Enabling dynamically centralized RAN architectures in 5G and beyond

Alberto Martínez Alba, *Student Member, IEEE,* Shakthivelu Janardhanan, and Wolfgang Kellerer, *Senior Member, IEEE*

***Abstract*—In order to deliver the high data rates promised for 5G networks, mobile base stations need to be deployed in dense layouts. This results in increased inter-cell interference, which can be mitigated by leveraging centralized architectures in radio access networks. Nonetheless, centralizing all the processing requires prohibitively high link capacities for the fronthaul network connecting centralized and distributed units. In contrast, a static, partially-centralized architecture yields poor performance as it fails to adapt to instantaneous interference situations. In this work, we show that a dynamically centralized architecture enables drastic interference reductions even when using a very limited fronthaul network. We propose multiple algorithms to find the optimal centralization option and evaluate their performance on operator-grade hardware. In addition, owing to the dynamicity of the problem being solved, we provide a framework to decide on the best algorithm based on the trade-off between performance, cost, and adaptation time.**

***Index Terms*—Dynamic, functional split, 5G, flexible, wireless networks and cellular networks, mathematical optimization, interference, centralization.**

I. Introduction

Every new generation of mobile networks introduces novel technologies with the intention of providing services to new markets. For example, in 2G, the possibility to extend mobile communications beyond voice calls is introduced. In 3G, the focus is to provide high-speed mobile broadband access. The 4G architecture supports all-IP networking, point-to-multipoint connections and the broadcast of alarms [1], [2]. In 5G, a substantial share of the research attention goes towards low-latency and machine-type communications [3]. Nonetheless, a common objective for all mobile generations is to provide faster connectivity than that of the previous generation. Indeed, being able to provide high data rates (up to 1 Gb/s) is still the main selling feature of 5G [4].

There are three strategies to increase user data rates over the air interface: improving spectral efficiency, allocating new spectrum, or increasing cell density. The first method is the goal in the development of improved modulations, but physical limits hinder the achievement of drastic improvements [5]. The second method is exploited in the research towards enabling millimeter communication in 5G, but spectrum scarcity and bad propagation properties also pose important challenges [6]. Finally, the third method can be straightforwardly applied by mobile operators so as to increase received signal power and thus data rates. However, increasing cell density also increases interference, which may counter the benefits of higher signal power. As a consequence, interference-mitigation techniques and efficient network management are required to take advantage of dense deployments.

Numerous interference-mitigation techniques have been proposed over the years, such as interference coordination [7], successive interference-cancellation [8], or coordinated multi-point [9]. In addition, the emergence of network function virtualization (NFV) along with the development of faster optical networks and more affordable data centers has led to more efficient architectures for the 5G radio access network (RAN). One example is Cloud-RAN [10], in which all the processing of the base stations (also referred to as gNodeBs, or gNBs, in 5G) is centralized into a data center. This enables fast inter-gNB coordination to easily implement the aforementioned interference-mitigation techniques.

Nonetheless, it was soon noticed that a fully centralized architecture requires a high-capacity *fronthaul* network connecting the central data center with the remote antennas, which renders this architecture infeasible in many cases [10]. To overcome this issue, the concept of *functional split* is proposed [11]: instead of centralizing all the processing of a gNB, we select a subset of its functions to be deployed in a central unit (CU), whereas the remaining functions run at the distributed unit (DU). There are multiple functional splits options that feature different interference-mitigation capabilities and require different fronthaul capacities [12], [13]. Namely, the larger the number of centralized functions, the higher the required fronthaul capacity, but also the higher the effectiveness of interference-mitigation techniques, owing to the faster coordination between centralized gNB functions. Therefore, the functional split of each gNB can be tailored to the specific requirements of the network.

Previous work tackles the problem of statically selecting the optimal functional split of each gNB based on its average traffic patterns and resource usage. This provides better average performance than a uniform selection of the functional split, but it still results in poor performance when the experienced traffic deviates from its average value. In this work, we go one step beyond and propose to adapt the functional split dynamically, during runtime, in accordance with the interference experienced by the user equipments (UEs). We model the dynamic problem, propose multiple algorithms to address it, and show that a dynamic adaptation of the functional split is feasible, cost-efficient, and leads to substantial data rate enhancements. We also provide a simple strategy to trigger the change in the functional split.

The rest of this paper is structured as follows. In Sec-

W. Kellerer, A. Martínez Alba, and S. Janardhanan are with the Chair of Communication Networks, Technical University of Munich, Munich, 80333, Germany.

tion II we briefly present the related work on this topic. Section III describes the system model. In Section IV, we formulate the main problem and derive several approaches to solve it. Section V describes the experimental equipment and setup. In Section VI we present the experimental results. We evaluate the cost of a dynamic RAN in Section VII. Finally, Section VIII concludes the paper.

II. Related work

Previous research works about flexible functional splits can be divided into two categories: those dealing mainly with theoretical aspects and those focusing on the implementation. In the former category, [11] is one of the first works to propose that the functional split could be selected differently, yet statically, for each gNB based on the fronthaul limitations. The authors argue that function centralization is desirable to reduce interference and cost, but limited by the fronthaul capacity. A similar idea is explained further in [14], where a more complete framework is presented.

Building upon the same idea, in [15] the authors formulate the problem of selecting the optimal functional split for the deployment phase. Their objective is to minimize network and computing costs while centralizing as many functions as possible. In order to do estimate the required fronthaul capacity, the expected average traffic of each gNB is used. The authors of [16] face a similar problem with a different objective: minimizing traffic delay. In [17] the idea of dynamically changing the functional split is introduced with the intention of allocating new slices within a virtual RAN framework. Inter-cell interference reduction and fronthaul bandwidth minimization are the main objectives when selecting the functional split, although this selection is not updated once the slice is implemented. Finally, the authors of [18] present for the first time the idea of adapting the functional split dynamically to cope with the instantaneous interference situation. Nonetheless, they tackle a simplified version of the problem and focus on confirming that the network changes slowly enough so that dynamic adaptation is possible, without providing a detailed strategy on how to select the functional split.

Regarding the implementation aspects, there are two main works that focus on realizing a flexible functional split. In [19], a comprehensive description of a platform supporting multiple functional splits is presented, although the capability of changing during runtime is not included. Conversely, in [20] the authors present a pioneer framework that enables to change the functional split of a gNB without stopping its operation or dropping packets. However, the motivation to trigger such a change is not studied. To the best of our knowledge, this is the first work addressing in detail the problem of reconfiguring the functional splits of all gNBs at runtime based on its experienced interference.

III. System model

In this section, we introduce the concepts required to formulate the dynamic functional split selection problem. We describe the network components, explain the considered functional split, and explain the adaptation framework.

