Journal Pre-proof

Resource allocation for cost minimization of a slice broker in a 5G-MEC scenario

Annisa Sarah, Gianfranco Nencioni

PII: S0140-3664(23)00407-3
DOI: https://doi.org/10.1016/j.comcom.2023.11.016
Reference: COMCOM 7681

To appear in: *Computer Communications*

Received date : 12 April 2023
Revised date : 20 October 2023
Accepted date : 16 November 2023

Please cite this article as: A. Sarah and G. Nencioni, Resource allocation for cost minimization of a slice broker in a 5G-MEC scenario, *Computer Communications* (2023), doi: https://doi.org/10.1016/j.comcom.2023.11.016.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Journal Pre-proof

Revised Manuscript

Click here to view linked References

Highlights

Resource Allocation for Cost Minimization of a Slice Broker in a 5G-MEC Scenario

Annisa Sarah,Gianfranco Nencioni

- **Motivation**: Slice broker (SB) is a new actor that supports a slice tenant in configuring network slices by buying resources from infrastructure providers. Challenges arise when an SB needs to decide the configuration of resources to buy while (1) suppressing the cost to increase SB profit and (2) maintaining the Service Level Agreement (SLA).
- **Contribution**: Accurate formulation of the problem of joint network slice allocation and configuration selection to minimize the SB cost. We provide an extensive evaluation to analyze the impact of various potential scenarios on the SB cost.
- **Method**: We use commercial optimization software (IBM CPLEX) to solve the formulated problem and generate the optimal solution.
- **Key Finding 1**: The main solution (Minimization of Cost, Min.Cost) can save $\sim$ 30% cost compared to reference solutions.
- **Key Finding 2**: The total cost is mainly due to buying the Computing Platform (CP) configurations, but minimizing the number of used CPs leads to the most expensive solutions. It means that doing a CP consolidation is not cost-efficient.
- **Key Finding 3**: The cheaper solution can be obtained from distributing the Virtual Network Function (VNF) requests throughout CPs and less CP sharing.

Journal Pre-proof

Journal Pre-proof

Resource Allocation for Cost Minimization of a Slice Broker in a 5G-MEC Scenario*

Annisa Sarah[a,*], Gianfranco Nencioni[a]

[a] University of Stavanger, postboks 8600, Stavanger, 4036, Norway

ARTICLE INFO

Keywords:
Resource Allocation
Slice Broker
MEC
Edge Computing
5G
Cost Minimization

ABSTRACT

The fifth generation (5G) of mobile networks may offer a custom logical and virtualized network called network slicing. This virtualization opens a new opportunity to share infrastructure resources and encourage cooperation between several Infrastructure Providers (InPs) to offer tailored network slices for the Slice Tenants (STs). The Slice Broker (SB) is emerging as intermediate entity that purchases resources from the InPs and it offers network slices to the STs. The main challenge of the SB is to jointly decide the purchase of heterogeneous (data and network) resources from multiple InPs and create the slices to meet the various requests from the STs. Being an economical entity, the target of the SB is to maximize its profit by minimizing the costs while satisfying all the ST requests. This paper formulated the SB cost minimization problem and used CPLEX to obtain the optimal solution. The problem formulation considers the realistic scenario that the InPs offer the computing, storage and network resources by using predetermined configurations. Therefore, for each of the computing platform and logical connection, the SB may select one of the configurations. The proposed cost-minimization problem is compared with three alternative problems that have three different objectives: computing platform consolidation, network connection consolidation, and both computing-network consolidation. The computing platform and network connection consolidation are currently the most common approaches for decreasing resource costs. However, the result shows that consolidating computing and network resources fails to reach the actual minimal cost. The proposed problem finds the cheapest solution, which can save at least 30% of the total cost of the other approaches in every evaluated scenario. Moreover, consolidating the number of computing platforms can lead to the most expensive solution, up to 40% higher than the optimal solution of our proposed problem.

1. Introduction

The fifth generation (5G) of mobile networks can provide different categories of services. Three service categories with distinct requirements are envisioned in International Mobile Telecommunications-2020 (IMT-2020) [1]. First, the enhanced Mobile Broadband (eMBB) services require a high-data-rate connection. Second, the massive Machine Type Communication (mMTC) supports high connection density. Third, Ultra-Reliable Low-Latency Communication (URLLC) requires high reliability and strict latency. Network slicing is one of the key techniques to deliver different service categories in 5G networks by creating multiple logical networks over the same physical infrastructure.

European Telecommunications Standards Institute (ETSI) defines the roles that have different responsibilities in the context of network slicing. Each role can be taken by one or more actors, and each actor can take one or several roles altogether [2]. We define the *Infrastructure Provider* (InP) as an actor that is responsible for providing and managing network equipment (network resources) or for providing and managing the data centre services (data resources). The InPs are also responsible for supplying virtualized (network or data) resource. Another actor called *Slice Broker (SB)* can cross-manage the virtualized resources to create network slices. The SB is responsible for providing and orchestrate the network slices. These network slices are then used by *Slice Tenants* (STs) that provide the communication services that end-users or consumers will access.

The SB concept has been introduced in [3]. The SB acts as an intermediary entity between STs and InPs and helps to manage the resources effectively. The SB is responsible for receiving requests from STs, selecting and leasing resources from InPs, providing slices to the STs, and monitoring the ongoing service performances. Ideally, the roles of an SB can be played by either the InPs or STs. However, this is impracticable due to the following reasons. First, network slicing is a complex technique that demands a deep understanding of both service requirements and the network infrastructure. STs often focus on service delivery such as subscriber management, applications management, and customer experience. The STs may lack expertise in network operation or resource orchestration. The SB ensures efficient data and network resource management and provides network slices ready to used by the STs. Second, in a multi-domain environment with multiple InPs and multi-tenant scenarios, it is complex and unpractical for a single InP to peer business relationships with the other InPs to get further resources and create network slices. In this scenario, the SB can act as a neutral element that builds the network slices on top of virtualized resources across multiple domains.

The establishment of an SB is confronted with various technological and economic challenges. The SB must decide on a resource allocation strategy that can meet the

*This work has been funded by the Research Council of Norway through the 5G-MODaNeI project (no. 308909).

*Corresponding author
annisa.sarah@uis.no (A. Sarah); gianfranco.nencioni@uis.no (G. Nencioni)
ORCID(s): 0000-0001-8503-1380 (A. Sarah); 0000-0002-9684-0375 (G. Nencioni)

Journal Pre-proof

service expectations of STs while ensuring profitability. In this context, a techno-economic analysis of SB is essential. This paper proposes a slice allocation problem with the objective of minimizing the overall cost of SB. The problem is formulated as a Mixed Integer Programming (MIP) problem that can be solved by a commercial solver such as CPLEX. Our approach models the network slice as a composition of Virtual Network Functions (VNFs) that can be deployed across multiple computing platforms, forming a chain known as the VNF Forwarding Graph (VNF-FG). The VNF-FG concept works similarly to the Service Function Chain (SFC) concept.

To summarize, the SB jointly selects the configurations of the InP' resources and decides where to allocate the VNFs of each network slice in order to minimize the costs of leasing resources from InPs. The primary motivation of this study is to investigate if the SB can implement a cost-effective slice allocation strategy and to explore potential methods to achieve this. This paper has the following main contributions:

- Accurate mathematical formulation of the problem of joint network slice allocation and configuration selection in order to minimize the SB cost. To the best of our knowledge, there have been no previous attempts to precisely formulate a similar problem.
- Extensive evaluation to analyze the behaviour of the proposed problem under various potential scenarios: variation of the infrastructure size and of the number and type of network slice requests. The problem is also compared with some reference approaches. The results allows to understand the following things:
  - The theoretical limit for SB cost minimization compared to the reference approaches, which are not cost-aware.
  - The characteristics of the selected resources and slice allocation when the SB cost is minimized.

The formulated problem is solved using a commercial solver; thus, no algorithmic contribution is provided. However, the optimal solution derived from the formulated problems provides a valuable means to analyze the problem and the potential benefits.

This paper is organized as follows. The problem description is provided on Section 2. The related works are introduced in Section 3. The problem formulation is presented on Section 4. The evaluation is available in Section 5. Lastly, the conclusions are discussed on Section 6.

2. Problem Description

This section describes the scenario and the assumptions considered in our resource allocation problem. First, the business model is briefly presented. Second, the InP resources and network slice requests are described in detail.

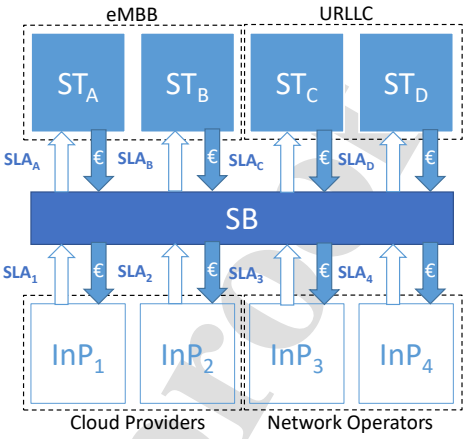


Figure 1: Business model

2.1. Business Model

As previously introduced in Section 1, there are two important terms in 5G with a network-slicing business model. A role is a duty that must be conducted to provide or deliver equipment or service. An actor is an entity that can perform a specific role(s). ETSI defined that one role can be played by one or more actor(s), and one actor can also take one or more role(s) [4, 2]. An SB is an actor that have a role to orchestrate a slice resources. Based on the [3], the SB has direct cooperation with at least two other actors, namely InPs and STs. In this scenario, the SB receives network slice requests from the STs. Then, the SB buys resources from InPs. This paper assumes two kinds of InPs: the *cloud providers*, who provide the *Computing Platforms* (CPs), and the *network operators*, who provide the *Logical Connections* (LCs) between CPs [5]. The SB composes the resources provided by the InPs to create the network slices for the STs. The STs manage the slice and provide services to the clients. The services can belong to different 5G service categories, such as eMBB and URLLC.

Given these three actors, the business model (depicted in Fig. 1) for the SB can be summarized as follows:

- the SB buys resources from the InPs;
- the SB sells slices to the STs.

The acquisition of resources from the InPs and the provision of network slices is defined by Service Level Agreements (SLAs). An SLA is a business contract binding one party (e.g., InP) to fulfill a set of measurable service-level expectations to its client (e.g., SB) [2, 6].

2.2. InP Resources

As we have already mentioned, there are two kinds of InP: cloud providers and network operators.The cloud providers have CPs, which have the *computing resources* to process a specific task (Unit of Measurement, UoM: vCPU) and the *storage resources* to store the data related to task processing (UoM: GiB) [7, 8]. There are two types of CP.

Journal Pre-proof

First, a *Data Center* (DC) acts at the same level as a cloud server. A DC is a CP that can be accessed on a core-level architecture and offers a large amount of vCPU and storage resources. Second, a *MEC Host* (MEH) is a CP which acts as a smaller server and is located on the edge of the network. A MEH can be accessed on an edge-level architecture and offers a smaller amount of vCPU and storage resources [7, 9]. The MEHs can be clusterized based on locations, provider company, or managerial reasons. The cluster is similar to the concept of MEC system by ETSI, which calls *MEC Federation* the union of multiple MEC systems and provides different use cases in [10].

The network operators provide LCs between the CPs. The LC is created by the network operator on top of the physical infrastructure. The network operator is able to guarantee the *data rate* and the *delay* [11]. The data rate is the guaranteed amount of bits transferred from one point to another during a specified time unit (UoM: Mb/s). Delay is the guaranteed maximum amount of time for transferring one bit from one point to another (UoM: ms). An example on how a network operator can provide LCs is by using Multi-protocol Label Switching (MPLS), which provides guaranteed services through traffic engineering [12].

