtualized edge infrastructure. Taking these elements into account, **Scope B** can be assumed to be the most cost-effective EN deployment scheme, regarding latency optimization.

Overall, when placing an EN under 5G latency constraints, the maximum allowable delay between any aggregation point and its serving EN is of critical importance. Assuming a $L_{max} = 1\ ms$ threshold for delay-sensitive use cases and $L_{radio} = 0.5\ ms$ [78], the ENPP solution should ensure $L_{max} = L_{radio} + L_{Fronthaul} + L_{processing} \leq 0.5\ ms$, considering $L_{processing}$ to be known in advance or accurately predicted (e.g., using machine learning techniques). This latency constraint is quite challenging [80]. However, a joint effort mixing radio-communication advances, EC placement optimization, service and management layer efficiency and other cutting-

edge technologies may result in turning such latency value a reality. In fact, recent research has set a promising starting point towards achieving this goal [25].

Further latency complexities are introduced by a federation of edge and centralized cloud platforms under a hierarchical arrangement, where the operations with a local scope are handled by edge platforms while broader decisions are centralized. Such architecture can be seen as an extension of the traditional cloud, allowing flexibility in service deployment and mobility, by enabling an elastic combination of different resources across separate platforms for particular application types. This deployment requires an orchestration system to manage, control and configure the corresponding services across the set of cloud platforms.

The capacity of an EN[2] directly depends on the traffic density. Such metric is tightly coupled with the 5G strict bandwidth requirements and expected ultra-dense device geo-distribution. In order to effectively consider traffic density for site selection purposes, the network **throughput** demanded by the TAPs must be considered as a placement parameter. One of the key ENPP trade-offs rises from the interrelation between throughput and EN sizing. In principle, allocating as much demand as possible to each EN is desirable. Following this approach, commonly used base station placement strategies and tessellation mechanisms become suitable solutions [15, 81].

However, latency restrictions could then lead to unmet requirements, performance issues and overlooked location-dependent costs. Furthermore, since ENs are envisioned as small-sized COTS infrastructure nodes, as the capacity demand over an EN rises, its CAPEX/OPEX increases. In fact, EN expenses do not follow the traditional data center cost patterns for this reason [24, 82]. As a result, maximizing performance and coverage through a higher number of ENs is desirable in terms of overall expenses, rather than condensing the throughput demand into fewer high-capacity nodes. This reasoning is also supported by the automation levels expected under 5G, as less complex and capacitated ENs will reduce CAPEX/OPEX by being highly automated, self-aware and remotely managed/maintained if needed. Nevertheless, given the trade-off regarding EN number, capacity allocation and throughput requirements, only multi-objective/multi-criteria optimization mechanisms can be used for this particular ecosystem.

[2]Computing, storage and networking resources available for service execution

3.1.2 Location-dependent costs

The list of location-dependent costs is significantly extensive, as they go from energy prices and land acquisition to installation expenses due to distance between the closest suitable power or networking source. Throughout this research, the following elements where identified regarding the site selection for an EN:

- Power line layout
- Network line layout
- Energy source
- Land acquisition
- Build costs
- Interconnection capabilities

Table 3.2 describe each location-dependent cost. The **power line layout** parameter accounts for the costs of bringing power to the EN site if needed. Similarly, **network line layout** refers to the cost of bringing networking. These two parameters entail significant cost reductions when selecting locations close to a power source or a network PoP.

In terms of energy, a self-sustainable location or a green-powered one (ecological energy sources in-use) is preferable. To guarantee this, the **energy source** parameter is defined. This parameter allows the placement strategy to assess each location regarding its energy capabilities. Any site with an in-use ecological power source or capable of using "*green*" energy without incurring in high extra expenses, is ranked higher than other locations with exclusive "*traditional*" energy capabilities.

On the other hand, **build costs** and **land acquisition expenses** are tied to the EN capacity. The former accounts for the cost of installing cooling and power delivery infrastructures and other support infrastructures, while the latter sums up the costs of renting or buying the required space. Commonly such expenses are computed in terms of the maximum power of the infrastructure which is basically determined by the computing resources.

Table 3.2: Location-dependent parameters

Parameter	Description
Power line layout	Cost of bringing energy to the location
Network line layout	Cost of bringing networking to the location
Energy source	Energy price according to the available power sources
Land acquisition	Cost of renting/buying the required space
Build costs	Cost of deploying the required infrastructure
Interconnection capabilities	EN-TG and EN-EN interconnecting costs

The location-routing nature of the ENPP is taken into account through assessing the **interconnection capabilities** per site. The communication path between any EN-TG and EN-EN pair is analyzed for each site in order to find those locations where less energy is consumed along the service channel and the lowest capital expenses are needed to ensure interconnection. This parameter should rank all locations according to already in-place communication infrastructure, IP-capable equipment, radio-wave communications feasibility.

3.1.3 Site capabilities/restrictions

Not all available sites are suitable for the placement of IT infrastructure. Political, social and environmental conditions should not be overlooked. If a set of potential locations is not identified and the entire geographical area is considered for EN placement, a tessellation method should be applied in order to exclude unfeasible areas and thus optimize the solution procedure. Moreover, each location capabilities should be analyzed as it could impact the costs. For example, an ISP-PoP location is expected to be far less expensive for deploying and EN than a base station site. This way, the ENPP solution method must evaluate each site according to its deployment capabilities.

3.1.4 Reliability

Service availability, tightly coupled with the architecture and system-level reliability, depends on the budget constraints and site-dependent properties (i.e., natural disaster exposure, networking PoPs available, site physical security). Intuitively, disaster-prone areas are to be avoided, but the trade-off on this matter must be carefully analyzed to avoid overpriced or unfeasible solutions. Overall, these elements can be grouped into a parameter conventionally called **site reliability**.

Table 3.3: Reliability parameters

Parameter	Description
Site reliability	Site characteristics such as natural disaster exposure and physical security
Service reliability	For some TAP and use cases, more than one EN is to be necessarily allocated in order to meet the service reliability requirements (i.e., mandatory backup ENs)

In certain scenarios such as mission critical systems or sensitive infrastructure communications, reliability must be ensured by guaranteeing the placement of additional or backup ENs within the communication range allowed by the latency constraints. Such consideration is considered in this work as a **service reliability** parameter. This placement criteria entails certain particularities as it basically refers to the user demands and not to the site itself. However, if a given set of users or demand scenario (i.e., a TAP) requires coverage from more than one EN, the placement strategy must place additional infrastructure in a different suitable location (in addition to the best location found). Such deployment would imply doubling the overall costs. Therefore, in a first step the placement mechanism must analyze the already placed ENs to check whether an existing EN can cover the "*high-reliability*" demand. If such EN is not found, the placement solution must propose an additional site.

A summary of the findings regarding the **reliability** placement parameter is presented in Table 3.3.

3.1.5 Energy Consumption

Power-consumption for the optimal placement of edge infrastructure has been poorly studied on mobile networking, namely, on 5G networking. The revisited literature mainly focuses on the access layer energy optimization -i.e., optimizing the radio resource allocation and usage [78, 83]- thus overlooking the need to somehow evaluate how the energy consumption can be considered when selecting where to place the edge servers. Additionally, the energy metrics related to the radio layer fall out of the scope of the ENPP as the EN placement is abstracted from the access layer details by the TAPs.

Datacenter placement strategies have considered energy as parameter in the past [24]. However, some core differences must be pointed out about datacenter and EN placement regarding

energy: 1) given the expected small-sized hardware capacity on ENs, the power consumption patterns and analyses on datacenters are inaccurate and cannot be followed, since they are commonly based on a power-per-rack metric and designed for large computing capacities, 2) when selecting a location for a datacenter, due to their limited number within large service areas (i.e., few cloud datacenters are placed even within a country-sized service area), the energy parameter is usually linked to the site-specific energy costs[3], whereas the energy price variation become irrelevant for the EN site selection as the EC nodes are to be scattered within areas where the energy costs are typically immutable (e.g., cities, towns) [24, 82].

In summary, a network-aware ENPP solution can take into account energy consumption as a parameter by following the principles of energy-aware routing, commonly based on link rate adaptation and sleeping mode [84, 85]. The reason is that the overall cost of the EC network is directly linked to the in-use underlying network capacities and resources. Any set of EN potential sites can be assessed and ranked considering the energy consumption on the possible network path(s) from the EN site to the TAPs it has to serve. Namely, the network paths are to be normalized and weighted in terms of energy usage based on the number of nodes, the in-use capacity on the links and the additional traffic load imposed by the EN-TAP pairing.

3.1.6 Service Area Type

Partitioning and classifying the service areas into urban and rural decreases the execution times of the proposed solving schemes while keeping accuracy, efficiency and performance. Moreover, given the significant difference among **service area type** characteristics, different placement parameters or schemes could be considered accordingly. Rural areas, for instance, are mainly prone to a co-location solution, where ENs are to be deployed in existing communication or computing facilities such as mobile macro cell sites. In contrast, the traffic density and use case mixture in urban and even suburban environments forces the ENPP solver to analyze a list of potential sites or the continuous placement space in order to propose EN optimal locations.

[3]Placing a datacenter in a given province or even a given country is analyzed based on the cost of the energy in that particular area, the energy price varies among different provinces, cities

3.2 Additional placement considerations

Capacity planning is undoubtedly a hard problem yet to be optimally solved given the numerous parameters, constraints and scenarios to be considered. For the ENPP, capacity allocation should consider, among others, the following elements: variable workload intensity and distribution (for both fixed and mobile networks), inter-tier communication, service and infrastructure involved platforms (for example FC, where some capacity could be leased from customer end devices and thus not be required on the ENs).

In spite of the lack of research found for the ENPP solution, the VNF placement problem has been exhaustively tackled and proved to be NP-hard [86–89, 89–94]. The VNF available placement methods should be fully understood and carefully considered. Among the factors taken into account for the VNF placement methods, the following directly influence the ENPP: latency, bandwidth, resource utilization and capacity. The scope of such parameters on the VNF placement problem differs when compared to the ENPP, as they are independent of the infrastructure location. Nevertheless, analogies can be made without loosing generality and by considering service chain and virtual functions placement, valuable hints on how to distribute physical resources can be defined.

3.3 Conclusion

A lack of placement criteria completeness was found from the literature revisited regarding the ENPP. For instance, to the best of our knowledge, Section 3.1.1 and Section 3.1.5 are the first step into defining the delay values and energy consumption parameters, respectively, to be considered when placing an EN. Similarly, reliability has been mainly overlooked in most placement research, although the foreseeable EN deployment density and 5G use cases pose complex requirements in this regard. Additionally, service area type is a novel parameter proposal that directly affects CAPEX/OPEX.

Overall, the EC deployment for next generation 5G networks requires innovative schemes and solutions. This chapter sets a starting point for the EN placement optimization towards a feasible 5G-EC ecosystem. By defining a potential list of parameters to solve the ENPP, the

groundwork for a cost-effective solution strategy has progressed further. Consequently, the following chapters focus on a deep understanding of the EN capacity planning requirements and placement guidelines, aiming at providing the required mathematical formulation and solution for the EN site selection problem on 5G networks.

CHAPTER 4

SINGLE-OBJECTIVE ENPP

This chapter is based on:

- A. Santoyo González and C. Cervelló Pastor, "Edge Computing Node Placement in 5G Networks: A Latency and Reliability Constrained Framework," in *2019 6th IEEE CSCloud / 2019 5th IEEE EdgeCom*, (Paris, France), pp. 183–189, IEEE, 2019.
- A. Santoyo-González and C. Cervelló-Pastor, "A Framework for Latency-constrained Edge Nodes Placement in 5G Networks," in *XXXIII URSI 2018*, (Granada, Spain), 2018.
- A. Santoyo-González and C. Cervelló-Pastor, "Latency-aware cost optimization of the service infrastructure placement in 5G networks," *Journal of Network and Computer Applications*, vol. 114, pp. 29–37, 2018.
- A. Santoyo-González and C. Cervelló-Pastor, "On the Optimal NFVI-PoP Placement for SDN-Enabled 5G Networks," in *Trends in Cyber-Physical Multi-Agent Systems. 15th PAAMS 2017*, 2017.

To build the foundations towards a realistic multi-objective ENPP formulation, this chapter dives into the single-objective formulation of the ENPP. Two mathematical

models are presented and a novel heuristic is proposed, evaluated and extended through a functional framework, to solve two Single-Objective ENPP (SO-ENPP) variants:

- Latency-constrained ENPP for pre-defined EN capacities.
- Latency and reliability-constrained ENPP for flexible EN capacities.

The first ENPP variant is analyzed in Section 4.1. It proposes a single-objective optimization MILP model where the main goal is to minimize the number of nodes deployed in an EC network. To model the computing, storage and networking capacities of the ENs, three EN sizes were defined: small, medium and large. Furthermore, this section details a novel heuristic to solve the ENPP called HSA, following the advantages of SA showcased in Section 2.3. This heuristic is evaluated against the exact model and a traditional SA implementation in a controlled simulated environment.

On the other hand, Section 4.2 thoroughly extends the problem model and heuristic solution showcased in Section 4.1. Namely, the latency-constrained MILP formulation is enhanced aiming at minimizing the deployment cost of the EN network: capacity-related costs, interconnecting expenses and fixed deployment costs. Additional parameters -i.e., reliability- are inserted into the model, adding more complexity to the problem and the HSA strategy is significantly extended through the development of a flexible framework to ensure the usability of our results.

4.1 Latency-constrained ENPP for pre-defined EN capacities

In order to model the SO-ENPP as a latency-constrained optimization problem for pre-defined EN capacities, we first assume that the service users distributed over a given area can be modeled as TGs[1] [15, 16]. Such simplification is made considering that the last-mile access infrastructure is envisioned to be wireless for most 5G usage scenarios. Thus, the aggregated cell structure composed by mobile base stations, wireless access points (macro cells, micro cells, femto cells), as depicted in Figure 4.1, is used as base entry data and these aggregation points are then defined as TGs.

[1]Hereinafter, this assumption is followed for all ENPP formulations unless stated differently

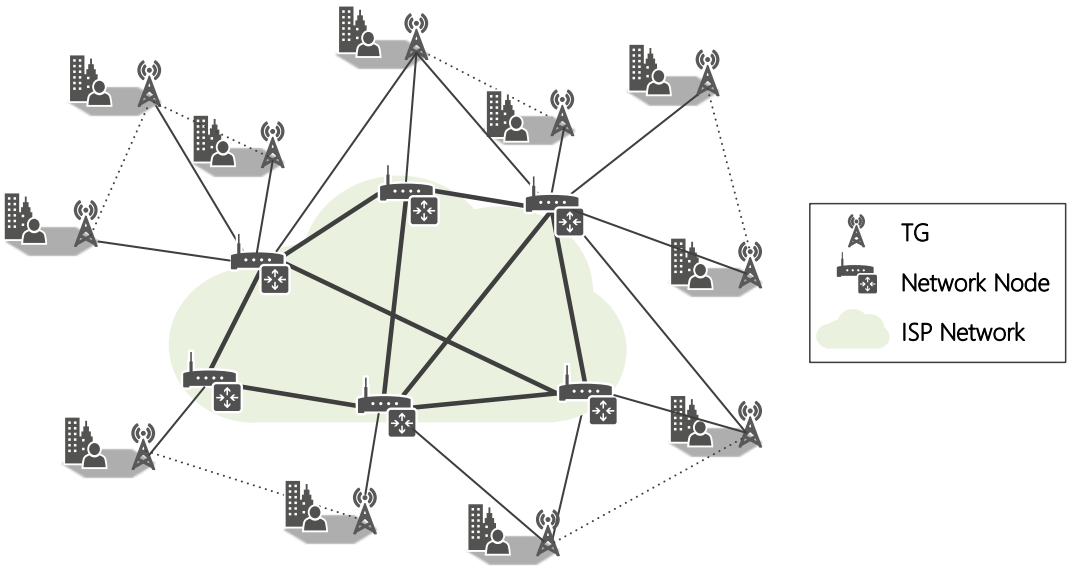


Figure 4.1: Traffic aggregation points defined as TGs

This simplification allows us to effectively model the end user demands regarding latency, reliability, throughput, without loss of generality. The reason is that the TAPs, hereinafter defined as TGs to represent their "*traffic demand point*" nature, abstract the requirements of the underlying users, acting as mandatory gateways for all data exchanged between any arbitrary end user-service host pair.

4.1.1 Problem model

The main objective of this model is to reduce costs by minimizing the number of ENs while considering a limited capacity for each EN. Therefore, the optimization problem is formulated as follows (glossary of terms in Table 4.1):

(4.1) Minimize: $\sum_{\forall e \in EN} v_e$

(4.2) s. t.: $\sum_{\forall e \in EN} u_{te} \geq 1$ $\forall t \in \text{TG}$

(4.3) $u_{te} = v_e$ if loc(t) = loc(e) $\forall t \in TG,\ \forall e \in EN$

(4.4) $u_{te} \leq v_e$ if loc(t) $\neq$ loc(e) $\forall t \in TG,\ \forall e \in EN$

Table 4.1: Glossary of symbols for the latency-constrained SO-ENPP

Symbol	Parameter	Variable	Description
M	✓		set of coordinate pairs forming the territory of interest
TG	✓		set of TG coordinates
EN	✓		set of coordinates for EN potential sites
D_{max}	✓		maximum allowed distance between a TG and its serving EN
td_t	✓		total demand of TG t
u_{te}		✓	1 if TG t is served by EN e, 0 otherwise
v_e		✓	1 if EN e is deployed, 0 otherwise
$loc(t)$ or $loc(e)$		✓	location of TG t or EN e
$c(e)$		✓	EN e capacity
d_{te}		✓	part of TG t demand served by EN e

$$\sum_{\forall e \in EN} d_{te} = td_t \qquad \forall t \in \text{TG} \tag{4.5}$$

$$d_{te} \leq td_t \cdot u_{te} \qquad \forall t \in TG, \forall e \in EN \tag{4.6}$$

$$\sum_{\forall t \in TG} d_{te} \leq c(e) \qquad \forall e \in EN \tag{4.7}$$

$$c(e) = \begin{cases} 0 & \text{if} \quad \sum_{\forall t \in TG} d_{te} = 0 \\ A & \text{if} \quad 0 < \sum_{\forall t \in TG} d_{te} < A \\ B & \text{if} \quad A \leq \sum_{\forall t \in TG} d_{te} < B \\ C & \text{if} \quad B \leq \sum_{\forall t \in TG} d_{te} \leq C \end{cases} \qquad \forall e \in EN \tag{4.8}$$

$$\text{if } u_{te} = 1 \Rightarrow \text{distance(t,e)} \leq D_{max} \qquad \forall e \in EN, \forall e \in EN \tag{4.9}$$

$$\text{if } u_{te} = 0 \Leftrightarrow d_{te} = 0 \qquad \forall t \in TG, \forall e \in EN \tag{4.10}$$

$$\text{if } v_e = 0 \Leftrightarrow \sum_{\forall t \in TG} d_{te} = 0 \qquad \forall e \in EN \tag{4.11}$$

$$u_{te}, v_e \in \{0,\ 1\} \qquad \forall t \in TG, \forall e \in EN \tag{4.12}$$

$$d_{te} \geq 0 \in \mathbb{R} \qquad \forall t \in TG, \forall e \in EN \tag{4.13}$$

The objective function in Eq. (4.1), seeks to minimize the number of ENs (i.e., v_e). The global aim is to select "good" EN locations in terms of delay, capacity and service load. Furthermore, by adjusting the EN capacity to the covered area demands, we also pursue a low-cost solution.

The first set of restrictions (4.2) specifies that any given TG t should be covered by one or

more ENs. The constraint set (4.3), refers to the case of EN e co-located at TG t position, while (4.4) ensures that no EN is placed unless there is a TG to *cover*. In (4.5), TG t demand is to be entirely covered by its serving ENs e. On the other hand, (4.6) defines the interrelation between the part of TG t demand served by EN e, in case the association between t and e exists. From (4.7), the summation of the covered TG demands under an EN, should not exceed the EN capacity, which is defined in (4.8). The linearization of the EN capacities as a piecewise constant function is detailed in Section 4.1.1.1.

To achieve latency-awareness, the parameter D_{max} is introduced in (4.9). This parameter is set as the maximum distance allowed between a TG t and its serving EN, such that a given latency value is not exceeded by the placement strategy. As a consequence, any EN location complies with the particular latency requirements imposed to the placement algorithm. The distance between any pair TG-EN was assumed to be the Euclidean distance, therefore, these implications have been linearized as shown in Section 4.1.1.2.

The set of restrictions (4.10) relates the part of the TG t demand served by EN e to the binary variable u_{te}, which determines if this relationship indeed exists. The same idea is applied on (4.11), guaranteeing that only deployed ENs cover the corresponding part of TG demand that is associated to them. Both set of constraints are linearized in Section 4.1.1.3.

Finally, (4.12) and (4.13) are variable-type or domain constraints that specify the type of values the decision variables can take.

4.1.1.1 Modeling EN capacities

The capacity of each EN is modeled as a piecewise-constant function of P pieces or sections (with $P = 4$, see Fig. 4.2), as shown in (4.8). In order to linearize such function, the binary variable δ_{ie} $\forall i \in P,\ e \in$ EN (a δ value per function section), is introduced to determine which capacity should be selected depending on the sum of the demands covered by EN e. The value of δ_{ie} is 1 at the i^{th} section and 0 otherwise. As result, the constraints (4.14)–(4.19) are added to the model, where A, B and C are the available EN capacities, being C the maximum and A the minimum value. To obtain the inequalities in (4.15) and (4.16), as required in the linearization procedures, the value ϵ is defined as an arbitrary small value.

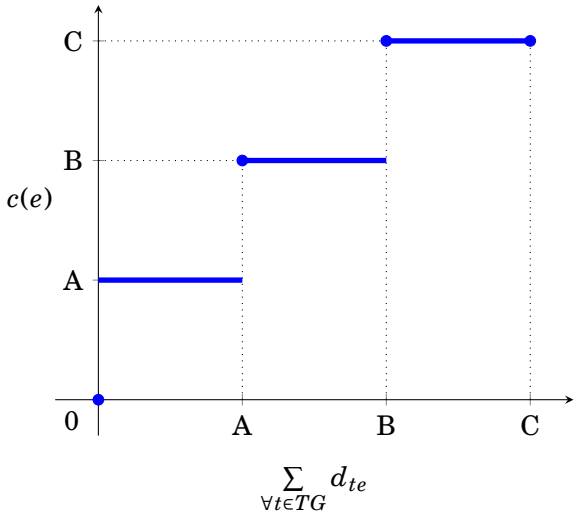


Figure 4.2: Piecewise-constant function modeling the EN capacity

$$\sum_{\forall t \in TG} d_{te} \leq C \cdot (1 - \delta_{1e}) \qquad \forall e \in \text{EN} \tag{4.14}$$

$$\sum_{\forall t \in TG} d_{te} \leq C \cdot (1 - \delta_{2e}) + \delta_{2e} \cdot (A - \epsilon) \qquad \forall e \in \text{EN} \tag{4.15}$$

$$\sum_{\forall t \in TG} d_{te} \leq C \cdot (1 - \delta_{3e}) + \delta_{3e} \cdot (B - \epsilon) \qquad \forall e \in \text{EN} \tag{4.16}$$

$$\sum_{\forall t \in TG} d_{te} \geq A \cdot \delta_{3e} \qquad \forall e \in \text{EN} \tag{4.17}$$

$$\sum_{\forall t \in TG} d_{te} \geq B \cdot \delta_{4e} \qquad \forall e \in \text{EN} \tag{4.18}$$

$$\sum_{\forall t \in TG} d_{te} \leq C \qquad \forall e \in \text{EN} \tag{4.19}$$

Moreover, variable $\delta_{ie},\ \forall i \in \{1,\ldots,P\},\ P = 4$ should comply the following condition:

$$\sum_{\forall i \in \{1,\ldots,P\}} \delta_{ie} = 1 \qquad P = 4, \forall e \in \text{EN} \tag{4.20}$$

Finally, the capacity of each EN is defined as:

$$c(e) = \delta_{1e} \cdot 0 + \delta_{2e} \cdot A + \delta_{3e} \cdot B + \delta_{4e} \cdot C \qquad \forall e \in \text{EN} \tag{4.21}$$

Overall, restrictions (4.14) to (4.21) replace the set of constraints (4.8), used in the model to determine the capacity value for each EN e, such that it will always be higher than the covered demand (otherwise, another EN is selected to cover the unsatisfied service requirements).