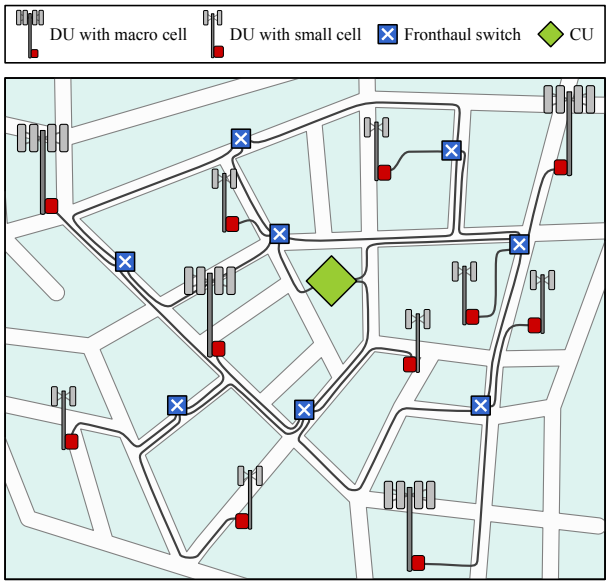


Figure 1: Example network with $G = 11$ gNBs (including macro and small cells) and eight fronthaul switches.

A. Network description

The considered network consists of G gNBs, whose operation is divided into a DU and a CU. The CUs of all gNBs are deployed in a single data center, whereas the DUs are located close to the radio equipment of the cells. As a result, there are G different DU locations and a single CU location.

CUs and DUs are connected by means of a packet-switched fronthaul network [15], which includes layer-2 or layer-3 switches. We assume that these switches are able to steer and divide the incoming flows as configured by a central controller at the CU, thus following the software-defined networking (SDN) paradigm. We model this fronthaul network by a directed graph $\mathcal{D} = (\mathbb{N}, \mathbb{E})$, where $\mathbb{N}$ is the set of network nodes (DUs, switches, and CU) and $\mathbb{E}$ is the set of network links. The total number of nodes and eges, that is, the cardinality of sets $\mathbb{N}$ and $\mathbb{E}$ is represented as N and E, respectively. We denote by n_0 the node corresponding to the CU and by n_g the node corresponding to DU g, such that $g \in \{1, ..., G\}$. A depiction of a simple network with $G = 11$ gNBs and eight fronthaul switches is shown in Fig. 1.

In addition to a CU and a DU, recent gNodeB architectures often include a remote unit (RU), which hosts the radiofrequency equipment and, optionally, the lower part of the physical functions. Their presence is not precluded in our system model, but we assume that DUs are connected to their corresponding RUs by means of dedicated, high-capacity links instead of a shared, capacity-limited network [21]. As a result, the RUs do not modify our problem formulation, since they do not share network resources with the fronthaul[1] network nor impact the interference mitigation capabilities. It is, nonetheless, possible to drop this assumption and extend the presented problem formulation for two split configurations

[1]When RUs are considered, it is usual to reserve the term *fronthaul* to the network connecting DUs to RUs, whereas that connecting DUs to CUs is called *midhaul*. Since our system model does not require the presence of RUs, we always refer to the network connecting CUs and DUs as *fronthaul*.

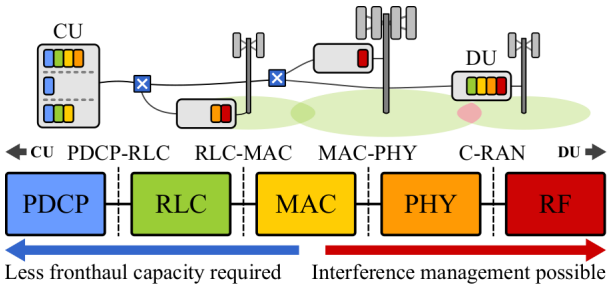


Figure 2: Scheme of the considered network, gNB functions, and functional splits.

(CU/DU and DU/RU splits) without severely affecting the proposed solution approach.

There are U simultaneously active UEs within the coverage area of all cells. Each UE u is connected to a serving gNodeB, which is denoted by h_u. The vector of serving gNodeBs for all UEs is $\mathbf{h} \triangleq [h_1, ..., h_U]$. Throughout the paper, we focus on the downlink data rates and on the downlink interference as perceived by the UEs. Nonetheless, an extension of the analysis to include the uplink is straightforward.

Finally, we assume that all gNodeBs operate in the same frequency bands, that is, a frequency reuse factor of 1. This is done to highlight the performance of the interference mitigation enabled by the dynamic functional split, although using a different reuse factor is not precluded.

B. Functional splits

The processing chain of every gNB can be divided into functions, which are often identified with the layer or sublayers of the RAN protocol stack [12]. For each pair of consecutive functions we define a functional split option. We denote by Q the number of possible functional split options, also referred to as *centralization levels*. For instance, in Fig. 2 there are five functions and $Q = 4$ centralization levels (PDCP-RLC, RLC-MAC, MAC-PHY, and C-RAN). The instantaneous centralization level of a gNodeB g is denoted by x_g, such that $x_g \in \{0, ..., Q-1\}$. We consider that $x_g = 0$ denotes the lowest centralization level, that is, the functional split option for which the least amount of functions are centralized. In Fig. 2, $x_g = 0$ corresponds to the PDCP-RLC split. Conversely, $x_g = Q - 1$ denotes the highest centralization level. In Fig. 2, $x_g = 3$ corresponds to the C-RAN split. The *centralization vector* of all centralization levels is defined as $\mathbf{x} \triangleq [x_1, ..., x_G]$.

As indicated in Fig. 2, low centralization levels require less fronthaul capacity, but their interference-mitigation capabilities are limited. Conversely, gNBs implementing high centralization levels are able to coordinate with each other to reduce the interference they cause to each other, at the expense of requiring higher fronthaul capacity [21].

1) Interference mitigation: The activity of neighboring cells causes downlink interference on nearby UEs, hence reducing user data rates. Proposed techniques to mitigate this interference (such as coordinated scheduling, coordinated beamforming, joint transmission, etc.) require some level of coordination between the involved gNB functions. This coordination is, in general, not possible between distributed functions, since the latency due to propagation, switching, and processing on the different units may prevent fast communication between coordinated functions. For example, in order to employ interference cancellation, gNBs need to generate, communicate, and apply the interference cancellation algorithm to their transmission slots in less than the duration of a 5G time slot [22], which can be as short as 62.5 μs when using high numerologies [23]. As result, each interference-mitigation technique requires a minimum centralization level to be applied, depending on which functions need to be centralized for the technique to operate properly. For instance, coordinated scheduling requires the centralization of the MAC layer [24], whereas joint transmission also requires the centralization of the physical layer [25].

Based on the analysis shown in [26], we model the effectiveness of an interference-mitigation technique between two gNBs as a constant factor multiplying their average received interference power. We relate each centralization level x with the interference-cancellation factor of the most effective interference-mitigation technique that is supported by means of function $c(x)$. The codomain of function $c(x)$ is $[0, 1]$, that is, it ranges from 0 (full interference cancellation) to 1 (no interference cancellation).

Since centralization levels are defined incrementally along the function chain, the higher the centralization level x, the lower its related interference-cancellation factor $c(x)$. Moreover, an interference-mitigation technique can only be used by two gNBs if both of them are operating at the required centralization level or higher. As a consequence, the resulting interference-cancellation factor between gNBs g and g' is $c(\min(x_g, x_{g'}))$, that is, the gNB with the lowest centralization level is the bottleneck to interference mitigation. Knowing this fact, we can compute the expected total interference power I_u experienced by UE u from all gNBs as:

$$I_u(\mathbf{x}) = \sum_{g=1}^{G} i_{u,g} \cdot c(\min(x_{h_u}, x_g)), \tag{1}$$

where $i_{u,g}$ is the interference power received by UE u from gNB g and $i_{u,h_u} \triangleq 0$, as the UE is not interfered by its serving gNB. Note that I_u is a function of the centralization vector $\mathbf{x}$, hence selecting the right values of $\mathbf{x}$ can be used to reduce overall interference.