If two VNFs are allocated within the same CP, there is a guaranteed data rate for the data transfer between them, referred to as the *Intra-Connection data Rate* (ICR), along with a guaranteed delay known as the *Intra-Connection Delay* (ICD). The limitation of ICR and ICD is based on two assumptions. First, a corporate data center usually hosts up to 24 servers interconnected by fiber cables. Thus, although several VNFs are placed on the same data center, two VNFs belonging to the same SFC can be physically placed on different servers, this requires a certain amount of time transferring the data from one VNF to the other [13]. Second, if the VNFs are located within the same machine, the VNFs are usually connected via a virtual switch, provided by a hypervisor. Transferring data via a virtual switch within the same machine takes time. Thus ICD and ICR need to be considered [14, 15].

The CP resources are sold by the InP by using different configurations that contain predetermined values of computing capacity [vCPU], storage capacity [GiB], ICR [Gb/s], ICD [ms], and hourly cost [€/h]. Similarly, each LC resource configuration contains a set of values of data rate [Gb/s], delay [ms], and hourly cost [€/h]. These configuration assumptions refer to the realistic subscriptions that are sold by cloud providers and network operators [7, 16, 8].

2.3. Network Slice Requests

In this paper, two types of network slices are considered, namely eMBB and URLLC. The eMBB has the characteristic of requiring a high data rate but not sensitive to the latency requirement, whereas the URLLC has the characteristic of requiring a low latency, but not sensitive to the data rate requirement. We aim to focus on the URLLC slices allocation, which shows the main uniqueness of 5G network to the previous generations. The eMBB slices has been considered to highlight the distinct requirement of latency and data rate. For simplicity, we do not consider mMTC slice in our problem.

In our problem, a network slice is assumed to be a Service Function Chain (SFC) [17]. An SFC is composed of (ingress and egress) *endpoints* and chained VNFs. Each VNF requires a different amount of vCPU and storage [18]. Two VNFs are interconnected by a *Virtual Link* (VL), which requires an amount of data rate [18]. Finally, a network slice requires also an end-to-end delay, which is determined by the sum of delay in each VL in the SFC from the ingress endpoint to the egress endpoint [19, 20]. We assume that the computation delay for running each VNF in the SFC is constant and does not depend on the CP where a VNF is allocated [21]. The InP guarantees dedicated computing resources for the VNF in any CPs, ensuring a constant computing delay regardless of the CP allocation decision. In contrast, the network delay varies based on the allocation decision and chosen configuration. For this reason, in our analysis, the end-to-end delay is only related to the network component.

Since this paper does not include the access network, the endpoints are assumed to be located at a MEH. The endpoints of an eMBB service can be located far away from each other (i.e., located in a different MEC cluster). On the contrary, the endpoints of a URLLC service are located close to each other, e.g., within the same cluster. The URLLC endpoints assumption is taken from a *low latency zone* concept from ETSI specification in [10].

A slice allocation scenario is illustrated in Figure 2. STs have requested two slices composed as SFCs. The first SFC (SFC-A) consists of one ingress endpoint (I_A), two VNFs ($VNF1_A$ and $VNF2_A$), and one egress endpoint (E_A). SFC-A is a URLLC slice, and both ingress and egress endpoints are located within the same cluster. The second SFC (SFC-B) consists of one ingress endpoint (I_B), three VNFs ($VNF1_B$, $VNF2_B$, and $VNF3_B$), and one egress endpoint (E_B). SFC-B is an eMBB slice, in which the ingress and egress endpoints can be located on different clusters. One of the possible solutions to allocate SFC-A and SFC-B is to spread the VNFs on different computing platforms. The $VNF1_A$ and $VNF2_A$, which are part of SFC-A, are allocated on different MEHs within the same cluster. The $VNF1_B$, $VNF2_B$, and $VNF3_B$ are allocated to one MEH and two DCs. The solution can vary depending on how many slices are requested and where the endpoints are located. In our problem, we allocate all the slice requests (and select the configurations of CPs and LCs) at the same time. We do not make assumptions about the dynamic organization of the allocation process, e.g. request-by-request allocation, time-slotted allocation, reallocation of previous requests, etc. In this way, we can better understand the theoretical limit of cost minimization. Moreover, we assumed that the SB will accept and allocate all requests. Consequently, cost minimization implies profit maximization.

The problem addressed by this paper can be summarized as follows. The STs generate the network slice requests to

Journal Pre-proof

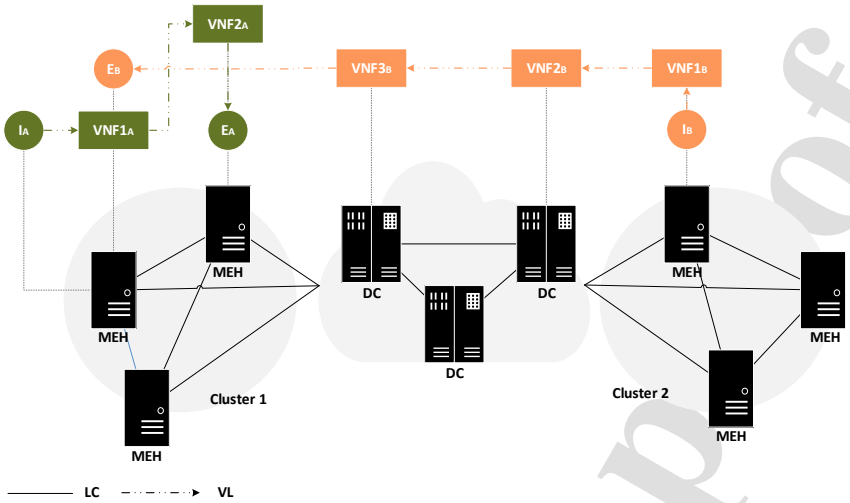


Figure 2: Example of network slice allocation

SB. The requests consist of the requirements for each VNF and VL, the required end-to-end delay, and the endpoints' location. The SB selects which configuration (if any) to buy for each CP and LC from InPs. The SB jointly allocates the network slices in the selected CPs and LCs to meet the network slice requirements. The objective of the joint configuration selection and network slice allocation is to minimize the SB's cost.

3. Related Works

An efficient resource allocation is essential for achieving cost-effectiveness in network slicing, helping InPs or SB to balance network capacity with user demand and optimize network performance while controlling costs. This section highlights the state of the art of the slice allocation problems that optimize the cost and how the researchers frame the problem. Key features of this problem that we would like to highlight include how the cost is formulated, whether a generic or economic cost is considered, the inclusion of a MEC system, and the modelling of the slice.

Some papers [22, 23, 24, 25] provide techno-economic analysis of slice allocation of MEC-enabled network, and find the trade-off between resource usage, service performances, cost and revenue. Other papers [26, 19] also provide a techno-economic analysis of a similar problem but do not consider the MEC system. Studies in [22, 26, 23] evaluate the slice admission, and focus on maximizing the provider's revenue by selecting which slices needs to be admitted with SLA constraints. Studying resource admission is a critical aspect of network slicing. However, it is equally essential to identify an optimal approach for allocating specific slices in the provided infrastructure. Some slice allocation problem involves managing resource fairness [19, 24] to minimize the slice-serving costs [25]. Other papers such as [27, 28, 29] study the generic cost of slice allocation, not the economic cost. The authors consider a generic cost that occurred from resource usage.

Most of the studies examine the cost-effectiveness of slice allocation from the perspective of InPs. Our problem considers a novel business model that involves three actors: STs, SBs and InPs, instead of only considering STs and InPs as in other related works. With this business model, an SB must carefully evaluate the slice allocation strategy to be profitable and stay in business for the long run. Some few works [30] and [31] analyze a slice allocation problem from the perspective of SB. In [30], the authors formulate a problem that can minimize the SB cost from buying resources of DC and MEH, which the InPs provide. In [31], the authors evaluate the SB revenue by resource usage and SLAs.

Understanding how researchers model the network slices is critical for assessing the effectiveness of these models in achieving cost efficiency and meeting the needs of InP and SBs. In practice, 5G network slices can be differentiated into three isolation types: Radio Access Network (RAN) slicing, RAN-transport slicing, and RAN-transport-core slicing [32]. RAN slicing refers to the isolation of radio functions related to the radio resources (i.e., physical resource block (PRB). RAN-transport slicing refers to isolating radio and transport functions, including network resources. RAN-transport-core slicing refers to the full isolation of an independent virtual infrastructure that includes core, network and radio functions.

Generally, studies have been carried out to evaluate RAN-slicing [26, 30, 31], or a high-layer perspective of slice that can fall into RAN-transport or RAN-transport-core slicing [33, 24, 25, 19, 27, 28, 29]. Based on the

Journal Pre-proof

slice isolation categories, the network slice can be modelled accordingly. The common approach of RAN-slicing is to model a network slice as a fluid entity that needs to decide the number of PRBs and the user association [26, 30, 31]. The common approach of high-layer isolation is to model the network slice as an SFC composed of VNFs [33, 24, 25, 19, 27, 28, 29]. The slice as SFC allocation problem is then further divided into SFC resource partition (e.g., deciding how many resource should be allocated to each VNF request [24, 25, 28]) and VNFs embedding (e.g., deciding which VNF should be allocated to which part of the infrastructure [22, 23, 27, 29]). However, most slice allocation and embedding problems consider a physical infrastructure, which a single InP offers. In our problem, we evaluate the slice allocation of high-layer isolation, which decides the VNF allocation to virtual resources that different InPs offer. Our assumption supports the real case scenario, where several InPs offer predetermined virtualised resource configurations to intermediary entities. The slice allocation among different InPs has a similar concept to a multi-domain Virtual Network Embedding (VNE) [34, 35, 36] or multi-domain slice allocation [37, 38]. The VNE concept has a slight difference compared to slice allocation. The virtual network is a logical representation of a network providing a generic purpose, e.g., a cloud network. The slice network is an isolated end-to-end network with a specific purpose, e.g., for vertical industries or MVNOs. In a multi-domain VNE, a broker is needed to customize resources provided by InPs, whether integrated within the Service Provider (SP) [34] or operating individually as a global controller [35, 36]. In [38], a network slice is configured directly by an SP, while in [37], there is a presence of a resource manager that configure the slice resources from various InPs and sell them to the slice tenants. However, the authors of [34, 35, 36, 37, 38] do not consider a broker as a third-party entity and consequently use a negotiation scheme to define the amount of resources and the price. Our paper adopts an alternative approach where each InP can sell predetermined resource configurations. Our approach is similar to selecting *flavour types* when buying a cloud instance [7], avoiding a negotiation among InPs. Moreover, VNE in [37] only considers cloud DC, whereas we consider both DC and MEH as the option of computing platforms.

VL allocation is another important aspect in modelling a slice as an SFC. In [23, 27, 29], the VLs are mapped to physical connections based on the allocation of VNFs. However, the authors primarily consider the SFC allocation strategy from the perspective of the InPs, thus assuming a physical infrastructure is relevant for the SFC allocation. In our problem formulation, the VL is mapped differently. We map the VLs to the LCs as an SB only observes virtual CPs offered by the cloud provider and the LCs provided by the network operators. Each LC has three predetermined resource and price configurations for the SB to select. The configuration follows a similar scheme to the data rate subscription guarantee offered by MNO [39].

For this assumption, we jointly allocate the computing (VNF allocation) and network (VL allocation) resources. Most existing works consider only computing resources when calculating the cost [22, 25, 26, 30], and only a few works jointly consider computing and network resources [40, 19, 23]. Since we assume that all the network slice requests are allocated, the revenue is not depending on the allocation. Consequently, cost minimization implies profit maximization.