4.1.1.2 Linearization of the euclidean norm

In this subsection, the linearization procedure for (4.9) is shown. The proposal of [95] is followed to linearize the computation of the Euclidean distance for continuous points in $\mathbb{R}^2$. The basis is to discretize the directions of the Euclidean plane, which is characterized by the continuous domain [0, 2π], by n_d directions of size $\frac{2\pi}{n_d}$. Thus, the i^{th} discretized direction is the following unit vector U_i:

$$U_i = \left[cos\left(\frac{2(i-1)\pi}{n_d}\right), sin\left(\frac{2(i-1)\pi}{n_d}\right)\right]^T \quad \forall i \in \{1,\dots,n_d\}, \text{ being } ||U_i|| = 1$$

To verify whether two points $\mathbf{p_A} = (x_A, y_A)$ and $\mathbf{p_B} = (x_B, y_B)$ are closer than a given distance d_{TEmax}, we check that all the projections of the $\mathbf{p_A} - \mathbf{p_B}$ vector on these directions are lower than $d_{TEmax} \cdot cos(\theta_{max})$, being $\theta_{max} = \frac{\pi}{n_d}$.

$$(x_A - x_B)\cdot cos\left(\frac{2(i-1)\pi}{n_d}\right) + (y_A - y_B)\cdot sin\left(\frac{2(i-1)\pi}{n_d}\right) \leq d_{TEmax}\cdot cos(\theta_{max}) \quad \forall i \in n_d, \forall t \in TG, \forall e \in \text{EN} \tag{4.22}$$

Moreover, we have to linearize the following proposition:

$$\text{If } u_{te} = 1 \Rightarrow \text{distance(t,e)} \leq d_{TEmax} \text{ is TRUE,} \tag{4.23}$$

which is equivalent to:

$$\text{distance}(t,e) - \text{MaxD}\cdot(1 - u_{te}) \leq u_{te}\cdot d_{TEmax}, \tag{4.24}$$

being MaxD the maximum distance between two locations. Thus, from inequalities (4.22) and (4.24), the following constraint is obtained:

$$(x_A - x_B)\cdot cos\left(\frac{2(i-1)\pi}{n_d}\right) + (y_A - y_B)\cdot sin\left(\frac{2(i-1)\pi}{n_d}\right) - \text{MaxD}\cdot(1 - u_{te}) \leq u_{te}\cdot d_{TEmax}\cdot cos(\theta_{max}). \tag{4.25}$$

4.1.1.3 Linearization of the TG-EN assignments

The constraint set in (4.10) relates the part of TG t demand served by a EN with the binary variable u_{te}, which determines if this relation really exists. Thus, (4.10) involves the following implications:

$$\text{if } u_{te} = 0 \Rightarrow d_{te} = 0 \qquad \forall t \in TG, \forall e \in \text{EN} \tag{4.26}$$

$$\text{if } u_{te} = 1 \Rightarrow d_{te} > 0 \qquad \forall t \in TG, \forall e \in \text{EN} \tag{4.27}$$

which are equivalent to the following constraints, being ϵ an arbitrary small value:

$$d_{te} \leq C \cdot u_{te} \qquad \forall t \in TG, \forall e \in \text{EN} \tag{4.28}$$

$$d_{te} \geq \epsilon \cdot u_{te} \qquad \forall t \in TG, \forall e \in \text{EN} \tag{4.29}$$

Repeating the same procedure for (4.11), the following must be linearized:

$$\text{if } v_e = 0 \Rightarrow \sum_{\forall t \in TG} d_{te} = 0 \qquad \forall e \in \text{EN} \tag{4.30}$$

$$\text{if } v_e = 1 \Rightarrow \sum_{\forall t \in TG} d_{te} > 0 \qquad \forall e \in \text{EN} \tag{4.31}$$

consequently equivalent to the restrictions below:

$$\sum_{\forall t \in TG} d_{te} \leq C \cdot v_{te} \qquad \forall e \in \text{EN} \tag{4.32}$$

$$\sum_{\forall t \in TG} d_{te} \geq \epsilon \cdot v_e \qquad \forall e \in \text{EN} \tag{4.33}$$

Therefore, (4.10) has to be replaced by restrictions (4.28) and (4.29), while (4.11) has to be replaced by constraints (4.32) and (4.33).

4.1.2 Solution Proposal: Hybrid Simulated Annealing

Since any variant of the ENPP can be derived to be NP-hard (see Section 2.3), a heuristic method based on the SA algorithm was developed in this thesis as placement solution: **Hybrid Simulated Annealing (HSA)**.

Overall, selecting SA as solution was a decision based on its flexibility to solve combinatorial problems and its proven solid performance to address FLPs, as showcased in Section 2.3. In spite of its benefits, SA showed a non-convergent behavior during our experiments. The obtained solutions were widely diverse in terms of cost and number of ENs despite varying the cooling parameters and iteration counters. To solve this problem and improve the obtained results, we decide to develop an SA-based strategy leveraging some of the core ideas behind efficient methods such as Tabu Search. The idea was to inherit the flexibility of SA and combine it with the use of memory structures as done in Tabu Search [71], and local search techniques. The indicated

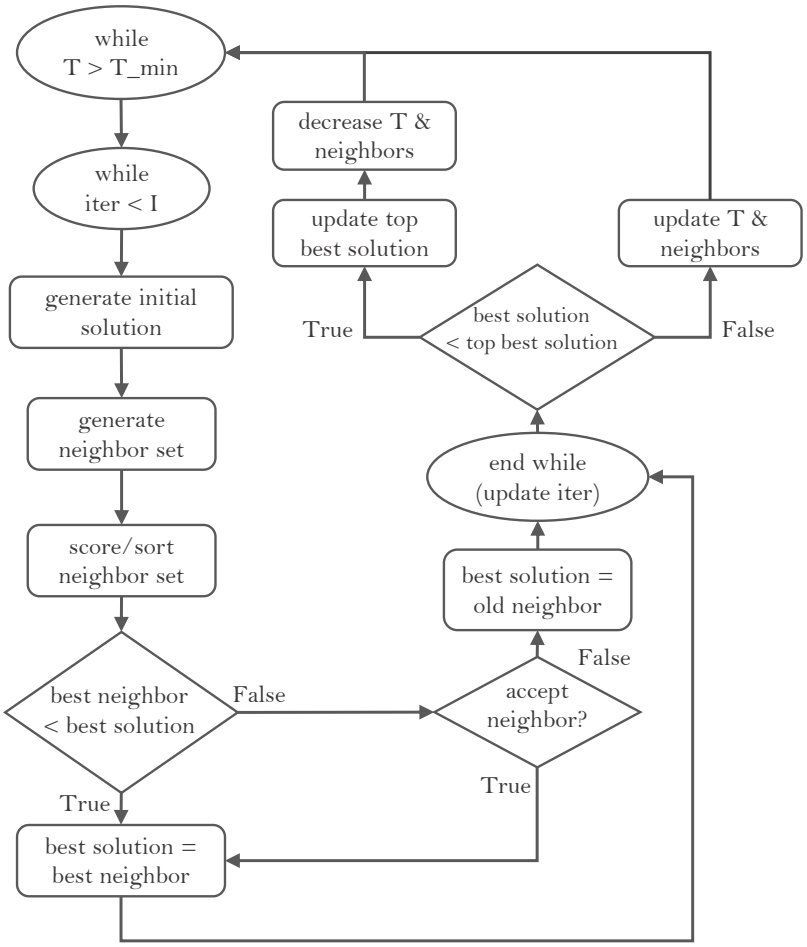


Figure 4.3: HSA flow diagram

method allowed us to improve our experimental results when compared to the traditional SA implementation (see Section 4.1.3).

In order to reduce the computation time without loss of generality and accuracy, the algorithm (see Fig. 4.3) starts by finding the isolated TGs, defined as follows:

> **Definition 2.** *A TG t is said to be* ***isolated*** *when there is no potential EN site e within the territory analyzed such that:*

(4.34) $$\text{delay}(t,e) \leq D_{max}$$

Following Definition 2, every isolated TG is to be necessarily upgraded to EN. This concept was extended to Pre-Optimized TG Areas (PTAs) which resulted in a significant reduction of the

search space size. A PTA is defined as follows:

> **Definition 3.** *A group of TGs is considered a PTA if regardless of the arbitrary TG upgraded to EN within the group there is no impact on the solution quality (i.e., cost, number of ENs, etc., according to the objective function of the mathematical model).*

The reason why no optimization can be made within PTAs is that every TG in a PTA is within the coverage area of every other TG while remaining isolated to any TG outside the PTA. Since our deployment strategy was thought for the ENs to be placed in both rural (where the TG density is expected to be lower) and urban areas, dealing with isolated TGs and PTAs in advance improves the overall performance of the proposed heuristic.

The flow diagram of the in-house developed algorithm is showed in Fig. 4.3 and detailed as pseudo-code in Algorithm 1. The first critical step is to create a good enough initial solution. For this purpose we develop a greedy strategy where random TGs are upgraded to ENs taking into account capacity and latency limitations and ensuring that no EN is assigned unneeded resources. Secondly, a set of neighbor solutions (called individuals) based on this initial step is obtained. The neighbor set contains a predefined number of individuals and is divided in a subset of solutions based on good, bad and randomly generated solutions. The overall idea was to widely explore the search space in each iteration.

Generating new solutions based on good previous individuals ensures the convergence of the algorithm into the best placement locations found (in terms of overall cost and number of ENs). For this purpose, it is crucial to ensure that a new individual resembles the previous obtained one. This is performed by selecting the TGs to be upgraded to ENs within the vicinity of the old selected ENs. Additionally, as a "*diversification*" strategy (based on Tabu Search techniques), random and bad solutions are generated to visit unexplored areas of the solution space. As the system "*cools down*", the number of neighbors generated in each iteration changes as part of the "*intensification*" process. As a result, less bad and random solutions are created while the number of good solutions is increased as long as there are cost improvements. If after an iteration cycle for a given temperature, the cost is better than the best cost ever recorded (short-term memory structure part of the "*intensification*" process), a penalty function based

Algorithm 1: Hybrid Simulated Annealing

Input: M, EN, TG, td_t, D_{max}, A, B, C, T maximum temperature, T_{min} minimum temperature, I counter, n_s number of neighbor solutions, α_f fast T decrease factor, α_s slow T decrease factor

Output: b_{se}

```
1
2  i_s ← gen_sol()                                   // generate initial placement solution
3  b_s ← i_s                                         // save best solution so far
4  b_se ← i_s                                        // save overall best solution
5  w_s ← ∅                                           // no bad solution at start
6
7  i ← 0
8  while T > T_min do
9      while i < I do
10         n_s ← num_neig(T, T_min, i)               // num. bad, random and good neighbor solutions
11         N ← neig_set(b_s, w_s, n_s)               // generate neighbor solution set
12         S ← score(N)                              // score neighbor set
13
14         if score(S[0]) < score(b_s) then
15             b_s ← S[0]                            // update best solution
16         else
17             p ← ap(T, score(S[0]), score(b_s))    // acceptance probability
18             if p > random_probability() then
                   // accept solution if p is greater than a random probability
19                 b_s ← S[0]                        // update best solution
20         w_s ← rand_sol(S)                         // random solution from S selected as bad solution
21         i ← i + 1                                 // increase iterator counter
22     if score(b_s) < score(b_se) then
23         T ← T * α_f                               // cost improvement, therefore, FAST temperature decrease
                   // decrease number of neighbors with HIGH probability as T decreases
24         p ← 1− ap(T, score(b_s), score(b_se))     // acceptance probability
25         if p > random_probability() then
26             n_s ← decrease(n_s, α_f)              // update best solution
27     else
28         T ← T * α_s                               // NO cost improvement, therefore, SLOW temperature decrease
29         p ← ap(T, score(b_s), score(b_se))        // acceptance probability
30         if p > random_probability() then
               // decrease number of neighbors with LOW probability as T decreases
31             n_s ← decrease(n_s, α_f)              // update num. neighbors
32 return b_se
```

on a random probability decreases the number of neighbors, and the speed of temperature reduction. The function probability gets higher as the temperature declines. Consequently, the system "*cools down*" quickly when there are continuous cost improvements and convergence to the global optima, while, otherwise, it slowly changes the temperature and aggressively finds more solutions.

The acceptance probability was obtained through $e^{\frac{\Delta}{T}}$, with $\Delta = s_1 - s_2$ (s_1: old solution score, s_2: new candidate solution score) [71]. To evaluate each solution, a scoring method was developed. Both the cost and the number of ENs had to be taken into account, but their values were in different orders of magnitudes. The solution was to normalize the values using logarithms and then estimate the Euclidean distance from both values, as a coordinate pair, to the coordinate origin $\mathbf{o} = (0,0)$ as follows:

$$\text{score}(n) = \text{distance}(\mathbf{n}, \mathbf{o}) \qquad \forall n \in N \tag{4.35}$$

Where

- $\mathbf{n}$: vector for the cost and number of ENs per solution such that $\mathbf{n} = (c, u)$, with $c = \log(n.cost)$ (normalized solution cost), $u = \log(n.num\_ens)$ (normalized solution number of ENs)
- N: neighbor solution set

This scoring method was inspired by the Hypervolume calculations used in multi-objective optimization [96]. The obtained score value was then used to evaluate the solutions found in each iteration and to score them accordingly.

To reduce computation times, facility locations are assumed to be co-locations of existing TGs. This approach offers a near-optimal solution in acceptable running times without extreme usage of computing resources for a fairly large number of TGs. Such assumption is supported by two main facts: capital and operational investments could be minimized by reusing already existing infrastructure and site conditions (e.g., space, networking and powering lines) on the high service demand locations. Additionally, placing the infrastructure as close as possible to the aggregation points on the service access layer significantly decreases end-to-end latency.

4.1.2.1 Complexity analysis

Since the core of our heuristic is the SA method, the traditional implementation goes through t temperature steps where the related complexity is O(log(t)). For each t the search is executed a fixed number of iterations and generates O(n) neighbor solutions. The solution generation method populates the neighbor set. For this function the worst case are the "solution-based individuals", as they loop through previous generated solutions, EN by EN (i.e., O(e), being e the number of ENs in the baseline solution), in search for random candidates (i.e., TGs suitable to be upgraded to ENs) within each EN coverage area. This iterative process is directly linked to the maximum number of TGs, conventionally called M, to be found under the most populated coverage area. M is determined by running a greedy algorithm (see Section 4.1.3) while assigning the maximum available capacity to each EN. It is straightforward to conclude that M cannot be found beforehand and that the overall algorithm complexity must be formulated based on it. Based on this analysis the complexity can be specified as O(n e M log(t)). The initial value of the number of neighbor solutions is relatively small and it is reduced as the system converges. Therefore, the overall algorithm complexity can be defined as O(e M log(t)).

4.1.3 Evaluation and results

In order to compare the performance of the placement strategy proposed, a traditional SA implementation, the HSA approach and the MILP were run for three latency values: 1 ms, 3 ms and 5 ms. These delay values were selected because they comply with the 5G latency requirements for a wide variety of use case scenarios. For the case study of mobile RANs and a Cloud-RAN (C-RAN) architecture, virtualized BBUs [97], are to be placed at the ENs. Consequently, from [75, 98] a backhaul transmission delay for Long-Term Evolution (LTE) networks is known to be around 250 μs. Therefore, for 5G networks and the proposed latency values, D_{max} was estimated to be 3 km,9 km,15 km (for transmission times of 31 μs,93 μs,156 μs). The input list can be observed in Table 4.2.

A map grid of 100 km x 100 km was used with a set of TG ranging from 100 to 500 TGs (with a 100 TGs increase step in each simulation). Figure 4.4 illustrates the simulated territory of interest. TGs are distributed in three *cities* and randomly in rural areas, consequently emulating

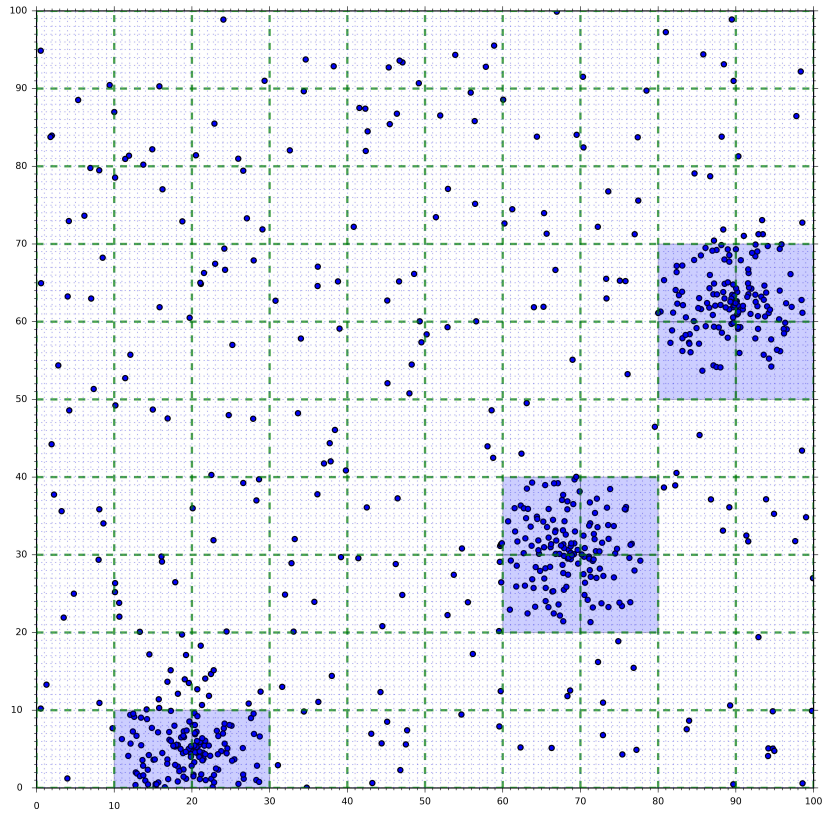


Figure 4.4: TGs randomly distributed in three *cities*

Table 4.2: Input parameter values

D_{max} (km)	TG number	Cap. L-EN	Cap. M-EN	Cap. S-EN
3	100	40	30	21
	200	59	41	32
	300	80	51	40
	400	105	74	58
	500	150	101	78
9	100	74	51	38
	200	138	94	71
	300	209	146	102
	400	291	205	153
	500	348	238	179
15	100	75	52	41
	200	154	101	79
	300	240	157	119
	400	326	214	160
	500	392	268	193

a reasonably realistic distribution of demand points, where urban areas present higher traffic density.

For the heuristic, the *temperature* ranged from 1.0 to 0.001 with a step size for the fast

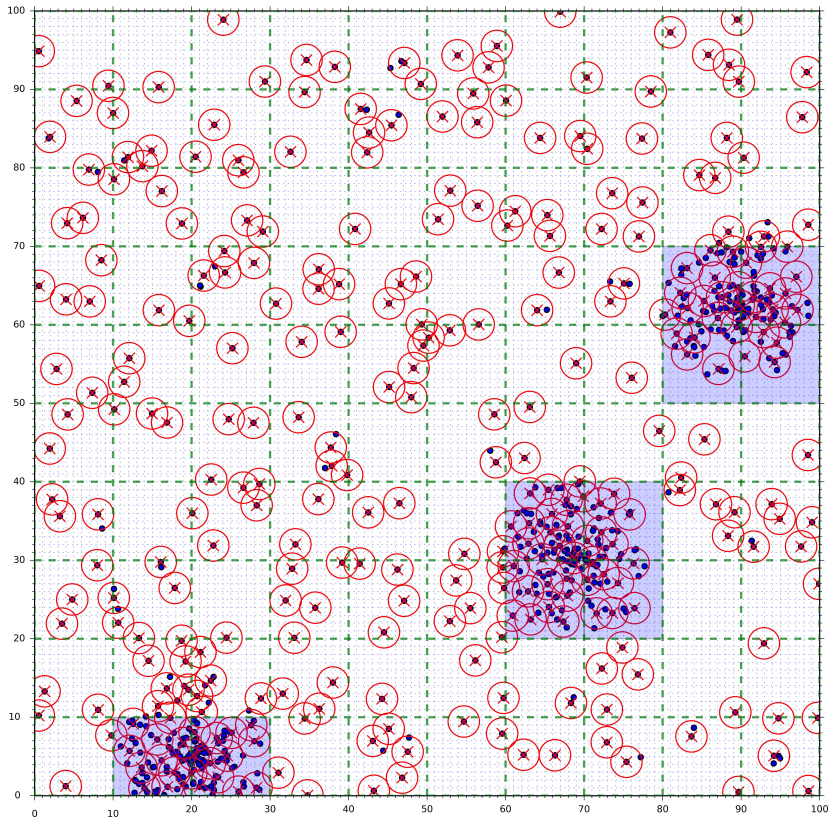
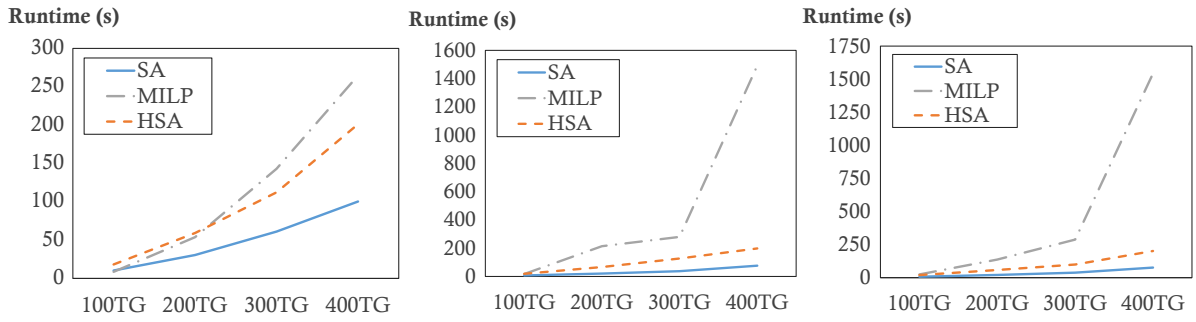


Figure 4.5: Solution obtained after running the algorithm

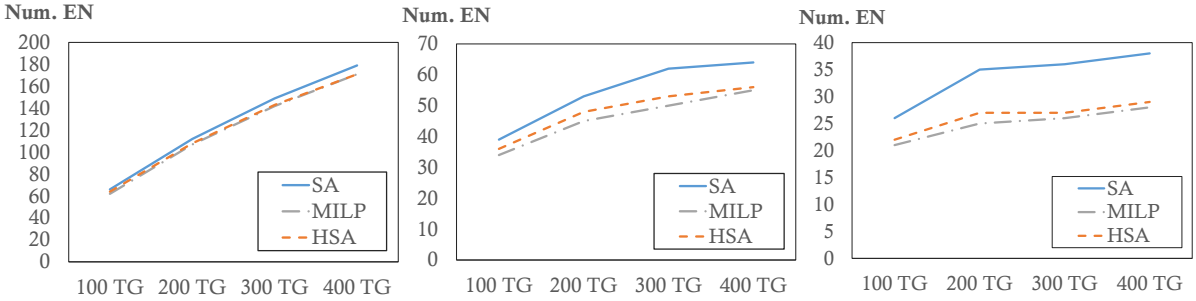
temperature reduction of $\alpha = 0.8$ and $\alpha = 0.9$ for the lower stepping process. The number of iterations per temperature was set to 10 for the HSA as several solutions are created and evaluated in each iteration (the number of neighbor individuals in each iteration was set to 8). The iteration counter for the traditional SA was set to 100. This value was empirically determined, aiming to ensure a fairly similar number of iterations when compared to the HSA proposed (in fact 100 is a bit higher to compensate SA lack of accuracy). All simulations were run in a computer with a 2.60 Hz 8-core CPU ($x64$ architecture) and 32 GB RAM. Pyomo [30, 31] was the python-based package selected to solve the optimization model proposed in Section 4.1.1, with GLPK as underlying solver.

To obtain the EN capacities showed in Table 4.2, an additional greedy algorithm was developed. It iteratively upgrades to EN the TG with the most populated coverage area (given D_{max}), and keeps on until no TG remains uncovered. As a result, the allowed EN capacities are found for any particular solution. Such greedy algorithm was run several times for each simulation setting



(a) Case $D_{max} = 3\ km.$ (b) Case $D_{max} = 9\ km.$ (c) Case $D_{max} = 15\ km.$

Figure 4.6: Execution times for the SA, HSA and MILP.



(a) Case $D_{max} = 3\ km.$ (b) Case $D_{max} = 9\ km.$ (c) Case $D_{max} = 15\ km.$

Figure 4.7: Number of ENs deployed the SA, HSA and MILP.

described above. Consequently, the final capacity values for the heuristics and the mathematical model were obtained through the statistical analysis of the results.

Figure 4.5 displays a final solution after running the heuristic. It can be observed how every TG is covered (ENs are depicted as × and the surrounding circles are the coverage area of D_{max} radius).

To validate the results both the HSA and the SA were run ten times for every D_{max} and TG combination. Meanwhile, for each D_{max} value the number of TGs was increased as mentioned above, aiming to calculate the execution time and the number of ENs of the optimal solution found. The findings are presented in Fig. 4.6 and Fig. 4.7.

A Significant difference in the running times for both heuristics and the MILP model can be noticed in Fig. 4.6. Despite the steady surge in the first stages for all cases, the mathematical model has a nearly impossible task in obtaining the optimal solution for $D_{max} = \{9, 15\}\ km$ and

TG = 500 (see Fig. 4.6). Therefore, the experimental results are just shown from 100 TG*s* to 400 TG*s* in both Fig. 4.6 and Fig. 4.7. In fact, the heuristics are able to find a near-optimal solution in significantly less time and with a maximum of a few ENs gap as shown in Fig. 4.7.

The MILP model execution time rapidly steeps to huge values after reaching 400 TG*s*, due to the exponential growth in the number of feasible solutions. In contrast, the running delay of both heuristics climbed regularly throughout the TG experimentation set. Due to the latency constraint variation, the number of EN decreases as D_{max} rises. The reason is that the EN coverage area becomes larger, thus less ENs are required to cover the existing TGs.

Regarding the performance of both heuristics compared to the exact model when minimizing the number of ENs, the HSA shows clear improvements at the cost of an increase in the execution time. Since the goal is to place physical infrastructure, the placement strategy is to be run during the planning phase of the deployment and thus this is not considered an issue.

The HSA performance regarding the number of EN deployed is quite promising. From Fig. 4.7, the difference between the number of ENs placed by the MILP and the HSA approach never surpassed a threshold of even less than 5 ENs.

Based on these results, the following sections adopt HSA as core solution for the ENPP in either variant. In this regard, HSA is thoroughly assessed throughout the remaining of this thesis, demonstrating the suitability of our algorithm to solve complex combinatorial problems such as the ENPP.

4.2 Latency and reliability-constrained ENPP for flexible EN capacities

The scope of the formulation and solution approach proposed in the previous section is not applicable to a real-life EN deployment. The primary reason for this is the simplicity of the mathematical model, where core CAPEX/OPEX sources -e.g., fixed and interconnection costs- were not considered. Moreover, the capacity allocation model lacked accuracy for next generation network deployments, where a more flexible strategy is required to avoid critical capacity issues.