2) Fronthaul network: The capacity required for a fronthaul link connecting the DU and CU of a gNB depends on its centralization level, that is, on its functional split. Namely, previous research has shown that high centralization levels, such as the Intra-PHY split or full centralization, require large link capacities (in the order of hundreds of Gb/s), whereas low centralization levels (such as the PDCP-RLC split) require capacities barely larger than the user data rate (in the order of a few Gb/s) [12], [13], [21]. Formally, we model the capacity required by gNB g with centralization level x_g as the function $r(x_g)$. For the sake of simplicity, we assume that all gNBs offer the same maximum user data rate, hence $r(x_g)$ does not depend explicitly on g. If required, extending $r(x_g)$ to include this dependency is straightforward.

Finally, we define Φ_e as the capacity of each fronthaul link $e \in \mathbb{E}$. For each gNB g producing a downlink flow between its DU and the CU, we denote by f_e^g the fraction of this flow that is carried over link e. For notation convenience, we also define $\mathbf{f}^g = \left[f_1^g, ..., f_E^g\right]\ \forall g \in \{1, ..., G\}$ and $\mathbf{f} = \left[\mathbf{f}^1, ..., \mathbf{f}^G\right]$ as the vectors of the flow generated by gNB g and all flows, respectively.

C. Adaptation framework

We have shown in (1) that the centralization vector $\mathbf{x}$ affects the interference experienced by all UEs in the network, thus being able to change it dynamically is desirable to adapt to varying UE traffic and mobility. Two components are required to implement a dynamically-adapting functional split on a real 5G implementation. First, we need a *decision-making entity* that continuously monitors the state of the network and selects the optimal functional split. Second, a *migration platform* is also required to realize changes in the functional split without stopping the operation of the network.

There are multiple strategies to construct the decision-making entity. For example, we could use reinforcement learning to derive a set of automatic action rules from the observed context to trigger a change in the functional split. In fact, related problems in the field of mobile networks have been tackled successfully with reinforcement learning [27]. Another option would be to employ dynamic evolutionary algorithms to continuously update a previously-selected centralization vector under a changing environment [28]. These methods are, nonetheless, metaheuristic approaches, whose effectiveness is not always close to optimal. Albeit these methods will be considered in future work, an optimal reference is needed in order to assess how effective any technique is. In this paper, we formulate the selection of the centralization vector as a non-linear optimization problem and propose efficient strategies to find near-optimal solutions. As a result, we assume that the decision-making entity works as follows. First, it receives periodic information regarding either the position of all active UEs, from which their received interference can be estimated, or a full report of this interference directly from the active UEs. Note that the former is already possible in 4G and 5G networks [29], whereas the latter can be extracted from CSI reports. Then, the decision-making entity periodically runs an adaptation algorithm and forward the potential decision to change the functional split to the migration platform. Note that, although not included in the theoretical approach, we can use stored solutions from the past as the starting point for computing the current solutions. This is allowed by modern MIP solvers in order to reduce the solving time.

Regarding the migration platform, we assume that there is an underlying technology that is able to change the functional split without stopping the network operation. Previous research has addressed the design of such platform. Indeed, the authors of [20] developed a dedicated framework that can change the functional split in less than 20 ms without packet losses. Alternatively, gNB functions can be implemented as virtual machines or containers, and off-the-shelf frameworks can be used to lively migrate them [30].

IV. Problem formulation

In this section, we present the problem of dynamically selecting the functional split and derive approximative reformulations and heuristics.

A. Proportionally-fair formulation

The objective of dynamically selecting the optimal functional split is to maximize the data rate of all users by minimizing interference. The data rate ρ_u achieved by UE u can be calculated as $\rho_u = B_u \eta_u$, where B_u is the bandwidth allocated to UE u and η_u denotes its downlink spectral efficiency. We can use Shannon's formula to estimate the latter as follows:

$$\eta_u(\mathbf{x}) = \log_2\left(1 + \frac{s_u}{\varsigma + I_u(\mathbf{x})}\right), \tag{2}$$

where s_u is the signal power received by UE u from its serving gNB h_u, $I_u(\mathbf{x})$ is the experienced interference as defined in (1), and ς is thermal noise power (assumed constant over all UEs). Using (2), we could formulate an optimization problem to find the centralization vector $\mathbf{x}^*$ that maximizes the sum of user data rates:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} \sum_u \rho_u, \tag{3}$$

where $\mathbb{X}$ is the set of vectors $\mathbf{x}$ whose required link capacities are supported by the fronthaul network. However, problem (3) may lead to unfair situations, since data rates of users with good signal-to-interference-and-noise ratio (SINR) may be prioritized over those with poor SINRs. In order to prevent that, it is a better practice to maximize the sum of the logarithm of the data rates, which, as shown in [31], provides a proportionally-fair prioritization. Thus, we define the optimal centralization vector $\mathbf{x}^*$ as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} \sum_{u=1}^{U} \log\left(\rho_u\right) = \arg\max_{\mathbf{x}\in\mathbb{X}} \sum_{u=1}^{U} \log\left(B_u \eta_u(\mathbf{x})\right) \tag{4}$$

$$= \arg\max_{\mathbf{x}\in\mathbb{X}} \sum_{u=1}^{U} \log(\eta_u(\mathbf{x}))\,. \tag{5}$$

Note that we can remove B_u from the formulation since it does not depend on $\mathbf{x}$. We refer to the problem of finding the centralization vector $\mathbf{x}^*$ as defined in (5) as the *Funtional Split Selection Problem* (FSSP).

This objective function in (5) is directly related to the *geometric mean* of the spectral efficiency over all UEs, which is defined as:

$$\widetilde{\eta}(\mathbf{x}) \triangleq \left(\prod_{u=1}^{U} \eta_u\right)^{\frac{1}{U}} = \exp\left(\frac{1}{U}\sum_{u=1}^{U} \log\left(\eta_u(\mathbf{x})\right)\right). \tag{6}$$

Therefore, an equivalent definition of the optimal, proportionally-fair centralization vector is:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} \widetilde{\eta}(\mathbf{x}). \tag{7}$$

This formulation allows us to use $\widetilde{\eta}(\mathbf{x})$ as performance indicator when comparing alternative solutions, as it is done in Sec. VI, where experimental results are shown.