In summary, this paper is the first to formulate a slice allocation problem that combines several characteristics, including: (i) direct economic cost minimization, (ii) perspective of a SB, (iii) integration of the MEC system, (iv) predetermined configurations of InPs' resources, and (v) optimal solutions. While [40] also considers predetermined configurations, their solution is not optimal.

4. Problem Formulation

In Table 1, all the parameters that will be considered in the formulation of the problem are defined. A network slice $n \in \mathcal{N}$) is modelled as an SFC, and composed by endpoints $e^I, e^E \in \mathcal{E}$, VNFs $\mathcal{V}_n$, and VLs $\mathcal{L}_{gh}$ where $g, h \in \{\mathcal{V}_n \cup \mathcal{E}\}$. The STs request network slices $\mathcal{N}$. Within the SLA between ST and SB, there are two parts of the agreement. First part of the agreement is that the SB guarantees four parameters: end-to-end delay of the slice δ_n, processing resource π_v and storage resources θ_v for each VNF $v \in \mathcal{V}_n$, and data rate ρ_{gh} for each VL $(g, h) \in \mathcal{L}_{gh}$. The second part is that the ST pays the hourly price of network slice n. The SLA between ST and SB can be summarized as follows:

$$SLA(\delta_n, \pi_v \text{ and } \theta_v \forall v \in \mathcal{V}_n, \rho_{gh} \forall (g,h) \in \mathcal{L}_n) \Rightarrow q_n^N \quad (1)$$

Based on these network slice requests $\mathcal{N}$, the SB needs to lease resources from InPs. Given the defined parameters, for each CP $d \in \mathcal{D}$, one of the configurations $k \in \mathcal{K}_d$ can be selected by the SB. The SB leases computing resources from several cloud providers, and pays the hourly price of the selected configuration q_k^K in Euro (€). Each configuration $k \in \mathcal{K}_d$ is characterized by the processing resource m_k^P, storage resource m_k^S, ICR m_k^H, and ICD m_k^F. Note that several VNFs can be allocated to one CP. Therefore, the data rate and delay of VLs that connecting VNFs within the same CP is important to be guaranteed as well. The SLA between the SB and a cloud provider can be summarized as follows:

$$SLA(m_k^P, m_k^S, m_k^H, m_k^F) \Rightarrow q_k^K \quad (2)$$

For each LC $(i, j) \in \mathcal{C}$, one of the configurations $w \in \mathcal{W}_{ij}$ can be selected by the SB. The SB leases network resources from several network providers, and pays the hourly price of the selected configuration q_k^W (in €). The network provider guarantee two parameters, which are the data rate m_w^R and delay m_w^T. Therefore, the SLA between the SB and a network operator can be defined as follows:

$$SLA(m_w^R, m_w^T) \Rightarrow q_w^W \quad (3)$$

Journal Pre-proof

Table 1
Parameter definitions

Symbol [UoM]	Description
Sets	
$\mathcal{E}$	Set of endpoints
$\mathcal{D}$	Set of CPs
$\mathcal{K}_d$	Set of configurations for $d \in \mathcal{D}$
$\mathcal{C}$	Set of LCs (i,j), $i \in \mathcal{D}$, $j \in \mathcal{D}$
$\mathcal{W}_{ij}$	Set of configurations for $(i,j) \in \mathcal{C}$
$\mathcal{N}$	Set of network slice requests
$\mathcal{V}_n$	Set of VNFs for $n \in \mathcal{N}$
$\mathcal{L}_{gh}$	Set of VLs (g,h), $g \in \{\mathcal{V}_n \cup \mathcal{E}\}$, $h \in \{\mathcal{V}_n \cup \mathcal{E}\}$ for $n \in \mathcal{N}$
Given Variables	
q_k^K [€/h]	Hourly price of $k \in \mathcal{K}_d$ for $d \in \mathcal{D}$
q_w^W [€/h]	Hourly price of $w \in \mathcal{W}_{ij}$ for $(i,j) \in \mathcal{C}$
q_n^N [€/h]	Hourly price of $n \in \mathcal{N}$
m_k^P [vCPU]	Amount of processing for $k \in \mathcal{K}_d$ in $d \in \mathcal{D}$
m_k^S [GiB]	Amount of storage for $k \in \mathcal{K}_d$ in $d \in \mathcal{D}$
m_k^H [Mb/s]	Amount of data rate for $k \in \mathcal{K}_d$ in $d \in \mathcal{D}$
m_k^F [ms]	Amount of delay for $k \in \mathcal{K}_d$ in $d \in \mathcal{D}$
m_w^R [Mb/s]	Amount of data rate for $w \in \mathcal{W}_{ij}$ in $(i,j) \in \mathcal{C}$
m_w^T [ms]	Amount of delay for $w \in \mathcal{W}_{ij}$ in $(i,j) \in \mathcal{C}$
δ_n [ms]	E2E delay needed from $n \in \mathcal{N}$
π_v [vCPU]	Processing needed from $v \in \mathcal{V}_n$ in $n \in \mathcal{N}$
θ_v [GiB]	Storage needed from $v \in \mathcal{V}_n$ in $n \in \mathcal{N}$
ρ_{gh} [Mb/s]	Data rate needed from $(g,h) \in \mathcal{L}_n$ in $n \in \mathcal{N}$
γ_{ed}	1 if $e \in \mathcal{E}$ located at $d \in \mathcal{D}$, 0 otherwise
Unknown Variables	
a_k	1 if $k \in \mathcal{K}_d$ is set on $d \in \mathcal{D}$, 0 otherwise
b_w	1 if $w \in \mathcal{W}_{ij}$ is set on $(i,j) \in \mathcal{C}$, 0 otherwise
x_v^d	1 if $v \in \mathcal{V}_n$ of $n \in \mathcal{N}$ is allocated on $d \in \mathcal{D}$, 0 otherwise
y_{gh}^{ij}	1 if $(g,h) \in \mathcal{L}_n$ of $n \in \mathcal{N}$ is allocated on $(i,j) \in \mathcal{C}$, 0 otherwise
z_{gh}^d	1 if $(g,h) \in \mathcal{L}_n$ of $n \in \mathcal{N}$ is allocated on $d \in \mathcal{D}$, 0 otherwise

We propose a mathematical formulation of a slice allocation problem that minimizes the overall SB cost, by selecting configuration of CPs and LCs, constrained to the slice requirements. The objective function is defined in Eq 4. The first addend is the cost for virtual resources in the CPs. The second addend is the cost for the LCs. The SB provides readily configured slices, which consume both computing and network resources. Therefore, jointly minimizing both computing and network resources is important.

For the configuration selection, two binary decision variables have been defined: CP selection a_k, and LC selection b_k (see Eq. 5 and Eq. 6). For the slice allocation, three other decision variables have been defined: VNF allocation x_v^d, VL allocation on LC y_{gh}^{ij}, and VL allocation within a CP z_{gh}^d.

The objective function is subject to conditional constraints Eq. 5,6,7,8,9,10,11,12,13,14,15,16,17,18,19 and capacity constraints Eq. 20,21,22,23,24. The SB can only select one configuration for each CP and LC (see. Eq.7 and Eq. 8). Each VNF of each network slice must be allocated to one and only one CP, as in Eq. 12. Each VL of each network slice must be allocated on one and only one LC or intra-connection of a CP, as in Eq. 13.

$$\min \sum_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}_d} q_k^K \cdot a_k + \sum_{(i,j)\in\mathcal{C}} \sum_{w\in\mathcal{W}_{ij}} q_w^W \cdot b_w \tag{4}$$

subject to:

$$a_k \in \{0,1\} \qquad \forall k \in \mathcal{K}_d, \forall d \in \mathcal{D} \tag{5}$$

$$b_w \in \{0,1\} \qquad \forall w \in \mathcal{W}_{ij}, \forall (i,j) \in \mathcal{C} \tag{6}$$

$$\sum_{k\in\mathcal{K}_d} a_k \leq 1 \qquad \forall d \in \mathcal{D} \tag{7}$$

$$\sum_{w\in\mathcal{W}_{ij}} b_w \leq 1 \qquad \forall (i,j) \in \mathcal{C} \tag{8}$$

$$x_v^d \in \{0,1\} \qquad \forall v \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \tag{9}$$

$$y_{gh}^{ij} \in \{0,1\} \qquad \forall (g,h) \in \mathcal{L}_n, \forall n \in \mathcal{N}, \forall (i,j) \in \mathcal{C} \tag{10}$$

$$z_{gh}^d \in \{0,1\} \qquad \forall (g,h) \in \mathcal{L}_n, \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \tag{11}$$

$$\sum_{d\in\mathcal{D}} x_v^d = 1 \qquad \forall v \in \mathcal{V}_n, \forall n \in \mathcal{N} \tag{12}$$

$$\sum_{(i,j)\in\mathcal{C}} y_{gh}^{ij} + \sum_{d\in\mathcal{D}} z_{gh}^d = 1 \qquad \forall (g,h) \in \mathcal{L}_n, \forall n \in \mathcal{N} \tag{13}$$

$$\begin{gathered} y_{gh}^{ij} \leq 0.5 \cdot x_g^i + 0.5 \cdot x_h^j \\ \forall (g,h) \in \mathcal{L}_n : g \in \mathcal{V}_n, h \in \mathcal{V}_n, \\ \forall n \in \mathcal{N}, (i,j) \in \mathcal{C} \end{gathered} \tag{14}$$

Journal Pre-proof

$$z^d_{gh} \leq 0.5 \cdot x^d_g + 0.5 \cdot x^d_h \quad \forall (g,h) \in \mathcal{L}_n : g \in \mathcal{V}_n, h \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \tag{15}$$

$$y^{ij}_{eh} = \gamma_{ei} \cdot x^j_h \quad \forall (e,h) \in \mathcal{L}_n : e \in \mathcal{E}, h \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall (i,j) \in \mathcal{C} \tag{16}$$

$$y^{ij}_{ge} = \gamma_{ej} \cdot x^i_g \quad \forall (g,e) \in \mathcal{L}_n : e \in \mathcal{E}, g \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall (i,j) \in \mathcal{C} \tag{17}$$

$$z^d_{eh} = \gamma_{ed} \cdot x^d_h \quad \forall (e,h) \in \mathcal{L}_n : \forall e \in \mathcal{E}, h \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \tag{18}$$

$$z^d_{ge} = \gamma_{ed} \cdot x^d_g \quad \forall (g,e) \in \mathcal{L}_n : e \in \mathcal{E}, g \in \mathcal{V}_n, \forall n \in \mathcal{N}, \forall d \in \mathcal{D} \tag{19}$$

$$\sum_{n \in \mathcal{N}} \sum_{v \in \mathcal{V}_n} \pi_v \cdot x^d_v \leq \sum_{k \in \mathcal{K}_d} a_k \cdot m^P_k \qquad \forall d \in \mathcal{D} \tag{20}$$

$$\sum_{n \in \mathcal{N}} \sum_{v \in \mathcal{V}_n} \theta_v \cdot x^d_v \leq \sum_{k \in \mathcal{K}_d} a_k \cdot m^S_k \qquad \forall d \in \mathcal{D} \tag{21}$$

$$\sum_{n \in \mathcal{N}} \sum_{(g,h) \in \mathcal{L}_n} \rho_{gh} \cdot z^d_{gh} \leq \sum_{k \in \mathcal{K}_d} a_k \cdot m^H_k \qquad \forall d \in \mathcal{D} \tag{22}$$

$$\sum_{n \in \mathcal{N}} \sum_{(g,h) \in \mathcal{L}_n} \rho_{gh} \cdot y^{ij}_{gh} \leq \sum_{w \in \mathcal{W}_{ij}} b_w \cdot m^R_w \qquad \forall (i,j) \in \mathcal{C} \tag{23}$$

$$\sum_{(g,h) \in \mathcal{L}_n} \sum_{(i,j) \in \mathcal{C}} \sum_{w \in \mathcal{W}_{ij}} b_w \cdot m^T_w \cdot y^{ij}_{gh} + \sum_{(g,h) \in \mathcal{L}_n} \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} a_k \cdot m^F_k \cdot z^d_{gh} \leq \delta_n \quad \forall n \in \mathcal{N} \tag{24}$$

Eq. 14 and Eq. 15 further define the condition of VL allocation to LC or intra-connection of a CP. Eq. 14 means that a VL can be allocated to a LC only if the corresponding VNFs are allocated on the corresponding CPs. For example, a VL (i, j) can be allocated to LC (g, h), only if the VNFs i and j are allocated to the CPs g and h, respectively. Similarly, Eq. 15 means that a VL (i, j) can be allocated to an intra-connection of the CP d only if the corresponding VNFs i and j are both allocated on the CP d.

