Optimal Deployment Framework for Multi-Cloud Virtualized Radio Access Networks

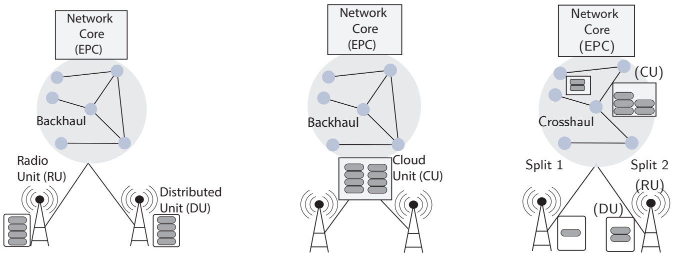
Fahri Wisnu Murti*, Jose A. Ayala-Romero*, Andres Garcia-Saavedra†, Xavier Costa-Pérez †‡, George Iosifidis*
*School of Computer Science and Statistics, Trinity College Dublin
†NEC Laboratories Europe, Heidelberg, Germany
‡ i2CAT Foundation and ICREA, Barcelona, Spain

***Abstract*—Virtualized radio access networks (vRAN) are emerging as a key component of wireless cellular networks, and it is therefore imperative to optimize their architecture. vRANs are decentralized systems where the Base Station (BS) functions can be split between the edge Distributed Units (DUs) and Cloud computing Units (CUs); hence they have many degrees of design freedom. We propose a framework for optimizing the number and location of CUs, the function split for each BS, and the association and routing for each DU-CU pair. We combine a linearization technique with a cutting-planes method to expedite the *exact* problem solution. The goal is to minimize the network costs and balance them with the criterion of centralization, i.e., the number of functions placed at CUs. Using data-driven simulations we find that multi-CU vRANs achieve cost savings up to 28% and improve centralization by 77%, compared to single-CU vRANs. Interestingly, we see non-trivial trade-offs among centralization and cost, which can be aligned or conflicting based on the traffic and network parameters. Our work sheds light on the vRAN design problem from a new angle, highlights the importance of deploying multiple CUs, and offers a rigorous optimization tool for balancing costs and performance.**

I. Introduction

A. Background

The Cloud Radio Access Network (C-RAN) has emerged as a promising solution for building low-cost high-performance RANs in next generation wireless networks. This idea has been motivated by the need to increase the densification of cellular networks, and is enabled by the *softwarization* of these systems [2]. Namely, today the rigid single-unit Base Stations (BS) are gradually replaced with softwarized implementations that can run even in commodity servers in different locations. In this context, the BS operation is essentially a chain of functions, each one performing specific tasks after the wireless radio signal is sampled.[1] This technological advancement has allowed pooling the BS functions in cloud servers within the RAN, known also as Cloud Units (CUs), aiming to reduce their execution cost (using more efficient servers) but also to improve the network performance through its centralized control. For instance, the co-location of functions of different BSs enables Coordinated Multi-Point transmissions, supports dynamic multi-cell spectrum management and also cross-cell interference control [3], [4].



(a) D-RAN (b) C-RAN (c) vRAN

Fig. 1: **RAN Architectures**. **(a)** D-RAN places all functions at the BSs and transfers only the payload to EPC; **(b)** C-RAN places all functions at the cloud units; **(c)** vRAN selects possibly a different split for each BS, placing functions both at the DUs and CUs.

The last few years there have been systematic efforts towards standardizing, and hence adopting, C-RANs [5], [6]. However, several technical issues still need to be carefully addressed before we can reap the benefits of these systems. One of the main open challenges is how to actually design the architecture of C-RANs. This is a difficult problem for many reasons. First, fully centralized RANs are typically not implementable [7] as they require high-capacity fronthaul networks that are not available in RANs, and are immensely expensive to deploy from scratch. This has motivated a shift from pure C-RAN solutions to flexible architectures where only a subset of the BS functions are placed at CUs. Indeed, it is technically possible to support different *functional splits*, by selecting which functions will be hosted at CUs and which will be kept at the distributed units[2] (DUs). The term *virtualized RAN* (vRAN) has been coined to describe these architectures which, in their most flexible version, allow a different split for each BS [8]. A schematic example of the different RAN architectures is presented in Fig. 1.

However, selecting the level of centralization (how many functions to place at CUs) for each BS is an intricate problem. Each split creates different computation load for the CUs and DUs, and also different data flows to transfer between these two ends. In some cases, one needs to deploy multiple CUs to increase the centralized functions. This comes at a cost

Part of this work appears in the proceedings of IEEE ICC 2020 [1]. This work was supported by Science Foundation Ireland (SFI) under Grant No. 17/CDA/4760. This work has been partially supported by EC H2020 5GPPP 5Growth project (Grant 856709). This publication has emanated from research supported in part by a research grant from SFI and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

[1] Two such platforms are OpenAirInterface (https://www.openairinterface.org/) and srsLTE (https://www.srslte.com/).

[2] Clarifying the terminology: in vRANs each BS consists of the CU, DU, and the Radio Units (comprising the antennas and A-D converters); and the DUs are typically co-located with the RUs at the network edge, see Fig. 1.

and thus it is necessary to carefully decide the number and deployment of CUs, and which DUs each of them will serve. However, the CU-DU *assignment* decisions are affected by the functional split of each BS. While initially the vRAN fronthaul was designed using point-to-point connections, the new Xhaul architecture consists of heterogeneous packet-switched links that share multiple flows [9]. This approach is cost-effective, yet compounds the vRAN design problem as one has, additionally, to decide the *routing* paths.

Our goal here is to tackle the vRAN design problem in this general form by optimizing jointly the number and location of CUs, the assignment of DUs to CUs, the functional split for each BS and the routing. These decisions are being made towards minimizing the operating expenditures of the network, subject to available computing and network resources. Going a step further, we examine if, and when, a trade off arises between minimizing costs and maximizing the function centralization, and how one can identify and achieve a sweet spot among these criteria.

B. Methodology & Contributions

We formulate a mathematical program that optimizes the vRAN design by using a measurement-based 3GPP-compliant model. The objective is to minimize the vRAN function execution and routing costs. Our framework is general and can be tailored to different networks and cost functions. For instance, we can use load-dependent computing or routing costs. Besides, we extend it to consider the important criterion of centralization, i.e., the number of functions placed at CUs. Using scalarization, we combine the two criteria and optimize them jointly, where a weight parameter $\eta \in [0, 1]$ determines their relative importance. This approach allows operators to design their networks giving priority either to ensuring low-cost implementation or high performance. As we will see, these criteria are not always aligned, but there is room for jointly orchestrating them.

The resulting formulation is a Mixed-Integer Quadratic Linear Problem (MIQLP), with prohibitive complexity in large networks. Our first finding is that the multi-cloud vRAN design problem is NP-hard to solve optimally, and, moreover, cannot be approximated within any constant factor. We prove this result through a polynomial-time reduction from the unsplittable hard-capacitated facility location problem [10]. In light of this result, we propose a novel two-stage solution process that makes no compromises in terms of optimality. First, we transform our problem to a Mixed-Integer Linear Problem (MILP) by linearizing its constraints [11], and then employ a cutting-planes method based on the seminal and well-known Benders' decomposition technique [12]. We prove that this approach delivers an exact solution; and, interestingly, maintains a feasible solution during its execution, which allows to terminate it earlier at the expense of optimality.

We evaluate our framework using measurements and a range of networks, including actual operational RANs. Following a detailed analysis we find that the benefits when departing from single-CU deployments can be as high as 24% when using 8 CUs, or 15% for 4 CUs, but these gains diminish as we add more CUs. This reveals a threshold effect, and the need to optimize the placement in order to avoid excessive deployment costs. We find that a multi-Cloud vRAN can support higher centralization (up to 77%) even for high routing costs, compared to single-CU vRANs. And we observe that it is preferable (up to 26% cost savings) to have multiple CUs with small computing capacity than few CUs with larger capacity. Our analysis shows that these findings are qualitatively persistent across different networks.

Furthermore, we launch a battery of tests to explore the Pareto fronts of the joint cost - centralization problem. We optimize the vRAN design for different η values (tuning parameter) and network architectures. We find that these two criteria are not always aligned; and even when they are, a careful selection of η is needed to avoid unnecessary losses to either of these dimensions. In certain cases, we can find sweet spots and, e.g., increase the centralization by 320% by accepting a small cost increase of 15%. Yet, in other scenarios the trade offs are less asymmetric and the operator needs to make hard choices between costs or performance; or invest in CU deployment. Indeed, we find that adding more CUs can change the Pareto front, not only expanding it (as one would expect) but also reshaping it and hence creating opportunities for joint optimizations. Our contributions can be summarized:

- We study the problem of multi-cloud vRAN design, considering multiple CUs, CU-DU assignments, functional split and routing. Our model takes into account some of the key practical dimensions of this network design problem; and considers jointly the criteria of centralization and cost, a relation that was hitherto overlooked.
- The complexity of the formulated optimization problems is characterized, and a rigorous solution framework is developed. Combining linearization, decomposition, and an iterative cutting plane method, we propose an algorithm that returns an exact solution, but can be also terminated in a sub-optimal point if execution time is of interest.
- In a series of evaluations using 3GPP-compliant parameters and real and synthetic RAN topologies we investigate: how the availability of CUs affect the vRAN cost; the impact of routing cost and traffic load; robustness of findings on vRAN topology; and we characterize the Pareto front of cost and centralization. Our findings reveal fresh opportunities for optimizing vRANs in terms of either criterion.

Paper Organization. In Section II we position our contributions with respect to prior work. Section III presents the technical background of the problem, e.g., 3GPP standards, and introduces the system model. We formulate the minimum cost vRAN design problem and study its properties in Section IV; while Section V develops the algorithmic framework for solving the problem. In Section VI we formulated the extended problem which balances costs and centralization, and discuss its properties and solution methods. A trace-driven numerical evaluation is presented in Section VII for a variety of networks and datasets, and we conclude in Section VIII.

II. Related Work

The idea of C-RAN was followed by the suggestion for implementing the BS functions in common hardware (cloudification) [13], where different functional splits are possible

	Split Point	UL BW (Mbps)	Delay (ms)	Description
S0	None	λ	30	D-RAN: all functions placed at the DU
S1	PDCP / RLC	λ	30	RRC, PDCP, and upper layer functions are deployed at CU; RLC, MAC, and PHY functions at DU
S2	MAC / PHY	$1.02\lambda + 1.5$	2	MAC and upper layer functions at CU; PHY and RF functions at the DU
S3	PHY / RF	2500	0.25	All functions at CU except RF layers

TABLE I: Data and delay requirements of our splits, when the traffic load is λ Mbps.

[14]. A detailed study of the split specifications can be found in [15], while [16] and [17] analyzed the split requirements and performance gains. However, only few works optimize the split selection, cf. [18]. The authors in [19] select splits to minimize inter-cell interference; [20] and [21] consider adaptive splits; [22] optimizes the splits for RAN slicing; and our previous work [7] optimized jointly the routing and splits. This is very important as vRANs use a shared packet-based network instead of dedicated CPRI links, and we refer the reader to our pertinent feasibility study [9].

The above works do not consider multiple CUs, nor the need to determine their location; yet, this is a key issue in vRAN design. [23] explores min-cost splits in tree networks with fixed CUs; while [24] selects the CU locations and formulates (but does not solve) a min-cost design problem. [25] and [26] consider multiple CUs but do not optimize routing. In [27] and [28] the DUs are assigned to co-located CUs aiming to reduce energy costs, thus the assignment decisions do not affect routing. Finally, our previous work considered fixed CUs [29]. Here, we decide the CU deployment by selecting their number and location, along with routing and assignments. Besides, we explore the trade-off between costs and centralization, where the latter is a good proxy for vRAN performance, see Table III. Interestingly, we find that these criteria are not always aligned, and explore how we can jointly orchestrate them.