On the other hand, Chapter 1 showcased that both reliability and latency-aware planning

Table 4.3: Glossary of symbols for the latency, reliability-constrained SO-ENPP

Symbol	Parameter	Variable	Description
EN	✓		Set of EN potential locations (where $e \in$ EN)
TG	✓		Set of TG locations (where $t \in$ TG)
C	✓		EN maximum allowable capacity
L_M	✓		Maximum allowed latency between any TG and its serving EN
L_m	✓		Maximum allowed latency between a TG with ultra-low latency requirements and its serving EN
F_e	✓		Fixed cost of deploying an EN at e
td_t	✓		Total demand of a TG at t
ω	✓		Cost per capacity unit
c_e		✓	Capacity of an EN placed at e
v_e		✓	1 if an EN at e is placed, 0 otherwise
L_{et}		✓	Cost of interconnecting an EN at e and a TG at t
u_{et}		✓	1 if a TG at t is covered by an EN at e, 0 otherwise
d_{et}		✓	Fraction of demand from a TG at t covered by an EN at e
$l(e,t)$		✓	Latency between an EN and TG at e and t respectively
x_t		✓	1 if TG at t requires ultra-low latency, 0 otherwise
y_t		✓	1 if TG at t requires ultra-high reliability and availability, 0 otherwise

are crucial for the EN efficient deployment and user requirement satisfaction in 5G scenarios and that reducing the EN network cost is directly linked to the EN deployed capacity. Taking this into account, this section significantly extends the results obtained in the previous one by proposing a framework for a cost-effective EN placement in 5G environments.

In summary, the main contribution of this section is to present a real-life cost optimization model based on: a) accurate capacity allocation, b) efficient site selection considering underlying fixed/interconnection costs and, c) an extended mathematical model through additional constraints considering both reliability and latency requirements.

4.2.1 Problem model

In order to formulate the problem (glossary of symbols available in Table 4.3), the assumptions made in Section 4.1.1 were followed, i.e., latency was translated into the Euclidean distance by assuming such delay to be the transmission latency between any EN-TG pair (the specific latency-distance equivalents are specified in Section 4.2.3); the demand aggregation points were modeled as TGs. The initial EN location set assumed to be known in order to reduce the computation time

and make the MILP formulation solvable comprised the following potential locations: ISP-PoPs, CDN-PoPs and TGs. Finally, a 10 ms delay was selected as the maximum EN coverage range (i.e., $L_M = 10\ ms$) in order to satisfy most identified 5G use cases. The model is presented below:

$$\text{(4.36)} \quad \text{Minimize:} \quad \sum_{\forall e} \omega \cdot c_e \cdot v_e + \sum_{\forall e,\, t} L_{et} \cdot u_{et} + \sum_{\forall e} F_e \cdot v_e$$

$$\text{(4.37)} \quad \text{s. t.:} \quad \sum_{\forall t \in TG} u_{et} \geq v_e \qquad \forall e \in EN$$

$$\text{(4.38)} \quad \sum_{\forall e \in EN} d_{et} = td_t \qquad \forall t \in TG$$

$$\text{(4.39)} \quad \sum_{\forall t \in TG} d_{et} \leq c_e \qquad \forall e \in EN$$

$$\text{(4.40)} \quad d_{et} \leq td_t \cdot u_{et} \qquad \forall e \in EN, t \in TG$$

$$\text{(4.41)} \quad \text{if } u_{et} = 1 \Leftrightarrow d_{et} > 0 \qquad \forall e \in EN,\ t \in TG$$

$$\text{(4.42)} \quad \text{if } v_e = 0 \Leftrightarrow \sum_{\forall t \in TG} d_{et} = 0 \qquad \forall e \in EN$$

$$\text{(4.43)} \quad \sum_{\forall e \in EN} u_{et} \geq 1 + y_t \qquad \forall t \in TG$$

$$\text{(4.44)} \quad \text{if } l(e,t) > L_M \Rightarrow u_{et} = 0 \qquad \forall e \in EN,\ t \in TG$$

$$\text{(4.45)} \quad \text{if } x_t = 1 \Rightarrow \begin{cases} u_{et} = 0 & \text{if } l(e,t) > L_m \\ u_{et} \leq 1 & \text{if } l(e,t) \leq L_m \end{cases} \qquad \forall e \in EN,\ t \in TG$$

$$\text{(4.46)} \quad u_{et}, v_e, x_t, y_t \in \{0,\ 1\} \qquad \forall e \in EN,\ t \in TG$$

$$\text{(4.47)} \quad c_e, L_{et}, F_e, d_{et} \geq 0 \in \mathbb{R} \qquad \forall e \in EN,\ t \in TG$$

The objective function in (4.1) represents the costs involved in the EN deployment following the findings in Chapter 3. The first addend accounts for the capacity related costs mainly determined by the capacity assigned to each EN (i.e., c_e). The second term represents the cost of interconnecting each TG with its serving EN, while the third addend accounts for the location-dependent costs. The constraints in (4.37) allows an EN to be deployed at location e only if there is at least one TG within its coverage range (determined by the latency constraints). The restrictions from (4.38) to (4.42) characterize the TG demand, as any TG can be fully or partially covered by one or more ENs. From (4.38) the total covered demand of a TG -i.e., the sum of all

the covered demand fractions- must be equal to the total demand of that TG. Additionally, each EN capacity to serve TG demands is limited in (4.39). The interrelation between each covered demand fraction and the existence of a TG at t to be covered by an EN at e is described in (4.40) and limited by td_t. Moreover, (4.41) forces to zero any fraction of TG demand covered by an EN (at e) in case the TG (at t) is not covered by the EN at e. From (4.42) if an EN is not placed at e, there must be no demand fraction covered from this site.

Each TG can be covered by one or more ENs (see Eq. (4.43)) considering the reliability requirements of each TG. This way, if an arbitrary TG requires ultra-high reliability ($y_t = 1$) our model ensures that it is covered by at least two ENs. Latency requirements are constrained in (4.44) and (4.45). The former ensures that any TG is only considered to be within the coverage range of an EN if the latency between locations e and t is less than a predefined threshold (see Section 4.2.3 for a case study analysis).

On the other hand, ultra-low latency is satisfied through (4.45). From this constraint if a TG has ultra-low latency requirements, the transmission delay between such TG and its covering EN should be less or equal than 1 ms (i.e., the following is assumed: $L_m = 10\ ms$, given the 5G latency requirements according to [4]), while forcing any TG beyond 10 ms from an EN to be outside its coverage range. In case all latency constraints are satisfied, a TG at t could be covered or not by an EN at e by setting the u_{et} value. Within the u_{et} and v_e value definition is where the optimization process takes place by deciding where to deploy an EN (according not only to the TG requirements but to the interconnection and location-dependent costs). The variable and remaining term domains are specified in (4.46) and (4.47).

The constraints in (4.44) and (4.45) can be implemented without additional linearization procedures. This is possible because the $l(e,t)$ value for each EN-TG pair can be estimated beforehand and easily called as a constant value during execution. However, (4.36) requires further transformation in order to be linearized. Thus, making $z_e = c_e \cdot v_e$ the following restrictions are added to the model (where C is the maximum EN capacity value):

$$z_e \leq C \cdot v_e \tag{4.48}$$

$$z_e \leq c_e \tag{4.49}$$

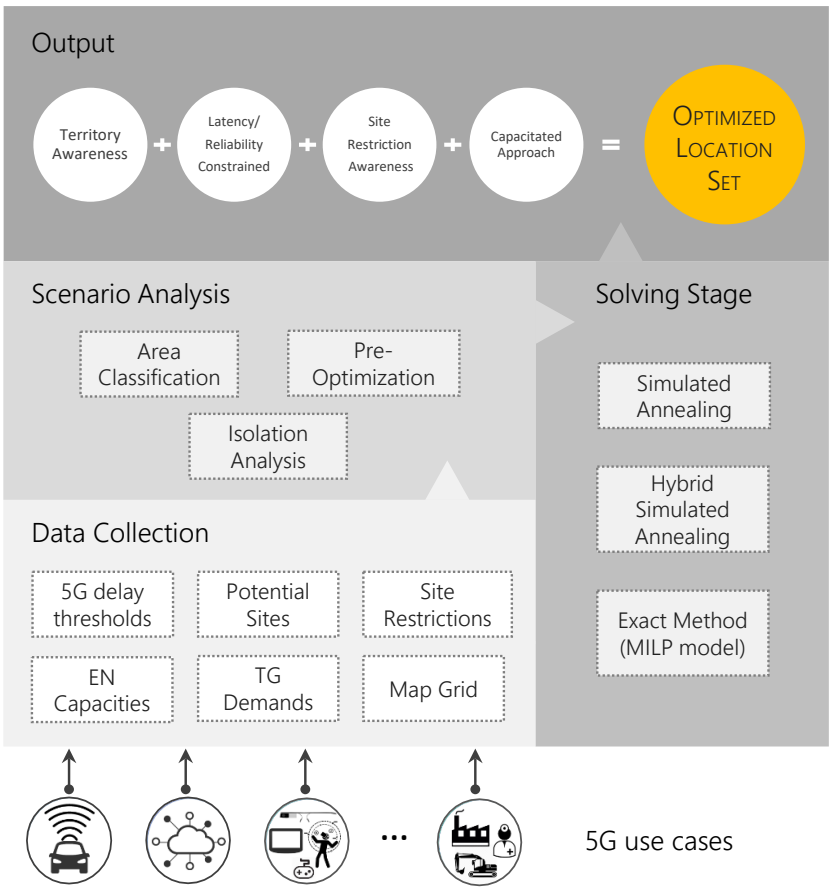


Figure 4.8: EdgeON framework architecture.

(4.50) $$z_e \geq c_e - C \cdot (1 - v_e)$$

(4.51) $$z_e \geq 0$$

4.2.2 Solution Proposal: EdgeON Framework

The framework proposal called **EdgeON** aiming at solving the modeled ENPP variant is outlined in Fig. 4.8. Overall, the goal is to output a ranking of potential EN sites such that the EN network cost is minimized by selecting the best deployment locations. To tailor **EdgeON** to 5G needs, the technical requirements (e.g., in terms of latency) identified for most 5G use cases were assumed, along with the benefits of current identified enabling technologies (e.g., NFV).

EdgeON goes through four main stages. The **Data Collection** analyzes and normalizes the input data to comply with the processing needs (e.g., data structure normalization, unit system

equivalencies). From this point on, the **Scenario Analysis** module checks the map grid and demand distribution, to divide the territory of interest and to classify the service areas as urban or rural. This stage allows us to simplify the problem in low populated areas where the ENs are arbitrarily co-located at the TG locations. Such approach reduces CAPEX and OPEX as the demand is primarily scattered over large underpopulated territories. Within this stage, further optimization is carried through the Isolation and Pre-Optimization phases. The former checks each service area for isolated TGs (see Section 4.1.2). The latter seeks for PTAs, according to Definition 3. Consequently, in each PTA a random EN site is selected. In addition, the **Solving Stage** runs at least one of its underlying placement strategies to solve the placement problem for the remaining uncovered TGs. After the problem is solved, the framework outputs a ranked set of restriction-free locations where the ENs should be placed, along with the allocated capacities, the demand covered per EN and additional relevant data regarding the performance of the executed solution methods.

The input data required by **EdgeON** is showed in Table 4.4. Each location is assumed to have known fixed deployment costs, while each TG has a known service demand value, latency and reliability requirements. Likewise, the cost of interconnecting any EN-TG pair is assumed to be known beforehand and estimated assuming a direct physical link. The PoP set added to the suitable EN locations (initially the set of TGs) was built with real data collected from Telegeography GlobalComms database[2] about ISP-PoPs operating in Spain and extrapolated to estimate the number of PoPs in an arbitrary-sized city.

The maximum capacity to be assigned to an EN was determined experimentally. By running a greedy algorithm selecting random locations as ENs and greedily assigning TGs to it the ENs typical capacity values were found. From this baseline data and attempting to represent a realistic scenario (where capacity has to be shared and coverage overlapping exists), the statistical mean of the capacity for a medium sized EN was selected. This approach ensures a thorough evaluation of our algorithm as it implies the worst deployment case where TG demands are usually split among several ENs.

[2]https://www2.telegeography.com/globalcomms-database-service

Table 4.4: Input parameters for the framework placement algorithm

Parameter	Meaning
$MapGrid$	Territory where the TGs are located
C	EN maximum allowable capacity
L_{et}	Cost of interconnecting an EN at e and a TG at t
L_M	Maximum allowed latency between any TG and its serving EN
L_m	Maximum allowed latency between a TG with ultra-low latency requirements and its serving EN
TG_{demand}	Set of TG coordinates (including data about fixed deployment costs and latency/reliability requirements)
$PoPs$	Set of existing potential EN locations (includes data about fixed deployment costs)
$Site\_Restrictions$	EN feasible location set of restrictions

As in Section 4.1, the ENPP tackled in this section could be derived to be NP-hard. Therefore, the HSA heuristic proposed in Section 4.1.2 was used as solution strategy. Although the inherited HSA core and behavior remained mostly unchanged, the solution generation method had to be significantly modified. The reason was the introduction of a new critical constraint (i.e., reliability), the enhanced problem model aiming at a more comprehensive cost function and the need to check whether each location was suitable for deployment (i.e., restriction checking per site). Additionally, the analysis of non-TG locations introduced additional complexities to the solving scheme.

While an initial solution is randomly generated by assigning each TG to randomly selected ENs, to find a new solution based on a previous one, the modified solution generator iterates over each coverage area[3]. For each coverage area a new EN is selected ($e2$ in Fig. 4.9) among the covered TGs or the available PoPs within the coverage area following the algorithm in Fig. 4.10. The idea behind this strategy is to minimize the "*dominoes effect*" that occurs when generating a neighbor solution by randomly selecting new ENs. Such randomized method entails a wide reallocation of the surrounding TGs resulting in higher execution times and solution evaluation inaccuracies.

In a nutshell, the TG-EN allocation is made considering the new latency-reliability constraint pair, the underlying TG demands, the maximum EN capacity and the PoPs available in the coverage area. As a result, each TG is covered by more than one EN if its reliability requirements

[3]Coverage Area: tuple formed by [EN, $covered\_TGs$]. Depicted as a red circle in Fig. 4.9.

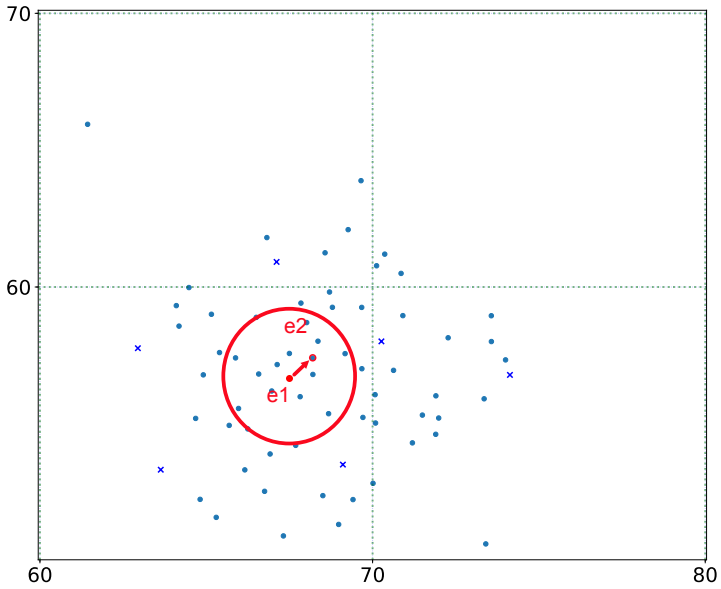


Figure 4.9: Coverage area analysis

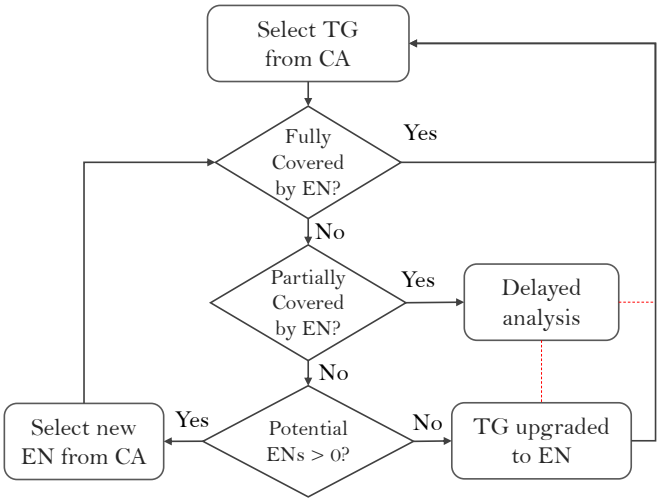


Figure 4.10: TG-EN allocation per CA after a new EN is selected

are high and similarly the TG-EN distance is ensured to comply with the delay demands of each TG. Namely, no coverage association is made if $l(e,t) > L_M$ for a given EN-TG pair. When a partial coverage is made (i.e., a TG is partially covered by the EN selected), the TG is queued for a Delayed Analysis phase, in case another TG is selected as EN. If the latter does not occur, the TG is either upgraded to EN or allocated to the nearest EN from another coverage area if possible and if this approach reduces the overall costs.

4.2.2.1 Complexity

The iterative process is directly linked to the number of EN locations available to be found under the most populated coverage area (as TG locations will always be higher than service providers PoPs and any other available sites), arbitrarily called M. To determine the M value, a greedy algorithm is run (see Section 4.2.3). The value of M cannot be found without generating at least an initial solution. What is more, it is certain that the heuristic complexity is directly impacted by this parameter. Since the placement heuristic is an extension of the algorithm presented in Section 4.1.2, the complexity is fairly similar.

Given the reliability constraint, a set of TGs must be covered by more than one EN (such TG set size is denoted as R). An additional component of the solution generator performs the covering steps for these particular TGs. This section of the code adds a complexity of O(R N), as each TG demanding ultra-high reliability has to be assigned to an available EN by searching the EN location set (of length N).

Globally, the complexity can be specified as O((G E M + R N) log(T)), where M is the number of EN locations available to be found under the most populated coverage area (as TG locations will always be higher than existing PoPs and any other available sites). The number of neighbor solutions is relatively small at first and it is substantially decreased periodically if there are solution improvements, thus G becomes negligible. Although $R \cdot N$ is comparatively small when compared to $E \cdot M$, a significant number of rejections and recursive steps are made due to the strict latency constraints. Therefore, the overall algorithm complexity can be defined as O((E M + R N) log(T)).

4.2.3 Evaluation and Results

The framework proposal described in Section 4.2.2 aims at ensuring the applicability and usability of the HSA in real-life scenarios. To assess the performance of this approach in terms of overall expenses and number of ENs deployed, a traditional SA and MILP model were implemented and ran as in Section 4.1.3.

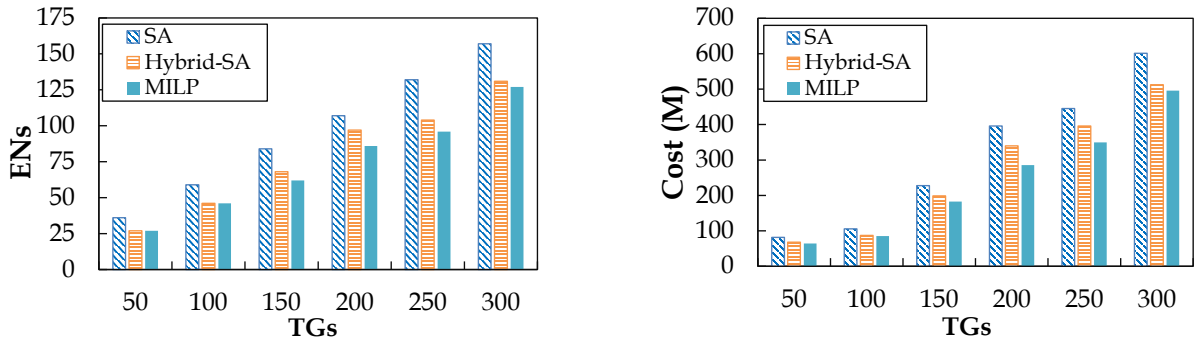
The latency parameters on the mathematical formulation (i.e., $l(e,t)$, L_m and L_M), were estimated based on the Euclidian distance. Therefore, for the proposed latency values in 5G

networks (1 ms for ultra-low latency and 10 ms for other scenarios), L_m was estimated to be 3 km and 15 km was found for L_M (assuming: transmission times of 31 μs, 300 μs, processing time of 200 μs due to routing/switching for long-distance links) extrapolated for a mobile scenario as in [75, 98]. The simulations varied the number of TGs from 50 to 300 TGs (with a 50 TGs increase step in each simulation).

Regarding the algorithms, the minimum *temperature* value was set to 0.001 (the initial *temperature* selected was 1.0). Meanwhile, the temperature reduction for the fast stepping process was 0.7, while 0.9 was selected as slow α for the HSA. For the traditional SA, the value was $\alpha = 0.95$. Each temperature cycle executed only 10 iterations for the HSA, since the neighbor set comprised a wide range of solutions to be assessed per iteration (the length of the neighbor set was 8). For the traditional HSA, 100 iterations were made for each temperature step, to guarantee a fairly similar number of iterations for the two placement methods. For the calculation of the solution expenses, the cost parameters were analyzed based on a generic measurement unit such that an arbitrary number of cost-units were assumed to be equivalent to a capacity unit (a conversion made through ω in the first addend of (4.36)). Consequently, the results presented below lack a specific currency, although this does not imply any loss of generality during the analysis.

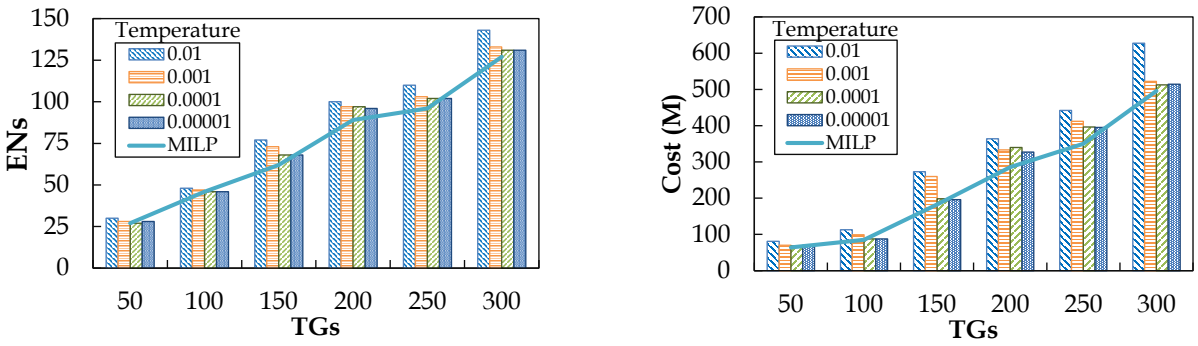
All simulations were ran in a computer with a 3.30 GHz 10-core CPU ($x64$ architecture) and 64 GB RAM. As in Section 4.1.3, Pyomo [30, 31] was the python-based package selected to implement the optimization model proposed in Section 4.2.1, along with Gurobi [32] as underlying solver. The numerical results were validated by running several times each heuristic and the MILP model for each parameter setting. Additionally, the TG count was periodically increased to estimate the solution costs, the running time and the number of ENs of the optimal solution found by each method. The findings are presented from Fig. 4.11 to Fig. 4.14.

Overall, the HSA clearly outperformed the traditional SA regarding the reduction of the ENs number and showed a very small gap when compared to the MILP model. The leftmost figure on Fig. 4.11 evidences an average gap of less than 5 nodes between the HSA and the exact method. In contrast, the traditional SA poorly performed when compared to the HSA and the MILP, resulting in an average of 15 and 22 additional ENs in each case, due to its inaccuracy



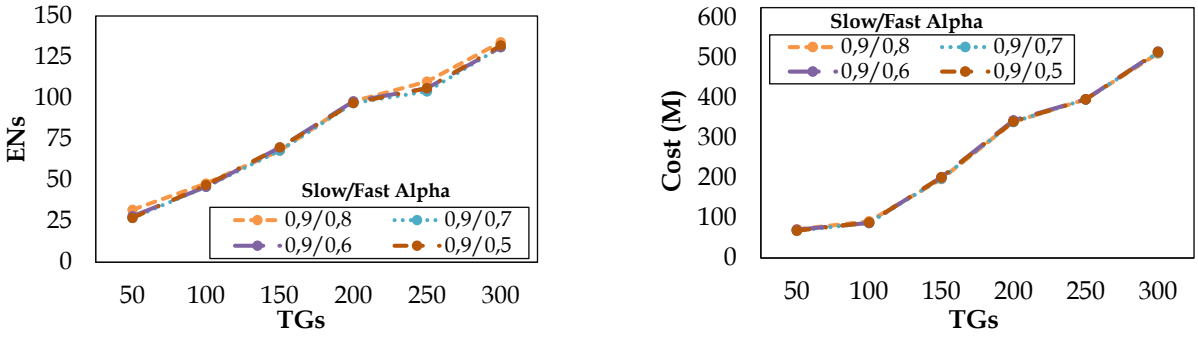
(a) Number of EN. (b) Solution cost.

Figure 4.11: Number of ENs and deployment cost obtained by SA, HSA and MILP.



(a) Number of ENs. (b) Solution cost.

Figure 4.12: HSA performance under temperature variation.



(a) Number of ENs. (b) Solution cost.

Figure 4.13: HSA performance under slow/ fast α variation.

and non-convergent behavior. The HSA significantly improves the traditional SA solution costs, while approaching the MILP.

From the rightmost image on Fig. 4.11, nearly 20% in average cost savings was achieved

through the HSA instead of the SA (when compared to the MILP results). On the other hand, the MILP cost values were only a 5% lower than the HSA. These cost reductions are mainly targeting the deployment expenses (as the location selection strategy considers location-dependent costs). Nevertheless, since the number of ENs and the capacity allocation is optimized, a substantial operational cost reduction is ensured.

Although the obtained results are promising, it is reasonable to argue that the variation of the internal heuristic parameters -i.e., temperature and slow/fast alpha- could significantly affect the performance values. Therefore, further simulations were conducted by changing the minimum temperature and α values (see Fig. 4.12 to Fig. 4.13) to analyze their impact on the solution. It is worth noticing that modifying these parameters leads to a higher number of iterations. Thus, considering that the SA core relies on the iteration count, relevant solution improvements were expected. Despite these intuitive assumptions, neither the temperature nor the slow/fast alpha variation significantly impacted the HSA outcome.