Eq. 16, Eq. 17, Eq. 18, and Eq. 19 define the VL allocation condition with respect to the ingress and egress endpoints. Eq. 16 means that a VL from the ingress endpoint can be allocated only to a LC from the CP where the endpoint is located ($\gamma_{ei} == 1$) to the CP where the first VNF is allocated. Eq. 17 means that a VL to an egress endpoint is allocated only in a LC from the CP where the last VNF is allocated to the CP where the endpoint is located ($\gamma_{ej} == 1$). Eq. 18 allows a VL from an ingress endpoint to be allocated on the CP where the endpoint is located ($\gamma_{ed} == 1$), if the first VNF is allocated on the same CP. Eq. 19 allows a VL to an egress endpoint to be allocated on the CP where the endpoint is located ($\gamma_{ed} == 1$), if the last VNF is allocated on the same CP.

Eq. 20, Eq. 21, Eq. 22, Eq. 23, Eq. 24 define the capacity constraints for both computing and network resources. Eq. 20 limits the allocations of the VNFs in a CP to the processing capacity of the CP. Eq. 21 limits the allocations of the VNFs in a CP to the storage capacity of the CP. Eq. 22 limits the allocations of the VLs in a intra-connection to the data rate capacity of the intra-connection. Eq. 23 limits the allocations of the VLs in a LC to the data rate capacity of the LC. Lastly, Eq. 24 guarantees that the end-to-end delay for each network slice respects the requirement.

The formalized problem is formulated as a Mixed Integer Quadratically Constrained Program (MIQCP) due to one non-linearity constraint (Eq. 24). This problem can be efficiently solved by CPLEX.

4.1. Problem Complexity

In this subsection, we compute the complexity of our proposed problem. The problem complexity analysis provides an insight into the upper bounds of time required for solving a problem. We calculate the problem complexity based on a number of unknown variables and constraints of the problem as formalized in [41]. The problem complexity is provided in Eq 25 and Eq 26.

$$N_{vars} = \sum_{d \in \mathcal{D}} |\mathcal{K}_d| + \sum_{(i,j) \in \mathcal{C}} |\mathcal{W}_{ij}| + |\mathcal{D}| \cdot \sum_{n \in \mathcal{N}} |\mathcal{V}_n| + (|\mathcal{C}| + |\mathcal{D}|) \cdot \sum_{n \in \mathcal{N}} |\mathcal{L}_n| \tag{25}$$

$$\begin{aligned} M_{const} &= 4 \cdot |\mathcal{D}| + 2 \cdot |\mathcal{C}| + \sum_{n \in \mathcal{N}} |\mathcal{V}_n| + \sum_{n \in \mathcal{N}} |\mathcal{L}_n| + |\mathcal{C}| \cdot \sum_{n \in \mathcal{N}} |\mathcal{L}_n| + |\mathcal{D}| \cdot \sum_{n \in \mathcal{N}} |\mathcal{L}_n| + |\mathcal{N}| \\ &= 4 \cdot |\mathcal{D}| + 2 \cdot |\mathcal{C}| + \sum_{n \in \mathcal{N}} (|\mathcal{V}_n| + 1) + (|\mathcal{C}| + |\mathcal{D}| + 1) \cdot \sum_{n \in \mathcal{N}} |\mathcal{L}_n| \end{aligned} \tag{26}$$

Eq. 25 depicts the complexity based on the unknown variables, where the first addend is of N_{vars} for a_k, the

Journal Pre-proof

second addend is for b_w, the third addend is for x_v^d, the fourth and five is for y_{gh}^{ij} and z_{gh}^d respectively. Eq. 25 depicts the complexity based on the constraints. The first part of Eq. 25 explains the more detailed version of constraints complexity, which is further simplified in the second part of the equation. The M_{const} can be explained as follows. The first addend is for Eq. 7, Eq. 20, Eq. 21, and Eq. 22, the second addend is for Eq. 8 and Eq. 23, the third addend is for Eq. 12, the fourth addend is for Eq. 13, the fifth addend is for Eq. 14, Eq. 16, and Eq. 17, the sixth addend is for Eq. 15, Eq. 18, and Eq. 19, and the seventh addend for Eq. 24. The M_{const} is simplified as the second part of the Eq 26.

The dominant term from the number of unknown variables is $(|\mathcal{C}| + |\mathcal{D}|) \cdot \sum_{n\in\mathcal{N}} |\mathcal{L}_n|$, whereas the dominant term from the number of constraints is $(|\mathcal{C}| + |\mathcal{D}| + 1) \cdot \sum_{n\in\mathcal{N}} |\mathcal{L}_n|$. The problem complexity for this scenario depends on the set of configurations in LCs and CPs and the set of VLs, which depends on the set of VNFs within each network slice. Therefore, the problem complexity can be stated to be in the order of $\mathcal{O}\left((|\mathcal{C}| + |\mathcal{D}| + 1) \cdot \sum_{n\in\mathcal{N}} |\mathcal{L}_n|\right)$.

5. Evaluation

In Section 4, we have introduced a novel mathematical formulation of the slice allocation problem from the perspective of the SB, aiming to minimize the costs associated with purchasing computing and network resources. To evaluate our problem, we compare our proposed problem with other reference problems. These reference problems are formulated based on the state of the art of cost minimization problems. Most of the papers minimize the cost by doing a resource consolidation. In the resource consolidation problem, the total cost depends on the number of MEH, or hops of the network path [22, 27, 23]. These papers indirectly target minimizing the computing and network cost by minimizing the number used MEH and choosing the shortest path. In contrast, our problem directly targets minimizing the cost of computing and network resources. We use the CPLEX optimization tool to obtain the optimal solution for all the problems. The optimal solution provides insight into the theoretical limits of achieving the objective in our problem formulation. Having the optimal solution can highlight the potential gains and benefits that can be achieved.

In this section, we present the evaluation settings, scenarios, and the discussion of the results and implications. All evaluations were performed using virtual machines with 16 vCPUs, 32 GB RAM. The operating system is Ubuntu 18.04.6 LTS (Bionic Beaver), with IBM ILOG CPLEX Optimization Studio 20.1.0 installed. To solve the MIQCP problem, CPLEX does a pre-processing to relax the problem and uses the branch-and-cut method to find the optimal solution [42].

5.1. Evaluation Settings

This subsection describes the assumptions and settings that implemented for this study. First, the settings for the InP resources are presented. Second, the settings for the network slice requests are presented.

Table 2
Configurations for the DCs [40]

Config. k	m_k^P [vCPU]	m_k^S [GiB]	m_k^H [Gb/s]	m_k^F [ms]	q_k^K [€/h]
1	36	7×10^3	10	1	1.521
2	42	4×10^3	10	1	1.954
3	64	14×10^3	25	1	4.173

Table 3
Configurations for the MEHs [40]

Config. k	m_k^P [vCPU]	m_k^S [GiB]	m_k^H [Gb/s]	m_k^F [ms]	q_k^K [€/h]
1	4	150	25	1	0.272
2	8	300	25	1	0.544
3	16	400	10	1	0.768

5.1.1. InP Resources

As presented in previous sections, the cloud providers sell CP configurations according to the SLA expressed in Eq. 2. We assume that all the DCs have the same configurations with the values shown in Table 2. All DCs are available on the core-level network. We assume that all the MEHs have the same configurations with the values shown in Table 3. The MEHs and DCs have different hardware characteristics, as the MEHs usually use a multi-core system instead of multiprocessor system. The multi-core system has characteristic of lower power consumption and lower cost. The configurations in the tables data are taken from [40], which have been extrapolated from the realistic configuration offered by cloud providers [7, 16, 8]. The MEH pricing structure, as indicated in Table 3, has been designed to reflect a higher cost per computing unit in comparison to the DC configuration price. This pricing structure is motivated by the unique attributes of MEH, which include limited computing capacity, enhanced feature accessibility (e.g., radio network information service, location service [43]), and lower latency compared to the DC. This pricing is also affected by the expensive deployment cost [44]. We split the MEHs in different clusters. We assume a homogeneous cluster size, meaning that each cluster have the same number of MEHs. The association of MEH and cluster is done randomly.

The network operators sell LC configurations between the CPs according to the SLA expressed in Eq. 3. We assume that the DCs can be connected to all the other DCs, and these LCs have the same configurations with the values shown in Table 4. We also assume that the DCs can be connected to all the MEHs, and these LCs have the same configurations with the values shown in Table 5. Furthermore, we assume the

Journal Pre-proof

Table 4
Configurations for LCs interconnecting DCs [40]

Config. w	m_w^R [Gb/s]	m_w^T [ms]	q_w^W [€/h]
1	10	2.8	0.247
2	20	2.1	0.342
3	25	4.7	0.288

Table 5
Configurations for LCs interconnecting MEHs and DCs [40]

Config. w	m_w^R [Gb/s]	m_w^T [ms]	q_w^W [€/h]
1	1	1.5	0.164
2	4	2.7	0.247
3	8	3.5	0.288

Table 6
Configurations for LCs interconnecting MEHs [40]

Config. w	m_w^R [Gb/s]	m_w^T [ms]	q_w^W [€/h]
1	1	1.3	0.205
2	10	1.9	0.288
3	10	3.2	0.260

Table 7
Service summary [40]

Services	Category	SFC
VIDS	eMBB	NAT-FW-TM-VOC-IDPS
PATM	eMBB	NAT-TM-WOC-IDPS
POWM	URLLC	TM-IDPS
ROBT	URLLC	NAT-TM

MEHs can be connected to only the other MEHs within the same cluster, and these LCs have the same configurations with the values shown in Table 6. Of course, a network operator could provide an LC between MEHs in different clusters. However, these LCs would not be profitable for both InPs and SB because they have expensive configurations with high delay and low data rates. It is important to note that the clusterization of MEHs in our study can be attributed to various factors, including geographical locations, provider companies, or managerial reasons [10]. Moreover, the interconnection of MEHs within a cluster is established through LCs, regardless of their physical proximity. All the LC configuration values are taken from [40], which have been computed from the realistic configuration offered by mobile network operators [45, 46].