Modeling-wise, the problem is reminiscent to server placement [30], Virtual Network Functions (VNF) chain embedding [31], network slicing [32], and wireless edge computing problems [33], [34]. However, there are also significant differences with those problems. Namely, the BS operation can be modeled as a varying-size chain of functions which can be deployed in different ways (splits). And each split has different bandwidth and delay requirements, which in turn affect which routing paths are eligible for each split. Moreover, the VRAN functions differ in their computing and memory requirements. Finally, each split brings different performance gains and capabilities to the network. On top of these distinct aspects of the VRAN splitting problem, we decide the number of CUs, which brings our formulation closer to *discrete location problems*. In fact, we prove that this is harder than the *unsplittable hard-capacitated facility location* problem [10], and hence does not admit any constant-approximation algorithm unless we violate the capacities. Besides, the existence of shared routing paths adds another layer of complexity and makes it a joint facility location-network design problem [35], [36]. In light of these observations, we opt to follow a different solution approach.

As a network design problem, suboptimal vRAN solutions have immense long-term cost impact. Hence, we avoid heuristic or approximation algorithms and employ the seminal technique of Benders' decomposition [12] that returns the optimal solution. This well-known method is extensively used in operations research [37], and has been recently employed in communication networks to design sensor networks [38], transmission power control algorithms [39], and routing in optical networks [40], among others. To the best of our knowledge, this is the first work using Benders' technique for RAN design, and to that end we formulated carefully the problem so as to avoid non-linearities and render it amenable to this decomposition. Finally, it is worth mentioning that machine-learning approaches, e.g., see [33], [34] and references therein, can be useful for this problem when there are unknown parameters, albeit they do not offer, in general, optimality guarantees.

III. System Model

A. Background and Preliminaries

We rely on 3GPP terminology [6] and the seminal white paper [15] for our model. A key block of each base station is the Base Band Unit (BBU) that hosts all functions except the radio signal reception/transmission which is implemented by the Radio Unit (RU). With the advent of C-RAN, the BBU functions are split into the Centralized (or, Cloud) Unit (CU) and the Distributed Unit (DU). Hence, while in previous generations of cellular networks a BS was composed of a RU and BBU, in vRANs a BS comprises the CU, DU, and RU, Fig. 1. Both CU and DU are computing equipment, but the CUs are typically bigger servers while the DUs are smaller units placed close to the RUs. The CUs and DUs host different functions based on the selected split, cf. [14], [20], where the prevalent splits are shown in Table I; see also [6]. Going from S0 to S3, more functions are placed at CUs. This increases the cost savings as, naturally, the CU servers are more efficient than the computing units of DUs[3]. Moreover, centralization increases network performance. For example, split S1 allows some L3 and L2 functionalities to be implemented in the same hardware; split S2 supports CoMP and effective MIMO implementation; and split S3 offers opportunities for power-saving, increases computation efficiency and enhances CoMP reception by combining uplink PHY levels.

On the other hand, centralizing functions increases the data load that needs to be transferred to CUs. For example, the load increases from λ Mbps (payload) in S1 to 2.5 Gbps in S3 for the considered configuration[4], see Table I. Similarly, the delay

[3] According to [16], a cloud-radio access network require approximately from 10% to 15% less capital expenditure per square kilometer compared to traditional fully decentralized deployments.

[4] Scenario: 1 user/TTI, 20 MHz bandwidth; Downlink: MCS (modulation and coding scheme) index 28, 2×2 MIMO, 100 Resource Blocks (RBs), 2 transport blocks of 75376 bits/subframe; Uplink: MCS 23, 1×2 SIMO, 96 RBs, 1 transport block of 48936 bits/subframe.

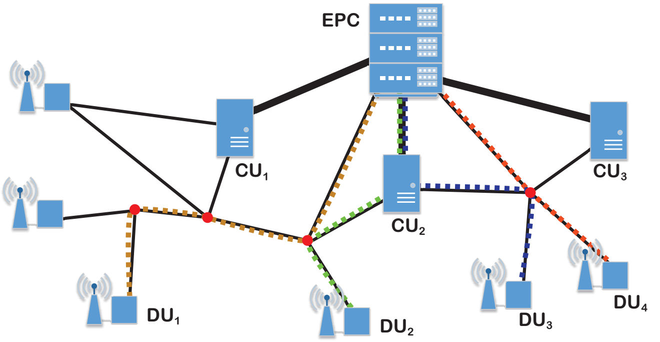


Fig. 2: The BS functions can be split between the DU and one of the CUs. DU$_1$ implements a full-stack BS (no split) and route its traffic directly to EPC. DU$_2$ selects a split and sends its traffic to CU$_2$ which further routes it to EPC. CUs have high-capacity links to EPC; and some CUs might not be activated (e.g., CU$_3$).

window within which the data transfers among the CU and DU must be completed changes by orders of magnitude, down to 250 μs for S3. These requirements restrict the paths one can use to connect the DUs and CUs, and might also increase the routing cost, hence offseting the computation cost savings.

Focusing on splits S1, S2 and S3, the BS operation can be modeled as a function chain ($f_0 \to f_1 \to f_2 \to f_3$). Function f_0 encapsulates the RF-related operations (e.g., A/D sampling) and is placed always at DUs, while f_3 is associated with PDCP and upper BS layers and can be placed at DU for D-RAN, or at CU for vRAN. Also, functions f_1 (PHY) and f_2 (RLC and MAC) can be deployed either in CUs or DUs depending on the RAN configuration. The current standards suggest a packet-based network with links that are shared by the DUs, instead of the point-to-point expensive CPRI links [15]. This architecture reduces the routing cost, but compounds the routing policy. Finally, the CUs are assumed to be directly connected to the network core through high-capacity (e.g., optical) links; see Fig. 2 for an example of our system.

B. Model

Network. The RAN is modeled with a graph $G = (\mathcal{I}, \mathcal{E})$ where the set of nodes $\mathcal{I}$ includes the subsets: $\mathcal{N}$ of the $N = |\mathcal{N}|$ DUs, $\mathcal{L}$ of the $L = |\mathcal{L}|$ routers, and $\mathcal{M}$ of the $M = |\mathcal{M}|$ locations for the CUs; and the core node (EPC) that we index with 0. We define $\mathcal{M}_0 = \mathcal{M} \cup \{0\}$ and assume that $M << N$. The nodes are connected with a set $\mathcal{E}$ of links and each link (i, j) has average data transfer capacity of c_{ij} (Mbps). DU-n is connected to each CU-m with a set of paths $\mathcal{P}_{nm}$, and to EPC with a set of paths $\mathcal{P}_{n0}$. We define the set $\mathcal{P}_m = \cup_{n=1}^{N} \mathcal{P}_{nm}$ of paths connecting all DUs to CU-m; and denote the set of all paths with $\mathcal{P} = \cup_{m \in \mathcal{M}_0} \mathcal{P}_m$. Each path p_k introduces end-to-end delay of d_{p_k} (secs). The BS functions are implemented in servers using virtual machines (VMs). We denote with H_n and H_m the processing capacity (cycles/s) of DU-n and CU-m, respectively, where naturally we assume that $H_m > H_n$; and define as ρ_1, ρ_2 and ρ_3 the processing load (cycles/Mbps) per unit of traffic of functions f_1, f_2 and f_3, respectively.

Demand & Cost. We focus on uplink where the users served by each DU-n, $n \in \mathcal{N}$, generate an aggregate data flow of $\lambda_n \geq 0$ (Mbps) . Hence, the RAN needs to admit, route and serve N different flows. The execution of the VRAN functions is considered more cost-effective when they are implemented at CUs, as opposed to DUs [16]. This is due to actors such as the highest computing capacity and improved energy efficiency of the CU servers compared to the smaller, and thus less efficient, DU servers [41]. Our model is general and does not make any strict assumptions on the relative computing efficiencies of CUs and DUs. Following the literature for modeling the processing costs in virtual machines, we denote with a_m and b_m the VM instantiating cost (monetary units) and the computing cost (monetary units/cycle) at CU-m, respectively. The former is fixed and paid for any VM that is installed, such as cooling costs, one-off expenses for leasing VMs from third parties, minimum resources required for starting the VM, and so on. On the other hand, b_m models the operating expenditures which depend on the load, e.g., the energy spent due to processing. We therefore define $\boldsymbol{a} = (a_m, m \in \mathcal{M})$ and $\boldsymbol{b} = (b_m, m \in \mathcal{M})$. The respective costs for the N DUs are $\boldsymbol{\alpha} = (\alpha_n, n \in \mathcal{N})$ and $\boldsymbol{\beta} = (\beta_n, n \in \mathcal{N})$. Finally, ζ_k is the routing cost (monetary units/Mbps) for each path $p_k \in \mathcal{P}$, and we define $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_{|\mathcal{P}|})$. Such costs might arise because the network links are leased from third-parties, or they can model the (average) expenditures of the network, or the amortized investments for maintaining the network links. The flows are routed to CUs and then to EPC through high-capacity links, or directly from the DUs to EPC in case no splitting has been decided (D-RAN implementation). Table II summarizes the model parameters.

Notation	Definition
$\mathcal{N}, N, n$	Set of DUs, number of DUs, DU index
$\mathcal{M}, M, m$	Set of CUs, number of CUs, CU index
$\mathcal{P}_{nm}$ ($\mathcal{P}_m$)	Set of paths between DU n (all DUs) and CU m
c_{ij}	Bandwidth (Mbps) of link between nodes i and j
d_{p_k}	End-to-end delay of the path p_k in seconds
λ_n	Traffic flow from DU-n in Mbps
H_n, H_m	Processing capacity (cycles/s) of DU-n and CU-m
ρ_1, ρ_2, ρ_3	Processing load (cycles) per Mbps of f_1, f_2, f_3
a_m, b_m	Cost of VM instantiation and computing at CU m
α_n, β_n	Cost of instantiation and computing at DU n
ζ_k	Routing cost of path k (monetary units per Mbps)
c_d	Routing cost per kilometer per Mbps

TABLE II: Notation table

C. Problem Statement

The objective is to minimize the operation costs by jointly optimizing the following decisions. *Deployment*: which of the available locations to use for deploying CUs? *Assignment*: which CU should serve each split DU? *Placement*: how to split the functions f_1, f_2, f_3 of each BS? *Routing*: how to route the data from the DUs to CUs? These decisions are inherently coupled, and this raises interesting trade-offs. Placing the functions at DUs reduces routing costs but increases computing costs due to the less-efficient DU servers. On the other hand, centralizing more functions reduces computing costs, but deeper splits restrict the routing options due to their tighter delay and bandwidth needs. Also, each DU-CU assignment affects the number of eligible paths for every other DU since the network links are capacitated; and similarly, the CU computing capacity couples the function placement across different DUs. All these decisions are conditioned on the number and location of the CUs. Clearly, as more CUs become available the RAN design space increases and, in turn, allows more cost-efficient configurations. However, this effect depends on the network topology and load, and deploying CUs comes at a cost that must not exceed the anticipated savings.