From Fig. 4.12 the HSA obtained better solutions for a minimum temperature under 10^{-4}, although the difference was not significantly high in most cases, with the exception of the 10^{-2} series. Similar findings arose from varying the slow/fast α pair. In spite of speeding up the heuristic by reducing the fast α value, the performance variation was mainly negligible. However, the highest gap for both the ENs number and the solution cost was obtained for slow/fast $\alpha = 0.9/0.5$.

In Fig. 4.14 the computation times for all solving methods are depicted. Despite the regular surge throughout the first stages, the MILP was unable to find a solution for any TG value above 300 nodes after more than a week running. When compared to the results obtained in Section 4.1.3, Fig. 4.14 evidences the impact of the reliability and latency modification parameters regarding the algorithm complexity.

Nevertheless, the significant increase in the HSA execution time is still not considered an issue due to the offline nature of the proposed method. In fact, a more important conclusion is that the execution time behavior remained consistent in spite of the objective function enhancement and the increased difficulty added by the stringent latency and reliability parameters.

What can be concluded from these analyses, is that the HSA conducts a thorough exploration

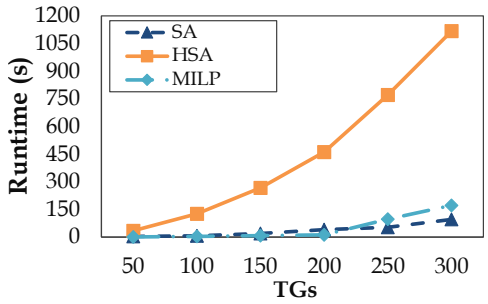


Figure 4.14: Runtime for SA, HSA and MILP.

of the search space based on the baseline minimum temperature (10^{-3}) and alpha values (slow/fast $\alpha = 0.9/0.7$) selected. Finally, from these experiments and towards further efficient analysis, an α combination of 0.9/0.7 and a temperature of 10^{-4} are recommended.

4.3 Conclusion

This chapter presented an in-house heuristic and framework to solve the SO-ENPP for strict 5G technical requirements.

The heuristic proposed in Section 4.1.2 was evaluated in order to determine its suitability to solve the ENPP. Overall, HSA showed promising results when compared to other heuristics and exact methods, thus encouraging its use to address complex multi-objective ENPP models. Consequently, this heuristic can be considered a core contribution of this thesis towards the solution of the ENPP and other complex combinatorial problems.

From this starting point and in order to build the foundations for a practical tool (i.e., for Telcos and operators) to solve real-life ENPP models, the problem presented in Section 4.1.1 was extended and a framework implementing several solving methods (for evaluation purposes) was designed to tackle the latency and reliability-constrained SO-ENPP. The proposed framework was tested showing the benefits of the in-house heuristic when compared to an approximation algorithm and an exact model.

CHAPTER 5

MULTI-OBJECTIVE ENPP

This chapter is based on:

- A. Santoyo-González and C. Cervelló-Pastor, "Network-aware Placement Optimization for Edge Computing Infrastructure under 5G," in *IEEE Access*, 2020.

In the previous chapters the foundations to address a real-life modeled ENPP were established. The core limitations of the mathematical formulations and solution strategies presented so far are mainly twofold: 1) the network-agnostic approach overlooking the underlying network capacities and capabilities when interconnecting TGs and ENs and, 2) the need to consider additional parameters of critical importance for 5G networking (e.g., energy consumption).

In this regard, this chapter presents a network-aware ENPP model and solution approach based on a re-design and improvement of the framework proposed in Section 4.2.2.

5.1 Network-aware Multi-Objective ENPP

Since a massive surge in data processing and bandwidth usage is envisioned under 5G networks, a network-aware strategy is mandatory to satisfy the required KPIs and avoid performance

Table 5.1: Complete glossary of symbols for the problem formulation

Symbol	Params.	Vars.	Description
α_i^t		✓	1 if a TG at t is served by an EN at i, 0 otherwise
v_i		✓	1 if an EN is placed at i, 0 otherwise
γ_{ij}^{te}		✓	fraction of the network demand of TG t served by EN e routed through link (i,j)
ψ_{ij}^{te}		✓	1 if link (i,j) is active and routing demand (i.e., $\gamma_{ij}^{te} > 0\ \forall (i,j) \in L, e \in E, t \in T$), 0 otherwise
χ_i		✓	ratio of in-use EN capacity such that $\chi_i \in (0,2]$
μ_i^t		✓	fraction of the compute demand of TG t served by an EN at i
κ_i^t		✓	fraction of the network demand of TG t served by an EN at i
F_i		✓	upfront costs of deploying an EN at i
ι_i^t		✓	cost of interconnecting an EN at i with a TG at t
θ_i		✓	cost of an EN with capacity (Cc_i, Cn_i) at i
τ	✓		cost per compute capacity unit
σ	✓		cost per network capacity unit
M_t	✓		computing demand of TG at t
K_t	✓		network demand of TG at t
A_t	✓		1 if a TG at t aggregates ultra-low latency services, 0 otherwise
R_t	✓		1 if a TG at t requires at least two serving ENs (i.e., main and backup) due to the reliability requirements of the aggregated services, 0 otherwise
Cc_i	✓		maximum compute capacity assigned to the EN at i
Cn_i	✓		maximum networking capacity assigned to (or available at) the EN at i
B_{ij}	✓		link bandwidth $(\forall (i,j) \in L)$
D_{ij}	✓		link delay $(\forall (i,j) \in L)$
P_i	✓		processing delay on node $i \in N$
D_M	✓		maximum delay allowed between a TG an its serving EN
D_U	✓		maximum delay allowed between a TG with ultra-low latency requirements an its serving EN

degradation in the long run. To address this open research question and achieve a core goal of this thesis, the following sections present a network-aware Multi-Objective ENPP (MO-ENPP) model and solution strategy tailored to 5G scenarios.

5.1.1 Problem Model

The MO-ENPP aims at reducing the cost of deploying an EC network while ensuring that the capacity usage ratio per EN is maximized and the number of deployed ENs is minimized. We assume that the underlying network topology (i.e., assumed to be a fully connected undirected graph) is composed by the set of nodes N and the set of links L. The set N is formed by the set of TGs, denoted as T, the nodes from the ISP backhaul network, existing Central Offices and ISP-PoPs amongst other suitable locations. Table 5.1 summarizes the variables and parameters used for the problem formulation.

Considering that any $i \in N$ is a potential EN site, the objective functions for the network-aware ENPP can be defined as follows:

$$\text{Min} \quad \sum_{\forall i \in N} \theta_i \cdot \upsilon_i + \sum_{\forall i \in N} \sum_{\forall t \in T} \iota_i^t \cdot \alpha_i^t + \sum_{\forall i \in N} F_i \cdot \upsilon_i \tag{5.1}$$

$$\text{Min} \quad \sum_{\forall i \in N} \upsilon_i \tag{5.2}$$

$$\text{Max} \quad \sum_{\forall i \in N} \chi_i \cdot \upsilon_i \tag{5.3}$$

where,

$$\theta_i = \tau \cdot (Cc_i - \sum_{\forall t \in T} \mu_i^t) + \sigma \cdot (Cn_i - \sum_{\forall t \in T} \kappa_i^t) \qquad \forall i \in N \tag{5.4}$$

$$\iota_i^t = \sum_{\forall (i,j) \in L} \sigma \cdot \gamma_{ij}^{te} \qquad \forall e, t \in N, T \tag{5.5}$$

$$\chi_i = \frac{\sum_{\forall t \in T} \mu_i^t}{Cc_i} + \frac{\sum_{\forall t \in T} \kappa_i^t}{Cn_i} \qquad \forall i \in N \tag{5.6}$$

$$\alpha_i^t, \upsilon_i, \psi_{ij}^{te} \in \{0,\ 1\} \qquad \forall i, e \in N, t \in T, (i,j) \in L \tag{5.7}$$

$$\theta_i, \iota_i^t, \chi_i, \beta_{ij} \geq 0 \qquad \forall i \in N, t \in T, (i,j) \in L \tag{5.8}$$

$$\kappa_i^t, \mu_i^t, \gamma_{ij}^{te} \in [0,1] \qquad \forall i, e \in N, t \in T, (i,j) \in L \tag{5.9}$$

$$Cc_i, Cn_i \geq 0 \qquad \forall i \in N \tag{5.10}$$

Equation (5.1) minimizes the overall cost of deployment. The first addend accounts for the operating costs of deploying an EN at i. These expenses are found through (5.4) based on two elements: 1) the processing capacity deployed at i, calculated by subtracting the maximum allowable processing capacity (Cc_i) and the capacity required to satisfy the processing demands of the TGs served by the EN at i and, 2) the networking capacity deployed, calculated following the same approach but considering the maximum allowable networking capacity (Cn_i) and the TG networking demands routed through the EN at i. Each addend in (5.4) is multiplied by a capacity-to-cost conversion factor to return a valid cost. The second addend in (5.1) comprises the cost of interconnecting an EN at i with a TG at t, calculated using (5.5) based on the bandwidth of the active links. The third addend in (5.1) represents all upfront deployment costs. These fixed expenses are estimated for each potential EN site selected as EN and it is calculated based on its interconnecting and operational costs when serving a TG (hence, F_i is defined as a

variable in Table 5.1). The objective function in (5.2) aims at minimizing the number of deployed ENs while (5.3) seeks to maximize the EN capacity usage ratio with χ_i calculated through (5.6). Restrictions (5.7) to (5.10) defining the variables and parameters on the model.

In order to solve the multi-objective optimization model, equations (5.1), (5.2) and (5.3) are linearly combined using a "weighted sum" approach to obtain a single objective function [99]:

$$\text{Min} \quad \omega_1 \cdot TC + \omega_2 \cdot NE - \omega_3 \cdot UR \tag{5.11}$$

where TC is the total cost of the EC network, calculated through (5.1), NE is the total amount of ENs deployed estimated using (5.2), UR is the capacity usage ratio of the ENs obtained through (5.3) and $\omega_1, \omega_2, \omega_3 \geq 0$.

The set of restrictions from (5.12) to (5.15) define how the model manages the TG demand and EN capacity interrelation. Both (5.12) and (5.13) ensure that the amount of demand of a TG served by one or more already selected ENs, does not exceed the TG total demand. Likewise, constraints (5.14) and (5.15) guarantee that the amount of demand served by an EN does not exceed the EN maximum capacity. The v_e variable ensure that restrictions from (5.12) to (5.15) are enforced for the locations where an EN has been already placed.

$$\sum_{\forall e \in N} \mu_e^t \cdot v_e = 1 \qquad \forall t \in T \tag{5.12}$$

$$\sum_{\forall e \in n} \kappa_e^t \cdot v_e = 1 \qquad \forall t \in T \tag{5.13}$$

$$\sum_{\forall t \in T} \mu_e^t \cdot v_e \cdot M_t \leq Cc_e \qquad \forall e \in N \tag{5.14}$$

$$\sum_{\forall t \in T} \kappa_e^t \cdot v_e \cdot K_t \leq Cn_e \qquad \forall e \in N \tag{5.15}$$

The restrictions required to define the behavior and interrelation among a selected EN at e (i.e., where $v_e = 1$), serving a TG at t (i.e., where $\alpha_e^t = 1$) and their capacities and demands, respectively, is regulated by the constraints from (5.16) to (5.18). Both (5.16) and (5.17) imply that if a TG is served by a given EN, that EN will serve a fraction of the TG demand higher than zero. Meanwhile, (5.18) forces to zero the compute demand served by any EN potential location where an EN is not placed.

$$\text{if } \alpha_e^t = 1 \Leftrightarrow \mu_e^t > 0 \qquad \forall e, t \in N, T \tag{5.16}$$

$$\text{if } \alpha_e^t = 1 \Leftrightarrow \kappa_e^t > 0 \qquad \forall e, t \in N, T \tag{5.17}$$

$$\text{if } v_e = 0 \Leftrightarrow \sum_{\forall t \in T} \mu_e^t = 0 \qquad \forall e \in N \tag{5.18}$$

Modeling the network-aware nature of the MO-ENPP under strict latency constraints was challenging. Our approach, showcased from (5.19) to (5.21), models the EN-TG interconnection using "*flow conservation*" conditions. Such strategy allowed us to significantly simplify the problem definition when compared to a traditional path-based analysis, while reducing the overall computation time. Through (5.19) and (5.20) the demand entering and exiting both source and destination nodes must be equal to the total demand of the source, considering the reliability requirements of the TGs. Similarly, (5.21) forces the amount of demand entering and exiting any node in between source and destination to be zero.

$$\sum_{\substack{\forall e \in N \\ | e \neq t}} \left(\sum_{\substack{(j,i) \in L \\ | i = t}} \gamma_{ij}^{te} - \sum_{\substack{(j,i) \in L \\ | i = t}} \gamma_{ji}^{te} \right) \geq 1 + R_t \qquad \forall t \in T \tag{5.19}$$

$$\sum_{\substack{\forall e \in N \\ | e \neq t}} \left(\sum_{\substack{(j,i) \in L \\ | j = e}} \gamma_{ij}^{te} - \sum_{\substack{(j,i) \in L \\ | j = e}} \gamma_{ji}^{te} \right) \geq 1 + R_t \qquad \forall t \in T \tag{5.20}$$

$$\sum_{(i,j) \in L | \substack{i \neq t \\ j \neq e}} \gamma_{ij}^{te} - \sum_{(j,i) \in L | \substack{j \neq t \\ i \neq e}} \gamma_{ji}^{te} = 0 \qquad \begin{array}{c} \forall e, t \in N, T \mid e \neq t, \\ n \in N \setminus \{e, t\} \end{array} \tag{5.21}$$

Since the amount of capacity for each link is limited, (5.22) guarantees that this capacity is not exceeded for any link in the EN-TG path selected. Restriction (5.23) defines a link as "*active*" (i.e., $\psi_{ij}^{te} = 1$) whenever it is used to route any amount of existing TG demands (i.e., $\gamma_{ij}^{te} > 0$). The constraint in (5.24) showcases the case where a TG is to be selected as EN in order to serve itself (in case it is required) and no "*active*" network link/path is therefore required. In the event of a TG at t being served by an EN at e (i.e., $\alpha_e^t = 1, v_e = 1$), (5.25) and (5.26) force the routed demand to be greater than zero and viceversa.

$$\sum_{\forall e \in N} \sum_{\forall t \in T} \gamma_{ij}^{te} \cdot K_t \leq B_{ij} \qquad \forall (i,j) \in L \tag{5.22}$$

$$\text{if } \gamma_{ij}^{te} > 0 \Leftrightarrow \psi_{ij}^{te} = 1 \qquad \forall e, t, (i,j) \in N, T, L \tag{5.23}$$

$$\text{if } e = t \Rightarrow \sum_{\forall (i,j) \in L} \psi_{ij}^{te} = 0 \qquad \forall e, t \in N, T \tag{5.24}$$

$$\text{if } \sum_{\forall (i,j) \in L} \gamma_{ij}^{te} > 0 \Leftrightarrow \alpha_e^t = 1 \qquad \forall e, t \in N, T \tag{5.25}$$

$$\text{if } \sum_{\forall (i,j) \in L} \gamma_{ij}^{te} > 0 \Leftrightarrow v_e = 1 \qquad \forall e, t \in N, T \tag{5.26}$$

The 5G latency requirements are comprehensively modeled through (5.27) and (5.28). A maximum latency is assumed in constraint (5.27) for any EN-TG assignment, such that most of the 5G use cases are met for every TG. In addition, (5.28) was defined to guarantee ultra-low

latency requirement satisfaction.

$$\sum_{\forall(i,j)\in L} (D_{ij} + P_i) \cdot \psi_{ij}^{te} + v_e \cdot P_e \leq D_M \qquad \forall e,t \in N,T \tag{5.27}$$

$$\text{if } A_t = 1 \Rightarrow \sum_{\forall(i,j)\in L} (D_{ij} + P_i) \cdot \psi_{ij}^{te} + v_e \cdot P_e \leq D_U \qquad \forall e,t \in N,T \tag{5.28}$$

The core aim with (5.27) and (5.28) is to ensure latency demand satisfaction for a comprehensive set of 5G use cases. For instance, setting $D_U = 1$ ms and forcing the RTT on the EN-TG service path -i.e., for TGs aggregating traffic from ultra-low latency 5G use cases- to be lower than D_U, enforces strict compliance of 5G requirements as presented in [4].

The propagation and processing delays for any path selected to interconnect e and t were considered in both (5.27) and (5.28) (further details on how the path delays are calculated are provided in Section 5.1.2.2).

5.1.2 Solution Proposal: extending EdgeON

By extending the framework presented in Section 4.2.2 to solve the ENPP we aim at providing a useful tool (fully adaptable and extensible) for operators to use when planning the deployment of an EC network.

The extended version of **EdgeOn** comprises a main (i.e., vertical) module containing all the base models used to ensure modularity and extensibility, three core processing stages, and an output/visualization phase (see Fig. 5.1). As in Section 4.2.2, the **Input Processing** stage takes as input and normalizes the 5G use case requirements data (e.g., latency, reliability, etc.) in order to tailor the EN ranked locations to pre-defined 5G demand values. Furthermore, a given territory of interest, network topology (see Fig. 5.2), EN maximum networking/computing capacity and aggregated traffic demands (i.e., TG demands) are assumed to be inputted. In addition to accepting real network topology data as input, the **Scenario Generation** stage of **EdgeOn** implements a network emulator based on the Python library Networkx[1], to provide test scenarios accepting as input an arbitrary number of TGs and network nodes, distributed

[1] `https://networkx.github.io/`

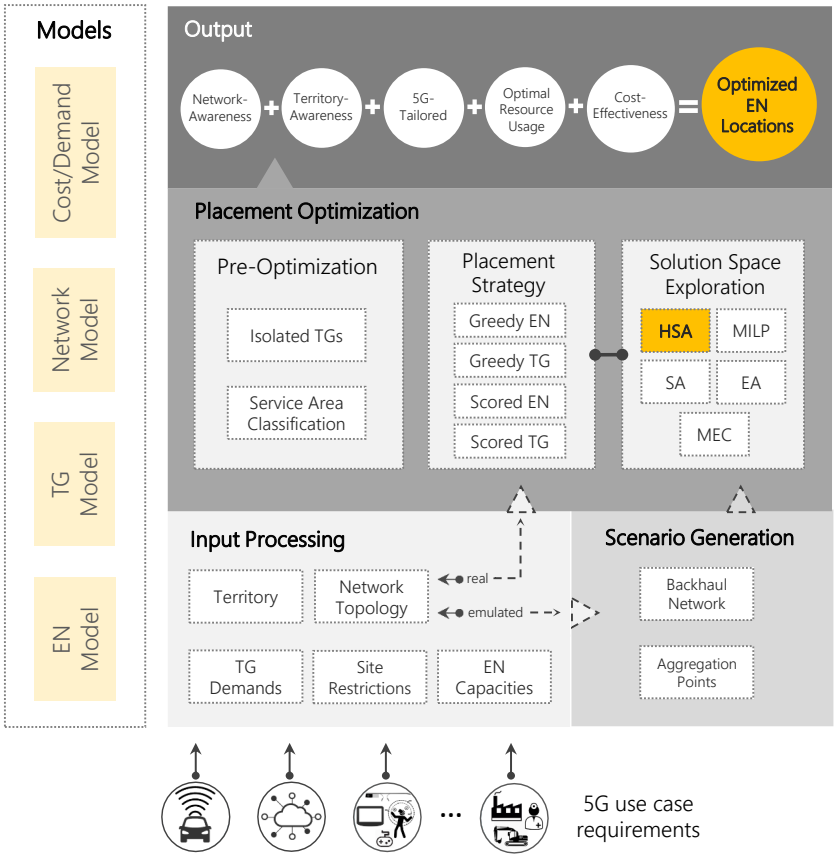


Figure 5.1: EdgeON Architecture.

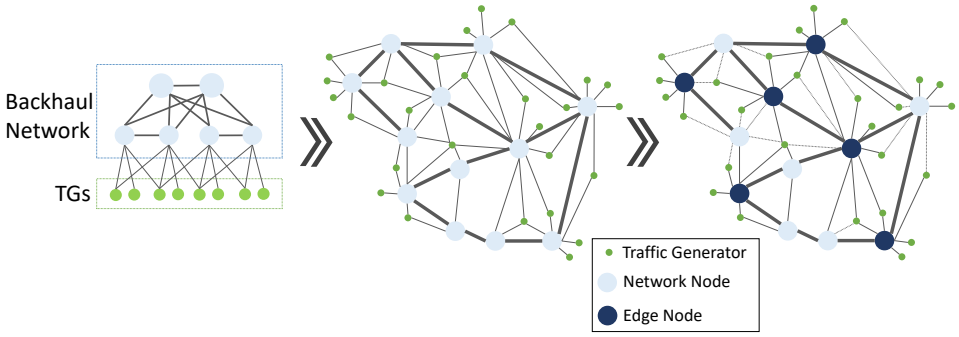


Figure 5.2: Network-aware solution process executed by EdgeON. Logical network diagram on the leftmost image, geographical node distribution on the center and rightmost images (the latter showing an example optimal EN site set).

over a given number of cities (i.e., the topology generator returns an arbitrary number of Wide Area Network (WAN) networks interconnected by a high-speed backbone, thus emulating a country-sized network).

In case a real-life topology is inputted, the **Scenario Generation** stage is bypassed and the framework moves on to the **Placement Optimization** phase. The key modules of **EdgeOn** -i.e., **Placement Strategy** and **Solution Space Exploration**- are executed within this stage. These two steps are tightly coupled, since any method on the **Solution Space Exploration** can use one or more algorithms from the **Placement Strategy** module to generate feasible placement solutions (i.e., TG-EN pairings considering all the underlying restrictions). The current version of **EdgeON** implements four placement algorithms and five solution space exploration methods. Finally, the framework returns an optimized placement solution within the final **Output** stage. All phases of the framework are detailed further in the following subsections.

5.1.2.1 Pre-Optimization Module

The **Pre-Optimization** module within this stage, aiming at reducing the overall problem complexity (as the number of TGs and potential ENs is decreased), seeks for "*isolated*" TGs and divides the territory of interest into Service Areas. A TG is said to be "*isolated*" according to Definition 2, where D_{max} can be either D_M or D_U according to the latency requirements of the TG). Checking the territory of interest in search for isolated TGs is done through Algorithm 2, where $delay(t,e)$ is calculated using the Networkx embedded $shortest\_path()$[2] function to estimate the shortest path delay between an EN at e and TG at t. Namely, after the shortest path between e and t is found, the path delay is calculated considering the sum of the processing and propagation delays of the links and nodes in the path (i.e., the former is assumed to be a fixed known value, the latter is calculated for each link based on the distance and assuming direct fiber connections, Section 5.1.3 specifies the values selected or each parameter). The directly connected nodes or "*neighbors*" for each TG -i.e., obtained by calling $t.neighbors$ in the pseudo-codes shown later in this section- are assumed to be known in advance based on the inputted (or generated) topology, although they can be easily found using Networkx available tools in case a generated topology is used. By determining the "*isolated*" TGs, the resources and execution time required to solve the problem can be effectively reduced as these nodes are immediately upgraded to ENs without loss

[2] `https://networkx.github.io/documentation/stable/reference/algorithms/shortest_paths.html`.

Algorithm 2: Isolated TG Check

Input: D_M, D_U
Output: T^s

```
1
2  for t ∈ T do
3      for e ∈ t.neighbors do
4          if A_t = 1 ∧ delay(t,e) < D_U then
               /* Save e as EN candidate for t                    */
5              t.candidates ← e
6          else if A_t = 0 ∧ delay(t,e) < D_M then
               /* Save e as EN candidate for t                    */
7              t.candidates ← e
8          if t.candidates = ∅ then
9              T^s ← t
10
11 return T^s
```

of generality and accuracy.

On the other hand, the **Service Area Classification** method within the **Pre-Optimization** module aims at a further reduction of the ENPP difficulty. We argue that in rural areas where the user density is typically low and thus TGs are scattered over large geographical areas, a co-location strategy can be used to deploy the ENs. This co-location approach reduces overall costs by minimizing CAPEX, as the required EN capacity is low with high probability and, for instance, a co-located cabinet-based EN-RAN solution, based on wireless connectivity, can be used.

After completing the pre-optimization phase, **EdgeON** is able to execute the core modules of the ENPP solution.

5.1.2.2 Placement Strategies

Although **EdgeON** only requires one placement strategy to solve the ENPP, the reasons to implement several in this thesis were twofold: a) to comprehensively evaluate different solving approaches in order to find the most suitable one for the ENPP as formulated in Section 5.1.1 and, b) to provide potential users of **EdgeON** with a flexible platform and set of methods to easily adapt to their needs and use cases. For this reason, two algorithm types (i.e., EN-TG

pairing methods) and two different implementations for each type were developed as placement strategies: "*greedy*" and "*scored*". The former greedily pairs TGs and ENs considering the TG requirements, available EN capacities, network usage. The latter enhances the greedy strategy by scoring either the TGs or ENs in order to consider the impact of the ENs selected so far over the new EN selection. The placement strategies developed are: Greedy EN (EN-G), Greedy TG (TG-G), Scored EN (EN-SG) and Scored TG (TG-SG).

The pseudo-code for the implementations of the"*greedy*" and "*scored*" strategies are showcased in Algorithm 3 and Algorithm 4. Both methods start by sorting the TG set T such that the more demanding TGs (e.g., $A_t = 1$ or $R_t = 1$) are processed first (Lines 2 and 3 in Algorithm 3 and Algorithm 4). From Line 5 to Line 19 in both placement strategies, each TG t is then analyzed and paired to any EN e to which a feasible path is found through $best\_path(e,t)$. This function is based on a modified version of the Depth-First Search algorithm implemented by Networkx and explained in [100]. It searches and scores all simple paths from e to t (i.e., simple paths with enough network capacity on nodes and links to route t demands) and returns the best path. The path scoring is executed considering three path attributes: total delay from source to target, number of hops, cost (i.e., according to the cost of the active links and the capacity required in the routing nodes), energy consumption (i.e., according to the number of hops, link usage, interconnection technology). In case a valid path is found and the EN at e has enough capacity to serve t (Lines 7-9 and 15 in Algorithm 3 and Lines 7-9 and 17 in Algorithm 4), the EN-TG pairing occurs. The reliability requirement satisfaction is checked in Lines 10-11 and 11-12 for Algorithm 3 and Algorithm 4 respectively. From Lines 13-17 and 15-19, for Algorithm 3 and Algorithm 4 respectively, the TGs with high reliability requirements not yet satisfied are served by greedily choosing suitable ENs. It is worth noticing that a feasibility check looking for non-technical limitations is performed for each e to guarantee that only restriction-free sites are evaluated. In summary, Algorithm 3 greedily selects a feasible EN site to serve each TG, while Algorithm 4 does the opposite process by greedily assigning TGs to each EN.