5.1.2. Network Slice Requests

Two categories of network slices are considered. The first category is eMBB, characterized by a high data rate and insensitive to the delay requirement. The second category is URLLC, characterized by a low data rate but a strict low latency. In this paper, we consider two services per category: Video Streaming (VIDS) and Patient Monitoring (PATM) for eMBB, Power Grid Monitoring (POWM) and Robot Tolling (ROBT) for URLLC. The VNFs in the SFC refer to Network Address Translator (NAT), Firewall (FW), WAN Optimization Controller (WOC), Intrusion Detection Prevention System (IDPS), Video Optimization Controller (VOC) and Traffic Monitor (TM). Each service is delivered as an SFC that contains a selection of the following VNFs: Network Address Translator (NAT), Firewall (FW), WAN Optimization Controller (WOC), Intrusion Detection Prevention System (IDPS), Video Optimization Controller (VOC), and Traffic Monitor (TM). Table 7 summarises the characteristics of each service. The requirements for each service, according to the SLA with the STs expressed in Eq. (1), are shown in Tables 8, 9, 10, and 11, which values are derived from [40, 19, 18, 47]. Note that I and E refer to the ingress and egress endpoints, respectively. The location of each endpoint, γ_{ed}, is generated randomly for each network slice request. We assume an ingress endpoint can be located only at MEHs, whereas an egress endpoint can be located at either MEHs or DCs. We assume different endpoints γ_{ed} generation for each service. The egress endpoints of VIDS and ROBT can be located at a DC. The ingress endpoints of VIDS and ROBT can be located at a MEH. The assumption is motivated by the real practical scenario, where the VIDS user is located at a MEH, and the video is located at a DC. The user of ROBT is also located at a MEH, and the ROBT database is located at a DC. The PATM and POWM have both ingress and egress endpoints at MEHs. However, the ingress and egress endpoints of PATM can be located at MEHs in a different cluster, whereas the ingress and egress endpoints of POWM can only be located at MEHs within the same cluster. The motivation is because for the PATM, the users, i.e., a patient and a doctor, can be quite far apart. The POWM, however, has both ingress and egress endpoints in the same area., i.e., a sensor and actuator for controlling the power plants. Based on the above location constraints, all endpoints are generated randomly.

5.2. Evaluation Scenarios

To evaluate the problem we have formalized in the previous section (in the rest of the paper, we will refer to it as *Min.Cost*), we compare its optimal solution with the optimal solution of problems that have another objective but the same constraints. We define three other reference problems that perform a resource consolidation, which is usually considered an indirect measure of minimizing the total cost. In the resource consolidation problem, the total cost depends on the number of CP, or hops of the network path [22, 27, 23]. We provide three distinct reference problems that specifically address resource consolidation, representing the state of the art in cost optimization. Our goal is to highlight the difference between a problem that directly minimizes cost (Min.Cost) and similar problems typically found in the

Journal Pre-proof

Table 8
Requirements for VIDS [40]

δ_n [ms]	300	
VL (g,h)	ρ_{gh} [Mb/s]	
(I,NAT)	850	
(NAT,FW)	680	
(FW,TM)	680	
(TM,VOC)	680	
$(VOC,IDPS)$	510	
$(IDPS,E)$	510	
VNF v	π_v [vCPU]	θ_v [GiB]
NAT	2.4	4.7
FW	7.8	8.3
TM	2.3	4.1
VOC	8.3	8.3
$IDPS$	4.6	2.3

Table 9
Requirements for PATM [40]

δ_n [ms]	100	
VL (g,h)	ρ_{gh} [Mb/s]	
(I,NAT)	200	
(NAT,TM)	160	
(TM,WOC)	160	
$(WOC,IDPS)$	112	
$(IDPS,E)$	112	
VNF v	π_v [vCPU]	θ_v [GiB]
NAT	6.7	3.3
TM	6.6	3.3
WOC	3.3	9.7
$IDPS$	3.3	6.6

Table 10
Requirements for POWM [40]

δ_n [ms]	30	
VL (g,h)	ρ_{gh} [Mb/s]	
(I,TM)	200	
$(TM,IDPS)$	200	
$(IDPS,E)$	200	
VNF v	π_v [vCPU]	θ_v [GiB]
TM	0.6	0.3
$IDPS$	3.3	0.6

literature that usually perform a resource consolidation. The Min.Cost approach finds a solution with the minimum cost. In contrast, resource consolidation problems find a solution with the minimum number of CPs or network hops without necessarily guaranteeing the minimum cost.

Table 11
Requirements for ROBT [40]

δ_n [ms]	5	
VL (g,h)	ρ_{gh} [Mb/s]	
(I,NAT)	500	
(NAT,TM)	400	
(TM,E)	280	
VNF v	π_v [vCPU]	θ_v [GiB]
NAT	16.7	8.3
TM	16.6	8.3

The first reference problem, *Min.CP*, has the target to minimize the number of selected CPs by using the following objective function.

$$\min \sum_{d\in D} \sum_{k\in\mathcal{K}_d} a_k \tag{27}$$

The second reference problem, *Min.LC*, has the target to minimize the number of selected LCs by using the following objective function.

$$\min \sum_{(i,j)\in C} \sum_{w\in\mathcal{W}_{i,j}} b_w \tag{28}$$

The last reference problem, *Min.CP-LC*, has the target to minimize the number of selected CPs and LCs by using the following objective function.

$$\min \Big(\sum_{d\in D} \sum_{k\in\mathcal{K}_d} a_k + \sum_{(i,j)\in C} \sum_{w\in\mathcal{W}_{i,j}} b_w \Big) \tag{29}$$

The Min.CP and Min.LC only targets one type of resource, meaning that these problems aim to minimize the number of used CP or used LC and neglect the other. However, as joint optimization is important, we also evaluate the Min.CP-LC problem, which can be fairly compared to our proposed problem Min.Cost. In terms of the configuration, all reference problems select the cheapest configuration that can fit the resource allocated. For example, if Config.2 is enough to allocate all VNF and VL requests, then Config.2 is selected (not the Config.3, although it also fulfills the resource requirements).

The solution of Min.Cost is compared with the solution of the reference problems (Min.CP, Min.LC and Min.CP-LC). We consider a reference setting with eight network slice requests (2 VIDS, 2 PATM, 2 POWM, 2 ROBT), 16 CPs (4 DCs, 12 MEHs), and a cluster size equal to 3. To understand the behaviour of the various problems, we vary several parameters of the reference setting considering four evaluation scenarios:

1. *Scenario I*: We vary the *number of network slice requests* from 4 to 12. We keep the ratio between network slice categories constant: 50% are eMBB and 50% are URLLC. Within each category, the requests

Journal Pre-proof

are equally divided for each service belonging to the category. For example, when the number of requests is 8, there are four eMBB requests and 4 URLLC requests, 2 for each service.

2. *Scenario II*: We vary the *number of CPs* from 8 to 24. We keep the ratio between DCs and MEHs constant: 25% are DCs, and 75% are MEHs. For example, when the number of CP is 8, there are 2 DCs and 6 MEHs. The cluster size is constant, which is 3. The number of clusters linearly increases as the number of MEHs increase. The number of network slice requests is equal to 8 slices, of which 4 requests are eMBB slices and 4 requests are URLLC slices.
3. *Scenario III*: We vary the *URLLC ratio*, which is defined as the percentage of URLLC requests with respect to the total number of requests, from 25% to 75%. We keep the total number of network slice requests to 8.
4. *Scenario IV*: We vary the *cluster size* of MEH clusters, from 1 to 6. This implies that the number of clusters is inversely proportional to the cluster size.

For each evaluation scenario, we find the solutions of 10 instances; we compute the average cost and its 95% confidence intervals. Based on the tight confidence interval, 10 instances are enough to highlight the result difference of each problem. To obtain results in a reasonable time for every scenario, we set the maximum deterministic time limit for CPLEX to 1.5 million computation ticks.

Although the derived solutions are not always optimal, both the solutions of Min.Cost and Min.CP+LC have an average optimality gap between 1.5% to 8%. The optimality gap measures how close the solution found by CPLEX is to the true optimal solution, calculated as a difference between the best-known solution and the best bound. The optimality gap is 0% when the generated solution is optimal. The achieved results of Min.CP and Min.LC are always optimal, as the solutions are found under the time limit.

In practice, CPLEX can generate the solutions within a reasonable time for a small problem, e.g., from around 1 minute to 10 minutes, when we have a small number of requests and size of the infrastructure. A network slice is a virtualized network infrastructure that may be changed occasionally but not often. The network slice allocation will be re-computed when there is a need for scaling up, scaling down or sometimes new requests [48, 49]. Several studies show the reconfiguration can happen in less than 30 minutes [49], or within 700 timeslots [50], because the traffic in a network slice can be changed dramatically within that period. Therefore, we could say that 1 to 10 minutes is a reasonable time, while 60 minutes is not feasible.

In the case of Scenario I, the problem with 4 network slice requests and 16 CPs can be solved by CPLEX in approximately 60 to 600 seconds (equivalent to 26,000 to 257,000 computation ticks). However, with 12 network slice requests, CPLEX takes around 3,700 seconds (1.5 million computation ticks) to find a solution, exceeding the deterministic time limit. For Scenario IV, cases involving 8 CPs and 24 CPs with 8 NSs exceed the time limit and return a solution within 6,200 to 7,000 seconds. The most sensitive parameter is the number of network slice requests. There is a drastic difference in processing time from 4 network slice to 6 network slice requests. The CPLEX needs less than 10 minutes to solve the 4 network slice problems, while it needs around 1 hour to solve the 6 network slice problem. Consequently, an alternative strategy that can find the trade-off between optimality and solving time is required to solve larger problem instances practically.

Figure 3 shows the results given by the solution of our problem and the reference problems for each of the evaluation scenarios. The y-axis is the hourly cost (€/h) that the SB needs to pay to the InP, whereas the x-axis plots the various configurations for each evaluation scenario. The cost is composed of the cost for the CPs and the cost for the LCs.

To understand the motivation of the behaviour in some evaluation scenarios, we have plotted the selected configurations for CPs and LCs. Figure 4 shows the configuration of the used CPs for the solution of each problem. The x-axis plots the various configurations for each evaluation scenario. The y-axis explains how many used CPs are selected and which configuration has been selected to allocate all the network slices. Configuration 1 (conf1) has the least capacity but is the cheapest one, and Configuration 3 (conf3) has the biggest capacity but is the most expensive one.

Figure 5 shows the number of LCs used in the solution of all the problems for each scenario. The y-axis represents the number of LC. The x-axis represents the various configurations for each evaluation scenario. configuration-1 has the smallest data rate capacity and the smallest latency. On the other hand, configuration-3 has the biggest data rate capacity and the biggest latency. As the LCs chosen between DC-DC, DC-MEH, and MEH-MEH connections are influenced by the chosen CPs, the connection type is not shown in the figure. Instead, the figure shows only the configuration taken among all connection types.

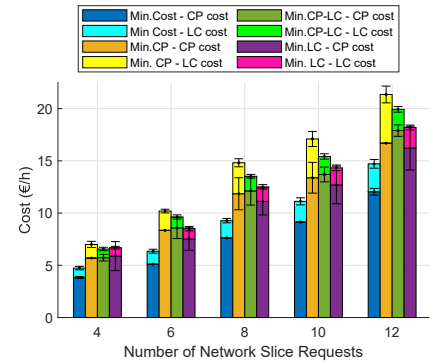
Moreover, to understand the allocation of the network slice requests, we have reported the number of SFCs and the number of VNFs per DC and MEH. Then, we have also reported the number of DCs and MEHs used to allocate an SFC for each service type. Noted that different service types have different number of VNFs and requirements.