IV. Problem Formulation

A. Decision Variables & Constraints

Function Placement. We denote with $x_{1n}, x_{2n} \in \{0,1\}$ the decisions for deploying f_1 and f_2, respectively, to DU-n. Similarly, $y_{1nm}, y_{2nm} \in \{0,1\}$ decide the deployment of these functions at CU-m. We define the function placement vectors for all DUs w.r.t. CU-m as $\boldsymbol{x}_1 = (x_{1n} : n \in \mathcal{N})$, $\boldsymbol{x}_2 = (x_{2n} : n \in \mathcal{N})$, $\boldsymbol{y}_{1m} = (y_{1nm} : n \in \mathcal{N})$, $\boldsymbol{y}_{2m} = (y_{2nm} : n \in \mathcal{N})$. We further define the vectors $\boldsymbol{y}_1 = (\boldsymbol{y}_{1m} : m \in \mathcal{M})$ for f_1, $\boldsymbol{y}_2 = (\boldsymbol{y}_{2m} : m \in \mathcal{M})$ for f_2, and the overall decision vectors $\boldsymbol{x} = (\boldsymbol{x}_1; \boldsymbol{x}_2)$ and $\boldsymbol{y} = (\boldsymbol{y}_1; \boldsymbol{y}_2)$. The function placements are coupled. Namely, f_1 cannot be deployed at a CU unless f_2 is also placed there; while f_2 can be at an DU only if f_1 is already placed there [7], [15], [42]. Hence, we obtain the following *chaining* constraints:

$$y_{1nm} \le y_{2nm}, \qquad x_{2n} \le x_{1n}, \qquad \forall n \in \mathcal{N}, m \in \mathcal{M}. \quad (1)$$

Also, duplicate deployments of each function should be prevented in order to enforce the consistent operation of every BS chain, i.e.,

$$x_{1n} + \sum_{m \in \mathcal{M}} y_{1nm} = 1, \;\; x_{2n} + \sum_{m \in \mathcal{M}} y_{2nm} = 1, \; \forall n \in \mathcal{N}. \quad (2)$$

CU Deployment. We assign DU-n to CU-m using the binary variable z_{nm}, where $z_{nm} = 1$ if at least one function of BS-n is deployed at that CU. The assignment matrix is $\boldsymbol{z} = (z_{nm} \in \{0,1\} : \forall n \in \mathcal{N}, m \in \mathcal{M})$. We model the configuration where f_1 and f_2 are at the DU-n, but f_3 at the CU-m, by setting $y_{1m} = y_{2m} = 1 - z_{nm} = 0$, hence we do not define explicit variables for f_3. Each DU can be assigned at most to one CU:

$$\sum_{m \in \mathcal{M}} z_{nm} \le 1, \forall n \in \mathcal{N}, \quad (3)$$

$$\text{and} \qquad z_{nm} \ge y_{2nm}, \;\; \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (4)$$

must hold to preserve the BS chain. Finally, the function placements and the assignment decisions need to ensure that the computing capacity at each location is not exceeded, hence:

$$\lambda_n \Big(x_{1n}\rho_1 + x_{2n}\rho_2 + (1 - \sum_{m \in \mathcal{M}} z_{nm})\rho_3 \Big) \le H_n, \;\; \forall n \in \mathcal{N}, \quad (5)$$

$$\sum_{n \in \mathcal{N}} \lambda_n (y_{1nm}\rho_1 + y_{2nm}\rho_2 + z_{nm}\rho_3) \le H_m, \;\; \forall m \in \mathcal{M}. \quad (6)$$

Data Routing. Variable r^k_{nm} (Mbps) decides the amount of DU-n traffic that will be routed over path $p_k \in \mathcal{P}_{nm}$ to CU-m, and we define $\boldsymbol{r} = \big(r^k_{nm} \in \mathbb{R}_+ : \; \forall p_k \in \mathcal{P}_{nm}, n \in \mathcal{N}, m \in \mathcal{M}_0 \big)$. The routing decisions must respect the link capacities:

$$\sum_{n \in \mathcal{N}} \sum_{p_k \in \mathcal{P}_n} r^k_{nm} I^k_{ij} \le c_{ij}, \quad \forall (i,j) \in \mathcal{E} \quad (7)$$

where $I^k_{ij} = 1$ if link (i,j) is included in path p_k and $I^k_{ij} = 0$ otherwise; and also have to satisfy the flow requirements of the splits:

$$\sum_{p_k \in \mathcal{P}_{nm}} r^k_{nm} = z_{nm} S_n(x_{1n}, x_{2n}), \;\; \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (8)$$

where $S_n(x_{1n}, x_{2n})$ is the data flow (Mbps) from DU-n, which is determined by the load and the selected split of BS-n, as summarized in Table I. In particular, we can use the following succinct formula for expressing each split's flow, $S_n(x_{1n}, x_{2n}) =$

$$x_{1n}(1.02\lambda_n + 1.5) - x_{2n}(0.02\lambda_n + 1.5) + 2500(1 - x_{1n}). \quad (9)$$

Equations (8) ensure there is no data flow from DU-n to CU-m unless the BS-m function f_3 is placed at that CU, namely n is assigned to m. This captures nicely the interaction between assignment and routing, but creates a quadratic constraint term as one can see by replacing the expression for $S_n(x_{1n}, x_{2n})$. Finally, in case of fully decentralized BSs (D-RAN, f_3 functions placed at the DUs), the flow needs to be routed directly to EPC, and hence it should hold:

$$\sum_{p_k \in \mathcal{P}_{n0}} r^k_{n0} = \big(1 - \sum_{m} z_{nm}\big) \lambda_n, \;\; \forall n \in \mathcal{N}. \quad (10)$$

We note that our model can be readily extended to include routing decisions from the CUs to the EPC, which is required if they are not connected with direct links as it is assumed.

Delay. The routing has to satisfy the delay requirements of the selected split [15]. To enforce this, we classify the paths into the following sets: $\mathcal{P}^A_{nm} \subseteq \mathcal{P}_{nm}$ with delay larger than 30 ms, $\mathcal{P}^B_{nm} \subseteq \mathcal{P}_{nm}$ with delay larger than 2 ms, and the set of paths $\mathcal{P}^C_{nm} \subseteq \mathcal{P}_{nm}$ with delay larger than 0.25 ms. There is also a set of paths with delay less than 0.25 ms for which we do not need to introduce notation. Clearly, it holds $\mathcal{P}^A_{nm} \subseteq \mathcal{P}^B_{nm} \subseteq \mathcal{P}^C_{nm}$, e.g., a path in $\mathcal{P}^B_{nm}$ (with delay > 2 ms) belongs also to set $\mathcal{P}^A_{nm}$ (its delay is also > 0.25 ms). Observe that this classification follows the delay thresholds of splits [15], shown in Table I. Using these sets, we can ensure that only paths that are eligible for each split are selected, by using the following constraints:

$$\sum_{p_k \in \mathcal{P}^A_{nm}} r^k_{nm} \le T(y_{1nm} + y_{2nm}), \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (11)$$

$$\sum_{p_k \in \mathcal{P}^B_{nm}} r^k_{nm} \le T(1 - y_{1nm} + y_{2nm}), \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (12)$$

$$\sum_{p_k \in \mathcal{P}^C_{nm}} r^k_{nm} \le T(2 - y_{1nm} - y_{2nm}), \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (13)$$

where $T >> 0$ is a big-M type of constraint. Indeed, whenever we select a split (a decision made by variables y_{1nm} and y_{2nm}) then we can only use the paths with delays smaller than the thresholds. We achieve this by enforcing the routing variables to be zero for the paths that do not satisfy this condition.[5]

B. Minimum Cost vRAN Design

The computation cost for each DU-n depends on which functions it implements and what is the load λ_n that it needs to serve. Following [43], we define:

$$V_n(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{z_n}) = \alpha_n \Big(x_{1n} + x_{2n} + \big(1 - \sum_{m \in \mathcal{M}} z_{nm}\big)\Big) \\ + \beta_n \lambda_n \Big(\rho_1 x_{1n} + \rho_2 x_{2n} + \big(1 - \sum_{m \in \mathcal{M}} z_{nm}\big)\rho_3 \Big), \quad (14)$$

where $\boldsymbol{z_n} = (z_{nm}, m \in \mathcal{M})$. The first term is a base offset depending on the deployment platform and the type of instantiation, and the second term is linear with the DU traffic. We can consider other cost in our framework, e.g., power consumption, which can be also modeled linearly following [27] and [44]. Similarly, the respective cost for CU-m is:

$$V_m(\boldsymbol{y}_{1m}, \boldsymbol{y}_{2m}, \boldsymbol{z}_m) = b_m \sum_{n \in \mathcal{N}} \lambda_n \Big(\rho_1 y_{1nm} + \rho_2 y_{2nm} + \rho_3 z_{nm} \Big) \\ + a_m \sum_{n \in \mathcal{N}} \Big(y_{1nm} + y_{2nm} + z_{nm} \Big) + \omega_m \sum_{n \in \mathcal{N}} z_{nm} \lambda_n, \quad (15)$$

where $\boldsymbol{z_m} = (z_{nm}, n \in \mathcal{N})$, and ω_m is the cost to use the CU and route data from there to EPC; and $\sum_n z_{nm} = 0$ when CU-m is not used. The cost of routing data from all DUs to CU-m is $U_m(\boldsymbol{r}_m) = \sum_{n \in \mathcal{N}} \sum_{p_k \in \mathcal{P}_{nm}} \zeta_k r_{nm}^k$. Putting the above together, we define the minimum-cost vRAN design problem:

$$\begin{aligned} \mathbb{P}: \min_{\boldsymbol{r}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}} \quad & \sum_{n \in N} V_n(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{z_n}) + \sum_{m \in M} V_m(\boldsymbol{y}_{1m}, \boldsymbol{y}_{2m}, \boldsymbol{z}_m) \\ & + \sum_{m \in \mathcal{M}_0} U_m(\boldsymbol{r}_m) \\ \text{s.t.} \quad & (1) - (13). \end{aligned}$$

This problem outputs the assignments $\boldsymbol{z}$, the function placements $\boldsymbol{x}$ and $\boldsymbol{y}$, and the routing variables $\boldsymbol{r}$ for each network graph $G = (\mathcal{I}, \mathcal{E})$ and load $\boldsymbol{\lambda}$. Moreover, as it will become clear in the sequel, one can use different computing and routing cost functions, as long as they remain within the family of convex functions[6]. We analyze the complexity of $\mathbb{P}$ next.

C. Complexity Analysis

We characterize the complexity of $\mathbb{P}$ through a polynomial time reduction from the *unsplittable hard-capacitated facility location* (UHCFL) problem.

UHCFLP. We are given a weighted graph $G = (\mathcal{V}, \mathcal{E})$ with a set of vertices $\mathcal{V}$ and a set of links $\mathcal{E}$. Set $\mathcal{V}$ is partitioned further in two possibly overlapping sets $\mathcal{J} \subseteq \mathcal{V}$ of facilities and $\mathcal{I} \subseteq \mathcal{V}$ of clients. Each client $i \in \mathcal{I}$ has demand d_i; and each facility j has activation cost v_j and can serve demand of u_j units (*hard capacitated*). Serving client i by facility j induces cost c_{ij} per unit of demand, and naturally $c_{ii} = 0, \forall i \in \mathcal{V}$. The costs satisfy the triangle inequality[7]; and each user is served by a single facility (*unsplittable*). Our goal is to activate a subset of facilities $\mathcal{J}^* \subseteq \mathcal{J}$ and assign clients to them using a rule $\pi : \mathcal{I} \to \mathcal{J}^*$, so as to minimize the aggregate cost $\sum_{j \in \mathcal{J}^*} v_j + \sum_{i \in \mathcal{I}} d_i c_{i\pi(i)}$ while not exceeding their capacity, i.e., $\sum_{\pi(i)=j} d_i \leq u_j, \forall j \in \mathcal{J}^*$.

The UHCFL problem is NP-hard to approximate within any constant factor, unless the facility capacities are violated by at least a factor of 3/2 [10]. Hence, the literature has been focusing on bi-criteria approximation algorithms. We prove next that our problem is harder than UHCFL.