In order to enhance the TG-EN pairing, both the Greedy EN and Greedy TG algorithms were modified resulting in the Scored EN and Scored TG algorithms (see the **Placement Strategy** in Fig. 4.8). These strategies rely on enhanced pairing methods scoring each EN potential site -i.e.,

Algorithm 3: Greedy EN (EN-G)

Input: N, L, D_M, D_U
Output: E

$T^{hr} = \{t \mid A_t = 1 \ \forall t \in T\}$
$sort(T)$

for $t \in T$ **do**
  $randomize(t.candidates)$ **for** $e \in t.candidates$ **do**
    if $is\_feasible(e) = True$ **then**
      $p_{et} = best\_path(e,t)$ **if** $p_{et} \neq \varnothing \wedge e.avail\_capacity > 0$ **then**
        $E \leftarrow e$
      if $K_t = 1 \wedge len(t \in [T^e, \ \forall e \in E]) \geq 1 + R_t$ **then**
        Remove t from T^{hr}

for $t \in T^{hr}$ **do**
  for $e \in N$ **do**
    if $is\_feasible(e) = True$ **then**
      $p_{et} = best\_path(e,t)$ **if** $p_{et} \neq \varnothing \wedge e.avail\_capacity > 0$ **then**
        $E \leftarrow e$

return E

based on its current usage ratio, capacity cost and non-technical limitations[3]- and each TG to be served, i.e, based on its demand (processing, networking, latency, reliability), impact on the EN capacity usage ratio and number of serving ENs. The path delay calculation includes the transmission and propagation delays corresponding to the links and network nodes traversed from source to target.

5.1.2.3 Solution Space Exploration

Given the strictly constrained and multi-objective nature of the ENPP, the key optimization procedure to be executed goes beyond the TG-EN pairing. Namely, the critical mechanism when solving the ENPP is the exploration of the solution space in order to determine the Pareto front.

[3]If the EN potential site is a PoP -e.g., a Central Office, a ISP-PoP- a score bonus is added to enforce using PoPs as ENs given their potential lower CAPEX/OPEX when compared to, for instance, deploying ENs at TG sites.

Algorithm 4: Greedy TG (TG-G)

Input: N, L, D_M, D_U
Output: E

$T^{hr} = \{t \mid A_t = 1 \ \forall t \in T\}$
$sort(T)$

while $T \neq \varnothing$ **do**
    Select random EN site e
    if $is\_feasible(e) = True$ **then**
        for $t \in T$ **do**
            $p_{et} = best\_path(e,t)$ **if** $p_{et} \neq \varnothing \wedge e.avail\_capacity > 0$ **then**
                $E \leftarrow e$
            if $K_t = 1 \wedge len(t \in [T^e, \ \forall e \in E]) \geq 1 + R_t$ **then**
                Remove t from T^{hr}
        Remove fully served t from T

for $t \in T^{hr}$ **do**
    for $e \in N$ **do**
        if $is\_feasible(e) = True$ **then**
            $p_{et} = best\_path(e,t)$ **if** $p_{et} \neq \varnothing \wedge e.avail\_capacity > 0$ **then**
                $E \leftarrow e$

return E

However, as mentioned before (see Section 2.3) the MO-ENPP defined in this research can be derived to be NP-hard due to its Multi-criteria Multi-attribute FLP nature. All this considered, although exact methods were discarded to solve any variant of the ENPP for mid to large amounts of nodes (cf. Fig. 5.3, showcasing the exponential growth in runtime for the MO-ENPP exact model), the MILP formulation presented in Section 5.1.1 is still included within **EdgeON** for evaluation purposes on small-sized and controlled testing scenarios.

Currently, **EdgeON** implements four solution space analysis methods (i.e., Traditional Simulated Annealing (TSA), HSA and, EA) and a widely used approach for EN placement (i.e., MEC, where the ENs are co-located with the RAN nodes). These algorithms are among the most used to solve complex placement problems and were selected based on their flexibility to be adapted to the particularities of the ENPP, namely, its non-convergent nature within the FLP problem set

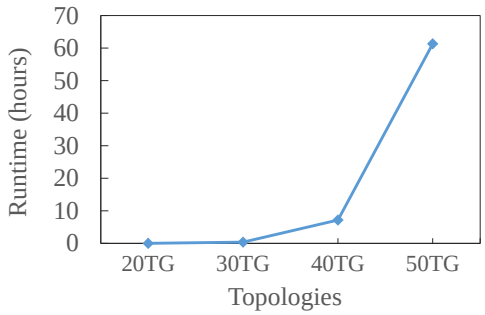


Figure 5.3: Runtime for the MILP model.

and the added difficulties of a network-aware formulation. Nevertheless, due to the promising results of the HSA placement solution this is the default ENPP mechanism used by **EdgeON**. As explained in Section 4.1.2, the key elements in HSA are the memory structures, "*intensification*" and "*diversification*" mechanisms, which combined with the SA core provide HSA with a strong ability to escape local optima and thoroughly explore the problem solution space.

5.1.2.4 Output

The last stage of the framework returns the best solution obtained containing the set of EN locations to place the service infrastructure at the edge of the 5G network and the network paths, link and node usage regarding the TG-EN interconnection. Additionally, **EdgeON** optionally provides both static and interactive charts depicting the deployment details and the performance of the selected placement solutions.

5.1.3 Evaluation and Results

To evaluate **EdgeON**'s suitability to solve the proposed MO-ENPP we conducted several experiments on emulated network topologies varying the number of TGs, the placement strategies and the solution space exploration mechanisms.

The testbed used was developed using the **Scenario Generation** tool embedded within **EdgeON**. Namely, we emulated a geographical area (i.e., a 2D map grid formed by (x, y) coordinate pairs with a 1 m separation step) and, in each experiment, we varied the network topology placed within this area. Each topology generated was formed by a scattered set of TGs and network

nodes (i.e., interconnecting the TGs) randomly scattered resembling a WAN network surrounded by rural territory (i.e., where TGs are separated by a higher distance). All network topologies were generated through the Python library Networkx (i.e., as mentioned in Section 4.2.2) as fully connected undirected graphs with all edges assumed to be fiber optic links. Overall, 9 topologies were tested, with the number of TGs ranging from 20 to 100 nodes (with an increase step of 10 nodes) and the number of network nodes assumed to be half the amount of the TGs within each topology.

The link delay was assumed to be calculated based on the distance between the vertices and each link was assigned either 1 or 10 Gbps capacity based on the link type, i.e., lower bandwidth for the links connecting the TGs to the core network nodes (i.e., access links) and higher bandwidth for the backbone network links (i.e., links where no vertex is a TG). In addition, each routing node within the network was assumed to have a typipcal processing delay of 0.05 ms (i.e., for IP forwarding) [101]. The maximum networking and processing capacities were set to 300 units (i.e., generic units were used to model the bandwidth/processing capacities for the ENs and network nodes) for each EN, while the same network capacity value was assigned to each network node. To obtain this capacity value we ran **EdgeON** 10 times for each topology with randomly selected capacity values. The goal was to find an arbitrary capacity value forcing the worst placement conditions for most of the topologies -i.e., when the majority of the TGs must be served by more than one EN, thus resulting in drastic capacity imbalance and complex EN-TG pairing. Moreover, each TG within each topology was assigned a random processing and networking demand ranging from 20 to 100 units, along with random latency and reliability requirements. The conversion factors τ and σ were set to 10000 $/unit and 700 $/unit to model the general operating costs of deploying an EN considering a realistic scenario [102]. Table 5.2 summarizes the parameter values used of the scenario generation, while Table 5.3 to Table 5.5 present the input parameter values used for the solution space exploration algorithms.

To simulate 5G heavily constrained use cases regarding, for instance, latency and reliability, we assumed a RTT of 1 ms for ultra-low delay requirements and 10 ms for the remaining 5G scenarios (i.e., $D_U = 0.5$ ms and $D_M = 5$ ms). The 1 ms RTT ensures compliance with the identified demands for 5G ultra-low latency use cases [4][7]. Meanwhile, the maximum RTT allowed of

Table 5.2: Parameter values

Model	Param.	Unit	Value	Details
Network	Cn_i	-	300	Generic capacity units were used
	B_{ij}	Gbps	1 - 10	Lower bandwidth for access links, higher bandwidth for core network links
	D_{ij}	ms	-	Estimated based on the distance between nodes assuming a direct fiber link and a propagation delay of 5 μs/km [103]
	P_i	ms	0.05	Typical processing delay for IP forwarding
EN	Cc_i	-	300	Generic capacity units were used
TG	M_t	-	20 - 100	A random processing demand is assigned to each TG
	K_t	-	20 - 100	A random networking demand is assigned to each TG
	A_t	-	0 - 1	Randomly set to 1 (ultra-low latency) or 0 for each TG
	R_t	-	0 - 1	Randomly set to 1 (ultra-high reliability) or 0 for each TG
Cost	τ	\$/unit	10000	Cost per generic capacity unit
	σ	\$/unit	700	Cost per generic capacity unit

Table 5.3: Input parameters for the EA.

Parameter	Value
Num. Generations	100.00
Num. Individuals	100.00
Mutation rate	0.01

Table 5.4: Input parameters for the HSA.

Parameter	Value
Minimum Temperature	0.0001
Maximum Temperature	1.0000
Temperature Iterations	10.000
Fast Alpha	0.8000
Slow Alpha	0.9500
Num. Neighbors	10.000

Table 5.5: Input parameters for the TSA.

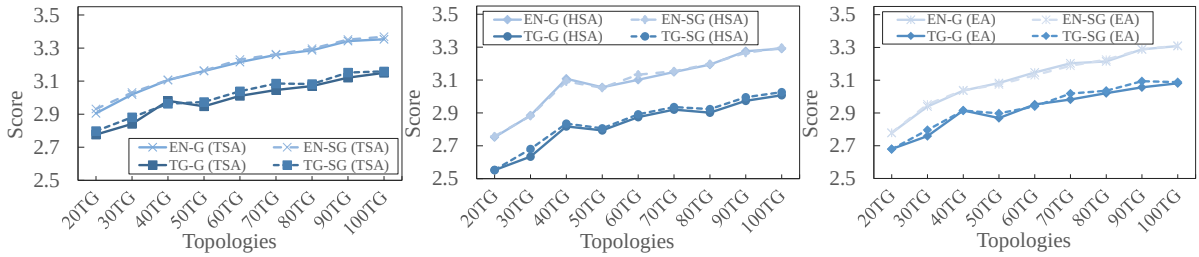
Parameter	Value
Minimum Temperature	0.0001
Maximum Temperature	1.0000
Temperature Iterations	10.000
Alpha	0.9500
Num. Neighbors	10.000

10 ms, for any EN-TG pairing, guarantees that most 5G use cases can be met for any TG and its serving ENs [4].

For the objective function we arbitrarily selected the normalized weights $\omega_1 = 0.35$, $\omega_2 = 0.33$, $\omega_3 = 0.32$. Similarly, arbitrary values were selected for the weights in the $best\_path(e,t)$ function.

The first step towards a comprehensive evaluation of **EdgeON**'s capabilities was to determine the best placement strategy to solve the ENPP, due to the critical impact of the EN-TG pairing on the overall performance of the solution. To this aim, we repeatedly ran the TSA, HSA and EA algorithms for all placement strategies and topologies. The results are showcased in Fig. 5.4.

Taking into account that the lower the score the better the performance, for all the topologies



(a) Performance using TSA. (b) Performance using HSA. (c) Performance using EA.

Figure 5.4: Evaluation of the placement strategies for all solution space exploration algorithms and network topologies with TG values ranging from 20 to 100 TGs. The naming convention is as follows: Greedy EN → EN-G, Greedy TG → TG-G, Scored EN → EN-SG and, Scored TG → TG-SG.

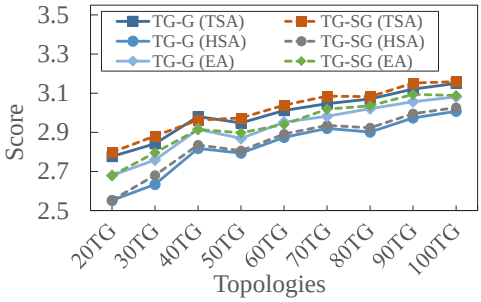


Figure 5.5: Performance results for the TG-G and TG-SG placement strategies for all topologies analyzed.

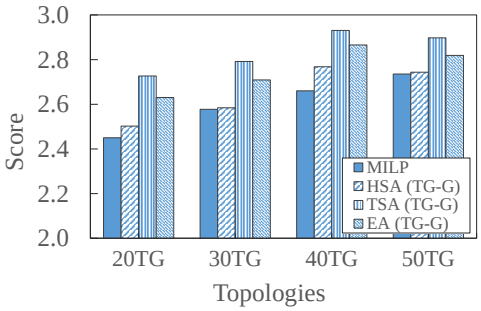
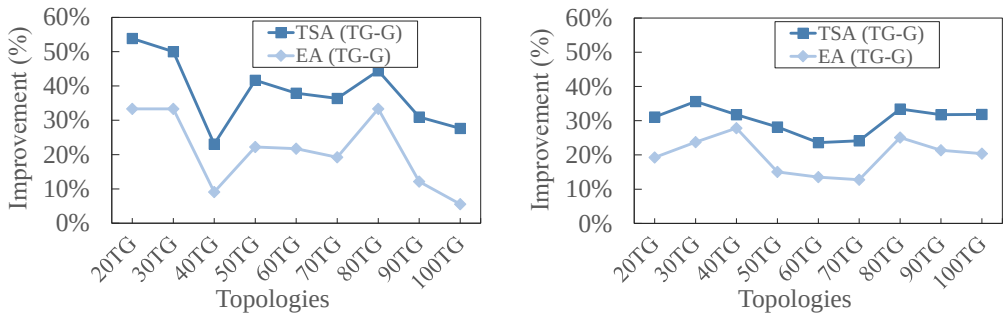


Figure 5.6: Performance results for the MILP model compared to the solution space exploration heuristics (i.e., HSA, TSA, EA).

analyzed (i.e., named after the number of TGs on the topology), the Greedy EN (EN-G) and Scored EN (EN-SG) were significantly outperformed by both the Greedy TG (TG-G) and Scored TG (TG-SG). The reason is that greedily assigning feasible ENs to each TG results in a poor usage ratio balance and higher number of ENs when compared to selecting random ENs and greedily pairing them with suitable TGs, considering the underlying capacities and TG requirements. Consequently, we discarded EN-G and EN-SG as placement strategies in favor of TG-G and TG-SG for the remaining of our experiments.

A different perspective to further analyze the placement strategies performance is shown in Fig. 5.5. Crosschecking the charts in Fig. 5.4 and Fig. 5.5 evidences the superiority of TG-G and TG-SG for any solution space exploration mechanism. For all topologies analyzed, both TG-G and TG-SG outperformed the remaining placement strategies, resulting in significantly lower



(a) Improvement achieved by HSA regarding the number of ENs deployed.

(b) Improvement achieved by HSA regarding the average EN capacity Usage Ratio.

Figure 5.7: HSA results for Num. ENs and Usage Ratio compared to TSA and EA (using TG-G in all cases). The HSA improvement percentage is depicted.

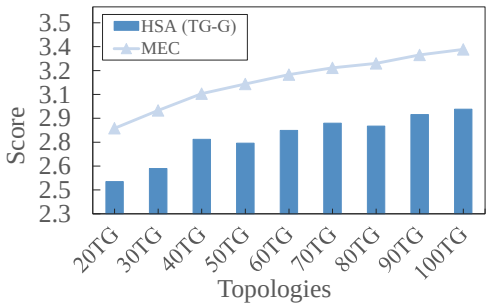


Figure 5.8: Performance results for HSA and MEC using TG-G as placement strategy.

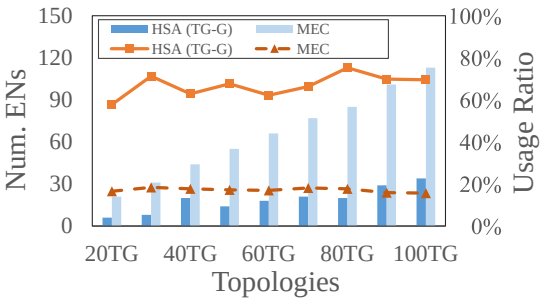


Figure 5.9: Performance results for HSA and MEC using TG-G as placement strategy in terms of Average Usage Ratio (lines, left y axis) and Num. ENs (columns, right y axis).

costs and number of deployed ENs and in higher average usage ratio, thus lowering the overall score. In addition, this figure evidences how TG-G performed slightly better than TG-SG for all algorithms and the majority of topologies analyzed. Consequently, we set TG-G as the default placement strategy to solve the ENPP using **EdgeON**.

The second step on **EdgeON**'s analysis was to thoroughly assess the solution space exploration strategies. The idea within this step was to evaluate **EdgeON**'s ability to find the best near-optimal solution using our in-house heuristic (i.e., HSA) tested against the exact method, in a controlled test scenario -i.e., reduce number of nodes- and against widely used heuristics commonly applied to other placement problems. The findings of these tests are depicted in Fig. 5.6 and Fig. 5.7. The former showcases the superiority of HSA when compared to the other

heuristics, with an average score offset of around 1.5% compared to the MILP model[4], i.e., with EA and TSA achieving a score offset of 6% and 8% respectively. The significantly better score offset obtained by HSA when compared to TSA and EA, showcased in Fig. 5.7, resulted from its performance improvements in terms of number of ENs and average capacity usage ratio. Overall, Fig. 5.7 illustrates that HSA deployed an average of nearly 40% less ENs than TSA and 20% less than EA. Moreover, HSA achieved a 30% and 20% better usage ratio when compared to TSA and EA respectively.

Finally, to further validate HSA's suitability for EN deployment within 5G networking, we tested it against a commonly preferred strategy to locate ENs: the MEC approach, where as mentioned above, the service infrastructure (i.e., the EN) is arbitrarily co-located with the RAN nodes. As expected, Fig. 5.8 evidences how using MEC can lead to a rather inefficient EC network deployment when compared to HSA, since it results in lower usage ratio, higher number of deployed ENs and performance degradation due to overlooking the in-place backhaul network capacity. In summary, the MEC approach placed an average of 71% more ENs than HSA (using TG-G as placement strategy) and resulted in 50% less average usage ratio for the vast majority of the analyzed scenarios (cf. Fig. 5.9).

The aforementioned results encourage the evaluation and test of **EdgeON** on real-life scenarios and network topologies. Furthermore, its modular implementation ensures an easy-to-use and extensible platform for operators to adapt to their requirements and use cases.

5.2 Conclusion

This chapter rigorously defines the network-aware ENPP under heavily constrained 5G scenarios and significantly extends **EdgeON** in order to solve the presented problem.

In Section 5.1.1 the mathematical definition and MILP model for the network-aware ENPP was presented, considering a 5G strictly constrained ecosystem. Flow conservation conditions were used to deal with the challenges derived from interconnecting ENs and TGs through

[4]Results shown for topologies with less than 50 TGs due to the exponential increase in runtime for the MILP model when applied to topologies with more than 50 nodes

a realistic network topology. In addition, a multi-objective model was developed to ensure a comprehensive optimization of the EN placement, addressing not only the overall expenses minimization, but the optimization of the number of EN and their capacity balance across the EC network.

Aiming at achieving a key goal of this thesis (i.e., to provide a useful tool for the deployment of an EC network), the **EdgeON** framework was redesigned and extended in Section 5.1.2. The new version of **EdgeON** was developed focusing on flexibility and extensibility, while comprising a thorough analysis of the technical and non-technical aspects and costs of the network-aware EN placement.

To validate the capabilities of this new version of the framework, the performance of its core placement optimization solution (i.e., based on HSA), was thoroughly assessed using several strategies as core placement methods. The promising results obtained encourage the use of **EdgeON** to solve the network-aware ENPP under strict 5G use case requirements. Namely, significant improvements were achieved regarding the number of ENs deployed and average usage ratio (i.e., around 30% and 25% on average, respectively, compared to the remaining tested heuristics). Moreover, an average score offset of just 2% was obtained when testing our heuristic against an exact method (i.e., MILP model).

CHAPTER 6

VNFs over optimally placed ENs: Case Studies

This chapter is based on:

- A. Santoyo-González, C. Cervelló-Pastor and D. P. Pezaros, "High-performance, platform-independent DDoS detection for IoT ecosystems," 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 2019, pp. 69-75.
- A. Santoyo-González and C. Cervelló-Pastor, "A Framework for Latency-constrained Edge Nodes Placement in 5G Networks," in *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, (Granada, Spain), 2018.
- I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "A Framework for the Joint Placement of Edge Service Infrastructure and User Plane Functions for 5G, *Sensors*, vol. 19, no. 18, p. 3975, 2019."

This chapter extends the evaluation and validation of the proposed solution to optimally place the ENs under 5G requirements. To achieve this goal, in the following sections two 5G scenarios are presented where core VNFs are placed over ENs.

In Section 6.1, a DDoS edge-based detection solution is presented for an IoT ecosystem, aiming

at line-rate processing, platform-independent and lightweight execution, taking advantage of the service infrastructure assumed to be optimally located at the users' premises, thus extending the work presented in the previous chapters.

Similarly, Section 6.2 showcases a solution to optimize the placement of UPFs through a modified version of **EdgeON** solving the joint placement of 5G UPFs and ENs. Namely, **EdgeON** is extended to integrate a novel solution for UPF placement over EC infrastructure under 5G latency and reliability requirements, with additional mobility constraints.

6.1 Case Study 1: High-performance, platform-independent IoT-DDoS edge-based detection

Given the nature of IoT deployments where millions of end devices acquire networking capabilities, IoT-DDoS attacks (i.e., IoT devices forming botnets) have emerged as a challenge due to the number of forecasted devices in 5G networks and their inability to be easily patched [104]. So far, solutions against DDoS attacks in this context have been implemented through complex, centralized software and hardware-based mechanisms [105]. Distributed detection and mitigation techniques have been studied, aiming at offering more efficient ways of dealing with scenarios such as IoT-DDoS attacks [106–108].

The vast majority of these approaches assume the use of purpose-built middleboxes deployed in the network, close to the victim, in order to be able to detect the attack through analyzing aggregated traffic features [104][107][109–111]. However, in an IoT environment, the cost of a purpose-built ecosystem to detect and mitigate DDoS poses complex deployment and operational challenges, for instance, if early detection and high-speed processing is to be achieved.

What is more, most current DDoS detection schemes rely on traffic redirection methods or aggregated flow statistics collection. These mechanisms introduce additional costs and performance issues into the network (e.g., longer flow completion times, bandwidth overhaul), degrading the system's effectiveness due to longer timeframes between detection and mitigation phases [112]. To partially overcome such problems, multi-stage distributed systems can be used to detect and mitigate the attacks. In these approaches, coarse-grained detection is to be executed upstream in

the network, closer to the attackers.

However, this leads to the use of dedicated middleboxes scattered across the network for scrubbing purposes [110][107]. For an IoT-DDoS detection solution (i.e., protecting the network against DDoS originated on IoT devices) to solve the above mentioned problem, it has to ensure: **a) lightweight processing**, by relying on traffic features and analysis methods targeting overhead minimization and coarse-grained anomaly detection; **b) platform-independence**, to minimize the need for purpose-built devices and the use of traffic redirection-based approaches; and **c) high-performance**, in order to achieve fast reaction through early detection while avoiding performance degradation.

The advent of paradigms such as EC and SDN can help overcome the aforementioned issues for DDoS detection and mitigation. Through EC and SDN dataplane programmability principles, lightweight functions can be placed at the edge, resulting in enhanced network capabilities. Namely, a programmable dataplane improves the network's agility and flexibility by allowing dynamic high-speed edge function allocation/deallocation. Thus, providing early detection capabilities and more efficient resource usage at the edge nodes. Additionally, edge network functions deliver the elasticity and scalability required to efficiently handle vast amounts of traffic in a distributed manner. IoT can directly benefit from the joint work of data-plane programmability and edge network functions. By placing the detection at the attackers' vicinity (see Fig. 6.1), fast reactive procedures can effectively isolate the IoT malicious devices while reducing bandwidth consumption typically produced by DDoS attack traffic, and avoiding the processing overhead of current remote centralized detection approaches.

In the following sections, we present a lightweight, platform-independent anomaly detection mechanism to be deployed at the edge of the network. To achieve true platform independence while ensuring high-performance levels, our proposal is based on BPFabric, a data-plane programmability architecture presented in [113]. In a nutshell, the BPFabric platform allows to program the data-plane of SDN network nodes and therefore, it can be partially considered complementary to other solutions, e.g., P4[1]. Unlike the latter, however, BPFabric focuses on

[1]https://p4.org/

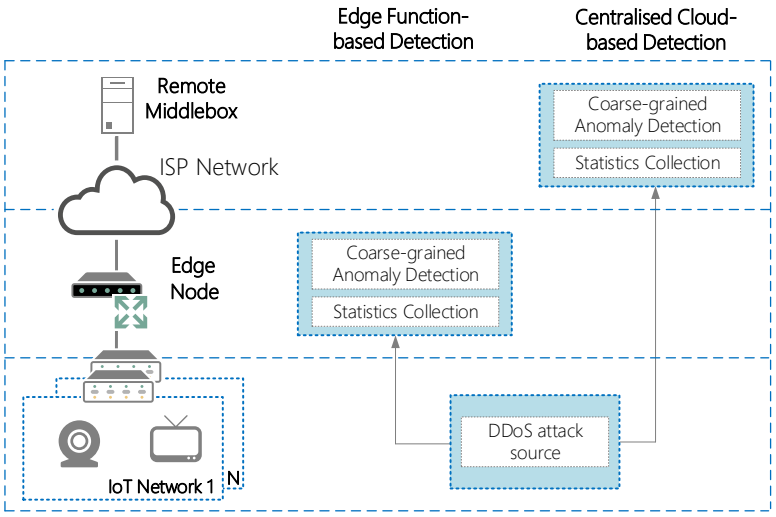


Figure 6.1: Centralised Cloud vs. Edge Functions-based Detection

high-speed processing and, for that reason, it is based on the eBPF [114] instruction set, rather than a higher-level Domain Specific Language (DSL).