5.3. Simulation Results: Scenario I

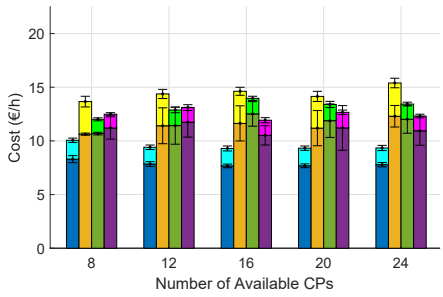
5.3.1. Cost Analysis

Figure 3a shows the cost for the solution of each problem when the number of network slice requests varies. When the number of network slice requests linearly increases, the cost for all the problems gradually increases. The Min.Cost provides the least cost compared to other problems, whereas the Min.CP has the most expensive cost. In case of a small number of requests, Min.CP, Min.CP-LC, and Min.LC have a cost 30% higher than Min.Cost. For a high number of requests, Min.LC has a cost 20% higher than Min.Cost, whereas Min.CP and Min.CP-LC still have a cost of around 30% higher. The cost of CPs is predominant to the cost

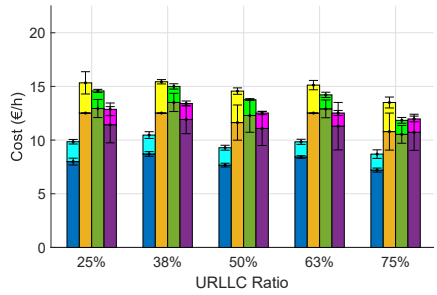
Journal Pre-proof



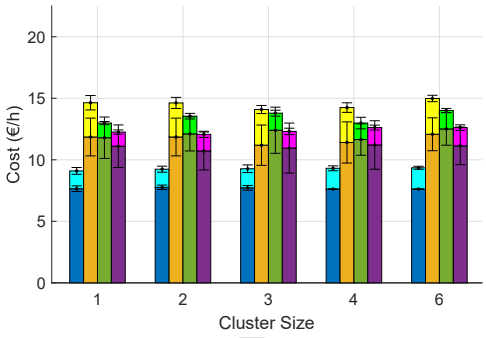
(a) Scenario I



(b) Scenario II



(c) Scenario III



(d) Scenario IV

Figure 3: Hourly cost for each evaluation scenario. Bars from left to right: Min.Cost, Min.CP, Min.CP-LC, Min.LC

of LCs since it is always more than 78% of the total cost. However, minimizing the number of CPs does not guarantee the least cost—instead, the Min.CP gives the most expensive solution compared to Min.Cost, Min.CP-LC, and Min.LC. The Min.LC gives the second-best performance regarding the total hourly cost.

5.3.2. CP Configuration Analysis

Figure 4a shows that the number of used CPs in our problem (Min.Cost) linearly increases when the number of slice requests increases. The Min.LC problem shows almost a similar behaviour to the Min.Cost. For Min.Cost and Min.LC problems, more slice requests imply that more MEHs are used. The other problems generate different solutions in terms of the used CP configuration. The number of used CPs for Min.Cost remains the same when the number of network slice requests changes from 8 to 12, and it only buys configurations from DCs. The Min. CP-LC has a behaviour similar to the Min.CP. Given the configuration options, purchasing more CPs with cheaper configurations can decrease the total cost. The Min.CP tends to buy the most expensive DC configuration, which offers the largest capacity (DC-conf3), and in most cases, never buys MEH configurations. The trade-off between selecting MEHs and DCs is also important, as a higher number of MEH does not necessarily give the least cost if we still buy an expensive DC configuration. The latter statement is based on comparing the results of Min.Cost and Min.LC. Thus, being aware of the available configurations is important as a slightly different CP configuration combination may affect the cost severely.

5.3.3. LC Configuration Analysis

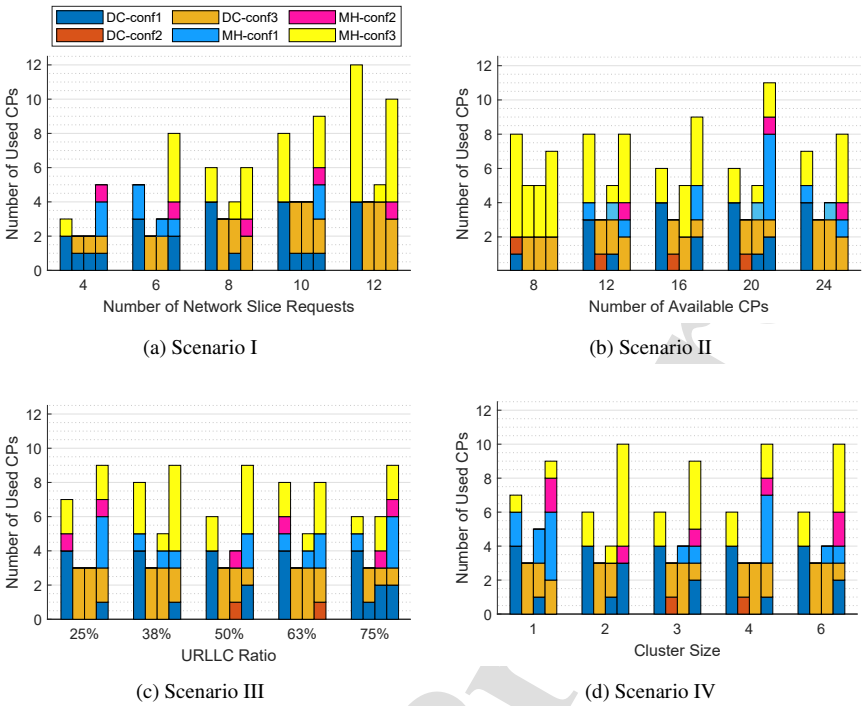
Figure 5a shows that the number of used LCs for all problems linearly increases when the number of network slice requests increases. Min.LC provides the least number of used LCs, whereas the Min.CP provides the highest number of used LCs. A higher number of network slice requests means more VNFs to be allocated. Increasing the number of VNFs means an increase in the number of VLs as well. The VLs are mapped to either intra-connection in a CP or logical connections between CPs. In Figure 5a, configuration-3 has never been selected to allocate the VLs. For almost all the problems, configuration-1 is the one most selected, and sometimes the, configuration-2 has also been selected. The Min.Cost has the second-worst performance regarding the number of used LCs, but it generates the least cost among all problems.

5.4. Simulation Result: Scenario II

5.4.1. Cost Analysis

The cost comparison of Scenario II is shown in Figure 3b. When the number of available CPs increases, the Min.Cost problem shows a small decrease in the total hourly cost. This result shows that when the infrastructure size gets bigger, i.e., more CPs are available, the Min.Cost problem can maintain almost the same cost. For the reference problems (Min.CP, Min.CP-LC, Min.LC), the cost performance

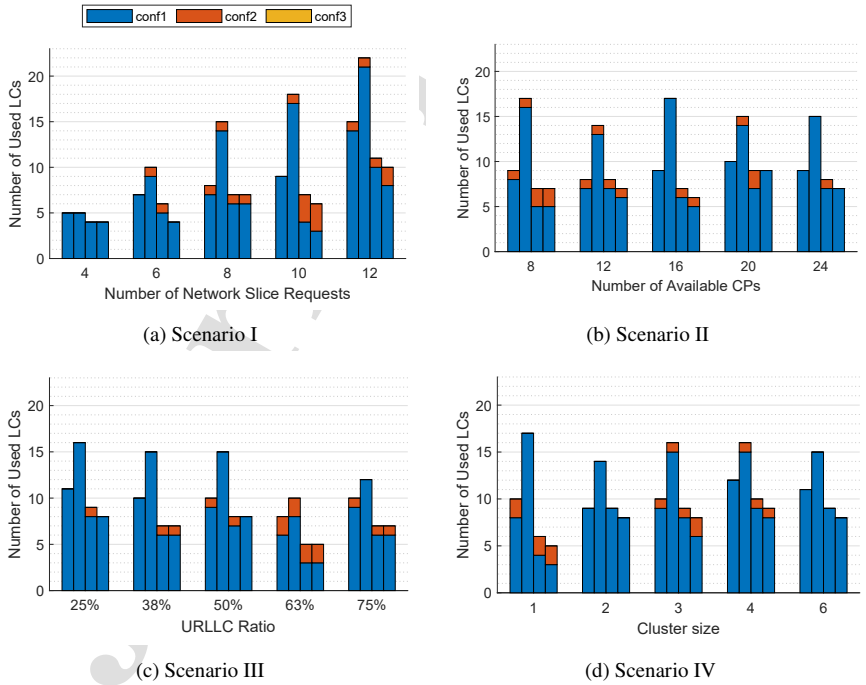
Journal Pre-proof



(a) Scenario I (b) Scenario II

(c) Scenario III (d) Scenario IV

Figure 4: Configuration of CPs for each evaluation scenario. Bars from left to right: Min.Cost, Min.CP, Min.CP-LC, Min.LC



(a) Scenario I (b) Scenario II

(c) Scenario III (d) Scenario IV

Figure 5: Configuration of LCs for each evaluation scenario. Bars from left to right: Min.Cost, Min.CP, Min.CP-LC, Min.LC

Journal Pre-proof

shows a more significant fluctuation. The reference problems are more affected by the change in the infrastructure magnitude. Min.CP has a slightly higher cost when the infrastructure grows, whereas the Min.CP-LC and Min.LC have a slightly cheaper expense. Aside from the slight fluctuation, Min.LC has the second-best cost performance among all the reference problems. The Min.CP has the worst performance in terms of cost, with almost 40% more expensive than the Min.Cost solution. In summary, a variation in the number of available CPs only slightly impacts the cost, especially for the Min.Cost.

5.4.2. CP Configuration Analysis

The result of the selected CP configurations when varying the number of available CPs is shown in Figure 4b. When we have 8 available CPs, all problems become capacity-limited, and the solutions to the various problems have only slight differences. However, when the number of available CPs increases, the solutions to the various problems differ more significantly. Min.CP has the least number of used CPs, and almost all the selected CPs are DCs with configurations 2 or 3. The Min.Cost and Min.LC have more used CPs, but their solution is cheaper than Min.CP and Min.CP-LC. The small cost of Min.Cost is since in a non-capacity-limited problem, Min.Cost never buys a DC-conf3 which is very expensive—instead, the Min.Cost uses a higher number of DC than the other problems but only buys the cheapest configuration, i.e., DC-conf1. Moreover, Min.Cost often buys the most expensive MEH configuration, i.e., MH-conf3, and cheaper MEH configurations less often.

5.4.3. LC Configuration Analysis

The result of the selected LC configurations when varying the number of available CPs is shown in Figure 5b. The number of used LCs for the Min.Cost, Min.CP-LC and Min.LC problems do not fluctuate significantly when the number of available CPs changes. Moreover, the Min.Cost problem has an almost steady performance regarding the number of used LCs. In contrast, the changes in the number of available CPs have significantly affected the number of used LCs for the Min.CP problem.

5.5. Simulation Results: Scenario III

5.5.1. Cost Analysis

The cost comparison of Scenario III is shown in Figure 3c. The increase in the URLLC ratio means we have a lower number of total VNFs but a higher number of network slices with a stricter latency requirement. When the URLLC ratio increases, the total hourly cost of all problems shows small fluctuations. However, if we compare the result for 25% and 75% URLLC ratios, we can see a slight decrease for all the problems. The Min.Cost always shows the best cost performance, whereas the Min.CP always shows the worst one. Min.LC has the second-best cost performance in almost all cases, except when the URLLC ratio is 75%. For 75%, the Min.LC costs slightly more compared to the Min.CP-LC. In summary, a change in the type of network slice has only a slight impact on the cost, especially for the Min.Cost.