From (2) and the definition of $\mathbf{x}^*$ in (5), we can obtain an expression for the objective function of the FSSP. Regarding the constraints of the FSSP, it is clear that the validity of a solution $\mathbf{x}$ is limited by the topology and capacity of the fronthaul network. In other words, a solution $\mathbf{x}$ is valid if and only if there exists a vector of flows $\mathbf{f}$ that satisfies the flow requirements for every gNB, as mentioned in Sec. III-B2, and can be implemented on the fronthaul network without exceeding the capacity of any link. As a result, we can formulate the FSSP as follows:

$$\max_{\mathbf{x},\mathbf{f}} \sum_{u=1}^{U} \text{llog}\left(\frac{s_u}{\varsigma + \sum_{g=1}^{G} i_{u,g} \cdot c(\min(x_{h_u}, x_g))}\right), \tag{P0a}$$

subject to

$$\sum_{e\in\mathbb{E}^+(n)} f_e^g - \sum_{e\in\mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n \in \mathbb{N}\setminus\{n_0, n_g\} \\ r(x_g) & \text{for } n = n_0 \\ -r(x_g) & \text{for } n = n_g \end{cases} \quad \forall g \in \mathbb{G}, \tag{P0b}$$

$$\sum_{g=1}^{G} f_e^g \le \Phi_e \qquad \forall e \in \mathbb{E}, \tag{P0c}$$

$$f_e^g \ge 0 \qquad \forall e \in \mathbb{E}, \forall g \in \mathbb{G}, \tag{P0d}$$

$$\mathbf{x} \in \{1, ..., Q\}^G, \tag{P0e}$$

where $\mathbb{G} \triangleq \{1, ..., G\}$, $\mathbb{E}^+(n)$ is the set of edges leaving node n, $\mathbb{E}^-(n)$ is the set of edges entering node n, and

$$\text{llog}(\xi) \triangleq \log\left(\log_2(1+\xi)\right), \tag{8}$$

is a shorthand function used for notational convenience. Constraint (P0b) is the *flow conservation* constraint, which ensures that the flow leaving the CU and entering the DU is $r(x_g)$ for each gNB g. In addition, (P0c) enforces the *link capacity* constraint for each link e.

The FSSP as formulated in (P0) is a mixed integer nonlinear problem (MINLP), which are, in general, NP-Hard. Moreover, the non-standard expression of the objective function (P0a) prevents the direct utilization of state-of-the-art techniques. In order to make it more tractable, we present two reformulations that simplify the problem structure at the expense of introducing additional variables.

We start with the following variable change, which allows us to replace the discrete functions $r(\cdot)$ and $c(\cdot)$ by polynomial expressions:

$$x_g = \sum_{q=1}^{Q-1} y_g^q, \tag{9}$$

such that $y_g^q \in \{0,1\}$ and $y_g^q \ge y_g^{q'}$ if and only if $q \le q'$. For compactness, we define $\mathbf{y}_g \triangleq \left[y_g^1, ..., y_g^{Q-1}\right]$ $\forall g \in \mathbb{G}$ and $\mathbf{y} \triangleq [\mathbf{y}_1, ..., \mathbf{y}_G]$. The purpose of this variable change is to convert the integer variables x_g into binary variables in a particularly useful fashion. For instance, given $x_g = 2$ and $Q = 4$, then $\mathbf{y}_g = [1,1,0]$. Note that this is not the conventional manner of performing an integer-to-binary conversion in integer programming, which usually consists in the binary representation of the numbers from 0 to Q and thus requires $\lceil \log_2(Q) \rceil$ new variables per original variable. Instead, conversion (9) requires $Q-1$ new variables, but this increment in the number of additional variables is very small (since $Q \le 8$ in real deployments [12]), and it is compensated by its useful implications. Indeed, by using the variable change in (9), we can rewrite function $c(\min(x_g, x_k))$ as:

$$c(\min(x_g, x_k)) = c(1) - \sum_{q=1}^{Q-1} \delta(q) y_g^q y_k^q, \tag{10}$$

where $\delta(q) = c(q-1) - c(q)$. As a result, we can reformulate the FSSP as:

$$\max_{\mathbf{y},\mathbf{f}} \sum_{u=1}^{U} \text{llog}\left(\frac{s_u}{\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{q=1}^{Q-1} \delta(q) y_g^q y_{h_u}^q\right)}\right), \tag{P1a}$$

subject to

$$\sum_{e\in\mathbb{E}^+(n)} f_e^g - \sum_{e\in\mathbb{E}^-(n)} f_e^g = \begin{cases} 0 & \forall n \in \mathbb{N}\setminus\{n_0, n_g\} \\ r_y(\mathbf{y}_g) & \text{for } n = n_0 \\ -r_y(\mathbf{y}_g) & \text{for } n = n_g \end{cases} \quad \forall g \in \mathbb{G}, \tag{P1b}$$

$$y_g^1 \ge y_g^2 \ge ... \ge y_g^{Q-1} \qquad \forall g \in \mathbb{G}, \tag{P1c}$$

$$\mathbf{y} \in \{0,1\}^G, \tag{P1d}$$

(P0c), and (P0d),

where $I_u = \sum_{g=1}^{G} c(1) i_{u,g}$ is the interference power received by UE u when the lowest centralization level is in operation on its serving gNodeB, and function $r_y(\cdot)$ is defined as follows:

$$r_y(\mathbf{y}_g) = r(1) - \sum_{q=1}^{Q-1} \left(r(q-1) - r(q)\right) y_g^q, \tag{11}$$

such that $r_y(\mathbf{y}_g) = r(\mathbf{x}_g)$.

Formulation (P1) replaces the integer variables and discrete functions $r(\cdot)$ and $c(\cdot)$ of (P0) by binary variables and polynomial functions. As a result, linearization techniques can be now applied to improve the tractability of the FSSP. Namely, the product of two y_g^q variables can be linearized via the variable change $z_{g,k}^q = y_g^q y_k^q$, which can be enforced with additional linear inequalities [32]. This leads to the following reformulation:

$$\max_{\mathbf{y},\mathbf{z},\mathbf{f}} \sum_{u=1}^{U} \text{llog}\left(\frac{s_u}{\varsigma + I_u - \sum_{g=1}^{G} i_{u,g}\left(\sum_{q=1}^{Q-1} \delta(q) z_{g,h_u}^q\right)}\right) \tag{P2a}$$

subject to

$$2z_{g,k}^q \le y_g^q + y_k^q \qquad \forall q \in \mathbb{Q}, \forall g,k \in \mathbb{G}, g < k, \tag{P2b}$$

$$1 + z_{g,k}^q \ge y_g^q + y_k^q \qquad \forall q \in \mathbb{Q}, \forall g,k \in \mathbb{G}, g < k, \tag{P2c}$$

$$\mathbf{z} \in \{0,1\}^{(Q-1)\binom{G}{2}}, \tag{P2d}$$

(P0c) – (P0d) and (P1b) – (P1d),

where $\mathbb{Q} \triangleq \{1, ..., Q-1\}$ and $\mathbf{z} \triangleq \left[z_{g,k}^q\right]$ $\forall q \in \mathbb{Q}$, $\forall g,k \in \mathbb{G}$ such that $g < k$. The number of additional $\mathbf{z}$ variables is $(Q-1)\binom{G}{2} = O(G^2)$, as one additional variable is required for every pair of gNBs and consecutive splits. In Section IV-B3, we exploit the characteristics of the network to reduce the number of these additional variables.

Formulation (P2) is still a MINLP, but its simpler objective function admits further analysis. Indeed, we observe that the continuous relaxation of the objective function is convex on $\mathbf{z}$, but since the FSSP is a maximization problem, this implies that we are in the realm of concave optimization. Thus, there may be multiple local maxima, making the problem hard to tackle. It is still possible to use exact global optimization techniques for concave MINLPs, such as those presented in [33], but these mainly consist on applying branch-and-bound or branch-and-cut algorithms, whose convergence time may be high. Since the FSSP is a real-time problem, we opt instead for deriving increasingly simpler approximations to the original FSSP, until a suitable approach within the speed-quality trade-off is found.