Theorem 1. *UHCFL problem can be reduced in polynomial time to $\mathbb{P}$, i.e., $UHCFL \leq_P \mathbb{P}$.*

Proof. We consider the decision versions of the problems which answer if a solution yields higher cost than a threshold. We assume there is a unit-time oracle for the following instance of our problem, which we denote $\widehat{\mathbb{P}}$: we can select only the split[8] S1; there are $\mathcal{N}$ DUs with λ_n demand and $H_n = 0, \forall n$ capacity; $\mathcal{M}$ CUs with $H_m, m \in \mathcal{M}$ capacity; each DU n is connected to a subset of CUs, to each one with a non-shared path of capacity $c_p > S_n$ and low delay so that it can support the split. Then, solving any UHCFL instance is equivalent to solving $\widehat{\mathbb{P}}$. To see this, notice that if we can solve $\widehat{\mathbb{P}}$, then we can solve any UHCFL problem $\widehat{\mathbb{U}}$ by setting: $\mathcal{I} = \mathcal{N}$ for the clients, $\mathcal{F} = \mathcal{M}$ for facilities, $c_{ij} = c_p$ for the serving costs of each link $(i, j) \in \mathcal{E}$, $u_j = w_m$ for the activation cost of each facility $j \in \mathcal{J}$ (the routing cost from CU-m to EPC), and $v_j = H_m$ for the server capacities. This way, answering the question if $\widehat{\mathbb{P}}$ exceeds the value Q_1 is equivalent to deciding if the solution of $\widehat{\mathbb{U}}$ exceeds some value Q_2. In other words, we can solve $\widehat{\mathbb{U}}$ by invoking the oracle that solves $\widehat{\mathbb{P}}$. And it is straightforward to see that this reduction is of polynomial time: for every deployed CU we activate the respective facility, and for every assignment of a DU to the CU we assign the respective client to that facility. □

In fact, our problem is much harder since we need to route the traffic over links that are hard-capacitated and shared among the different DUs. Such problems are known also as *facility location-network design* problems, see overview in [35]; or can be explicitly modeled as *multi-level facility location* problems, cf. [36], where each link is modeled as a facility, and the CU servers as the top-level facilities. Besides, in $\mathbb{P}$ there are many different splits from which we need to select exactly one. Concluding, since we have proved that there is no constant approximation algorithm for our problem, we follow a different solution strategy in the next section.

[5] For example, if we select split $S2$ (delay 2 ms) for the DU-CU pair (m, n), we will set $y_{1nm} = 0$ and $y_{2nm} = 1$ and the RHS of (12) will become 0, hence the variables r_{nm}^k for all paths $p_k \in \mathcal{P}_{nm}^B$ (those with delay larger than 2 ms) will also be 0.

[6] For example, in case the computation costs are non-linear increasing functions of the load, we will then have a convex instead of linear objective function. Our framework will still deliver the optimal solution, since the Benders' decomposition technique has been generalized by Geoffrion [45] to non-linear convex problems.

[7] This standard assumption requires $c_{ij} + c_{jk} \geq c_{ik}, \forall i, j, k \in \mathcal{V}$, and is inherent in networks where the link costs represent delays or other spatial-based parameters. If the assumption does not hold, the UHCFLP becomes even more challenging.

[8] Equivalently, one can consider that all splits are possible, but the paths can support only S1. The problem remains the same.

V. Algorithmic Framework

In order to solve this challenging problem we follow a two-stage approach. First, we reformulate $\mathbb{P}$ using a linearization technique that replaces the intricate constraint (8). Then, we decompose the problem and employ a cutting-planes method that expedites the solution and finds, provably, an *exact optimal* point. Since this is an NP-hard problem, we naturally do not to expect any guarantees on the convergence time. However, as we will see in the numerical evaluation, our solution approach is remarkably fast in practice.

A. Linearization of constraints

The product of two integer variables in (8) can be modeled with auxiliary variables $v_{1nm} = x_{1n}z_{nm}$, and $v_{2nm} = x_{2n}z_{nm}, \forall(n,m)$, where the auxiliary variables belong to set:

$$\mathcal{V}=\Big\{\boldsymbol{v_1}:v_{1nm}\in\{0,1\},\ \boldsymbol{v_2}:v_{2nm}\in\{0,1\}, n\in\mathcal{N}, m\in\mathcal{M}\\ \mid v_{1nm}\leq x_{1n}; v_{1nm}\leq z_{nm}; v_{1nm}\geq x_{1n}+z_{nm}-1;\\ v_{2nm}\leq x_{2n}; v_{2nm}\leq z_{nm}; v_{2nm}\geq x_{2n}+z_{nm}-1\Big\}.$$

Using a reformulation similar to [11], we define the new problem:

$$\boxed{\begin{aligned}\mathbb{P}_2: \min_{\substack{\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}\in\mathcal{V},\\ \boldsymbol{y},\boldsymbol{r}\succeq 0}} \quad & J(\boldsymbol{r},\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})\\ \text{s.t.} \quad & (1)-(7),(10)-(13),\\ & \sum_{p_k\in\mathcal{P}_{nm}} r_{nm}^k = S_{nm}(v_{1nm},v_{2nm}), \forall n\in\mathcal{N}, m\in\mathcal{M}\end{aligned}}$$

where $J(\cdot)$ is the objective of $\mathbb{P}$, and we have set $S_{nm}(v_{1nm},v_{2nm})=z_{nm}S_n(x_{1n},x_{2n})$. It is important to stress that problems $\mathbb{P}$ and $\mathbb{P}_2$ are equivalent, meaning that when the optimal solution for $\mathbb{P}_2$, constitutes also an optimal solution for our initial problem $\mathbb{P}$. To see this, it suffices to observe that due to the constraint set $\mathcal{V}$ the newly introduced variables are tightly coupled with the original variables.[9]

B. Applying Benders' Decomposition

We use the Benders' method [12] which decomposes $\mathbb{P}_2$ to a *Master* sub-problem $\mathbb{P}_{2M}$ that optimizes the binary variables for fixed routing; and to a *Slave* program $\mathbb{P}_{2S}$ that optimizes routing for fixed split and assignment decisions:

$$\begin{aligned}\mathbb{P}_{2S}: & \min_{\boldsymbol{r}\succeq\boldsymbol{0}} J(\boldsymbol{r},\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{v}})\\ & \text{s.t.} \quad (7),(10)-(13),\\ & \qquad \sum_{p_k\in\mathcal{P}_{nm}} \bar{r}_{nm}^k = S_{nm}(\bar{v}_{1nm},\bar{v}_{2nm}), \forall n\in\mathcal{N}, m\in\mathcal{M}.\end{aligned}$$

Following the standard practice in Benders decomposition, we use the dual of the Slave problem:

$$\mathbb{P}_{2SD}: \max_{\boldsymbol{\pi}} \ h(\boldsymbol{\pi},\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}},\bar{\boldsymbol{v}}) \quad \text{s.t.} \quad H^\top\boldsymbol{\pi}\preceq\boldsymbol{\zeta}, \tag{16}$$

where $\boldsymbol{\pi}$ is the vector of dual variables and matrix H collects the pertinent coefficients.

Algorithm 1: Benders' Decomposition Algorithm

```
Initialize: $\tau = 0$; $\mathcal{C}_O^{(0)} = \mathcal{C}_F^{(0)} = \emptyset$; $UB^{(0)} =$
        $-LB^{(0)} >> 1$; $\epsilon$
1  repeat
2      Solve $\mathbb{P}_{2M}(\mathcal{C}_O^\tau, \mathcal{C}_F^\tau)$ to get $\{\theta^\tau, \boldsymbol{x}^{\boldsymbol{\tau}}, \boldsymbol{y}^{\boldsymbol{\tau}}, \boldsymbol{z}^{\boldsymbol{\tau}}, \boldsymbol{v}^{\boldsymbol{\tau}}\}$
3      Set $LB^\tau =$
         $\sum_n V_n(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{\tau}}, \boldsymbol{x}_{\boldsymbol{2}}^{\boldsymbol{\tau}}, \boldsymbol{z}_n^{\boldsymbol{\tau}}) + \sum_m V_m(\boldsymbol{y}_{1m}^\tau, \boldsymbol{y}_{2m}^\tau, \boldsymbol{z}_m^\tau) + \theta^\tau$
4      Solve $\mathbb{P}_{2SD}(\boldsymbol{x}^{\boldsymbol{\tau}}, \boldsymbol{y}^{\boldsymbol{\tau}}, \boldsymbol{z}^{\boldsymbol{\tau}}, \boldsymbol{v}^{\boldsymbol{\tau}})$ to obtain $\boldsymbol{\pi}^\tau$.
5      if $UB^\tau < UB^{\tau-1}$ then
6          $UB^\tau =$
             $\sum_n V_n(\boldsymbol{x}_1^\tau, \boldsymbol{x}_2^\tau) + \sum_m V_m(\boldsymbol{y}_{1m}^\tau, \boldsymbol{y}_{2m}^\tau, \boldsymbol{z}_m^\tau) +$
             $h(\boldsymbol{\pi}^{\boldsymbol{\tau}}, \boldsymbol{x}^{\boldsymbol{\tau}}, \boldsymbol{y}^{\boldsymbol{\tau}}, \boldsymbol{z}^{\boldsymbol{\tau}}, \boldsymbol{v}^{\boldsymbol{\tau}})$;
7      end
8      if $h(\boldsymbol{\pi}^\tau, \boldsymbol{x}^{\boldsymbol{\tau}}, \boldsymbol{y}^{\boldsymbol{\tau}}, \boldsymbol{z}^{\boldsymbol{\tau}}, \boldsymbol{v}^{\boldsymbol{\tau}}) < \infty$ then
9          $\mathcal{C}_O^{\tau+1} = \mathcal{C}_O^\tau \cup \{\boldsymbol{\pi}^m\}$  % add extreme point;
10     else
11         $\mathcal{C}_F^{\tau+1} = \mathcal{C}_F^\tau \cup \{\boldsymbol{\pi}^m\}$.  % add extreme ray;
12     end
13     $\tau = \tau + 1$.
14 until $\big(UB^{(\tau)} - LB^{(\tau)}\big)/LB^{(\tau)} \leq \epsilon$
15 Set optimal configuration, assignment, deployment:
     $\boldsymbol{x}^* = \boldsymbol{x}^\tau; \boldsymbol{y}^* = \boldsymbol{y}^\tau, \boldsymbol{z}^* = \boldsymbol{z}^\tau$
16 Obtain optimal routing $\boldsymbol{r}^*$ from $\mathbb{P}_{SD}(\boldsymbol{x}^{\boldsymbol{\tau}}, \boldsymbol{y}^{\boldsymbol{\tau}}, \boldsymbol{z}^{\boldsymbol{\tau}}, \boldsymbol{v}^{\boldsymbol{\tau}})$
```

The Master problem $\mathbb{P}_{2M}$ optimizes the discrete decisions and a proxy continuous variable:

$$\begin{aligned}\mathbb{P}_{2M}: \min_{\theta\geq 0,\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}\in\mathcal{V},\boldsymbol{y}} \quad & J(\bar{\boldsymbol{r}},\boldsymbol{x},\boldsymbol{y},\boldsymbol{v})+\theta\\ \text{s.t.} \quad & (1)-(7),\\ & h(\boldsymbol{\pi}^\xi,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{v})\leq\theta, \ \forall\boldsymbol{\pi}^\xi\in\mathcal{C}_O, \qquad (17)\\ & h(\boldsymbol{\pi}^\xi,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{v})\leq 0, \ \forall\boldsymbol{\pi}^\xi\in\mathcal{C}_F, \qquad (18)\end{aligned}$$

where (17)-(18) are the optimality and feasibility cuts, respectively, which gradually construct the entire constraint set of $\mathbb{P}_2$. The intuition behind this method is that the optimal solution can be found before a full re-construction is built.