Previous work has successfully tested the line-rate capabilities of eBPF [113–115]. Namely, it has been demonstrated that eBPF-based packet processing, by acting at the socket level, significantly improves both throughput and latency, while still offering the advantages of kernel integration (i.e., full network stack processing) when required. BPFabric provides true platform-independent execution on account of eBPF, avoiding the Protocol Independent Switch Architecture (PISA)-based device restriction imposed by P4. Furthermore, BPFabric goes beyond the data-plane programming capabilities of P4 by defining a fully developed architecture specifying: the SDN controller and remote agent behaviors, the controller-agent interactions, mandatory core packet processing functionalities and message exchange procedures. The framework hence allows to define and deploy diverse network functions as part of the forwarding behavior of each switching element, from a remote centralized location.

Overall, the main contributions presented in the following sections are: 1) the design and implementation of a lightweight, platform-independent anomaly detection mechanism based on edge functions defined as part of the BPFabric architecture, exploiting SDN-based data-plane programmability, 2) the implementation of an eBPF-based detection method using Shannon's Entropy and Exponentially Weighted Moving Averages (EWMA) and, 3) the evaluation of our

edge-based anomaly detection scheme against a fully centralised cloud-based approach considering carefully selected traffic features (see Section 6.1.1) matching the particularities of traffic anomalies in IoT ecosystems.

6.1.1 Solution proposal

Typical DDoS attacks are said to be characterized by high frequency of incoming packets, endpoint communications asymmetry, and high number of source Internet Protocol (IP) addresses [107, 110]. However, such characteristics are directly linked to a close-to-target detection approach and fail to describe IoT-DDoS if they are to be detected at the attackers' vicinity [116]. For instance, by pushing the detection mechanism to the network edge, the benefits from traffic aggregation are lost and hence, outliers such as high packet rate/volume and source IP diversity cannot be considered. For a joint scenario mixing IoT and upstream attack identification, a tailored set of metrics is required. The works in [110, 111, 116, 117] provide a solid baseline for a set of metrics in order to identify anomalous traffic in IoT environments. Based on these findings, the set of IoT-DDoS detection parameters used in this work is summarized below:

Destination/Source IP Address Distribution: given their reduced functionality scope, IoT devices usually communicate with a small set of endpoints. Therefore, anomalous traffic can be identified by analyzing the destination IP address distribution [110, 111]. Furthermore, DDoS attacks usually employ forged source IPs to communicate with a victim host. Therefore, to effectively identify abnormal traffic, the destination/source address space entropy can be used. According to the findings from [111], IoT devices should mostly have an overall low entropy. As a consequence, any change in the entropy value over a given timeframe can be considered a sign for an ongoing attack.

Flow Asymmetry: during a DDoS attack, the interaction between the attacker and the target has been found to be asymmetrical [110]. Under a DDoS attack from an IoT botnet, the underlying IoT devices send a high number of requests to the victim. Eventually, the target capacity is exceeded and the symmetry of outgoing requests and incoming responses is affected, a situation that can be identified by detection methods placed at the attacker's vicinity. To use traffic asymmetry as a detection feature, the method presented in [110] is adapted and used in

this paper (see Section 6.1.2).

Inter-packet Interval: within the time domain, the traffic patterns of IoT devices are often quite stable with each device sending information to, for example, remote control systems at clearly pre-defined, arbitrary, and immutable time intervals. In contrast to regular IoT traffic, DDoS attack incoming traffic from an IoT device is often characterized by high burstiness in short and periodic timeframes [109, 111, 116].

Packet Size: under a DDoS attack, the packet size distribution for IoT devices varies greatly over time. Typically, malicious traffic comprises bursts or steady flows of incoming small packets around 100 bytes, while normal traffic packet sizes are unevenly distributed from 100 to more than 1000 bytes [111]. This behavior allows us to detect anomalous traffic by analyzing the packet size variation -i.e., number of packets with length under 100 bytes- over arbitrary controlled timeframes.

Packet Volume: the transferred data volume is a key parameter when detecting volumetric DDoS [111]. Given the reduced and typically fixed amount of traffic periodically sent by IoT devices, analyzing the packet volume at the network edge can effectively lead to detect an ongoing attack.

6.1.1.1 Edge-based detection

Leveraging SDN data-plane programmability and EC principles, coarse-grained detection mechanisms can be deployed at the network edge close to potential attackers (i.e., IoT devices). This can be achieved through in-line edge functions and technologies tailored to the edge node resources and characteristics. BPFabric allows functions to be implemented at the network edge, encoded as part of the data-plane behavior of the device (e.g., a switch). Therefore, BPFabric provides the added flexibility of being able to deploy the system on a wide variety of devices already in use at the user's vicinity (e.g., home gateways, access routers).

When selecting the anomaly detection mechanism, the inherent limitations of the edge nodes (e.g., limited resources, rigid programmability), the goal of achieving line-rate performance to avoid throughput or latency degradation, for instance, and the need for fast detection, significantly reduce the list of mechanisms that can be used. For instance, complex detection techniques based

on machine learning require high processing capabilities unavailable on the data path of an edge node. Moreover, the use of BPFabric, based on the eBPF instruction set, introduces additional particularities (e.g., limited program size) that should be taken into account in order to achieve high-speed and bound execution time. Taking the above into consideration, the coarse-grained detection running at the edge is forced to be fairly simple, while still ensuring an adequate level of accuracy.

In an IoT context, we believe such tradeoff can be sorted out through adequate traffic statistics collection combined with multi-feature analysis. Such an idea is supported by the nature of the attack traffic characteristics identified above (see Section 6.1.1). For example, let us consider a DDoS Transmission Control Protocol (TCP) SYN flood attack with variable packet burstiness and overall low packet volume. Through a joint evaluation of the volume, destination IP addresses and inter-packet intervals, the attacker could be pinpointed. This is possible because the anomaly detection system is able to conclude that, for instance, for a certain target IP, the traffic burstiness and packet volume (e.g., average packet size under 100 bytes) do not follow the expected behavior.

Nevertheless, as the detection method is forced to be simple and even following the above multi-feature analysis approach, the detection accuracy will highly depend on the attack complexity and the dynamic adjustment of the mechanisms (e.g., thresholds) used to detect a suspicious event. To overcome these limitations and based on the promising results found on previous work [106–108], we envision a multi-stage detection architecture where the coarse-grained detection is carried out close to the attackers, and the upper and more advance analysis layers can be executed in more powerful edge nodes scattered within the service provider network (either in a centralized or distributed fashion).

Overall, the idea behind such scheme is to periodically collect traffic information through eBPF filtering rules on the IoT network gateways (see Fig. 6.1). The detection analysis is carried out through a pipeline of condition evaluation steps injected into the IoT gateway running BPFabric (the data collection is also part of the eBPF program inserted). If any anomalous behavior is found, an alarm is then sent to the upper detection layers on the architecture via the controller (assuming an SDN implementation). In case an anomaly is found, the upper layer executes further processing (after requesting additional information if required) and confirms

if an attack has been made. In the event of a false positive, the detection parameters on the coarse-grained mechanism are to be adjusted to increase its accuracy.

In order to comply with the above mentioned system limitations, we decided to use Exponential Weighted Moving Average (EWMA) and Shannon's Entropy for outlier detection. After the data collection interval finishes, the EWMA (according to Eq. 6.1) is calculated for the following features: packet count, rate, volume, and size distribution. Flow completion (for the traffic asymmetry feature) is determined through source/destination IP address pairs, keeping track of the number of outgoing communications and the associated responses. Finally, the endpoint/source variation over time is checked based on the source/destination IP entropy (referred to as $H(X)$) calculated using Eq. 6.2 [107].

$$EWMA = \alpha \cdot value + (1-\alpha) \cdot last\_prediction \tag{6.1}$$

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i) \tag{6.2}$$

6.1.2 Evaluation and Results

We conducted experiments to evaluate the suitability of the proposed EWMA and Shannon's Entropy-based detection within BPFabric architecture (for convenience this method is hereinafter referred to as "*ESE-Detection*") to effectively detect IoT-DDoS.

The testbed used in our experiments is shown in Fig. 6.2. A set of IoT networks is emulated, connected through access routers to the WAN/MAN, and finally to the remote cloud where the attack targets are located. The metrics for the scenario analysis are presented in Table 6.1. They are selected in order to thoroughly assess the behavior of our solution and its overall performance.

The detection pipeline collecting traffic data and executing EWMA and Shannon's Entropy-based detection is run in node GW1 (i.e., the IoT gateway). The cloud-based detection is executed within the emulated remote cloud collecting traffic data from node R1. Two additional detection methods were implemented for evaluation purposes: Cosine Similarity [109] and Shannon's Entropy [107, 118]. Both detection strategies were adapted to use the metrics presented in Section 6.1.1, and were selected considering their previous use in coarse-grained DDoS detection [107, 109]. Since a thresholding approach was adopted for all detection strategies, the methodology

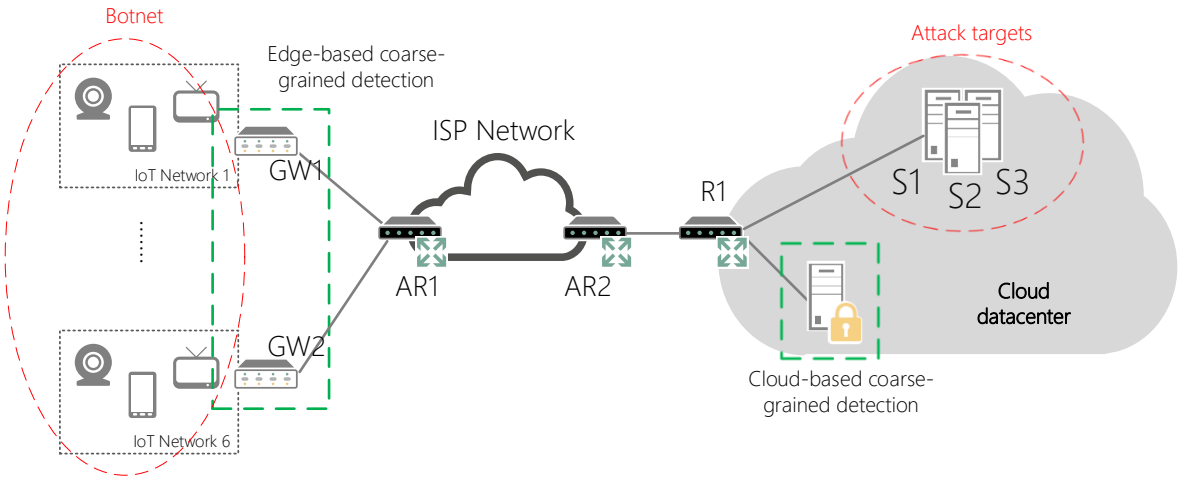


Figure 6.2: Testbed architecture used for edge-based IoT-DDoS detection

Table 6.1: Evaluation Parameters

Parameter	Description
Bandwidth	Bandwidth overhaul caused by traffic redirection or data collection in centralized cloud-based approaches.
Detection time	Time elapsed between the anomaly occurrence until an alarm is raised.
Attack penetration	Amount of anomalous traffic (attack packets) inserted into the network until an alarm is raised.
Accuracy	The false positive and false negative ratio achieved.
Cost	Overall expenses based on [119]: cost of information gathering, data processing and detection implementation.

presented in [118] was used to optimize the threshold selection process and enhance the overall accuracy. To emulate an attack, we developed a Python script using the Scapy library[2] to generate spoofed source address and destination ports, targeting an arbitrary server within the remote cloud in Fig. 6.2 simulating a TCP SYN flood attack.

To generate the IoT traffic for the experiments, the tool "*Distributed Internet Traffic Generator*" [120] was selected due to its flexibility and granularity in controlling the traffic characteristics. Furthermore, the findings presented in [121] and [122] allowed us to model IoT traffic of a home network, assuming each setup comprises the following elements: 3 smart appliances (e.g.,

[2]https://scapy.net/

Table 6.2: IoT traffic simulation details

Feature	Smart Appliance	Climate & Lighting Control
Dst. IPs	2 ~ 10	2 ~ 5
Num. Dst. Ports	2 ~ 5	2 ~ 5
Avg. Load (Kbps)	5 ~ 25	5 ~ 15
Packet Size (bytes)	100 ~ 600	100 ~ 200
Act./Idle intervals (s)	2 ~ 10 / 80 ~ 100	2 ~ 5 / 10 ~ 20

refrigerators, washers), 4 climate control sensors and 6 lighting control devices.

The values in Table 6.2 were assumed to generate the device data flows and model the device normal behavior. Mininet[3] was employed to emulate the IoT network due to its simplicity and flexibility. Overall, a round-trip delay of 100 ms was assumed for the end-to-end communication from the IoT networks to the servers in the remote cloud, accounting for the processing, routing/switching, and propagation delays involved.

Estimating the entropy of the IP distribution was quite challenging considering the limitations of the envisioned underlying hardware (e.g., no support for float point operations) and the eBPF instruction set characteristics. Consequently, Eq. 6.2 was adapted to overcome these restrictions. To efficiently find the base 2 logarithm we adapted the Taylor Series expansion method described in [123], hence approximating the base 2 logarithm through Eq. 6.3. The K constant value defined in Eq. 6.4 was calculated beforehand and predefined in the eBPF program.

$$\log_2(\frac{x}{y}) = K \cdot (-\log(\frac{x}{y})) \qquad K \in \mathbb{R},\ \ x, y \in \mathbb{Z} \tag{6.3}$$

$$K = \frac{-1.0}{\log(2.0)} \tag{6.4}$$

Eq. 6.5 allowed us to effectively approximate the logarithm of x/y (e.g., x: destination IP count, y: total destination IPs).

$$\log(\frac{x}{y}) = a + \frac{a^2}{2} + \frac{a^3}{3} + .. + \frac{a^n}{n} \qquad a \in \mathbb{R},\ \ n \in \mathbb{Z} \tag{6.5}$$

$$a = \frac{(y-x)}{y} \tag{6.6}$$

$$\frac{y^{N-1} \cdot (2 \cdot 3 \cdot 4 \cdot ... \cdot N) \cdot (y-x) + y^{N-2} \cdot (1 \cdot 3 \cdot 4 \cdot ... \cdot N) \cdot (y-x)^2 + ...}{y^N \cdot (1 \cdot 2 \cdot 3 \cdot .. \cdot N)} \tag{6.7}$$

[3] http://mininet.org/

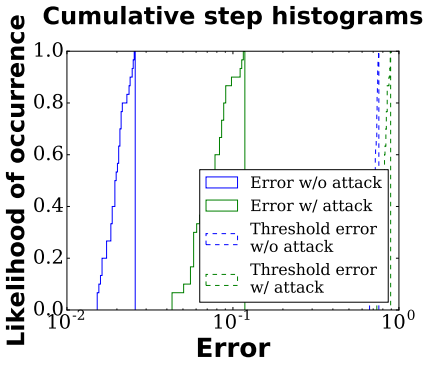


Figure 6.3: Entropy estimation error.

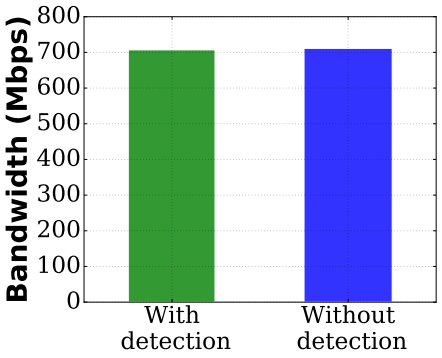


Figure 6.4: Bandwidth consumption with and without detection processing.

Unrolling and computing Eq. 6.5 as a running product allowed us to handle all operations using integers and comply with the unbounded loop restrictions in eBPF. As a result, -log(x/y) was implemented as shown in Eq. 6.7. Where N is an arbitrarily chosen integer ($N = 2$ was empirically selected, the estimation error analysis is shown in Fig. 6.3) providing the desired precision. Finally, to compute the entropy for the destination IP variation, for instance, fast map iteration through eBPF `bpf_map_get_next_key`[4] was employed and eBPF maps were used as immutable global counters when needed.

To enhance the accuracy while avoiding register overflow (i.e., likely to occur for large N values), we decided to multiply the numerator by 1000. Consequently, the entropy estimation resulted in an integer comprising up to three of the decimal values of the real result (e.g., for an entropy equal to 1.123, the estimated entropy found was 1120). The performance and resource overhead is minimized by removing user-kernel space interactions, as all computations on the detection pipeline are executed within kernel space. The bandwidth analysis depicted in Fig. 6.4 shows the minimum performance impact of ESE-Detection processing, causing a bandwidth reduction of around 1%.

Although the estimation error is thoroughly described in [123], we decided to conduct experiments to determine the impact of the entropy estimation over the detection accuracy. The findings can be observed in Fig. 6.3 where the cumulative step histogram for the estimated entropy error

[4]http://man7.org/linux/man-pages/man2/bpf.2.html

is shown. Overall, the estimation error fell most of the times within 1% to 3% for typical IoT traffic and within 6% to 11% in the event of an attack. This precision level gave us sufficient margin to effectively determine an anomaly was occurring. The estimation error was significantly lower than the error required to incur in a miss-detection, represented by the right-most dashed lines in Fig. 6.3, for both regular and attack traffic. In order to result in a false positive/negative, our estimation error should have been at least 60% higher in any case.

6.1.2.1 Results

Fig. 6.5 shows the cumulative step histogram for the detection delay for each of the implemented schemes. As expected, the performance of the BPFabric approach is significantly better due to the low processing delay introduced by the enhanced data plane pipeline. The eBPF-based detection engine is able to reduce over 80% of the anomaly identification time when compared to both Entropy and Cosine Similarity, lowering the detection delay to an average of less than $5ms$. Such results show the potential of BPFabric for early anomaly detection. Powered by its high-speed and lightweight processing potential, BPFabric-based detection is capable of greatly reduce the data processing overhead, thus resulting in significantly lower detection timescales. Since the BPFabric detection is running in kernel space, an inherent limitation is the lack of access to a proper timer due to the absence of an eBPF in-kernel function for this purpose (within the scope of the eBPF program type we are running). As a solution, the timing is followed using the incoming packet timestamps provided by BPFabric.

The accuracy of the detection methods was measured using the typical False Positive Ratio (FPR) and False Negative Ratio (FNR) definitions, as shown in Equation 6.8 and Equation 6.9. Maintaining per-flow data statistics using in-kernel processing on an resource-constrained IoT gateway is unfeasible due to: memory requirements to hold the generated data in the event of an attack, eBPF map limitations and performance degradation due to longer processing timeframes. Consequently, the detection analysis was not performed considering the benign/malicious flow count. Instead, we decided to run several experiments executed at both fixed and random time intervals, in order to emulate a more realistic botnet scenario, while measuring the accuracy through the number of attacks detected by the implemented methods. The results are presented

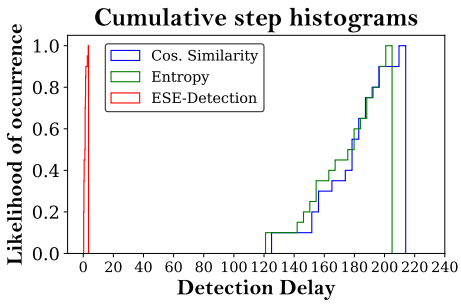


Figure 6.5: Detection delay analysis for the proposed methods: ESE-Detection, Entropy and Cosine Similarity.

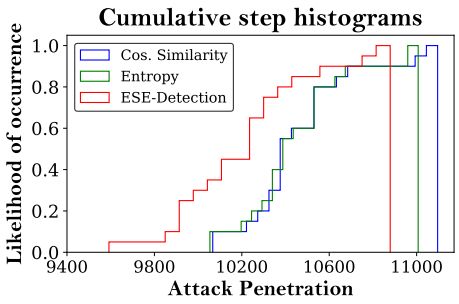


Figure 6.6: Attack penetration (in packets) for the proposed methods: ESE-Detection, Entropy and Cosine Similarity.

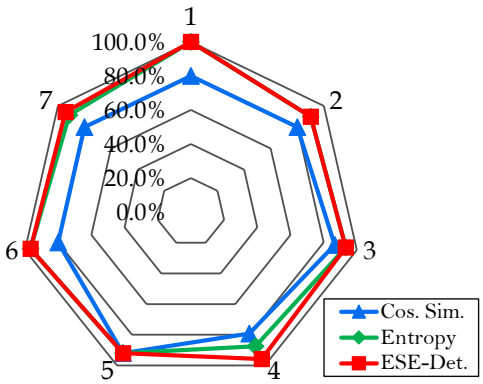


Figure 6.7: Accuracy for attacks executed at fixed intervals (number of attacks varying from 1 to 7) for the proposed methods: ESE-Detection, Entropy and Cosine Similarity.

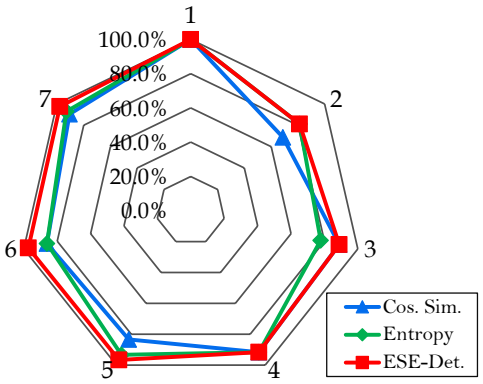


Figure 6.8: Accuracy for attacks executed at random intervals (number of attacks varying from 1 to 7) for the proposed methods: ESE-Detection, Entropy and Cosine Similarity.

in Fig. 6.7 and Fig. 6.8, where the number of executed attacks is depicted, alongside the attacks detected by each algorithm.

$$\text{FPR} = \frac{FP}{(FP + TN)} \tag{6.8}$$

$$\text{FNR} = \frac{FN}{(FN + TP)} \tag{6.9}$$

Where:

- FP: Number of false alarms
- FN: Number of undetected attacks

- TN: True benign traffic
- TP: Number of detected attacks

Throughout our experiments, ESE-Detection had an average accuracy of around 95%, superior to the entropy and cosine similarity strategies (93% and 83%, respectively) for the experiments where the attacks were executed at fixed intervals (see Fig. 6.7). For the randomly timed SYN flood attacks, ESE-Detection outperformed again the remaining methods, achieving an average of 93% versus 88% and 86% of entropy and cosine similarity respectively (see Fig. 6.8). Overall, ESE-Detection superior accuracy can be expected for this ecosystem given its segregated view of the traffic (i.e., detection executed closer to the attackers). Conversely, the typical traffic patterns of IoT devices cannot be effectively analyzed through cloud-based scrubbing due to the traffic convergence.

Some interesting facts were found when estimating FNR/FPR. The ESE-Detection engine was able to ensure less than 20% FNR for both fixed and randomly timed attacks in the worst case scenario, surpassing the maximum of 50% found for cosine similarity and entropy. On the contrary, both these methods performed slightly better, overall, than ESE-Detection regarding the FPR. ESE-Detection reached a maximum of 33%, equaling the cosine similarity results and below the 25% reached by the entropy method. However, ESE-Detection showed higher FPR values in more experiments when compared to the tested strategies. This is actually expected, because of the EWMA limitations, i.e., the time it takes for the moving average to adapt to significant changes in the input data. A solution to this problem is to force the upper layers of the detection architecture to continuously monitor and update the analysis thresholds.

The attack penetration was tested to determine how much malicious traffic could be injected into the network before an alarm was raised by the detection engine. In Fig. 6.6, the results show that the BPFabric edge-based approach performed better than both cloud-based methods. Less attack packets were inserted into the network, due to the lower detection delays of BPFabric detection. Moreover, attack penetration values are directly linked to the in-place mitigation strategies. Consequently, BPFabric-based early detection provides the network with enhanced flexibility and efficiency in reducing the amount of attack traffic, by allowing upstream mitigation procedures to be executed in a fast and reactive manner.

Similar results were obtained when evaluating the bandwidth consumption, i.e., the network capacity required by the detection method to collect and analyze the data. This metric was measured checking the data message size sent to the detection algorithms by the collecting device (R1 and GW1 in Fig. 6.2). For a detection interval of 30s, the entropy mechanism employed an average of 3.3 MB of data while the cosine similarity was fed with around 1 MB. Conversely, BPFabric underlying ESE-Detection collected all required traffic statistics at line-rate on the IoT network gateway. Overall, BPFabric edge-based detection avoids the need to continuously poll traffic counters from the network nodes, thus preventing unnecessary bandwidth usage and enhancing scalability.

From the aforementioned results, the BPFabric edge-based detection approach stands out as the less costly solution to implement and deploy, when compared to adding a dedicated detection server/middlebox at the remote cloud or even paying for anomaly detection as a service. In a nutshell, BPFabric significantly decreases core operational/capital costs (e.g., power, cooling, processing/networking hardware), and allows an administrator or orchestration entity to easily and remotely control upstream packet processing and detection mechanisms for a significantly large number of nodes with minimum effort and low error rate. Regarding the monitoring and analysis expenses, the edge-based detection through BPFabric clearly outperforms the cloud-based scheme, as it introduces almost null overhead into the network while ensuring line-rate performance even for demanding scenarios and infrastructures.

6.2 Case Study 2: Optimal 5G User Plane Functions and EN placement

The need for ultra-low latency and ultra-high reliability for several 5G use cases, can only be satisfied (in terms of UPF response time) by placing UPFs closer to the users and assigning redundant UPFs to the access nodes placed on the service path. As a consequence, the number of UPFs required to satisfy existing service demands rapidly increases, thus rising the CAPEX/OPEX of the entire network. Moreover, a higher number of UPFs results in a significant increase in the number of UPF relocations mainly enforced by user mobility and handover. The amount of

relocations directly impacts the users' QoE by introducing additional delays during handover and signaling overhead for bearer establishment [124]. In this context, a significant reduction in the overall network costs can be achieved through an optimized placement strategy for the UPFs.