B. Fractional approximations

The main obstacle when tackling formulations (P0)–(P2) is the $\mathrm{llog}(\cdot)$ function, which prevents the application of simplifying reformulations. Fortunately, we can exploit the slow growth rate of this function, as it can be very well approximated by a rational function:

$$\mathrm{llog}\,(\xi) \approx \frac{\alpha}{\beta+\xi} + \gamma. \tag{12}$$

Coefficients α, β and γ can be obtained from rational fitting within the desired interval. In our case, we choose the interval $0.1 \leq \xi \leq 100$, that is, an SINR ranging from -10 dB to 20 dB. After applying (12) to (P2) and some straightforward algebra, we obtain the following reformulation:

$$\min_{\mathbf{y},\mathbf{z},\mathbf{f}} \sum_{u=1}^{U} \frac{s_u}{\beta\,(\varsigma + I_u) + s_u - \beta \sum_{g=1}^{G} i_{u,g} \left(\sum_{q=1}^{Q-1} \delta(q) z^q_{g,h_u} \right)} \tag{P3a}$$

subject to (P0c)–(P0d), (P1b)–(P1d), and (P2b)–(P2d).

Problem (P3) is now a multiple-ratio fractional mixed-integer optimization problem, which can be tackled with state-of-the-art techniques. Indeed, since the continuous variables $\mathbf{f}$ do not appear on the objective function, we can directly apply existing techniques to reformulate it into an MILP. We present two such techniques: the Li-Wu-Tawarmalani transformation and the Borrero-Gillen-Prokopyev transformation.

1) Li-Wu-Tawarmalani transformation: The Li-Wu-Tawarmalani transformation (LWT transformation) reformulates a 0-1 multiple-ratio fractional program into an MILP by introducing continuous variables $\mathbf{w} \triangleq [w_u]$ and $\mathbf{v} \triangleq \left[v^q_{u,g}\right]$ via the following variable changes [34], [35], [36]:

$$w_u = \frac{1}{\pi_u - \beta \sum_{g=1}^{G} i_{u,g} \left(\sum_{q=1}^{Q-1} \delta(q) z^q_{g,h_u} \right)} \qquad \forall u \in \mathbb{U}, \tag{13}$$

$$v^q_{u,g} = w_u z^q_{g,h_u} \qquad \forall q \in \mathbb{Q}, \forall u \in \mathbb{U}, \forall g \in \mathbb{G}, \tag{14}$$

where $\pi_u = \beta\,(\varsigma + I_u) + s_u$. These identities can be enforced by additional constraints, resulting in the following equivalent formulation:

$$\min_{\mathbf{y},\mathbf{z},\mathbf{w},\mathbf{v},\mathbf{f}} \sum_{u=1}^{U} s_u w_u \tag{P4a}$$

subject to

$$\pi_u w_u - \beta \sum_{g=1}^{G} i_{u,g} \left(\sum_{q=1}^{Q-1} \delta(q) v^q_{g,h_u} \right) = 1 \qquad \forall u \in \mathbb{U}, \tag{P4b}$$

$$W^-_u z^q_{g,h_u} \leq v^q_{u,g} \leq W^+_u z^q_{g,h_u} \quad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \forall u \in \mathbb{U}, \tag{P4c}$$

$$v^q_{u,g} \leq w_u + W^-_u \left(z^q_{g,h_u} - 1 \right) \quad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \forall u \in \mathbb{U}, \tag{P4d}$$

$$v^q_{u,g} \geq w_u + W^+_u \left(z^q_{g,h_u} - 1 \right) \quad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \forall u \in \mathbb{U}, \tag{P4e}$$

(P0c) – (P0d), (P1b) – (P1d), and (P2b) – (P2d),

where $\mathbb{U} \triangleq \{1, ..., U\}$, and W^-_u and W^+_u are lower and upper bounds for w_u, respectively, which can be obtained trivially by setting the $\mathbf{z}$ variables in (13) to $\mathbf{0}$ and $\mathbf{1}$.

Formulation (P4) is an MILP that requires $U + 4(Q-1)GU$ new constraints, U additional $\mathbf{w}$ variables and $(Q-1)(G-1)U$ additional $\mathbf{v}$ variables with respect to (P3). Note that, in the general case, the number of required $\mathbf{v}$ variables would be $(Q-1)\binom{G}{2}U$ as these variables originate from the product of $\mathbf{z}$ and $\mathbf{w}$ variables. However, in our case it is clear that the interference coefficient corresponding to a triple (u, g, k), $u \in \mathbb{U}$, $g, k \in \mathbb{G}$ is zero unless $g = h_u$ or $k = h_u$ and $g \neq k$, hence we can remove the variables indexed by those triples. As a result, the number of variables of this reformulation grows with $O(G^2)$, assuming that U scales linearly with G [37], instead of $O(G^3)$.

2) Borrero-Gillen-Prokopyev transformation: The Borrero-Gillen-Prokopyev transformation (BGP transformation) is a recent improvement on the LWT transformation, which aims at reducing the number of required variables and constraints by approximating all coefficients in the objective function with integers [38]. This is accomplished by introducing the new variables $\mathbf{a} = [a_u]$ defined as:

$$a_u = \frac{\sigma}{\lambda_{u,0} + \sum_{g=1}^{G} \sum_{q=1}^{Q} \lambda^q_{u,g} z^q_{g,h_u}} \qquad \forall u \in \mathbb{U}, \tag{15}$$

where

$$\lambda_{u,0} = \left\lfloor \frac{\sigma\beta(n_u + I_u)}{s_u} \right\rceil + \sigma, \quad \lambda^q_{u,g} = -\left\lfloor \frac{\beta\sigma i_{u,g}\delta(q)}{s_u} \right\rceil,$$

σ is a constant factor used to scale the integer coefficients into the desired range, and $\lfloor\cdot\rceil$ represents the rounding operation to the nearest integer. In addition, new binary variables $\mathbf{b} = \left[b_{u,p}\right]$, $b_{u,p} \in \{0, 1\}$ are defined as:

$$\sum_{j=1}^{P_u} 2^{j-1} b_{u,p} = \Lambda_u + \sum_{g=1}^{G} \sum_{q=1}^{Q} \lambda^q_{u,g} z^q_{g,h_u} \quad \forall u \in \mathbb{U}, \forall p \in \{1, ..., P_u\}, \tag{16}$$

where $\Lambda_u = -\sum_{g=1}^{G} \sum_{q=1}^{Q} \lambda^q_{u,g}$ and $P_u = \left\lfloor \log_2(\Lambda_u) \right\rfloor + 1$. Finally, new variables $\mathbf{d} = \left[d_{u,j}\right]$ are defined as

$$d_{u,p} = b_{u,p} a_u \quad \forall u \in \mathbb{U}, \forall p \in \{1, ..., P_u\}. \tag{17}$$