C. Iterative Solution Algorithm

The details are presented in Algorithm 1 which runs until convergence. Firstly, it solves the Master problem $\mathbb{P}_{2M}$ to find the currently optimal integer decision variables $(\boldsymbol{x}^\tau, \boldsymbol{z}^\tau, \boldsymbol{v}^\tau, \boldsymbol{y}^\tau)$ and the value of the surrogate variable (θ^τ) for the current iteration τ (Step 2). These values are used to set the lower bound $LB^{(\tau)}$ (Step 3). Then, we solve $\mathbb{P}_{2SD}$ and obtain $\boldsymbol{\pi}^\tau$ by using $\boldsymbol{x}^\tau, \boldsymbol{z}^\tau, \boldsymbol{v}^\tau, \boldsymbol{y}^\tau$ (Step 4). Next, using the Master problem, we can obtain a new upper bound (Step 5-7). Regarding the cuts, we use the set $\mathcal{C}_O$ to collect the optimality cuts, and the set $\mathcal{C}_F$ to collect the feasibility cuts. Initially, these sets are empty, $\mathcal{C}_O^{(0)} = \mathcal{C}_F^{(0)} = \emptyset$. In each iteration τ, we inspect the value of the dual slave problem, and if it is bounded, we add the respective solution vector, denoted $\boldsymbol{\pi}_m$, to the set of optimality cuts $\mathcal{C}_O^{(\tau)}$, as it gives us information for where the optimal solution lies (Step 9); while if it is unbounded, then we add the vector to the set of feasible cuts $\mathcal{C}_F^{(\tau)}$, as it gives us information for where feasible solutions

[9] This linearization comes at the cost of increasing the number of variables, which is preferable in terms of computing cost.

	Split Point	Performance Gains
S1	PDCP / RLC	Enables Layer 3 (RRC, PDCP) and Layer 2 (RLC, MAC) operations in the same server.
S2	MAC / PHY	Enhancements to CoMP with RU frame alignment and centralized HARQ.
S3	PHY / RF	Power-saving opportunities; Enhancements to joint reception CoMP with uplink PHY level combining

TABLE III: Performance Gains of Different Splits [9], [15].

lie (Step 11)). In every iteration, the *new cuts* generate new constrains, and this shrinks the solution space. These steps are repeated until the proportional difference between the upper and lower bounds becomes smaller than the set threshold ϵ (Step 14). If we select $\epsilon = 0$ we obtain the exact optimal solution. This result is formally proved in the next theorem.

Theorem 2. *Algorithm 1 converges within finite iterations to the optimal solution of* $\mathbb{P}_2$*.*

Proof. We prove the result using Theorem 2.5 (*Finite ϵ-convergence*) of [45]. It suffices to verify that the properties of our problem satisfy conditions C1-C3 which are outlined in [46, Theorem 6.3.4]. We first define the set, $\mathcal{O} =$

$$\Big\{\boldsymbol{r} = \big(r^k_{nm} \geq 0\big)_{n,m,k} \mid \sum_{n\in\mathcal{N}} \sum_{p_k\in\mathcal{P}_n} r^k_{nm} I^k_{ij} \leq c_{ij}, \forall (i,j) \in \mathcal{E}\Big\}$$

where we simply used (7) to upper-bound the routing variables. Observe that this means our continuous variables are contained in a non-empty convex set. Furthermore, fixing the discrete variables $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{v})$ the objective and constraint functions become convex on $\boldsymbol{r}$ (namely, linear). Hence, $\mathbb{P}_2$ meets condition C1. Next, we can see directly that $\mathcal{O}$ is bounded and closed, and for fixed discrete variables the constraints (11)-(13) and (8) - (10) are linear on $\boldsymbol{r}$. Recall that the remaining constraints do not include $\boldsymbol{r}$. Therefore, $\mathbb{P}_2$ meets condition C2. Finally, for any feasible set $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{v})$, $\mathbb{P}_2$ admits a solution with bounded value. To see this, observe that all parameters in our linear (in $\boldsymbol{r}$) functions and constraints are bounded, and that $\mathcal{O}$ is bounded. I.e., $\mathbb{P}_2$ and its dual have bounded solution for any set of feasible discrete variables. Since conditions C1-C3 are satisfied, Algorithm 1 converges to the optimal solution of $\mathbb{P}_2$. Clearly, this result presumes that $\mathbb{P}$ has an optimal solution. □

VI. Balancing Costs and Centralization

In this section we extend the framework to include the centralization of functions, denoted R, together with the operating costs. At a first glance, one would expect that these are two perfectly aligned optimization criteria: minimizing the vRAN cost ensures the maximum number of functions are placed at CUs. After all, C-RAN has been envisioned as a solution for lowering the costs of D-RAN. It turns out that in certain cases, i.e., combination of network costs, capacities, and traffic loads, these two metrics might be actually competing.

The centralization level is defined as the aggregate number of functions f_1, f_2, and f_3 that are deployed in CUs:

$$R(\boldsymbol{y}, \boldsymbol{z}) = \frac{1}{3N} \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} \left(y_{1nm} + y_{2nm} + z_{nm}\right), \tag{19}$$

which takes values in the interval $[0, 1]$ with $R(\boldsymbol{y}, \boldsymbol{z}) = 1$ when all functions are placed at CUs (C-RAN), and $R(\boldsymbol{y}, \boldsymbol{z}) = 0$ when we have a D-RAN. The first term in (19) counts how many f_1 functions are deployed in any CU-m, $m \in \mathcal{M}$, the second term counts the centrally deployed f_2 functions, and the last term the centralized f_3 functions. The performance gains of centralization are summarized in Table III. For each split, we have more benefits if more base stations adopt that split (compared to D-RAN); and the benefits increase further, for the same number of split BSs, if we select deeper splits. Function $R(\boldsymbol{y}, \boldsymbol{z})$ captures qualitatively these aspects of the vRAN design problem.

In this framework, we aim to minimize the expenditure cost and maximize simultaneously the network centralization. We can express this combined criterion using scalarization:

$$\boxed{\begin{aligned} \mathbb{Q}: \quad & \min_{\boldsymbol{r},\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \ \eta J(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{r}) - (1-\eta) R(\boldsymbol{y}, \boldsymbol{z}) \\ & \text{s.t.} \quad (1) - (13) \end{aligned}}$$

The tuning scalar parameter $\eta \in [0, 1]$ determines the relative priority the operator wishes to place at each criterion. Increasing η, will prioritize architectures that minimize operating costs, and increasing $(1 - \eta)$ shift focus on maximizing centralization. Clearly, different operators have different priorities for their networks, and this can be reflected in the selection of the weight parameter η. Furthermore, problem $\mathbb{Q}$ can be enriched with additional explicit constraints in order to achieve certain cost-centralization values. For example, an operator might enforce the cost not to be more than twice the minimum J^*, where the latter can be found by setting $\eta = 1$.

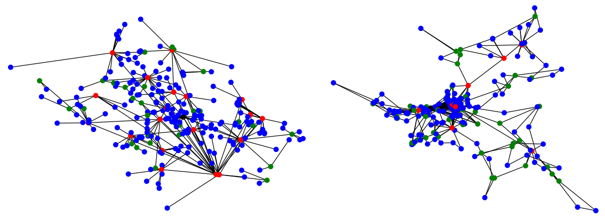
It is not surprising, of course, that $\mathbb{Q}$ is computationally harder than $\mathbb{P}$. This is formally stated in the following lemma.

Lemma 1. *Problem $\mathbb{Q}$ is NP-hard and cannot be approximated within any constant factor.*

Proof. It suffices to see that $\mathbb{P} \leq_L \mathbb{Q}$. Indeed, if we could obtain an oracle in polynomial time for solving $\mathbb{Q}$, then we could call it by setting $\eta = 1$ and obtain the respective optimal solution for $\mathbb{P}$; which, in turn, implies tractability of UHCFL according to Theorem 1. □

Nevertheless, Algorithm 1 can be extended to solve this new problem as well, only with minimal alterations. Indeed, we can see that the Benders' decomposition impacts only the definition of the Master problem, which in the case of problem $\mathbb{Q}$ will additionally include the discrete terms that count the centralization objective. Hence, the steps of the algorithm will not change, except the definition of $\mathbb{P}_{2M}$.[10] This also manifests the power of the proposed framework.

[10] Due to lack of space, we do not repeat the new version of the algorithm, given that the changes are minimal.



(a) N1

(b) N2

Fig. 3: **Radio Access Networks**. Real network topologies from operational networks in different countries [7]. Black, blue, green, and red-colored dots represent the core network, DUs, routers, and CU potential locations, respectively. The RANs are visualized according to their coordinate location (longitude and latitude).

Finally, it is worth emphasizing that we can consider different formulations for function $R(\boldsymbol{y}, \boldsymbol{z})$. For instance, an operator that values more the performance gains of having splits S2 than split S1, can add larger weights in variables $\{y_{2nm}\}_{n,m}$. Similarly, we can imagine cases where it matters if the functions are centralized in few CUs[11]; and such solutions can be driven by adding the discounting term $(-\gamma \sum_{n,m} z_{nm})$ which will activate fewer CUs. Our model and solution algorithm can directly accommodate these changes.

VII. Results and Discussion

In this section we present numerical tests using both real and synthetic datasets to increase the robustness of our findings aiming to examine the: *(i)* optimal multi-CUs vRAN configurations; *(ii)* benefits of deploying multiple CUs; *(iii)* cost-effectiveness of optimizing the CU locations and DU-CU assignments, compared to non-optimized decisions; and *(iv)* the effect of routing cost and DU traffic. Moreover, *(v)* we study the interaction of cost and centralization, where we find that a small cost increase can boost centralization, and explore the effect of routing cost and traffic on this trade-off.

A. Network Topology and Evaluation Setup

We use two real and two synthetic RANs, Fig. 3. RAN N1 consists of a core node, 198 DUs and 15 CU locations; RAN N2 has a core node, 197 DUs and 15 CU locations. Network parameters such as delay, location, distance, and link capacities are derived from the actual data or using other measurement studies. We calculate the DU-CU paths by applying the k-th shortest path algorithm [47]. The distance of DUs to CUs and the core network ranges from 0.1km to 25km and the path delays reach 257.61μs (N1) and 1152.69 μs (N2). As candidate CU locations, we select the nodes with the highest network degree[12]. We also use random networks generated using the Waxman algorithm [48] in order to test our findings in different than the real networks N1 and N2. These random networks comprise 272 nodes and have been generated using the parameters $\alpha = 0.2$ which indicate the link probability, and $\beta = 0.05$ that controls the edge lengths.

We set the system parameters according to actual testbed measurements and previous works [7], [16], [42]. The default DU load is $\lambda_n = 150$ Mbps for each DU[13]. For CPU capacity, we use a *reference core* (RC), Intel i7-4770 3.4GHz, and set the maximum computing capacity to 75 RCs for each CU, and 2 RCs for each DU. We assume that the default cost of CU-m VM instantiation is a half[14] of DU-n ($a_m = \alpha_n/2$) [7], [16], [42] but we also explore the impact of different ratios. The cost of deploying of the CUs and its routing CU-EPC takes values in the interval $\omega_m \in [15, 22]$, where $\omega_m \leq \omega_{m+1}$ [16]. The CU processing cost is set to $b_m = 0.017\beta_n$ according to our measurements in [42]. The CU deployment cost is assumed at least 30 times higher than the CU processing cost. This value is calculated based on the comparison server setup price ($20K) and data processing cost ($653.54) [16]. Finally, the routing cost per path grows linearly with distance, $\zeta = c_d \times d_{km}$, where c_d is the cost per Km per Mbps and captures how expensive is each link (can be different for each network).