5.5.2. CP Configuration Analysis

The number of used CPs for all problems when varying the URLLC ratio is shown in Figure 4c. Min.CP has the least number of CPs in all different URLLC ratios, whereas the Min.LC has a higher number of used CPs. Min.CP has the same behaviour as in the previous scenarios, i.e., it mostly uses 3 CPs with DC-conf3. The number of used CPs has only slight differences between Min.LC and Min.Cost. However, the selected configurations are highly different. The Min.Cost tends to buy more DCs but with DC-conf1, whereas the Min.LC tends to buy fewer DCs but with the combination of DC-conf1, DC-conf2 and DC-conf3. Moreover, Min.LC uses a higher number of MEHs, whereas Min.Cost uses a slightly less number of MEHs. For some cases, both problems buy a combination of MEH configurations (MH-conf1, MH-conf2, and MH-conf3). Figure 4c shows the significant cost difference between Min.LC and Min.Cost comes from the different selected configurations for the DCs, not for the MEHs, as the two problems have almost similar configurations for the MEHs.

5.5.3. LC Configuration Analysis

Figure 5c shows the number of used LCs for all problems when we increase the URLLC ratio. The number of used LCs decreases when the URLLC ratio increases. As previously mentioned, the increase in the URLLC ratio means a decrease in the total number of requested VNFs; hence the number of requested VLs will decrease. Thus, the decrease in used LCs for all problems is caused by the decreased number of requested VLs. Min.CP has the highest number of used LCs, whereas Min.CP-LC and Min.LC have the least number of used LCs. However, the Min.Cost, which gives the least cost in Figure 3c, has a high number of used LCs, almost close to the number of used LCs in the Min.CP problem. In summary, even with many used LCs, the Min.Cost problem still has the lowest cost. Therefore, the number of used LCs has a low impact on the total cost.

5.6. Simulation Results: Scenario IV

5.6.1. Cost Analysis

Figure 3d shows the cost comparison of Scenario IV. When the cluster size increases, there are no significant changes in cost for all problems. The figure highlights that a different cluster size does not affect all problems' costs. However, if we compare the different problems, Min.Cost still provides the lowest cost, whereas the Min.CP has the highest cost.

5.6.2. CP Configuration Analysis

The CP configurations when the cluster size varies for all problems are available in Figure 4d. As the cost between different cluster sizes remains stable, the number of used CPs also only shows slight differences. The number of used CPs for all problems remains the same. The only difference is when we have a cluster size 1, where Min.LC uses a lower number of DCs and Min.Cost uses a higher number of MEHs. In conclusion, the cluster size does not affect the

Journal Pre-proof

number of used CPs but might slightly change the chosen configuration for MEHs or DCs.

5.6.3. LC Configuration Analysis

The LC configurations when the cluster size varies for all problems are available on Figure 5d. When the cluster size equals 2 - 6, the number of used LCs remains steady for all problems. The only difference is when the cluster size is 1, especially for the Min.LC and Min.CP-LC problems. When each MEH is isolated, i.e., no MEH-MEH LCs available, the Min.CP-LC and Min.LC uses a lower number of LCs, whereas the Min.CP uses a higher number of LCs to allocate the VLs. However, whichever cluster size has been set, the Min.Cost problem has no significant changes concerning the number of used LCs.

5.7. Conclusion of the Scenario Analysis

If we analyze the four scenarios together, some important observations can be concluded. First, the Min.Cost always has the lowest cost since the reference problems have a cost of at least 30% higher. Second, the Min.CP has the highest cost, meaning that consolidation of CPs always leads to a costly solution in this evaluation setting. Third, the Min.LC has the second-best cost performance. Therefore decreasing the number of used LCs may give us a good solution if we cannot solve the Min.Cost problem. These three trends are valid for all scenarios, from Scenario I-IV. The Min.Cost problem has the lowest cost by allocating VNFs to more CPs and buying cheap DC configurations and expensive MEH configurations. If the InP provides several configurations, purchasing more CPs does not always lead to a higher cost. Instead, element consolidation, which is depicted by Min.CP problem may lead to a higher cost. However, solving the Min.Cost problem can be challenging depending on the size of the scenario, as the computation time might be high. The Min.Cost minimizes the cost directly, i.e., calculating the cost of each possible selection; thus, it has a bigger solution set than the other problems. It is solving a Min.LC problem that can be an alternative option, as the Min.LC has the second-best performance among all the reference problems. Min.LC tends to exploit the MEHs more compared to the DCs. In this way, the total hourly cost can be kept low.

Our findings are closely tied to the input data or configurations used in our study, and as such, may not be directly transferable to other configurations. However, it is important to note that our data was sourced from real-world information provided by network, cloud, and service providers [16, 7, 11, 40]. Additionally, we conducted a sensitivity analysis to gauge the potential impact of any changes to our results.

5.8. SFC Allocation Analysis

SFC allocation analysis helps to understand how the optimal solution can be achieved. The different costs (shown in Figure 3) and the different CP and LC configurations (shown in Figures 4 and 5, respectively) are also motivated by the different strategies to allocate network slices and their components (i.e., VNFs in the SFC). This subsection analyses the SFC and VNF distributions among the used CPs. Table 12 shows the average number of SFCs allocated on one CP (DC or MEH) for Scenario II, which allows us to investigate how the same number and type of network slices are allocated on an infrastructure with increasing size (i.e., number of available CPs). Note that the average is computed by considering only the used CPs, i.e., the CPs with a selected configuration. The results show that Min.CPs tend to allocate many SFCs in the DCs, and in most of the cases, never allocate any SFC to MEHs. The Min.Cost, Min.CP-LC and Min.LC distributes the SFC into several DCs and MEHs. The main difference between Min.Cost and both Min.CP-LC and Min.LC is the Min.Cost put less SFC in DC and more SFC in MEH in most cases. The Min.Cost strategy also maintains almost the same number of SFC on the DC and MEH, regardless of the infrastructure size, e.g., different CPs available. To conclude, all problems except Min.CP allocate, in average, from one to two SFCs in each MEH. The Min.Cost has the minimum average number of SFCs in the DCs. The Min.CP buys only DC configurations if possible, as also highlighted in Figure 4b.

The number of SFCs in one CP on Table 12 does not mean that all VNFs from the same SFC are allocated on the same CP. The VNF distribution of one SFC is further explained in Table 13. When allocating the VNFs in one SFC, there are two opposite options. In the first option, the VNFs of one SFC are allocated onto the same CP. In the second option, the VNFs of one SFC are spread into several CPs. A high average number of VNFs in Table 13 with a low average number of SFCs in Table 12 indicates that the first option is predominant. In contrast, a high average number of VNFs with a high average number of SFCs indicates the second option. For the other combinations, e.g., a low average number of VNFs with a low average number of SFC, or a high average number of SFC, it is difficult to explain the trend and classify it as the previous two options. Min.Cost and Min.LC tend to spread the VNFs and SFC among the CPs. Min.Cost and Min.LC have a slight difference in the SFC average but has quite a significant difference in the VNF average. The Min.Cost maintains the same proportion of VNFs on each DC and MEH, whereas Min.LC allocate more VNFs in one DC and fewer VNFs in one MEH. This behaviour shows that Min.LC allocated a significant portion of SFC into one CP, whereas Min.Cost tends to spread the VNFs of an SFC into several CPs.

Table 14 shows the average number of DCs and MEHs used to allocate the VNFs of one SFC, depending on the service type. Note that each service type has a different number of VNFs, e.g., VIDS has five VNFs, and PATM has four VNFs, as shown in Table 7. When the available CPs is 8, all the problems have a similar approach because of the limited alternatives.

However, the differences become more significant for a higher number of CPs. Min.Cost tends to use approximately one DC and one MEH to allocate one VIDS, PATM, or POWM. The ROBT must always be placed in DCs, as a MEH cannot fulfil the required capacity. The result of

Journal Pre-proof

Table 12
Average number of SFCs allocated in a DC and a MEH for each problem

	8 CPs		12 CPs		16 CPs		20 CPs		24 CPs	
	DC	MEH	DC	MEH	DC	MEH	DC	MEH	DC	MEH
Min.Cost	2.50	2.00	1.33	1.8	2.00	2.00	2.00	2.00	2.00	1.67
Min.CP	5.00	3.33	5.00	-	5.00	-	4.67	-	3.67	-
Min.CP-LC	4.00	1.67	2.33	1.50	3.50	1.67	2.00	2.00	2.67	1.00
Min.LC	3.00	1.80	2.50	1.33	2.00	1.33	2.00	1.00	2.50	1.83

Table 13
Average number of VNFs allocated in a DC and a MEH for each problem

	8 CPs		12 CPs		16 CPs		20 CPs		24 CPs	
	DC	MEH	DC	MEH	DC	MEH	DC	MEH	DC	MEH
Min.Cost	3.50	3.17	3.33	3.20	4.25	4.50	4.75	3.50	4.25	3.00
Min.CP	8.00	3.33	8.67	-	8.67	-	8.67	-	8.67	-
Min.CP-LC	9.00	2.67	6.33	3.50	8.00	3.33	6.67	3.00	8.00	2.00
Min.LC	6.00	2.80	5.50	2.50	4.75	2.33	4.67	1.50	5.50	2.50

Min.Cost shows that when we have more CPs, a network slice is less shared among CPs. In contrast, the Min.CP and Min.CP-LC prefer to use 1 or 2 DCs to allocate one slice. The Min.LC prefers to use fewer DCs but uses more MEHs to allocate one slice. The less sharing solution used by Min.Cost also helps to decrease the chance of a Single Point of Failure (SPOF). If we share SFC on several CPs (one CP serves many VNFs from a high number of SFC), one CP failure damages all the services. However, if the SFC is less shared, one CP failure may damage only one service instead of all the services.

6. Conclusions

In the paper, we have described and formulated a problem that performs a joint NFV-based slice allocation and selection of CP and LC configurations. The problem jointly considers computing and networking resources and minimizes the cost of the SB to buy resources from the InPs, while the requirements of the network slices requested by the STs are satisfied. Our formulated problem (Min.Cost) is evaluated and compared with three reference problems, i.e., Min.CP, Min.CP-LC and Min.LC. All the problems have been optimally solved by using CPLEX. In the evaluation, four scenarios have been considered by varying the following parameters: (I) number of network slice requests, (II) number of available CPs, (III) URLLC ratio, and (IV) MEH cluster size. The results have pointed out the following significant conclusions:

- *Reference solutions lead to a cost that is at least* $\sim$ *30% higher than Min.Cost.* This result highlights the importance of directly optimizing the costs and selecting the CP and LC configurations.
- *Min.Cost leads to a steady cost in Scenarios II, III, and IV.* Min.Cost leads to solutions with a cost that stays the same while varying the number of available CPs, the URLLC ratio, and the MEH cluster size. The reference problems lead instead to solutions that have relevant cost fluctuations.

Table 14
Average number of CPs used to allocate an SFC for each service type

8 CPs: 2 DCs, 6 MEHs								
Service	Min.Cost		Min.CP		Min.CP-LC		Min.LC	
	DC	MEH	DC	MEH	DC	MEH	DC	MEH
VIDS	1.00	2.50	2.00	2.50	1.00	1.50	1.00	2.00
PATM	1.00	2.50	1.50	2.00	1.00	1.00	1.00	2.00
POWM	1.00	1.00	1.00	1.5	1.00	1.00	1.00	1.00
ROBT	1.00	-	1.20	-	1.00	-	1.00	-
24 CPs: 6 DCs, 18 MEHs								
Service	Min.Cost		Min.CP		Min.CP-LC		Min.LC	
	DC	MEH	DC	MEH	DC	MEH	DC	MEH
VIDS	1.50	1.00	2.00	-	1.50	-	1.00	2.50
PATM	1.00	1.00	1.00	-	1.00	-	1.00	2.00
POWM	1.00	1.00	1.00	-	1.00	1.00	-	1.00
ROBT	1.00	-	1.50	-	1.00	-	1.00	-

Journal Pre-proof

- *Min.CP paradox, using the least CPs is the most expensive solution.* Even if the total cost is mainly due to the cost of buying the CP configurations, minimizing the number of used CPs (Min.CP) leads to the most expensive solutions.