The UPF placement under 5G can significantly benefit from the deployed EC service infrastructure, since the ENs reduce the round trip delay while enhancing infrastructure-level reliability [125]. For these reasons, we argue that the placement of these VNFs over previously optimally placed ENs can guarantee the satisfaction of the network and computing infrastructure capacity requirements. Furthermore, the joint solution to both EN and UPF placement problems can significantly increase CAPEX/OPEX savings, while effectively achieving 5G demands. Such joint optimization is possible due to the null inter-dependence among the parameters and variables required by the placement of the physical infrastructure (i.e., ENs) and the UPFs.

In this context, we present in the following sections an adapted version of **EdgeON** to cost-effectively place ENs and UPFs, aiming at reducing overall network expenses and achieve end-user demand satisfaction (for 5G envisioned use cases). In summary, our main contribution is a framework proposal designed to jointly solve the EN and UPF placement optimization problems, considering user mobility, latency and reliability requirements.

It is worth clarifying that the work developed in this section was carried out as a collaboration amongst several authors and can be fully consulted in [126]. Furthermore, although the findings presented in Section 6.2.2.2 are included in this manuscript to demonstrate the use and benefits of the optimized EN placement for 5G VNF placement, these results are the exclusive work of the other author in [126] and they were included and properly referenced here with express authorization.

6.2.1 Solution proposal

In order to solve the aforementioned problem certain assumptions were made. Namely, the UPFs are said to be placed at previously optimally placed ENs. The extended version of **EdgeON** proposed is depicted in Figure 6.9, while Figure 6.10, depicts a sample diagram showcasing a possible outcome of the framework, where both ENs and UPFs have been effectively placed within the available sites.

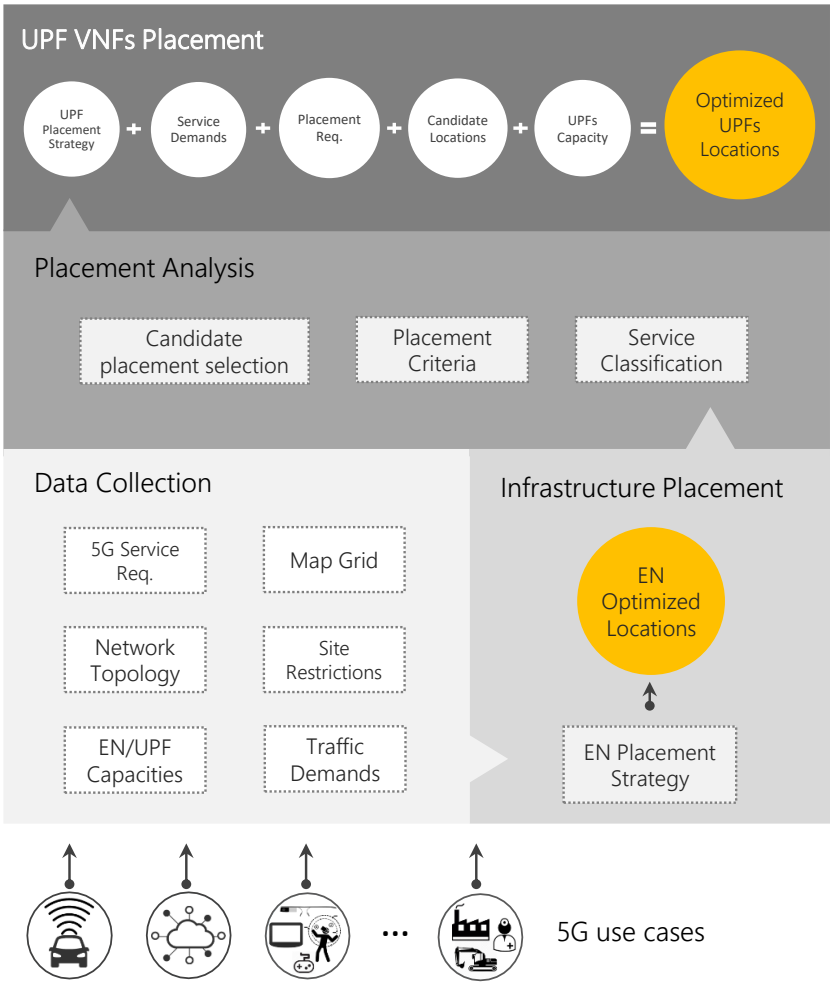


Figure 6.9: EdgeON extended version for the joint EN and UPF placement problem.

The first stage of the framework ensures data collection and normalization according to each underlying problem requirements: non-technical site restrictions, 5G use case requirements, territory to be analyzed, infrastructure capacities (i.e., EN and UPF maximum allowed capacities), current network topology and traffic demand model. Within the next step, the ENPP is solved using the model and placement strategy detailed in Section 4.2. Meanwhile, the UPF placement problems is addressed in the last two phases.

The UPF placement consists of two main stages: **Placement Analysis** and **UPF VNF Placement**. The **Placement Analysis** phase processes the data regarding the UPF placement through three main sub-stages: *Service Classification*, *Placement Criteria* and *Candidate Placement Selection*. The former clusters the services with similar placement demands into categories for further

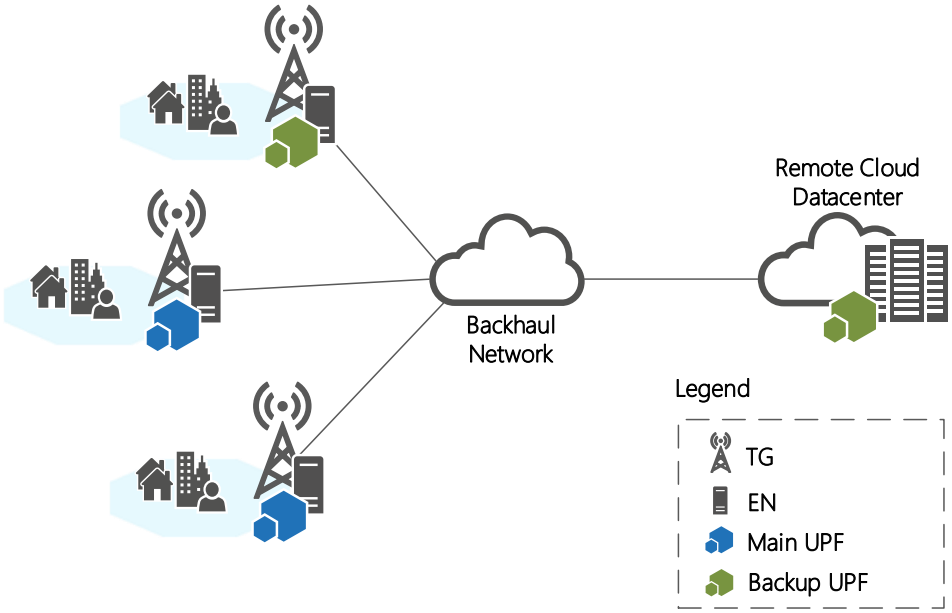


Figure 6.10: Outcome after placing both the ENs and UPFs

processing (thresholds levels for each parameter involved -i.e., mobility, latency, reliability- are set up). From the defined categories, the *Placement Criteria* stage determines the criteria to consider within the optimization strategy (e.g., number of backup UPFs for each service category according to its reliability level). The *Candidate Placement Selection* stage selects the ENs to place the UPFs according to the EN available capacities, UPF maximum allowed capacity and latency demands. Finally, the **UPF VNFs Placement** stage deals with the UPF placement problem taking into account underlying service and placement requirements, candidate sites and UPF capacity (for details about the optimization procedure to place the UPFs, please refer to [126]).

To optimize the UPF placement we developed two core strategies detailed in [126]: an exact mathematical model named “Optimal UPF Placement (OUP)” and a heuristic named “Near-Optimal UPF Placement (NOUP)”. The former mathematically formulates the UPF placement problem focusing on minimizing the overall deployment expenses, UPF number and UPF re-locations, considering users with and without mobility requirements. However, since the UPF placement problem is inherently NP-hard [126] and its complexity grows under 5G ultra-dense networking, a heuristic-based solution was devised (i.e., NOUP).

Table 6.3: 5G service requirements.

Service	**Latency (ms)**	**Data Rate (Mb/s)**	**Density (users/km^2)**	**Reliability (%)**	**m**[a]
Automated Factories	≤1	1	10^4 (R[b]), 0 (U[c])	99.999	0
mIoT	≤ 1	1	10^3 (R), 10^4 (U)	99.999	0/1
Cooperative Sensing	≤1	5	10 (R), 100 (U)	99.999	1
Home & Office	≤10	50 (R), 300 (U)	100 (R), 10^3 (U)	90	0
Traffic Efficiency	≤5	25	5 (R), 50 (U)	90	1
50 Mb/s everywhere	≤10	50	50 (R), 400 (U)	90	1

[a] Mobility requirement such that $m = 0$: no mobility, $m = 1$: mobility
[b] Rural
[c] Urban

6.2.2 Evaluation and Results

For evaluation pruposes, a 10000 km^2 map grid was employed, with a randomly placed set of TGs (i.e., fixed/radio access nodes with bandwidth demands ranging from 0 to 1 terabit per second) emulating both rural and urban areas. For urban areas, the radio access nodes were assumed to be BBUs with a maximum 3 km coverage area radius. For rural scenarios the BBUs were distributed with a coverage area radius ranging from 10 to 20 km. Six different services with arbitrary bandwidth, reliability, and latency requirements were used[127, 128] to generate the underlying TG demands. Table 6.3 summarizes the use cases and requirements analyzed.

The model in Section 4.2 was followed to solve the ENPP in this context. The latency constraints were translated into Euclidean distances considering the propagation times of a direct link between any TG-EN pair. Namely, for ultra-low latency requirements, a lower bound was fixed in 2 km while the upper bound was set to 6 km (considering an approximate propagation time of 5 μs/km [129]), for ultra-low latency requirements under 1 ms and low latency requirements around 5 ms.

6.2.2.1 EN placement evaluation

The results for the conducted experiments are shown in Figure 6.11 and Figure 6.12. Both the EA and the HSA (see Section 5.1.2.3) were tested for an arbitrary range of TGs varying between 200 and 400 (considering a representative number of nodes for envisioned 5G networks in mid to large city deployments). The hardware used to run the experiments has a 3.30 GHz CPU, $x64$ architecture (with 10 physical cores and 2 threads per core) and 64 GB RAM. The set of input

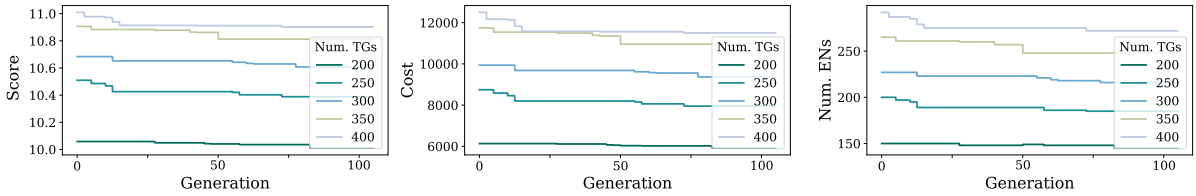


Figure 6.11: Evolutionary Algorithm performance on the joint UPF/EN placement problem.

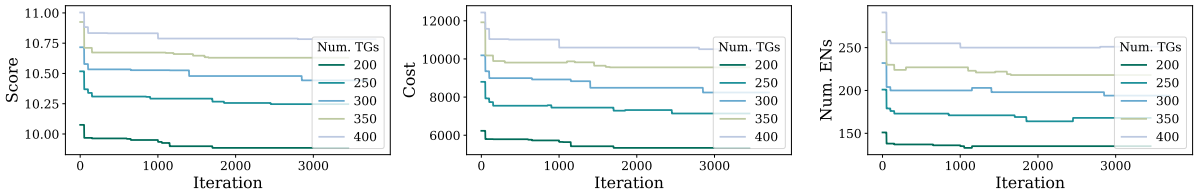


Figure 6.12: HSA performance on the joint UPF/EN placement problem.

parameters used for each algorithm is presented in Tables 6.4 and 6.5.

Figure 6.11 and Figure 6.12 showcase the score, cost and number of ENs deployed by each placement strategy for all input TG values. To estimate the score, the procedure in Section 4.1.2 was followed. In summary, the Score plots (leftmost plots in both Figure 6.11 and Figure 6.12) demonstrate that HSA outperformed EA significantly (considering that a logarithmic scale was used to normalize the score estimation values). Namely, HSA achieved cost savings over 15% when compared to EA in every case, reaching around 20%–30% for more than 300 TGs. Similar values can be observed for the number of ENs deployed by each mechanism (rightmost plots in both Figure 6.11 and Figure 6.12), where the HSA reached a maximum of over 30% less ENs deployed.

Based on the work carried out in this section and in Section 5.1.2.3 we found that when applying EA to the ENPP, the coverage nature of the problem forces a high probability of occurrence for a "*dominoes effect*", where a continuous EN-TG re-arrange is caused after changing a previously selected EN-TG pairing solution. Since the node density is significantly high (as expected in 5G ultra-dense networks), changing a valid EN-TG assignment, i.e., through the mutation and crossover techniques applied by evolutionary techniques, commonly results in a large chain of EN-TG reassignments throughout the complete service area. As a consequence, invalid solutions are typically generated and "*repair*" procedures have to be executed. This

Table 6.4: Input parameters for the EA.

Parameter	Value
Num. Generations	100.00
Num. Individuals	100.00
Mutation rate	0.0100

Table 6.5: Input parameters for the HSA.

Parameter	Value
Minimum Temperature	0.0001
Maximum Temperature	1.0000
Temperature Iterations	10.000
Fast Alpha	0.8000
Slow Alpha	0.9500
Num. Neighbors	10.000

situation leads to a lower probability of a child solution enhancing a parent valid EN placement.

The cost savings achieved by the placement strategies, in particular by the HSA, are directly linked to the EN network deployment costs. Nevertheless, the problem formulation and solving scheme used ensure a significant reduction in the operating expenses as well, due to the capacity assignment optimization and the reduction in the total number of ENs deployed.

6.2.2.2 UPF placement evaluation

After optimizing the EN placement the UPF placement problem had to be solved. This section presents a brief summary of the findings regarding the UPF placement stage of the proposal. As mentioned before, a comprehensive evaluation and detailed analysis of the results can be found in [126].

In summary, an arbitrary EC network -i.e., selected amongst the ones used in the previous section- was used as baseline for the evaluation of the UPF placement optimization strategies. Namely, a map grid with 100 TGs (i.e., access nodes in this context) and its corresponding EN sites (selected by HSA) was employed. The services presented in Table 6.3 were classified according to latency and reliability demands, in order to ensure similar placement conditions when evaluating the performance of the placement solutions under user mobility requirements. After executing the **Placement Analysis** phase two main service classifications were obtained: high-demand services (i.e., automated factories, Mobile IoT (mIoT) and cooperative sensing), low-demand services (comprising the remaining services). Both categories exhibiting various levels of mobility requirements.

When analyzing the UPF placement for high-demand services we arbitrarily fixed 1 ms user-plane delay and 1 backup UPF. Furthermore, 5 ms latency was assumed for the low-demand

Table 6.6: Network nodes distribution.

Region	Candidate Nodes		Access Nodes		Total Demand (Tb/s)	
	EN	PoP	Radio	Fixed	Group 1	Group 2
City_1	13	12	10	22	2.67	17.93
City_2	12	12	11	21	2.34	14.62
Rural	33	0	16	20	6.34	15.66

service category with no backup UPF. Equation 6.10 was used to determine the number of UPFs to which any given access node must be assigned (K_u), with p_r representing the access node failure probability and p_u the failure probability for UPFs.

$$R[r] = (1-p_r)\Big[1-\prod_{\forall u \in K_u[r]}\Big[1-(1-p_u)\Big]\Big] \tag{6.10}$$

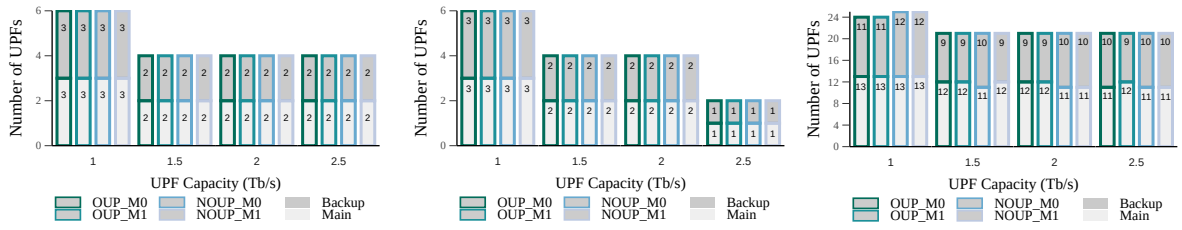
In our experiments, $p_r = 10^{-6}$ and $p_u = 10^{-3}$ were arbitrarily chosen (thus ensuring over 99.999% reliability). Moreover, to avoid exceeding a 0.5 ms end-to-end delay (required for 1 ms delay demand satisfaction)[7], a processing time of 0.3 ms was assumed for the co-located UPFs and application servers and 0.2 ms total delay between access nodes and UPFs was defined. For low-demand service the delay requirement was extended to 1 ms.

Three main scenarios, characterized in Table 6.6, were analyzed to place the UPFs. For each scenario all ENs and PoPs were revisited to optimally place the UPFs, although the latter were only considered whenever the existing ENs could not satisfy the service requirements. The proposed solutions -i.e., OUP_M1, OUP_M0, NOUP_M1 and NOUP_M0[5]-, were evaluated considering both mobility and no-mobility requirements and compared regarding four metrics[6]: number of UPFs, execution time, UPF utilization and, UPF relocations.

The number of UPFs for every scenario and service category is depicted in Figures 6.13 and 6.14. The proposed placement method performed significantly close to the exact model for all mobility requirements. Overall, the same number of main and backup UPFs were placed for both low and high-demand services, regardless of the capacity variation and the service area type.

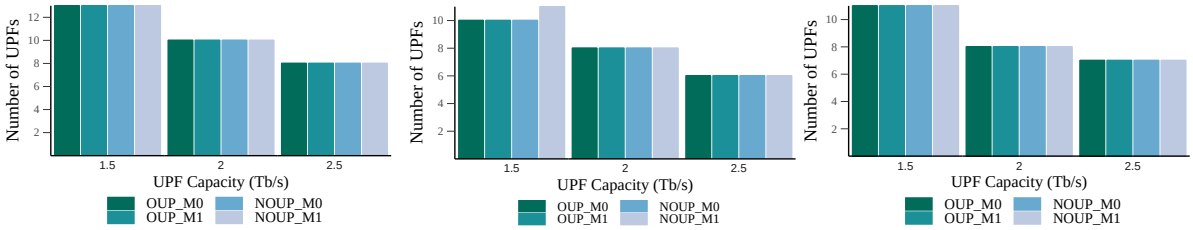
[5]Optimal and Non-Optimal placement, where M0: no mobility requirements and M1: mobility requirements are present, please refer to [126].

[6]The exact models were implemented using Pyomo and Gurobi as solver, with zero optimality gap.



(a) City_1 scenario. (b) City_2 scenario. (c) Rural scenario.

Figure 6.13: Number of UPFs vs. capacity for high-demand services.



(a) City_1 scenario (b) City_2 scenario. (c) Rural scenario.

Figure 6.14: Number of UPFs vs. capacity for low-demand services.

Some variation was obtained only in rural areas (where the number of main UPFs was always higher than the amount of backup UPFs due to existing isolated nodes [126]) for high-demand services, although a maximum of 1 extra UPF was required.

Throughout the above results, a clear trend can be observed: a rise in the UPF capacities and the opposite effect regarding the number of UPFs deployed. Such positive outcome is significantly higher for those services with relaxed latency demands, in any case showcasing the effectiveness of the proposed placement mechanisms.

Table 6.7 showcases our results regarding the analysis of the execution times for the proposed solutions. When compared to OUP, NOUP significantly reduces the processing times. By reaching a runtime reduction of over 80% and 30% (for urban and rural areas, respectively), our heuristic is able to outperform OUP, forcing the latter to be discarded for online placement purposes. What is more, NOUP showed a considerably lower computation time variation, ranging from 0.08 seconds to a maximum of 0.56 seconds. In comparison, OUP was quite sparse, ranging from 0.32 seconds to even 30058 seconds. Finally, it can be noticed that the computing times are mainly not significantly affected by the introduction of mobility requirements for the case of the non-optimal

Table 6.7: Execution Time.

Scenario	Model	Execution Time (s)						
		C_u for Group 1				C_u for Group 2		
		1.0	1.5	2.0	2.5	1.5	2.0	2.5
City_1	OUP_M0	3.41	0.37	0.43	0.45	1.11	1.18	0.47
	OUP_M1	10,428	8352	537	2378	244	190	121
	NOUP_M0	0.11	0.11	0.11	0.12	0.17	0.17	0.10
	NOUP_M1	0.16	0.14	0.15	0.13	0.21	0.16	0.13
City_2	OUP_M0	3.16	0.43	0.45	0.38	0.56	0.52	0.48
	OUP_M1	36,065	17192	4757	5.73	1420	176	30,058
	NOUP_M0	0.10	0.12	0.14	0.08	0.17	0.11	0.09
	NOUP_M1	0.12	0.14	0.14	0.14	0.16	0.14	0.12
Rural	OUP_M0	0.61	0.59	0.52	0.57	0.58	0.51	0.32
	OUP_M1	13.30	13.15	13.04	13.13	20,440	182,811	526
	NOUP_M0	0.37	0.36	0.33	0.29	0.33	0.25	0.09
	NOUP_M1	0.40	0.29	0.33	0.31	0.56	0.43	0.18

placement methods, although the optimal models were heavily impacted as expected (i.e., due to the model complexity).

As mentioned before, a detailed analysis of the results regarding the remaining two metrics analyzed -i.e., UPF utilization and relocations- can be consulted in [126].

6.3 Conclusion

This chapter presented two case studies as a final evaluation of the ENPP solution methods proposed in the previous chapters of this thesis.

In Section 6.1 a lightweight, platform-independent and high-performance DDoS detection architecture for IoT ecosystems was proposed, based on the BPFabric programmable data plane. This case study showed how DDoS detection in IoT can benefit from upstream executed mechanisms. The use of BPFabric and eBPF-based detection proved to effectively minimize the overall network overhead and provide early detection capabilities. The results obtained showed that the proposed solution introduced a bandwidth reduction of less than 1% and reduced several times the detection delay when compared to other methods. Moreover, the overall accuracy of our strategy was at least 5% higher than the other evaluated mechanisms.

The case study of Section 6.2.2.1 provided a framework to optimize the UPFs placement based on a previous optimal EC network site selection. The developed placement solutions

were thoroughly evaluated, with the EN placement method outperforming by over 20% in cost savings, mainly due to its adaptability to the problem and its better exploration of the solution space. These expense savings, enhanced the UPF placement by reducing their deployment costs and improving QoS, since the UPF candidate locations (i.e., ENs) were optimally determined according to users' traffic demands. Regarding the UPF placement itself, the devised solutions minimized not only the running time and computing resources required to solve the problem and the UPF deployment costs (measured in terms of the number of UPFs) but also the operational costs related to UPF relocations. Concretely, up to 55% and 70% in reductions were achieved for the UPF relocation rate of high and low-demand services respectively.

CHAPTER 7

FINAL REMARKS

Placing service infrastructure at the network edge has become a key enabler for 5G use cases requiring millisecond latency, highly reliable and flexible infrastructure and secure distributed service platforms at the users' vicinity. Although some CDN providers have been leveraging edge infrastructure for years (e.g., AWS CloudFront service), their edge locations can be considered as remote sites for the ultra-dense scenarios and ultra-high latency needs envisioned in future 5G networks. Cost-effective scalability is another key advantage of EC over the traditional remote datacenter model. Since expanding the capacities of dedicated datacenters is quite an expensive endeavor, computing/storage/networking resources bundled into devices with smaller footprints that can be placed at the network edge allow companies to leverage these ENs to expand their business reach and capabilities avoiding critical up-front construction costs and cyclic maintenance expenses.

In spite of the benefits, deploying and managing a 5G-EC ecosystem is extremely challenging and involves critical tradeoffs regarding CAPEX, OPEX, QoS, QoE and directly depends on the placement of the underlying physical infrastructure. This thesis focused on solving the above mentioned issues by providing a practical tool to optimize the placement of ENs for heavily constrained 5G use cases ensuring cost minimization and service requirement satisfaction. To this aim, both theoretical and practical (i.e., simulated) work were carried out in a bottom-up

approach, seeking to iteratively model and solve the EN placement problem, increasing the complexity and scope of the formulation and solution throughout the research process.

The core contributions of the solution proposed in this work to address the identified research questions are threefold: 1) a rigorous definition of the ENPP in a wide variety of scenarios leveraging a proposed set of parameters tailored to a 5G-EC ecosystem under ultra-dense networking conditions, 2) a novel heuristic based on SA to deal with the complexity of the ENPP variants (i.e., NP-hard for both problem types: network-agnostic and network-aware) and, 3) a framework called **EdgeON** implementing a set of algorithms and placement strategies offering an easy-to-use extensible platform to solve the ENPP. The next section briefly examines how these contributions answer the proposed research goals for this thesis and the key results obtained.

7.1 Research Contributions

The need for an adequate set of parameters to consider when solving the ENPP for 5G use cases is a key issue targeted in this thesis. For instance, most papers found during our literature review lacked a detailed analysis of the network delays involved in the placement of service infrastructure at the network edge for upcoming 5G networks (e.g., for a Cloud RAN (C-RAN) architecture). Chapter 3 presented a tailored set of optimization parameters in order to accurately evaluate any EN potential site aiming at reducing the overall deployment and even operational costs of the EC network. A key benefit of our set of parameter definition is its flexibility to be adapted to several ENPP core problem type (e.g., coveraqe, Weber), thus avoiding the rigidness of other proposals found in current literature.