B. Increasing the CU candidate locations

Our first experiment increases the number of available CU locations and studies how this impacts the total cost (in monetary units) and centralization. The deployment cost per CU is assumed to be higher as we add more locations, i.e., $\omega_m \leq \omega_{m+1}$, since, naturally, the cheapest locations are selected first. Fig. 4a shows that there are $6.65\%, 8.79\%, 13.10\%$, and 16.36% average cost savings when we increase the number of CU candidate locations from one to $M = 13$, for various routing costs and CU/DU computing efficiency gains. Similar findings hold for N2, Fig. 4b. The trend of cost reduction continues as the number of CU locations increases, albeit the additional gains are fast decreasing and we reach a state (after $M = 13$ in this network) where adding more CU locations does not bring further cost savings.

Figures 4c-4d show the distribution of deployed and non-deployed CUs, i.e., not used locations. In N1, the deployment is 100% when the candidate locations are up to $M = 4$. Beyond that, this percentage decreases, with a constant number of 11 deployed CUs when $M = 13$. However, in N2, we use all locations until $M = 14$, showing that the deployment is affected by the specific candidate locations and network topology. That is, these results are qualitatively persistent but the actual gains depend on the network and CU locations.

Findings: *1) As the CU locations increase, our framework saves significant costs (16.36% in N1 and 28.79% in N2), before the gains diminish. 2) Higher routing costs and lower compute costs of CUs lead to substantial gains. 3) Network characteristics affect the deployment of CUs.*

C. Impact of Routing Cost and Traffic

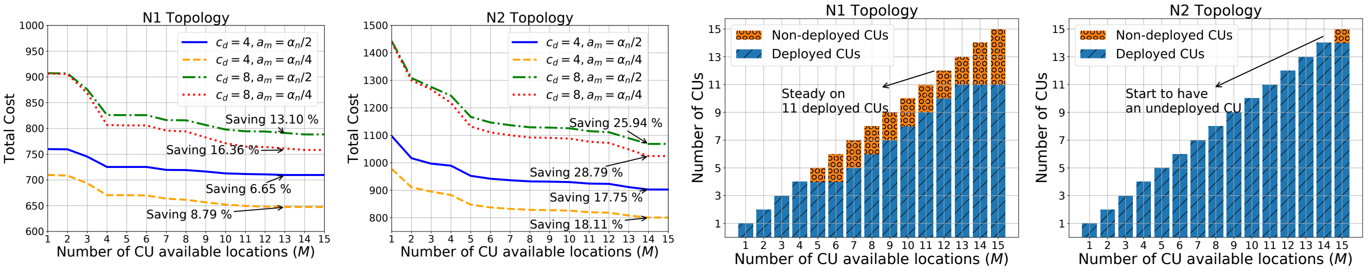
We study the effect of routing cost on the total vRAN cost and the centralization that our solution achieves i.e., the number of functions deployed at the CUs. Aside from

[11] Naturally, having all functionalities of base stations in the same CU gives better control for some operations, while dispersing the BS functions in many CUs might induces again coordination issues.

[12] The RANs do not have CUs, and we followed this intuitive approach to select locations that we then feed to our framework.

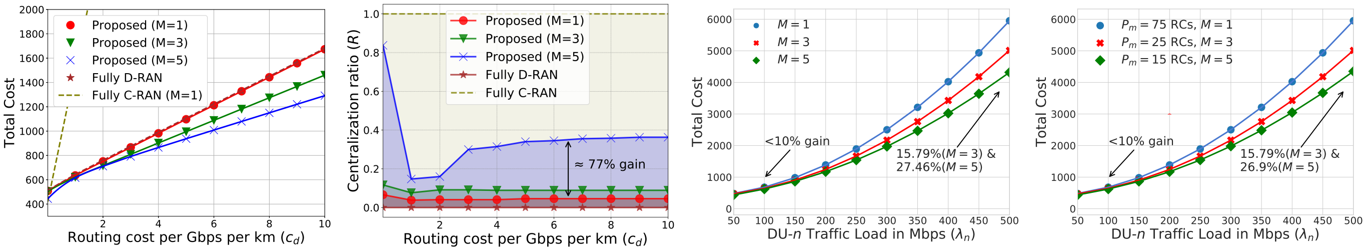
[13] This value corresponds to 2×2 MIMO, 1 user/TTI, 20 Mhz, 2 TBs of 75376 bits/subframe and IP MTU 1500B.

[14] D-RAN BSs cost twice as much as the C-RAN BSs, both for macro ($50K vs $25K) and micro BSs ($20K vs $10K) [16].



(a) (b) (c) (d)

Fig. 4: **Impact of increasing CU locations**: (a-b) on system cost with traffic load $\lambda_n = 150$ Mbps, and different values for the routing costs c_d and ratios of CU$-m$ vs DU-n computing costs a_m/α_n; (b-c) on the number of deployed CUs ($\lambda_n = 150$ Mbps and $c_d = 5$).



(a) (b) (c) (d)

Fig. 5: **Impact of routing cost**: (a) on the total cost; and (b) the centralization ratio. **Impact of DU traffic**: (c) on cost for different CU locations; (d) for heterogeneous CU capacities. Results are for topology N1, $\lambda_n = 150$ Mbps for (a) and (b), and $c_d = 5$ for (d) and (e).

comparing single and multiple CU candidate locations, we also benchmark against C-RAN and D-RAN solutions (two extreme cases). For this experiment, we have used the network N1. In fully C-RAN, all functions except f_0 are placed at the CU, while in D-RAN they are all deployed at the DU. In this case, the deployment cost of CU is set to zero ($\omega_m = 0$) for D-RAN (we do not need to deploy any CU), and the traffic load of each DU is directly routed to the core network. In this experiment, the routing cost (per Km and per Gbps) increases from $c_d = 0.01$ (very low) to $c_d = 10$ (very high). Please note that networks N1 and N2 cannot support fully C-RAN solutions and hence the results for this configuration are hypothetical and presented for reference.

Fig. 5a compares our system cost to C-RAN and D-RAN for different routing cost values c_d. The C-RAN cost increases significantly with c_d, and eventually exceeds the D-RAN cost (for $c_d \approx 0.01$ per Gbps). For a single CU location and the specific network parameters[15], the optimized configuration is slightly better than D-RAN ($0-1\%$) because the CU is near to the core and far from the DUs, thus most functions cannot be deployed at the CU, Fig. 5b. By considering higher number of candidate locations we can obtain approximately 13.10% ($M = 3$) and 23.15% ($M = 5$) cost savings at $c_d = 10$. Fig. 5b shows the impact of the routing cost on centralization ratio. For $M = 5$ the centralization is higher (up to 77%) than the ratio for smaller M values, but we also observe that, for fixed M, the centralization, unlike the cost, is not affected significantly by the routing cost. Clearly, this result is conditioned on the relative values of the routing and computing costs.

Next, we evaluate the impact of DU-n traffic on the cost. We consider two scenarios. Firstly, the computational capacity of each CU is 75 RCs. Secondly, the aggregate CU capacity is P_{tot} = 75 RCs. Consequently, every CU has different capacity, i.e., $P_m = P_{tot}/M$, in each scenario as M changes. Our goal here is to evaluate two design options: to have a single CU with high computational capacity, or multiple CUs with lower capacity. We use the same deployment and VM computational cost for both scenarios. Fig. 5c shows that increasing the values of M implies a reduction in the total cost. This saving gap increases with higher values of λ_n, achieving for $\lambda_n = 500$ Mbps a gain of 15.79% and 27.46% for $M = 3$ and $M = 5$, respectively, compared to $M = 1$ configuration. Similar results are obtained in Fig. 5d, in which the computational capacity of each CU decreases as M increases. In that case, we observe savings of 15.79% and 26.9% for $M = 3$ and $M = 5$, respectively, compared to $M = 1$ configuration.

Findings: *1) The cost savings of having multiple CUs increase with the routing cost and traffic load. 2) Higher values of M imply higher centralization ratio, up to 77% gain. 3) Multiple CUs in different locations can achieve up to 26.9% more cost savings compared to one CU with the same capacity.*

D. Random vs Optimized Deployment of CUs

Next, we evaluate the gains of optimized as opposed to random CU deployments. Fig. 6 shows the total cost as a function of routing cost. For this evaluation the problem has been slightly modified since we consider $M = 15$ and we force our framework to optimize the location of only 3 CUs (blue curve). For the random deployment (green curve), we select 3 random locations from the M available locations and we then optimize the function placement and routing. We

[15] This result is affected by, e.g., the cost of deploying a CU, the distance of this location compared to the core, etc. In general, even a single CU location can significantly reduce costs, see [7] for examples.

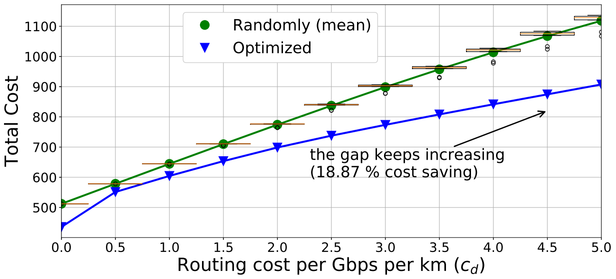


Fig. 6: **Impact of Optimizing CU Deployment**: Total cost J under randomized and optimized deployments, for Toplogy N1 and different routing cost values.

average over 10 iterations get the average total cost. Fig. 6 shows that it is crucial not only to use multiple CUs but to carefully optimize their placement. The cost-saving gap increases with the routing cost up to 18.87% for this setting.

Finding: *The location of the CUs in a multi-CUs vRAN is a crucial aspect to reduce the total cost. The evaluations show that when the CUs locations are optimized a* 18.87% *cost reduction can be achieved with respect to a random CU deployment for a specific setting.*

E. Pareto Analysis of the Multi-objective Problem

We study the problem $\mathbb{Q}$ of Sec. VI that includes both the cost and centralization, which are scalarized through parameter η. As we change η, the priority of these objectives changes and we obtain a new optimal solution. We have observed that there is a trade-off between the criteria; that is, minimizing the total cost does not necessarily imply the maximization of centralization and vice versa. Fig. 7-10 show this trade-off as a Pareto front, for different values of routing cost (c_d), DU demand (λ_n) and for various networks. In addition to the real RANs (N1 and N2), we consider here two random networks aiming at assessing the generality of our solution. Although the Pareto front values change in different scenarios, we always observe the same trade-off in all of them. In any case, our framework achieves all possible optimal network configurations that fall along the Pareto front, which is defined by the total cost and centralization ratio.

Fig. 7-8 evaluate the routing cost impact (c_d) on the total cost vs. centralization trade-off, for different M. As expected, when the routing cost increases (Fig. 8), the trade-off between centralization and operating cost is sharper, in other words, the cost of increasing the centralization ratio (by reducing η) becomes higher. For example, for the network topology N1, we can achieve $R = 1$ for $M = 5$ with an increment of 285% on the total cost (Fig. 7a), while with double routing cost (Fig. 8a) an increment of 507% is needed. We can also observe that, in Waxman topology 1, the configuration with $R = 0.85$ requires incrementing 310% the minimum cost for $M = 5$ (7c), while with double routing cost (8c) an increment of 498% in required. Note that the value of η can be adjusted by the operator depending on the centralization requirements of the network and its monetary budget.

Finding: *The cost we need to pay for increasing centralization, increases with the routing cost (c_d). For example, for topology N1 and $M = 5$, the cost J should be increased by* 285% *to attain $R = 1$, while if we double the value of c_d the increase of J is* 507%.

Fig. 9-10 evaluate the impact of the DU traffic (λ_n) on the Pareto front. These figures show that, although the sharpness of the Pareto front does not change, the curve is shifted in the y axis to higher cost values. This implies that to configure the same centralization ratio in the network the cost to pay is higher as the DU traffic increases. Specifically, comparing Fig. 9a-9d and Fig. 10a-10d we can observe that for the same values of R, doubling the DU traffic implies multiplying the cost by a factor between 2 and 3. Besides, we observe in Fig. 9-10 that the value of M has a higher influence on the Pareto front with higher DU traffics. The annotations in Fig. 9b-9c and Fig. 10b-10c show that for the same value of R the cost reduction associated with the increment of M is higher when the DU traffic increases.