The resulting reformulation, including additional constraints to enforce (15)–(17), is

$$\min_{\mathbf{y},\mathbf{z},\mathbf{a},\mathbf{b},\mathbf{d},\mathbf{f}} \sum_{u=1}^{U} a_u \tag{P5a}$$

subject to

$$(\lambda_{u0} - \Lambda_u) a_u + \sum_{p=1}^{P_u} 2^{p-1} b_{u,p} = -\sigma \qquad \forall u \in \mathbb{U}, \quad \text{(P5b)}$$

$$\sum_{g=1}^{G} \sum_{q=1}^{Q} \lambda_{u,g} z^q_{g,h_u} - \sum_{p=1}^{P_u} 2^{p-1} b_{u,p} = -\Lambda_u \quad \forall u \in \mathbb{U}, \quad \text{(P5c)}$$

$$a^L_u b_{u,p} \le d_{u,p} \le a^U_u b_{u,p} \qquad \forall u \in \mathbb{U}, p \in \mathbb{P}_u, \quad \text{(P5d)}$$

$$d_{u,p} - a_u \le a^L_u b_{u,p} - a^L_u \qquad \forall u \in \mathbb{U}, p \in \mathbb{P}_u, \quad \text{(P5e)}$$

$$d_{u,p} - a_u \ge a^U_u b_{u,p} - a^U_u \qquad \forall u \in \mathbb{U}, p \in \mathbb{P}_u, \quad \text{(P5f)}$$

(P0c) – (P0d), (P1b) – (P1d), and (P2b) – (P2d),

where $\mathbb{P}_u = \{1, ..., P_u\}$, $a^L_u = \frac{-\sigma}{\lambda_{u,0}}$ and $a^U_u = \frac{-\sigma}{\lambda_{u,0} - \Lambda_u}$.

Formulation (P5) is an MILP that requires $2U + 4\sum_{u\in\mathbb{U}} P_u$ new constraints, $U + \sum_{u\in\mathbb{U}} P_u$ additional continuous variables (**a** and **d**) and $\sum_{u\in\mathbb{U}} P_u$ additional binary variables (**b**) with respect to (P3). Since $\sum_{u\in\mathbb{U}} P_u = O(U \log(G))$ [38] and assuming again $U = O(G)$, this implies that the number of additional variables and constraints compared to (P3) grow with $O(G \log(G))$, at the expense of losing accuracy in the problem coefficients. Nonetheless, the overall size of the problem instances still grows with $O(G^2)$, due to the presence of variables **z**.

3) Punctured transformations: The fractional reformulation (P3) relies on the addition of **z** variables to be tractable by the LWT and the BGP transformations. These variables replace the product of **y** variables by single binary variables, which eventually enables these MILP reformulations. As there must be a $z_{g,k}$ variable for each pair of gNBs $[g, k]$, their number grows quadratically with the number of gNBs G.

However, in our problem not every pair of gNBs is worth considering. The interference between two gNBs that are far apart is negligible, so any variable modeling it contributes little to the overall solution. Knowing this fact, we can remove unnecessary variables so that the problem size is reduced without noticeably affecting the optimal solution. To do so, we define $j_{g,k}$ as the combined interference caused by gNBs g and k:

$$j_{g,k} = \sum_{u\in\mathbb{H}_g} i_{u,k} + \sum_{u\in\mathbb{H}_k} i_{u,g} \tag{18}$$

where $\mathbb{H}_g = \{u \mid h_u = g\}$ is the set of the UE indices served by gNB g. Now we sort coefficients $j_{g,k}$ and remove those gNB pairs $[g, k]$ whose combined interference is below a configurable threshold. For instante, in our experiments, we remove those gNB pairs with the smallest $j_{g,k}$ such that their addition contributes less than 5% to the total interference. This removes the related $z_{g,k}$ variables and all additional variables and constraints that are defined from them, hence simplifying the problem. Since removing these variables may impact the performance of the obtained solutions, we evaluate them separately for the LWT and BGP transformations, and refer to them as *punctured* LWT and BGP transformations, respectively.

C. Quadratic formulation

Instead of the approximation shown in (12), we can consider a simpler fractional approximation of the $\mathrm{llog}(\cdot)$ function:

$$\mathrm{llog}(\xi) \approx \alpha - \frac{\beta}{\xi}. \tag{19}$$

This approximation is less tight than (12), but in return it produces a much simpler problem formulation. Indeed, after combining (19) with (P1), we arrive at the following equivalent problem:

$$\max_{\mathbf{y},\mathbf{f}} \sum_{u=1}^{U} \sum_{g=1}^{G} \frac{i_{u,g}}{s_u} \left(\sum_{q=1}^{Q-1} \delta(q) y^q_g y^q_{h_u} \right), \tag{P6a}$$

subject to (P0c)–(P0d) and (P1b)–(P1d),

As this reformulation is a quadratic integer problem (QIP), we refer to it as the *quadratic formulation*. Its standard form enables the use of off-the-shelf solvers to tackle it. Nonetheless, its structure can be further exploited to reformulate it into a simple MILP. For this, we first introduce these new coefficients:

$$\epsilon^q_{g,k} = \begin{cases} \delta(q) \left(\sum_{u\in\mathbb{H}_g} \frac{i_{u,k}}{s_u} + \sum_{u\in\mathbb{H}_k} \frac{i_{u,g}}{s_u} \right) & \text{if } g \neq k, \\ 0 & \text{if } g = k, \end{cases} \quad \forall g, k \in \mathbb{G} \tag{20}$$

Finally, we introduce variables $\mathbf{t} = \left[t^q_g\right]$ via the following variable change:

$$t^q_g = y^q_g \left(\sum_{k=1}^{G} \epsilon^q_{g,k} y^q_k \right) \qquad \forall q \in \mathbb{Q}, g \in \mathbb{G}, \tag{21}$$

which can be enforced into our optimization problem by adding new constraints, as follows [39]:

$$\max_{\mathbf{y},\mathbf{t},\mathbf{f}} \sum_{q=1}^{Q-1} \sum_{g=1}^{G} t^q_g \tag{P7a}$$

subject to

$$0 \le t^q_g \le T^q_g y^q_g \qquad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \quad \text{(P7b)}$$

$$t^q_g \ge \sum_{k=1}^{G} \epsilon^q_{g,k} y^q_g - (1 - y^q_g) T^q_g \qquad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \quad \text{(P7c)}$$

$$t^q_g \le \sum_{k} \epsilon^q_{g,k} y^q_g \qquad \forall q \in \mathbb{Q}, \forall g \in \mathbb{G}, \quad \text{(P7d)}$$

(P0c) – (P0d) and (P1b) – (P1d),

where $T^q_g = \sum_{k=1}^{G} \epsilon^q_{g,k}$.

Formulation (P7) requires $(Q-1)G$ additional variables and $4GQ$ additional constraints with respect to formulation (P1). As a result, problem instances grow with $O(G)$, leading to substantially smaller problems when compared to the previous MILP formulations, which grow with $O(G^2)$. The drawback of this approach is the approximation (19) is less tight than (12), which may impact the quality of the solutions.