Based on the findings in Chapter 3, Chapter 4 presented two solution approaches for the network-agnostic ENPP building the foundations for a real-life problem formulation and solution. The proposed approaches targeted the limited formulation scopes of the placement problem models found in the revisited literature, avoiding unrealistic assumptions, simplistic problem definitions and rigid solution algorithms. The first solution presented targeted a coverage-based ENPP formulation where the EN capacities were assumed to be divided in three sizes (i.e., small, mid and large-sized nodes) to emulate a latency-constrained scenario where predefined

bundled resources (i.e., computing, storage and networking) packaged as an EN were to be used to satisfy user demands. A key finding in this proposal was to abstract the EN placement from the underlying user distribution, mobility characteristics, through the construct of TGs referred to the existing traffic aggregation points. In the subsequent formulation of the problem a thorough evaluation of the costs involved in the deployment of an EN was developed, alongside the analysis of the reliability requirements of the TGs and a segregated schema to model the latency restrictions, seeking to answer the needs of 5G delay-sensitive and location-aware use cases.

Chapter 5 dove deeper into the ENPP complexities by formulating a real-life network-aware problem variant. Within this chapter, the framework proposed in Chapter 4 was significantly extended and re-architected to thoroughly assess each EN location in a self-aware manner, where the placement of each EN directly depended on the EN sites already selected to serve the underlying TGs. The problem definition and solution described in Chapter 5 ensures a flexible solution scheme based on a fully extensible framework, avoids rigid initial assumptions such as knowing the number of servers to be deployed in advance and, presents a novel network-aware platform to reduce the overall costs of deploying and operating an EC network, considering both technical and non-technical requirements.

Finally, Chapter 6 leveraged the findings presented in prior chapters of this thesis and presented two core case studies where the optimized placed infrastructure (i.e., the ENs) was used to deploy edge-based services. Namely, a novel DDoS detection solution was described, to be placed at the attacker's vicinity and using eBPF as packet processing engine in order to achieve high-performance and platform independence. In addition, a solution based on **EdgeON** aiming at jointly optimizing the placement of both ENs and UPFs was showcased. With these case studies, the benefits of placing service infrastructure at the network edge following our optimization strategy were assessed. Moreover, due to the cost savings derived from a cost-effective EC network these analyses can be used as foundations for new service and infrastructure business models.

7.2 Future Work

In spite of the promising results obtained in this thesis, there are some open research questions that encourage further research and development tasks. These open questions may be summarized in four categories:

- Non-discrete locations set analysis
- Placement execution mode
- EC architecture model
- Framework functionality

Non-discrete locations set analysis

To deal with the NP-hard nature of the edge server placement problem, the revisited literature and this thesis both start by assuming that a list of potential EN sites is known beforehand. Therefore, a discrete analysis is performed aiming at selecting the near-optimal locations among the potential sites set.

However, in upcoming 5G scenarios such approach may lead to inaccurate near-optimal results. There are several reasons this situation may occur: 1) there may be suitable sites -i.e., with in-place IT or even IP infrastructure- that may not be operating as TG and thus end up being overlooked and, 2) non-suitable locations -i.e., no in-place IT or IP infrastructure- may result in lower overall expenses if considered, depending on the problem model and the location-dependent costs.

Therefore, a remaining open research question is to design a non-discrete approach to avoid overlooking IT-capable unforeseen location and other feasible sites (i.e., physically suitable non-IT-capable locations such as businesses, buildings, street cabinets). A suggestion to solve this challenge is to extend the Input Processing or Pre-Optimization phases of **EdgeON** to include the analysis of both fixed and mobile network traffic models in order to find non-TG feasible sites that may reduce the overall expenses, according to certain parameters and a pre-defined probability.

Placement execution mode

The ENPP considered in this research is modeled to be solved offline, i.e., **EdgeON** is to be executed during the planning phase for the EC network deployment. However, considering that mobile base stations (e.g., drone-based base stations) are already being used for emergency situations, an online placement solution -i.e., continuously running to dynamically select the optimal locations- would allow a mobile EN optimized placement. Additionally, such strategy should aim at merging current VNF placement research and edge server placement methods into a joint optimization solution, thus leading to an end-to-end deployment/operation optimization scheme.

EC architecture model

There is still room for improvement regarding the problem modeling. The ENPP modeled in this thesis assumed a flat EC architecture with a unique layer of ENs deployed in between the traffic aggregation points and the remote cloud datacenter. Nevertheless, upcoming ultra-dense networking scenarios may result in extreme networking and processing demands requiring a multi-layered hierarchical EC architecture to avoid performance degradation and ensure top-level QoS and QoE. Taking this into consideration, further work could be made to improve the problem modeling and placement strategies used by **EdgeON** in order to adapt the framework to a hierarchical ecosystem with multi-level task offloading demands.

Framework functionality

Regarding the functionalities provided by **EdgeON**, developing a fully functional web/desktop application is a first step towards improving the framework's applicability. Namely, a web/desktop application will definitely set up a turning point in the decision making process based on the framework's output, by enhancing output visualization, portability and the overall business impact of the optimized results. Moreover, adding new features such as the processing of layered maps, including network topology information, could lead to a more accurate and flexible placement outcome.

APPENDIX A

APPENDIX A: PUBLICATIONS

- A. Santoyo-González and C. Cervelló-Pastor, "Network-aware Placement Optimization for Edge Computing Infrastructure under 5G," in *IEEE Access*, 2020.
- A. Santoyo-González, C. Cervelló-Pastor and D. P. Pezaros, "High-performance, platform-independent DDoS detection for IoT ecosystems," 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 2019, pp. 69-75.
- I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "A Framework for the Joint Placement of Edge Service Infrastructure and User Plane Functions for 5G," *Sensors*, vol. 19, no. 18, p. 3975, 2019.
- A. Santoyo González and C. Cervelló Pastor, "Edge Computing Node Placement in 5G Networks: A Latency and Reliability Constrained Framework," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud) / 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, (Paris, France), pp. 183–189, IEEE, 2019.
- A. Santoyo-González and C. Cervelló-Pastor, "Edge Nodes Infrastructure Placement Parameters for 5g Networks," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN 2018)*, (Paris, France), p. 6, IEEE, 2018.

- A. Santoyo-González and C. Cervelló-Pastor, "Latency-aware cost optimization of the service infrastructure placement in 5G networks," *Journal of Network and Computer Applications*, vol. 114, pp. 29–37, 2018.
- A. Santoyo-González and C. Cervelló-Pastor, "A Framework for Latency-constrained Edge Nodes Placement in 5G Networks," in *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, (Granada, España), 2018.
- A. Santoyo-González and C. Cervelló-Pastor, "On the cost optimization of NFVI-PoP placement for 5G networks," in *International Summer School on Latency Control for Internet of Services*, Poster Presentation, (Karlstad, Sweden), 2017.
- A. Santoyo-González and C. Cervelló-Pastor, "On the Optimal NFVI-PoP Placement for SDN-Enabled 5g Networks," in *Trends in Cyber-Physical Multi-Agent Systems*. The PAAMS Collection - 15th International Conference, PAAMS 2017, 2017.

BIBLIOGRAPHY

[1] Cisco, "Cisco Global Cloud index: Forecast and methodology 2016–2021," Tech. Rep. C11-738085-02, Cisco, San Jose, CA, USA, 2018.

[2] S. K. Sharma and X. Wang, "Towards massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2019.

[3] W. Xiang, K. Zheng, and X. Shen, *5G Mobile Communications*. Springer International Publishing, 2017.

[4] B. Blanco *et al.*, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Computer Standards & Interfaces*, vol. 54, pp. 216–228, 2017.

[5] S. E. Elayoubi *et al.*, "5G service requirements and operational use cases: Analysis and METIS II vision," in *2016 European Conference on Networks and Communications (EuCNC)*, pp. 158–162, 2016.

[6] H. Shariatmadari, *Enabling Ultra-Reliable Low-Latency Communications in 5G Networks*. PhD Thesis, Aalto University, 2019.

[7] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5g: RAN, core network and caching solution," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, 2018.

[8] B. Varghese, N .Wang, D. S. Nikolopoulos, and R. Buyya, "Feasibility of Fog Computing," *arXiv preprint arXiv:1701.05451*, 2017.

[9] M. M. Hussain, M. S. Alam, and M. M. S. Beg, "Feasibility of Fog Computing in smart grid architectures," in *Proceedings of 2nd International Conference on Communication, Computing and Networking* (C. R. Krishna, M. Dutta, and R. Kumar, eds.), vol. 46, pp. 999–1010, Singapore: Springer Singapore, 2019.

[10] P. Bellavista and A. Zanni, "Feasibility of Fog Computing deployment based on Docker containerization over RaspberryPi," in *Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN '17)*, (Hyderabad, India), pp. 1–10, ACM Press, 2017.

[11] C. Kuo, V. Chang, and C. Lei, "A feasibility analysis for edge computing fusion in LPWA IoT environment with SDN structure," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[12] I. Hadžić, Y. Abe, and H. C. Woithe, "Server placement and selection for Edge Computing in the ePC," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 671–684, 2019.

[13] F. Zeng, Y. Ren, X. Deng, and W. Li, "Cost-effective edge server placement in wireless Metropolitan Area Networks," *Sensors*, vol. 19, no. 1, p. 32, 2019.

[14] T. Kuo, B. Liou, K. C. Lin, and M. Tsai, "Deploying chains of Virtual Network Functions: On the relation between link and server usage," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1562–1576, 2018.

[15] S. Wang and C. Ran, "Rethinking cellular network planning and optimization," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 118–125, 2016.

[16] S. Wang, W. Zhao, and C. Wang, "Budgeted cell planning for cellular networks with small cells," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4797–4806, 2015.

[17] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of CDNs," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 460–468, 2014.

[18] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, "Efficient gateways placement for internet of things with QoS constraints," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2016.

[19] Y. Zhang, D. Li, and M. Tatipamula, "The freshman handbook: A hint for server placement in online social network services," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pp. 588–595, 2012.

[20] Z. Ulukan and E. Demircioglu, "A survey of discrete facility location problems," *International Journal of Social Behavioral, Educational, Economic, Business and Industrial Engineering*, vol. 9, no. 7, pp. 2487–2492, 2015.

[21] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, "Multiple criteria facility location problems: A survey," *Applied Mathematical Modelling*, vol. 34, no. 7, pp. 1689–1709, 2010.

[22] M. Barbati, *Models and Algorithms for Facility Location Problems with Equity Considerations*.
PhD thesis, Universita degli Studi di Napoli Federico II, 2013.

[23] A. B. Arabani and R. Z. Farahani, "Facility location dynamics: An overview of classifications and applications," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 408–420, 2012.

[24] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in *2011 31st International Conference on Distributed Computing Systems (ICDCS)*, pp. 131–142, 2011.

[25] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, 2018.

[26] N. Mohan, A. Zavodovski, P. Zhou, and J. Kangasharju, "Anveshak: Placing edge servers in the wild," in *Proceedings of the 2018 Workshop on Mobile Edge Communications - MECOMM'18*, (Budapest, Hungary), pp. 7–12, ACM Press, 2018.

[27] H. Yin *et al.*, "Edge provisioning with flexible server placement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1031–1045, 2017.

[28] W. Zhang, B. Lin, Q. Yin, and T. Zhao, "Infrastructure deployment and optimization of fog network based on MicroDC and LRPON integration," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 579–591, 2017.

[29] N. di Pietro, M. Merluzzi, E. C. Strinati, and S. Barbarossa, "Resilient design of 5G Mobile-Edge Computing over intermittent mmWave links," *arXiv:1901.01894 [cs, math]*, 2019.

[30] W. E. Hartand, J. Watson, and L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[31] W. E. Hartand, C. D. Laird, J. Watson, D. L. Woodruff, G. A. Hackebei, B. L. Nicholson, and D. Siirola, *Pyomo–optimization modeling in python*, vol. 67.
Springer Science & Business Media, second ed., 2017.

[32] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016.

[33] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[34] M. Satyanarayanan, "The emergence of Edge Computing.," *IEEE Computer*, vol. 50, no. 1, 2017.

[35] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *2017 Global Internet of Things Summit (GIoTS)*, pp. 1–6, 2017.

[36] O. C. A. W. Group, "OpenFog reference architecture for Fog Computing," tech. rep., OpenFog Consortium, 2017.

[37] O. C. A. W. Group, "OpenFog architecture overview," tech. rep., OpenFog Consortium, 2016.

[38] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana, A. Jukan, G. Ren, and J. Zhu, "Do we all really know what a fog node is? Current trends towards an open definition," *Computer Communications*, vol. 109, pp. 117–130, 2017.

[39] ETSI GSNFV, "Mobile Edge Computing (MEC); Framework and reference architecture," Tech. Rep. v1.1.1, ETSI, 2016.

[40] M. T. Beck, M. Werner, S. Feld, and T. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*, Citeseer, 2014.

[41] J. O. Fajardo *et al.*, "Introducing mobile edge computing capabilities through distributed 5G cloud enabled small cells," *Mobile Networks and Applications*, vol. 21, no. 4, pp. 564–574, 2016.

[42] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[43] GR MEC ETSI, "Mobile edge computing (MEC); End to end mobility aspects," Tech. Rep. 018 V1.1.1, ETSI, 2017.

[44] Pavel Mach and Zdenek Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[45] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[46] R. Vilalta *et al.*, "TelcoFog: A unified flexible fog and cloud computing architecture for 5G networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 36–43, 2017.

[47] M. S. Elbamby, C. Perfecto, C. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, 2019.

[48] R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh, "Covering problems in facility location: A review," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 368–407, 2012.

[49] J. Qin, H. Xiang, Y. Ye, and L. Ni, "A simulated annealing methodology to multiproduct capacitated facility location with stochastic demand," *The Scientific World Journal*, vol. 2015, pp. 1–9, 2015.

[50] R. Z. Farahani, M. Hekmatfar, B. Fahimnia, and N. Kazemzadeh, "Hierarchical facility location problem: Models, classifications, techniques, and applications," *Computers & Industrial Engineering*, vol. 68, pp. 104–117, 2014.

[51] S. Zhou *et al.*, "On the spatial distribution of base stations and its relation to the traffic density in cellular networks," *IEEE Access*, vol. 3, pp. 998–1010, 2015.

[52] J. Kosmerl and A. Vilhar, "Base stations placement optimization in wireless networks for emergency communications," in *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 200–205, 2014.

[53] T. Taleb and A. Ksentini, "On efficient data anchor point selection in distributed mobile networks," in *Communications (ICC), 2013 IEEE International Conference on*, pp. 6289–6293, 2013.

[54] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014.

[55] F. Lobillo *et al.*, "An architecture for mobile computation offloading on cloud-enabled LTE small cells," in *2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–6, 2014.

[56] I. Giannoulakis *et al.*, "The emergence of operator-neutral small cells as a strong case for cloud computing at the mobile edge," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1152–1159, 2016.

[57] M. Yannuzzi *et al.*, "A new era for cities with Fog Computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 54–67, 2017.

[58] P. Maiti, J. Shukla, B. Sahoo, and A. K. Turuk, "QoS-aware fog nodes placement," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, pp. 1–6, IEEE, 2018.

[59] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[60] C. C. Byers, "Architectural imperatives for Fog Computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 14–20, 2017.

[61] M. Bouet and V. Conan, "Mobile Edge Computing resources optimization: A geo-clustering approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 787–796, 2018.

[62] W. Ma, C. Medina, and D. Pan, "Traffic-aware placement of NFV middleboxes," in *2015 IEEE Global Communications Conference (GLOBECOM)*, (San Diego, CA, USA), pp. 1–6, IEEE, 2015.

[63] E. Balevi and R. D. Gitlin, "Optimizing the number of fog nodes for Cloud-Fog-Thing networks," *IEEE Access*, vol. 6, pp. 11173–11183, 2018.

[64] F. J. F. Silva and D. S. D. la Figuera, "A capacitated facility location problem with constrained backlogging probabilities," *International journal of production research*, vol. 45, no. 21, pp. 5117–5134, 2007.

[65] L. Wu, X. Zhang, and J. Zhang, "Capacitated facility location problem with general setup cost," *Computers & Operations Research*, vol. 33, no. 5, pp. 1226–1241, 2006.

[66] Z. Zhu, F. Chu, and L. Sun, "The capacitated plant location problem with customers and suppliers matching," *Transportation Research Part E Logistics and Transportation Review*, vol. 46, no. 3, pp. 469–480, 2010.

[67] Z. Ho and C. Wu, "Application of simulated annealing algorithm to optimization deployment of mobile wireless base stations," in *Advances in Computer Science and Information Engineering*, pp. 665–670, Springer, 2012.

[68] J. Qin, L. Ni, and F. Shi, "Combined simulated annealing algorithm for the discrete facility location problem," *The Scientific World Journal*, vol. 2012, 2012.

[69] M. A. Arostegui, S. N. Kadipasaoglu, and B. M. Khumawala, "An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems," *International Journal of Production Economics*, vol. 103, no. 2, pp. 742–754, 2006.

[70] H. Delmaire, J. A. Díaz, E. Fernández, and M. Ortega, "Comparing new heuristics for the pure integer capacitated plant location problem," *Investigacion Operativa*, vol. 8, no. 1,2,3, 1999.

[71] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*.
Boston, MA: Springer US, 2003.

[72] D. B. Fogel, "The advantages of evolutionary computation," in *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1–11, 1997.

[73] I. Litvinchev and E. L. Ozuna, "Lagrangian heuristic for the facility location problem," *IFAC Proceedings Volumes*, vol. 46, no. 24, pp. 107–113, 2013.

[74] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 50, no. 12 Supplement, pp. 1861–1871, 2004.

[75] F. Musumeci, C. Bellanzon, N. Carapellese, M. T. A. Pattavina, and S. Gosselin, "Optimal BBU placement for 5G C-RAN deployment over WDM aggregation networks," *Journal of Lightwave Technology*, vol. 34, no. 8, 2016.

[76] G. M.-I. ETSI, "Mobile Edge Computing; Market acceleration; MEC metrics best practice and guidelines," Tech. Rep. 006 V1.1.1, ETSI, 2017.

[77] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, 2018.

[78] A. Mukherjee, "Energy efficiency and delay in 5G ultra-reliable low-latency communications system architectures," *IEEE Network*, vol. 32, no. 2, 2018.

[79] V. G. Nguyen, A. Brunstrom, K. J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey.," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, 2017.

[80] J. Zhang, W. Xie, F. Yang, and Q. Bi, "Mobile edge computing and field trial results for 5G low latency scenario," *China Communications*, vol. 13, no. Supplement2, pp. 174–182, 2016.

[81] M. Emara, M. C. Filippou, and D. Sabella, "MEC-aware cell association for 5G heterogeneous networks," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 350–355, 2018.

[82] L. A. Barroso, J. Clidaras, and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*.
No. 6 in Synthesis lectures on computer architecture, San Rafael, Califronia: Morgan & Claypool, 2 ed., 2013.

[83] S. Zhang, X. Xu, Y. Wu, and L. Lu, "5G: Towards energy-efficient, low-latency and high-reliable communications networks.," in *2014 IEEE ICCS*, vol. 14, IEEE, 2014.

[84] A. Fernández-Fernández and C. Cervelló-Pastor, *Energy-Aware Routing Techniques for Software-Defined Networks*.
PhD Thesis, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 2018.

[85] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, (San Francisco, California), pp. 323–336, USENIX Association, 2008.

[86] X. Li and C. Qian, "A survey of network function placement," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 948–953, IEEE, 2016.

[87] M. Ghaznavi *et al.*, "Elastic virtual network function placement," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pp. 255–260, IEEE, 2015.

[88] M. C. Luizelli *et al.*, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 98–106, 2015.

[89] S. Mehraghdam and M. K. H. Karl, "Specifying and placing chains of virtual network functions," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pp. 7–13, 2014.

[90] S. Clayman *et al.*, "The dynamic placement of virtual network functions," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9, 2014.

[91] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *2015 IEEE International Conference on Communications (ICC)*, pp. 3879–3884, 2015.

[92] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pp. 171–177, 2015.

[93] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for load balancing in NFV networks," *arXiv preprint arXiv:1610.08266*, 2016.

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

[95] J. Camino, C. Artigues, L. Houssin, and S. Mourgues, "Linearization of euclidean norm dependent inequalities applied to multibeam satellites design," *LAAS Publications*, vol. 1, no. 16116, pp. 1–18, 2016.

[96] N. Riquelme, C. Von Lücken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)*, pp. 1–11, 2015.

[97] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84–91, 2016.

[98] C. Chang, N. Nikaein, and T. Spyropoulos, "Impact of packetization and scheduling on C-RAN fronthaul performance," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, pp. 1–7, IEEE, 2016.

[99] J. H. Cho, Y. Wang, I. R. Chen, K. S. Chan, and A. Swami, "A Survey on Modeling and Optimizing Multi-Objective Systems," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1867–1901, 2017.

[100] R. Sedgewick, *Algorithms in C, Part 5: Graph Algorithms*. Addison-Wesley Professional, 3rd ed., 2001.

[101] R. Ramaswamy, N. Weng, and T. Wolf, "Characterizing network processing delay," in *2004 IEEE Global Telecommunications Conference (GLOBECOM'04)*, vol. 3, pp. 1629–1634, IEEE, 2004.

[102] D. Hardy, M. Kleanthous, I. Sideris, A. G. Saidi, E. Ozer, and Y. Sazeides, "An analytical framework for estimating TCO and exploring data center design space," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 54–63, IEEE, 2013.

[103] M. H. Eiselt and F. Azendorf, "Accurate Measurement of Propagation Delay in a Multi-Span Optical Link," in *2019 International Topical Meeting on Microwave Photonics (MWP)*, (Ottawa, ON, Canada), pp. 1–3, IEEE, 2019.

[104] K. Bhardwaj, J. C. Miranda, and A. Gavrilovska, "Towards IoT-DDoS prevention using edge computing," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, USENIX Association, 2018.

[105] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425–441, 2017.

[106] A. Pamukchiev, S. Jouet, and D. P. Pezaros, "Distributed network anomaly detection on an event processing framework," in *2017 14th IEEE CCNC*, pp. 659–664, 2017.

[107] M. Iordache, S. Jouet, A. K. Marnerides, and D. P. Pezaros, "Distributed, multi-level network anomaly detection for datacentre networks," in *2017 IEEE ICC*, pp. 1–6, IEEE, 2017.

[108] S. Simpson *et al.*, "An inter-domain collaboration scheme to remedy DDoS attacks in computer networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 879–893, 2018.

[109] D. Yin, L. Zhang, and K. Yang, "A DDoS attack detection and mitigation with software-defined internet of things framework," *IEEE Access*, vol. 6, pp. 24694–24705, 2018.

[110] H. Liu, Y. Sun, and M. S. Kim, "A scalable DDoS detection framework with victim pinpoint capability," *Journal of Communications*, vol. 6, no. 9, 2011.

[111] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 29–35, IEEE, 2018.

[112] C. Zhang and R. C. Green, "Communication security in internet of thing: Preventive measure and avoid DDoS attack over IoT network," *Simulation Series*, vol. 47, pp. 8–15, 2015.

[113] S. Jouet and D. P. Pezaros, "BPFabric: Data plane programmability for software defined networks," in *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 38–48, IEEE, 2017.

[114] D. Scholz, D. Raumer, P. Emmerich, A. Kurtz, K. Lesiak, and G. Carle, "Performance implications of packet filtering with linux eBPF," in *2018 30th ITC*, pp. 209–217, IEEE, 2018.

[115] M. Xhonneux, F. Duchene, and O. Bonaventure, "Leveraging eBPF for programmable network functions with IPv6 segment routing," in *Proceedings of the 14th CoNEXT '18*, pp. 67–72, ACM Press, 2018.

[116] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *USENIX Security Symposium*, pp. 1092–1110, 2017.

[117] P. Redekar and M. Chatterjee, "Hybrid technique for DDoS attack detection," *International Journal of Computer Science and Information Technologies*, vol. 8, pp. 377–379, 2017.

[118] K. Singh, K. S. Dhindsa, and B. Hushan, "Threshold-based distributed DDoS attack detection in ISP networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, no. 4, pp. 1796–1811, 2018.

[119] A.K. Marnerides, A. Schaeffer-Filho, and A. Mauthe, "Traffic anomaly diagnosis in internet backbone networks: A survey," *Computer Networks*, vol. 73, pp. 224–243, 2014-11.

[120] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[121] I. Cvitić, D. Peraković, M. Perisa, and M. Botica, "Smart home IoT traffic characteristics as a basis for DDoS traffic detection," in *Proceedings of the 3rd EAI International Conference on Management of Manufacturing Systems*, 2018.

[122] J. Mocnej, A. Pekar, W. K. G. Seah, and I. Zolotova, "Network traffic characteristics of the IoT application use cases," Tech. Rep. ECSTR18-0, School of Engineering and Computer Science, Victoria University of Wellington, 2018.

[123] T. H. Black, *Derivations of Applied Mathematics*.
Debian Project, 2018.

[124] T. Taleb and A. Ksentini, "Gateway relocation avoidance-aware network function placement in carrier cloud," in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems - MSWiM '13*, (Barcelona, Spain), pp. 341–346, ACM Press, 2013.

[125] A. Leivadeas, I. Lambadaris, and G. Kesidis, "VNF placement optimization at the edge and cloud," *Future Internet*, p. 23, 2019.

[126] I. Leyva-Pupo, A. Santoyo-González, and C. Cervelló-Pastor, "A framework for the joint placement of edge service infrastructure and user plane functions for 5G," *Sensors*, vol. 19, no. 18, p. 3975, 2019.

[127] 5G Americas, "5G Network Transformation," tech. rep., 5G Americas, 2017.

[128] N. Alliance, "Perspectives on vertical industries and implications for 5G by ngmn alliance," tech. rep., NGMN Alliance, 2016.

[129] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5G mobile core network based on SDN and NFV," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1061–1075, 2017.