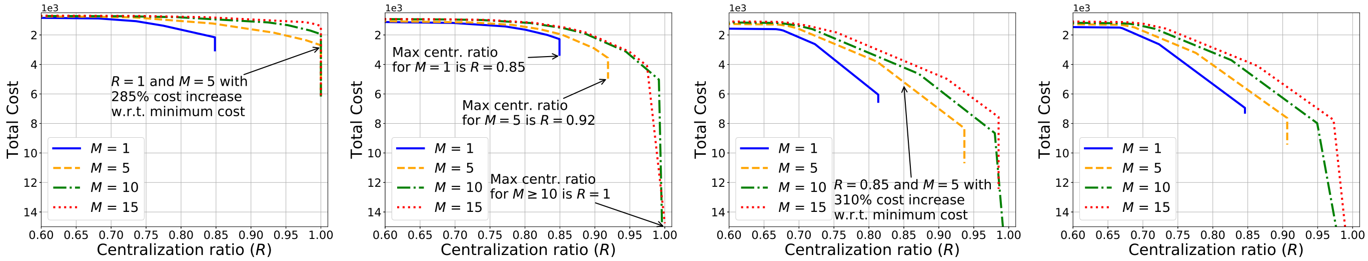
Findings: *1) When the DU traffic (λ_n) doubles, the Pareto front is multiplied by a factor in* $[2, 3]$, *leading to higher costs. 2) The gap between the Pareto curves for different values of M becomes more significant as the DU traffic increases.*

Fig. 11 shows the impact of ω_m on the Pareto front. We evaluate our framework for $\omega'_m = C \cdot \omega_m$, where $C \in [0.01, 10]$ is a constant. As expected, higher values of ω'_m imply that the centralization is more expensive. We also observe that the network configuration at minimum cost ($\eta = 0$) changes with the value of C. This means that the operator can achieve higher centralization at minimum cost as the deployment is cheaper.

Finally, Fig. 12 shows the average centralization ratio that can be achieved for different values of M (x-axis) and its associated increase of the total cost J (y-axis) compared to the minimum achievable cost (i.e., when we set $\eta = 1$). The results are averaged over multiple network instances. We find that for $M = 1$ we obtain $R = 0.48$ on average with minimum cost (y-axis value is $J - J^*/J^* = 0$). And by accepting a cost increase of 41% we can achieve centralization $R = 0.7$. Fig. 12 shows more examples associating the increment in R and its respective increment on the cost, with respect to the minimum attainable (for $\eta = 1$ in each case). The plots average the results for all networks (two real; two synthetic) presented above, and different values of routing cost and traffic load. Hence, it offers general insight into how the increase of network centralization affects the cost. We summarize the key numerical findings in Table IV.

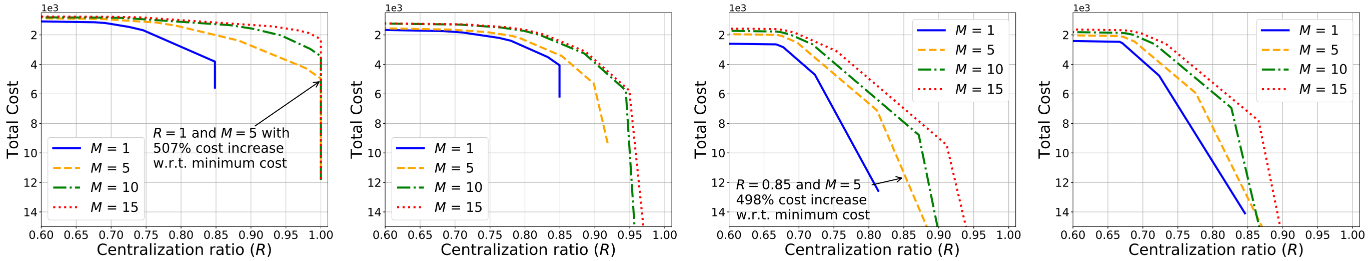
F. Evaluation of the computational time

We evaluate the execution time of Algorithm 1 in a range of scenarios. We used a small laptop with an Intel(R) Core(TM) i7-8750H CPU@2.20GHz, and 8 Gb of RAM memory. Algorithm 1 has been implemented using IBM ILOG CPLEX Optimization Studio 12.9.0.0 and its Python API [49]. We have found that the only parameter having an impact on the execution time is M. Higher values of M imply an increase of the solution space and therefore affect the computational time. However, Algorithm 1 is lightweight and converges fast (despite the lack of theoretical guarantees) since it requires at most 10 secs of computational time for $M = 15$, down to 2 secs for $M = 5$. Moreover, we do not find any significant dependence on the value of the gap (ϵ), other than increasing



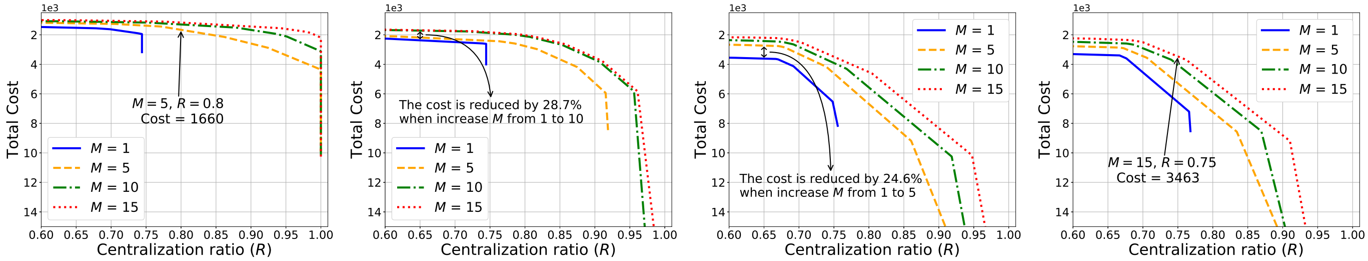
(a) Topology N1 (b) Topology N2 (c) Waxman topology 1 (d) Waxman topology 2

Fig. 7: **Pareto Analysis in Low Routing Cost Scenario**: Study of the cost J and centralization R in two real and two synthetic networks; with routing cost $c_d = 5, \forall p \in \mathcal{P}$, and traffic load $\lambda_n = 150, \forall n \in \mathcal{N}$.



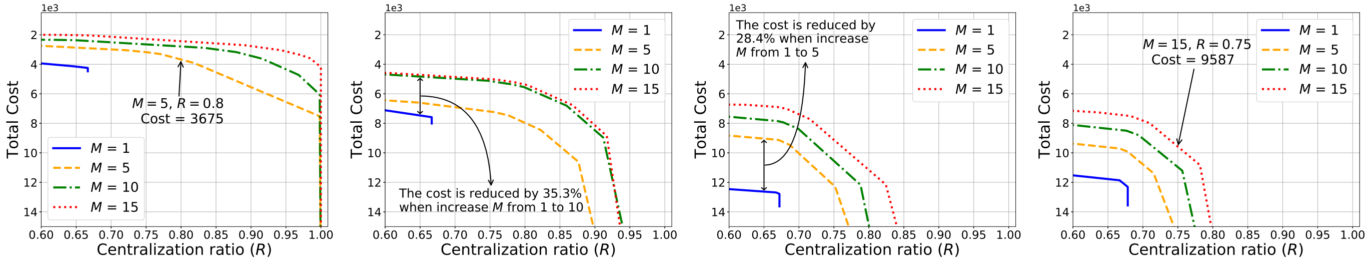
(a) Topology N1 (b) Topology N2 (c) Waxman topology 1 (d) Waxman topology 2

Fig. 8: **Pareto Analysis in High Routing Cost Scenario**: Study of the cost J and centralization R in two real and two synthetic networks; with routing cost $c_d = 10, \forall p \in \mathcal{P}$, and traffic load $\lambda_n = 150, \forall n \in \mathcal{N}$.



(a) Topology N1 (b) Topology N2 (c) Waxman topology 1 (d) Waxman topology 2

Fig. 9: **Pareto Analysis in Low DU Traffic Scenario**: Study of the cost J and centralization R in two real and two synthetic networks; with routing cost $c_d = 5, \forall p \in \mathcal{P}$, and traffic load $\lambda_n = 250, \forall n \in \mathcal{N}$.



(a) Topology N1 (b) Topology N2 (c) Waxman topology 1 (d) Waxman topology 2

Fig. 10: **Pareto Analysis in High DU Traffic Scenario**: Study of the cost J and centralization R in two real and two synthetic networks; with routing cost $c_d = 5, \forall p \in \mathcal{P}$, and traffic load $\lambda_n = 500, \forall n \in \mathcal{N}$.

the number of outliers (executions with 3-4 times longer convergence time).The reduced computational budget required by our proposal makes it suitable for being executed at any CU of the actual network.

VIII. Conclusion

There is a recent flurry of standardization and research activities to make vRAN the de facto solution for next generation access networks. To this end, our work contributes towards filling an important gap as it tackles the multi-cloud vRAN

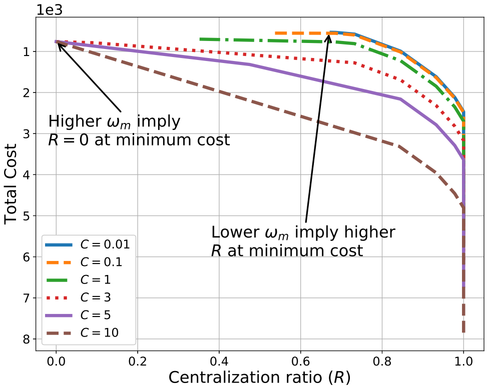


Fig. 11: **Pareto analysis for different CU deployment and routing costs**: We evaluate our framework for $\omega'_m = C \cdot \omega_m$, where $C \in [0.01, 10]$, using topology N1, $M = 5$, $c_d = 5$, and $\lambda_n = 150$ Mbps.

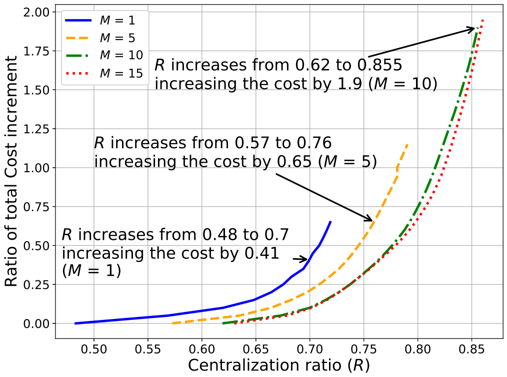


Fig. 12: **Centralization R as a function of cost increment** $(J - J^*/J)\%$: Plots correspond to different values of CU locations $M = \{1, 5, 10, 15\}$, and present mean values for all topologies by varying $c_d \in [1, 10]$ and $\lambda_m \in [50, 500]$, creating 864 experiments.

	Experiment	**Key Findings**
Cost Minimiz.	Impact of adding CU locations (Fig. 4)	• The gains are significant (up to 29%), especially when routing is expensive, and diminish gradually to zero.
	Impact of routing cost and traffic (Fig. 5)	• Adding CUs can increase centralization up to 77% for high routing costs; and for high traffic, it is preferable to disperse computing capacity to multiple CUs.
	Optimized versus random CU placement (Fig. 6)	• Optimizing the deployment saves significant costs, up to 19% when routing is costly.
Cost - centraliz. Trade-off	Relation of centralization and cost (Fig. 7-12)	• These two objectives can be aligned, independent (flat Pareto curve), or conflicting (sharp Pareto curve), based on routing and CU costs, and the traffic.
	Effect of routing cost on trade-off (Fig. 7-8)	• The cost to pay for higher network centralization increases with the routing cost.
	Effect of traffic on the trade-off (Fig. 9-10)	• When traffic (λ_n) doubles, the Pareto front is multiplied by a factor in [2, 3], leading to higher costs. CUs become more important for cost savings when λ_n is high.
	Effect of CU deployment cost on trade-off (Fig. 11)	• Higher CU deployment costs increase the cost of network centralization. Lower CU costs are associated with higher centralization at minimum cost.