D. Heuristic approaches

The previous reformulations approximate the original problem into MILPs, which can be tackled by dedicated off-the-shelf software. Nonetheless MILPs are still NP-Hard, thus exact techniques to solve them may be slow, exhibiting exponential worst-case performance. Since we are interested in promptly solving the problem so as to lively adapt to changes in the interference situation, we propose two heuristics with the intention of finding good-quality solutions with low convergence time. The first technique exploits the correlation between the interference caused by a gNB and its centralization level. The second technique is a local-search approach to improve the solutions of the quadratic reformulation.

1) Heuristic 1: As the objective of centralizing gNBs is to mitigate interference, it intuitively holds that, given an optimal solution $\mathbf{x}^*$, the gNBs with the highest centralization levels may tend to be those causing the most interference. From this, we can derive a heuristic rule that assigns the centralization level of a gNB g based on the total interference $\hat{I}_g$ that it causes to all UEs in the absence of interference-mitigation:

$$\hat{I}_g \triangleq \sum_{u=1}^{U} i_{u,g}. \tag{22}$$

Such a heuristic must not only produce a solution that is proportional to $\hat{I}_g \ \forall g \in \mathbb{G}$, but it must also satisfy constraints (P0b)–(P0d) and be as centralized as possible.

We propose the following method to accomplish these objectives. First, we define the *accumulated centralization level* X given a (possibly infeasible) solution $\mathbf{x}$ as:

$$X \triangleq \sum_{g=1}^{G} x_g. \tag{23}$$

An upper bound $X^+ \geq X$ for all feasible $\mathbf{x}$ can be obtained by solving the following MILP:

$$X^+ = \max_{\mathbf{x},\mathbf{f}} \sum_{g=1}^{G} x_g \tag{24}$$

subject to (P0b)–(P0d). Note that (24) only depends on the fronthaul network configuration, therefore it can be solved offline before the network is put into operation. In addition, a lower bound $X^- \leq X$ for all feasible $\mathbf{x}$ can be also easily found, for example as:

$$X^- = \left\lfloor \frac{\min_{e\in\mathbb{E}}(\Phi_e)}{c(Q)} \right\rfloor, \tag{25}$$

which is the maximum number of fully-centralized gNBs that can be supported by the weakest link.

Now, given an initial guess of X within these bounds, we need a method to assign a value to each $x_g \ \forall g \in \mathbb{G}$ in accordance to the values of $\hat{I}_g$. We use the Webster/Sainte-Laguë method (W/S-L method) to do this, an algorithm originally designed to proportionally allocate seats in party-list voting systems [40]. This method is selected since it preserves the proportionality of the original interference levels better than alternative approaches [41], yet it is simple to implement. In a nutshell, the W/S-L method starts from $\mathbf{x} = \mathbf{0}$, finds the index g of the maximum interference $\hat{I}_g$, increments x_g by 1, updates I_g to $\frac{I_g}{2x_g+1}$ (or to $-\infty$ if $x_g = Q$) and repeats the process until the desired value of X is achieved. A more detailed algorithmic description of this method is included in Appendix A.

After running the W/S-L method, we have a candidate centralization vector $\mathbf{x}$ for the desired X. At this point, we can solve the following minimization problem to find the corresponding flow vector $\mathbf{f}$:

$$\min_{\mathbf{f}} \sum_{e=1}^{E} q_e \tag{26a}$$

subject to (P0b)–(P0d). Note that vector $\mathbf{x}$ is not present in the objective function, but in constraint (P0b). This problem is a linear program (LP) with E variables, and thus it can be tackled very efficiently by modern solvers. However, it may happen that our initial guess of X yields a vector $\mathbf{x}$ such that (26) is infeasible. In that case, we need to find a different value of X and try again until a feasible value of X is found. This process can be performed efficiently by exploiting the properties of our objective function and the W/S-L method via a binary search of the largest feasible X. This is explained in Appendix B.

2) Heuristic 2: This heuristic exploits the properties of the FSSP by using local search so as to improve solutions provided by a previous algorithm. We start with an initial solution $[\mathbf{x}, \mathbf{f}]$ provided by the quadratic approach and we compute the following parameters from it:

$$J_q = \frac{1}{|\mathbb{G}_q|} \sum_{g\in\mathbb{G}(q)} \hat{I}_g \tag{27}$$

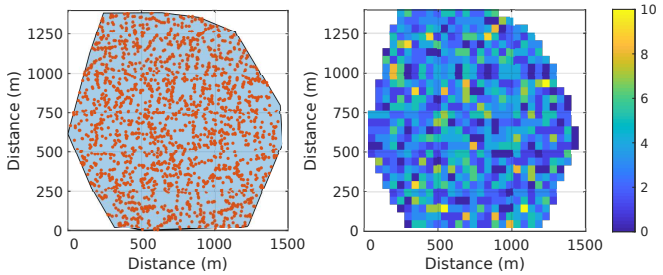
where $\mathbb{G}_q = \{g \in \mathbb{G} \,|\, x_g = q\} \ \forall q \in \mathbb{Q}$. The value of J_q is the average interference power caused by those gNBs whose centralization level is q. We then calculate deviations $\Delta\hat{I}_g = \hat{I}_g - J_{x_g} \ \forall g \in \mathbb{G}$, which represent how far the total interference caused by gNB g is from the average value among those with the same centralization level. Based on these deviations, we can identify two types of gNBs: those producing relatively high interference (which may benefit from a higher centralization level) and those producing relatively low interference (which might accept a lower centralization). Consequently, we select pairs of gNBs $(k, k') \in \{1, ..., Q-1\} \times \{2, ..., Q\}$ such that k belongs to the former type and k' belongs to the latter, and generate a new candidate solution $\mathbf{x}'$ whose elements are as follows:

$$x'_g = \begin{cases} x_g + 1 & \text{if } g = k, \\ x_g - 1 & \text{if } g = k', \\ x_g & \text{otherwise.} \end{cases} \tag{28}$$

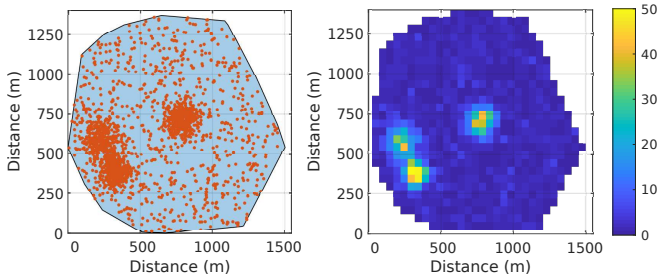
Then, the mean spectral efficiency $\widetilde{\eta}(\mathbf{x}')$ of the solution is evaluated. If $\widetilde{\eta}(\mathbf{x}') > \widetilde{\eta}(\mathbf{x})$, the feasibility of $\mathbf{x}'$ is evaluated with (26). If $\mathbf{x}'$ is both feasible and better than $\mathbf{x}$, it is taken as new initial solution and the procedure repeats until no better solution is found.