- *More distributing VNFs throughout CPs and less CP sharing lead to cheaper solutions.* Based on the SFC allocation analysis, Min.Cost tends to distribute the VNFs among different CPs, and in a CP the allocated VNFs belong to fewer SFCs.

References

[1] Minimum Requirements Related to Technical Performance for IMT-2020 radio interface(s) - ITU-R M.2410-0 (2017) 11.

[2] ETSI, 5G; management and orchestration; concepts, use cases and requirements (release 15), 2018.

[3] K. Samdanis, X. Costa-Perez, V. Sciancalepore, From network sharing to multi-tenancy: The 5G network slice broker, IEEE Communications Magazine 54 (2016) 32–39.

[4] J. S. Bedo, M. Filippou, A. Gavras, et al, Innovations for New Business Opportunities, 2015. URL: https://5g-ppp.eu/wp-content/uploads/2017/03/5GPPP-brochure-final-web1-with-Author-credits.pdf.

[5] Aspects; management and orchestration; concepts, use cases and requirements (release 17), 2021.

[6] An introduction to network slicing, 2017. URL: https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf.

[7] Amazon web services, amazon ec2 instance, 2022. URL: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html.

[8] Pricing - windows virtual machines: Microsoft azure, 2022. URL: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/#h-series.

[9] AWS for the Edge, 2022. URL: https://aws.amazon.com/edge/.

[10] Mec federation: Deployment considerations, 1st edition, 2022.

[11] 5G-ACIA White Paper: Service-Level Specifications (SLSs) for 5G technology enabled connected industries, 2021.

[12] Deploying Guaranteed-Bandwidth Services with MPLS, 2002. URL: https://www.racf.bnl.gov/Facility/TechnologyMeeting/Archive/06-30-04-CISCO/Guaranteed-Bandwidth-Services-with-MPLS.pdf.

[13] Data Center Infrastructure Resource Guide, Anixter Inc. World Headquarters (2012) 64.

[14] G. Jason, Hyper-V Virtual Switch, 2022. URL: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v-virtual-switch/hyper-v-virtual-switch.

[15] What is a Virtual Switch (vSwitch)?, 2022. URL: https://www.techtarget.com/searchitoperations/definition/virtual-switch.

[16] Amazon EC2 Pricing - Amazon Web Services, 2020. URL: https://aws.amazon.com/ec2/pricing/, library Catalog: aws.amazon.com.

[17] X. Li, J. Rao, H. Zhang, A. Callard, Network slicing with elastic sfc, in: 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), 2017, pp. 1–5. doi:10.1109/VTCFall.2017.8287914.

[18] J.-J. Pedreno-Manresa, P. S. Khodashenas, M. S. Siddiqui, P. Pavon-Marino, Dynamic QoS-QoE assurance in realistic NFV-enabled 5G Access Networks, in: 2017 19th International Conference on Transparent Optical Networks (ICTON), 2017, pp. 1–4. doi:10.1109/ICTON.2017.8025149, iSSN: 2161-2064.

[19] A. Chiha, M. Van der Wee, D. Colle, S. Verbrugge, Network Slicing Cost Allocation Model, Journal of Network and Systems Management 28 (2020) 627–659.

[20] Y. Yue, B. Cheng, X. Liu, M. Wang, B. Li, J. Chen, Resource optimization and delay guarantee virtual network function placement for mapping sfc requests in cloud networks, IEEE Transactions on Network and Service Management 18 (2021) 1508–1523.

[21] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, L. P. Gaspary, 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), IEEE, 2015, pp. 98–106.

[22] A. Fendt, C. Mannweiler, K. Ludwig, L. C. Schmelz, B. Bauer, End-to-end mobile network slice embedding leveraging edge computing, in: NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1–7. doi:10.1109/NOMS47738.2020.9110442.

[23] D. Xiao, S. Chen, W. Ni, J. Zhang, A. Zhang, R. Liu, A sub-action aided deep reinforcement learning framework for latency-sensitive network slicing, Computer Networks 217 (2022) 109279.

[24] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, X. Costa-Perez, How should i slice my network? a multi-service empirical evaluation of resource sharing efficiency, in: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, MobiCom '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 191–206. URL: https://doi.org/10.1145/3241539.3241567. doi:10.1145/3241539.3241567.

[25] B. Ojaghi, F. Adelantado, C. Verikoukis, SO-RAN: Dynamic RAN slicing via joint functional splitting and MEC placement, IEEE Transactions on Vehicular Technology (2022) 1–16.

[26] A. Rago, S. Martiradonna, G. Piro, A. Abrardo, G. Boggia, A tenant-driven slicing enforcement scheme based on pervasive intelligence in the radio access network, Computer Networks 217 (2022) 109285.

[27] H. Feng, Z. Shu, T. Taleb, Y. Wang, Z. Liu, An aggressive migration strategy for service function chaining in the core cloud, IEEE Transactions on Network and Service Management (2022) 1–1.

[28] B. E. Mada, M. Bagaa, T. Tale, H. Flinck, Latency-aware service placement and live migrations in 5g and beyond mobile systems, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6. doi:10.1109/ICC40277.2020.9148940.

[29] D. Harutyunyan, R. Fedrizzi, N. Shahriar, R. Boutaba, R. Riggio, Orchestrating end-to-end slices in 5g networks, in: 2019 15th International Conference on Network and Service Management (CNSM), 2019, pp. 1–9. doi:10.23919/CNSM46954.2019.9012732.

[30] Y. Gong, S. Sun, Y. Wei, M. Song, Deep reinforcement learning for edge computing resource allocation in blockchain network slicing broker framework, in: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), 2021, pp. 1–6. doi:10.1109/VTC2021-Spring51267.2021.9449081.

[31] V. Sciancalepore, L. Zanzi, X. Costa-Pérez, A. Capone, Onets online Network Slice Broker From Theory to Practice, IEEE Transactions on Wireless Communications 21 (2022) 121–134. Conference Name: IEEE Transactions on Wireless Communications.

[32] Z. Shu, T. Taleb, A novel qos framework for network slicing in 5g and beyond networks based on sdn and nfv, IEEE Network 34 (2020) 256–263.

[33] D. Xiao, S. Chen, W. Ni, J. Zhang, A. Zhang, R. Liu, A sub-action aided deep reinforcement learning framework for latency-sensitive network slicing, Computer Networks 217 (2022) 109279.

[34] M. Chowdhury, F. Samuel, R. Boutaba, Polyvine: policy-based virtual network embedding across multiple domains, in: Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures, 2010, pp. 49–56.

[35] W. Yi, W. Muqing, H. Xiaolan, An effective strategy of centralized multi-domain virtual network embedding, in: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), IEEE, 2019, pp. 1186–1191.

[36] Y. Ni, G. Huang, S. Wu, C. Li, P. Zhang, H. Yao, A pso based multi-domain virtual network embedding approach, China Communications 16 (2019) 105–119.

[37] R. A. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, H. Flinck, Optimization model for cross-domain network slices in 5g networks, IEEE Transactions on Mobile Computing 19 (2019) 1156–1169.

[38] I. Kovacevic, A. S. Shafigh, S. Glisic, B. Lorenzo, E. Hossain, Multi-domain network slicing with latency equalization, IEEE Transactions

Journal Pre-proof

on Network and Service Management 17 (2020) 2182–2196.

[39] Telia Company, Telia subscriptions with one contract, https://business.teliacompany.com/global-solutions/mobility/telia-subscriptions-with-one-contract, 2023. Accessed on March 20, 2023.

[40] A. Gohar, G. Nencioni, Minimizing the cost of 5G network slice broker, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021, pp. 1–6. doi:10.1109/INFOCOMWKSHPS51825.2021.9484590.

[41] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, A. Pattavina, Virtual network function placement for resilient service chain provisioning, in: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), 2016, pp. 245–252. doi:10.1109/RNDM.2016.7608294.

[42] I. Documentation, Features of the mip optimizer for miqcp, ???? URL: https://www.ibm.com/docs/en/icos/12.9.0?topic=constraints-features-mip-optimizer-miqcp.

[43] ETSI, Multi-access Edge Computing (MEC); Framework and Reference Architecture, 2022.

[44] Servers | Rack, Tower, Edge amp; Data Center Servers | Lenovo US — lenovo.com, https://www.lenovo.com/us/en/servers-storage/servers/, ???? [Accessed 18-Jul-2023].

[45] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, M. Roughan, The internet topology zoo, IEEE Journal on Selected Areas in Communications 29 (2011) 1765–1775.

[46] Clare, Top 41 Leased Line Providers: 2020 Price Comparison – BusinessFibre.co.uk, ???? URL: https://businessfibre.co.uk/leased-lines/, library Catalog: businessfibre.co.uk.

[47] ATIS, IOT Categorization - Exploring the Need for Standardizing Additional Network Slices, White Paper ATIS-I-0000075, ATIS, 2019. URL: https://api.govwhitepapers.com/wp-content/uploads/2020/06/ATIS-I-0000075.pdf.

[48] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, W. Tan, Reconfiguration in network slicing—optimizing the profit and performance, IEEE Transactions on Network and Service Management 16 (2019) 591–605.

[49] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, A. Banchs, Mobile traffic forecasting for maximizing 5g network slicing resource utilization, in: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9. doi:10.1109/INFOCOM.2017.8057230.

[50] J. Zhou, W. Zhao, S. Chen, Dynamic network slice scaling assisted by prediction in 5g network, IEEE Access 8 (2020) 133700–133712.

Annisa Sarah is Ph.D. student with the University of Stavanger, Norway, from 2021. She received her M.Sc. degree in Wireless Systems from KTH Royal Institute of Technology, Sweden (2017), and her B.Sc degree in Telecommunication Engineering from Telkom University, Indonesia (2014). Previously, Annisa worked as an assistant professor at Atma Jaya Catholic University of Indonesia (2018-2021) for electrical engineering program, responsible to teach and research in telecommunications. Her current research activity regards resource allocation in 5G-Multi-Access Edge Computing ecosystem. Her past works were mainly for wireless network, network optimization, and rural network.

Gianfranco Nencioni is Associate Professor with the University of Stavanger, Norway, from 2018. He is received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in information engineering from the University of Pisa, Italy, in 2008 and 2012, respectively. In 2011, he was a visiting Ph.D. student with the Computer Laboratory, University of Cambridge, U.K. He was a Post-Doctoral Fellow with the University of Pisa from 2012 to 2015 and the Norwegian University of Science and Technology, Norway, from 2015 to 2018. He is currently the head of the Computer Networks (ComNet) research group and leader of the 5G-MODaNeI project funded by the Norwegian Research Council. His research activity regards modelling and optimization in emerging networking technologies (e.g., SDN, NFV, 5G, Network Slicing, Multi-access Edge Computing). His past research activity has been focused on energy-aware routing and design in both wired and wireless networks and on dependability of SDN and NFV.

Journal Pre-proof

Journal Pre-proof

CRediT author statement

Annisa Sarah: Conceptualization, Methodology, Software, Writing – Original Draft . **Gianfranco Nencioni**: Conceptualization, Writing – Review & Editing, Supervision

Journal Pre-proof

Journal Pre-proof

Declaration of Interest Statement

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Annisa Sarah reports financial support was provided by Research Council of Norway. Gianfranco Nencioni reports a relationship with Research Council of Norway that includes: funding grants.

Journal Pre-proof