TABLE IV: Key Findings of the Numerical Evaluation.

design problem. Using a standards-compatible model we develop a rigorous optimization approach that selects jointly the number and location of CUs; assigns to them the DUs; and finds the optimal split levels and routing paths for each flow. Our framework can be tailored to other scenarios, e.g., when the operator needs to enforce a level of centralization or when the cost functions are non-linear; can account for energy-related or other costs and extended to include decisions such as user association policies. These are interesting directions for future work. The numerical evaluation showed a higher number of CUs implies a cost saving up to 28% and an improvement in the network centralization by 77%. The deployment of multiple CUs implies a cost saving up to 26% with respect to a single CU scenario with the same aggregated computational capacity. We also observed a trade-off between the minimization of the total cost and the maximization of the centralization in the network. Our framework is able to find all the optimal solutions in the Pareto front subject to any centralization or cost constraint.

References

[1] F. W. Murti, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "On the Optimization of Multi-Cloud Virtualized Radio Access Networks," in *Proc. of IEEE ICC (to appear)*, 2020.

[2] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," in *Proc. of ACM MobiCom*, 2019.

[3] I. F. N. T. R. W. Group, "IEEE 5G and Beyond Technology Roadmap White Paper," Institute of Electrical and Electronics Engineers (IEEE), Tech. Rep., 2017.

[4] N. Alliance, "5G White Paper," Next Generation Mobile Networks (NGMN), Tech. Rep., October 2017.

[5] A. Checko *et al.*, "Cloud RAN for Mobile Networks-A Technology Overview," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 1, 2015.

[6] 3GPP, "Study on New Radio Access Technology: Radio Access Architecture and Interfaces," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.801, 03 2017, version 14.0.0.

[7] A. G.-Saavedra et al., "FluidRAN: Optimized vRAN/MEC Orchestration," in *Proc. of IEEE INFOCOM*, 2018.

[8] Nokia, "Mobile Anyhaul White Paper," Nokia, Tech. Rep., 2017.

[9] S. Gonzalez-Diaz et al., "Integrating Fronthaul and Backhaul Networks: Transport Challenges and Feasibility Results," *IEEE Trans. on Mobile Computing*, 2019.

[10] M. Bateni and M. Hajiaghayi, "Assignment Problem in Content Distribution Networks," *ACM Trans. on Algorithms*, vol. 8, no. 3, 2012.

[11] A. Gupte *et al.*, "Solving Mixed Integer Bilinear Problems Using MILP Formulations," *SIAM Journal on Optimization*, vol. 23, no. 2, 2013.

[12] J. F. Benders, "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numer. Math.*, vol. 4, 1962.

[13] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, "Wireless Network Cloud: Architecture and System Requirements," *IBM Journal*

of Research and Development, vol. 54, April 2010.

[14] U. Dötsch *et al.*, "Quantitative Analysis of Split Base Station Processing and Determination of Advantageous Architectures for LTE," *Bell Labs Technical Journal*, vol. 18, no. 1, 2013.

[15] S. C. Forum, "R6.0. Small Cell Virtualization Functional Splits and Use Cases, document 159.07.02," Tech. Rep. Release 7, 2016, version 14.0.0.

[16] V. Suryaprakash, P. Rost, and G. Fettweis, "Are Heterogeneous Cloud-Based Radio Access Networks Cost Effective?" *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, 2015.

[17] K. Garikipati, K. Fawaz, and G. Shin, "RT-OPEX: Flexible Scheduling for Cloud-RAN Processing," in *Proc. of ACM CoNEXT*, 2016.

[18] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks," *IEEE Comm. Surveys & Tutorials*, vol. 21, no. 1, 2019.

[19] D. Harutyunyan and R. Riggio, "Flex5G: Flexible Functional Split in 5G Networks," *IEEE Trans. on Network & Service Management*, vol. 15, no. 3, 2018.

[20] A. M. Alba, J. H. G. Velasquez, and W. Kellerer, "An Adaptive Functional Split in 5G Networks," in *Proc. of IEEE INFOCOM*, 2019.

[21] Y. Alfadhli, M. Xu, S. Liu, F. Lu, P. Peng, and G. Chang, "Real-Time Demonstration of Adaptive Functional Split in 5G Flexible Mobile Fronthaul Networks," in *Proc. of IEEE OFC*, 2018.

[22] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "Sliced-RAN: Joint Slicing and Functional Split in Future 5G Radio Access Networks," in *Proc. of IEEE ICC*, 2019.

[23] N. Mharsi, M. Hadji, D. Niyato, W. Diego, and R. Krishnaswamy, "Scalable and Cost-Efficient Algorithms for Baseband Unit (BBU) Function Split Placement," in *Proc. of IEEE WCNC*, 2018.

[24] O. Arouk *et al.*, "Cost Optimization of Cloud-RAN Planning and Provisioning for 5G Networks," in *Proc. of IEEE ICC*, 2018.

[25] N. Mharsi and M. Hadji, "Edge Computing Optimization for Efficient RRH-BBU Assignment in Cloud Radio Access Networks," *Computer Networks*, vol. 164, p. 106901, 2019.

[26] L. Wang and S. Zhou, "Flexible Functional Split and Power Control for Energy Harvesting Cloud Radio Access Networks," *IEEE Trans. on Wireless Communications*, 2019.

[27] H. Gupta, M. Sharma, A. Franklin A., and B. R. Tamma, "Apt-RAN: A Flexible Split Based 5G RAN to Minimize Energy Consumption and Handovers," *IEEE Trans. on Network Services and Management*, vol. 17, no. 1, pp. 473–487, 2020.

[28] D. Mishra, H. Gupta, B. R. Tamma, and A. A. Franklin, "KORA: A Framework for Dynamic Consolidation and Relocation of Control Units in Virtualized 5G RAN," in *Proc. of IEEE ICC*, 2018.

[29] A. Garcia-Saavedra et al., "WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul," *IEEE Trans. on Mobile Computing*, vol. 17, no. 10, pp. 2452–2466, 2018.

[30] G. Rodolakis *et al.*, "Replicated Server Placement with QoS Constraints," *IEEE Trans. on Parallel and Distr. Sys.*, vol. 17, no. 10, 2006.

[31] R. Cohen, L. Eytan, J. Naor, and D. Raz, "Near Optimal Placement of Virtual Network Functions," in *Proc. of IEEE INFOCOM*, 2015.

[32] S. Vassilaras *et al.*, "The Algorithmic Aspects of Network Slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[33] H. Guo, J. Liu, and J. Lv, "Toward Intelligent Task Offloading at the Edge," *IEEE Network*, vol. 34, no. 2, pp. 128–134, 2019.

[34] J. Liu *et al.*, "Smart and Resilient EV Charging in SDN-Enhanced Vehicular Edge Computing Networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 217–228, 2020.

[35] I. Contreras and E. Fernandez, "General Network Design: A Unified View of Combined Location and Network Design Problems," *European Journal of Operational Research*, vol. 219, 2012.

[36] C. Astorquizaa, I. Contrerasa, and G. Laporte, "Multi-level facility location problems," *European Journal of Operational Research*, vol. 219, no. 3, 2018.

[37] R. Rahmaniani, T. Gabriel, C. Gendreau, and Rei, "The Benders Decomposition Algorithm: A Literature Review," *European Journal of Oper. Res.*, vol. 259, 2017.

[38] H. Lin and H. Uster, "Exact and Heuristic Algorithms for Data-Gathering Cluster-Based Wireless Sensor Network Design Problem," *IEEE/ACM Trans. on Networking*, vol. 22, no. 3, pp. 903–916, 2014.

[39] L. P. Qian, Y. J. A. Zhang, Y. Wu, and J. Chen, "Joint Base Station Association and Power Control via Benders' Decomposition," *IEEE Trans. on Wireless Communications*, vol. 12, no. 4, pp. 1651–1665, 2013.

[40] A. Elwalid and D. M. dn Q. Wang, "Distributed Nonlinear Integer Optimization for Data - Optical Internetworking," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1502–1513, 2006.

[41] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible Centralization through Cloud-RAN," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 35–44, 2014.

[42] A. G.-Saavedra *et al.*, "Joint Optimization of Edge Computing Architectures and Radio Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, 2018.

[43] N. Nikaein, "Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling," in *Proc. ACM Mobisys*, 2015.

[44] 3GPP TR 36.887, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Study on Energy Saving Enhancement for E-UTRAN (Release 12)," Tech. Rep., 2014.

[45] A. Geoffrion, "Generalized Benders Decomposition," *Journal of Opt. Theory and Applications*, vol. 10, no. 4, 1972.

[46] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York, N.Y.: Oxford University Press, 1995.

[47] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, no. 11, 1971.

[48] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[49] "IBM ILOG CPLEX Optimization Studio," version 12.9.0.0. [Online]. Available: http://www.cplex.com

Fahri Wisnu Murti received his B.S. from Telkom University, Bandung, Indonesia (2010-2014) and completed his M.S. at WENS Lab, Kumoh National Institute of Technology, South Korea (2016-2018). He was a student internship at Huawei Technologies, Shenzhen & Beijing, PR China (2013), a network engineer at Nokia Networks, Jakarta, Indonesia (2014-2016), and a research assistant at School of Computer Science and Statistics, Trinity College Dublin, Ireland (2019). He is currently pursuing a Ph.D. at the Centre for Wireless Communication (CWC), University of Oulu, Finland. His research interests include in the application of machine learning and optimization for wireless networks.

Jose A. Ayala-Romero received his M.Sc. and Ph.D. degrees from Technical University of Cartagena (UPCT), Spain, in 2015 and 2019, respectively. He was a visiting Ph.D. student with the DEI Department, University of Padova (2017), and NEC Laboratories Europe, Germany (2018). Since November 2019, he is a post-doctoral researcher with School of Computer Science and Statistics at Trinity College Dublin.

Andres Garcia-Saavedra received his PhD degree from the University Carlos III of Madrid (UC3M) in 2013. He then joined Trinity College Dublin (TCD), Ireland, as a research fellow. Since July 2015, he is a senior researcher with NEC Laboratories Europe. His research interests lie in the application of fundamental mathematics to real-life wireless communication systems.

Xavier Costa-Pérez (M'06–SM'18) is Head of Beyond 5G Networks R&D at NEC Laboratories Europe, Scientific Director at the i2Cat R&D Center and Research Professor at ICREA. His team contributes to products roadmap evolution as well as to European Commission R&D collaborative projects and received several awards for successful technology transfers. In addition, the team contributes to related standardization bodies: 3GPP, ETSI NFV, ETSI MEC and IETF. Xavier has been a 5GPPP Technology Board member, served on the Program Committee of several conferences (including IEEE Greencom, WCNC, and INFOCOM), published at top research venues and holds several patents. He also serves as Editor of IEEE Transactions on Mobile Computing and Transactions on Communications journals. He received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia (UPC) in Barcelona and was the recipient of a national award for his Ph.D. thesis.

George Iosifidis received the Diploma degree in telecommunications from the Greek Air Force Academy, Athens, in 2000, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Thessaly, in 2012. He was an Assistant Professor at Trinity College Dublin (2016-2020) and is currently an Assistant Professor at Delft University of Technology. He is serving as an Editor for IEEE Transactions on Communications and IEEE/ACM Transactions on Networking.