V. Experimental setup

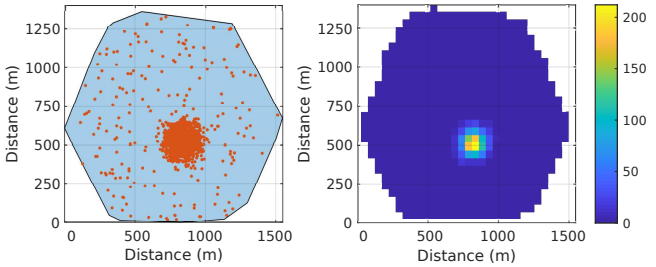
In this section, we present the setup used to evaluate the performance of the adaptation algorithms described in the last



(a) $\theta = 0.5$



(b) $\theta = 0.75$



(c) $\theta = 0.95$

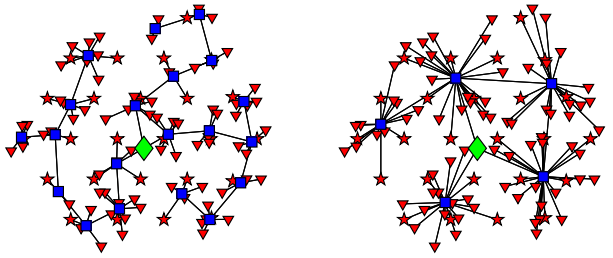
Figure 3: Visualization of UE concentration index for 2000 UEs. The red dots on the left figure represent the positions of each UE on the considered area, which is shown in light blue. The right figure is a color map of the 2-dimensional empirical distribution of the UEs, showing the UE density of each 50×50 m square bin.

section. We use a MATLAB simulator to create the interference coefficients $i_{u,g}$, required by all presented approaches, based on simulated UE and gNB positions. Then, we evaluate the algorithms on operator-grade hardware using a commercial optimization solver.

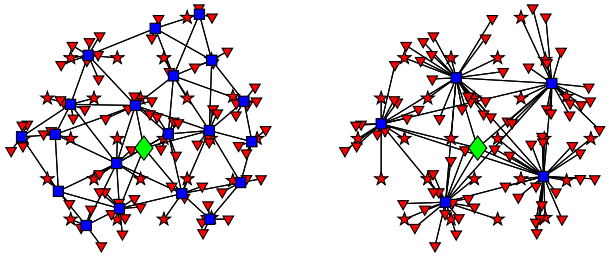
A. Simulated mobile coverage

In order to produce realistic instances of the FSSP, we follow the recommendations for simulating dense urban scenarios as described in 3GPP TS38.193 [37]. According to this specification, gNBs are divided into two layers: macro and micro layer. The macro layer follows an hexagonal layout with an inter-site distance of 200 m. In addition, the number of micro gNBs should be three times that of the macro gNBs. As a result, the average cell density is roughly 115 gNBs/km^2.

Regarding the UEs, its recommended number is 10 UEs per gNB on average, which results in 1150 UEs/km^2. Since the UE distribution may impact the performance of the algorithms, we derive a dedicate metric to model it. We divide the considered



(a) $\psi = 2,\ \nu = 5$ (b) $\psi = 2,\ \nu = 20$



(c) $\psi = 5,\ \nu = 5$ (d) $\psi = 5,\ \nu = 20$

Figure 4: Visualization of the average fronthaul network degree and gNB clustering parameters for the same gNB distribution. Black lines represent links, red stars represent macro DUs, red triangles are micro DUs, blue square are fronthaul switches, and green diamonds are CUs.

network area into square bins of side 50 m. Then, we count the number of UEs in each bin to compute its 2-dimensional empirical distribution. From this distribution, we compute its Gini coefficient and use it as the *UE concentration index* θ. It is observed that $\theta = 0.5$ corresponds to a uniformely-distributed random distribution of UEs, as it can be seen in Fig. 3a. Higher values correspond to higher UE clustering, (Fig. 3b and Fig. 3c), with a maximum value of $\theta = 1$ when all UEs are within the same bin. Once the position of the gNBs and the UEs is generated, we compute the received signal and interference power by using the log-distance path model for urban scenarios [42].

B. Fronthaul network

For our performance evaluation, we set the number of functional split options to $Q = 4$ (such as C-RAN, Intra-PHY, MAC-PHY, PDCP-RLC). Based on the analytical and experimental results shown in [26], we use the following interference-cancellation vector: $\mathbf{c} = [1, 0.6, 0.2, 0.01]$, such that there is no interference mitigation when using the lowest centralization level (as with PDCP-RLC) and a cancellation factor of 20 dB when using the highest centralization level. Note, however, that other values or split options are also possible, since they do not affect the problem formulation.

Regarding the fronthaul capacity vector, we use the values $\mathbf{r} = [4, 8, 80, 160]$ Gb/s, as provided in [21]. In order to simulate realistic fronthaul network layouts, we follow the descriptions provided in [15], which are based on real mobile networks on Italy, Romania, and Switzerland. Accordingly, we set the maximum link capacities to 0.5, 1 or 2 Tb/s (the higher value that makes full centralization infeasible, to prevent trivial

results). Furthermore, we simulate several types of fronthaul networks by controlling two parameters: the *average fronthaul network degree* ψ, defined as the ratio of links to fronthaul switches, and the gNB clustering parameter ν, defined as the average number of gNBs connected to the same switch. According to [15], the average degree of a typical fronthaul network ranges from $\psi = 2$ (a tree graph) to $\psi = 5$, and the gNB cluster ranges from $\nu = 5$ to $\nu = 20$. In Fig. 4 we show the some exemplary layouts resulting from varying these parameters.

C. Computing platform

After creating the interference coefficients and the fronthaul network with the MATLAB simulator, we have all the required components to run our adaptation algorithms. Since the convergence time of this algorithms is very relevant to decide on their viability, we use an operator-grade hardware platform consisting of 48 Intel Xeon E5 cores distributed over six computing servers [43]. As optimization solver, we use the commercial Gurobi software [44]. This software is able to divide large MILP instances and efficiently process them in parallel.

VI. Performance evaluation

In this section, a collection of comprehensive experimental results is shown. We first look into the convergence time of the proposed approaches, as this is a major factor limiting its applicability. We then evaluate and compare their performance, which, in combination with their convergence time, allow us to select the most appropriate approaches for each case. Finally, based on the observed results, we provide a strategy to implement a dynamic adaptation of the functional split.

We depict observed values of convergence time and spectral efficiency by means of boxplots, with the intention of representing their distribution. In order to provide as much information as possible in little space, we use a compact version of standard boxplots, whose interpretation is as follows: the central dot represents the median, the box contains the data between the first and third quartiles, and the whiskers extend to the lowest and highest values contained in 1.5 times the inter-quartile range. Occasionally, we show the (arithmetic) mean of the distribution as a horizontal bar. For the sake of clarity, outliers are not shown.

A. Convergence time

The applicability of the seven approaches presented in this work is heavily influenced by their convergence times, that is, the time required for the approaches to reach a (sub)optimal solution. This is due to the fact that the position and activity of the UEs are used as inputs of the optimization algorithms and not updated during the solving process. Consequently, it is possible that the solutions yielded by the algorithms are outdated if their convergence time is too long. Previous work has shown that mobile traffic is highly variable, and may often sharply deviate from average patterns [45]. In addition, in certain region types (such as entertainment or transport areas), even the average patterns may be fast-changing [46]. In fact, the analysis of recent mobile traffic traces show that the user traffic experienced by a 5G RAN may abruptly change in few minutes [47], [48]. As a result, approaches with long convergence times will not be useful for dynamic adaptation. In this work, we allow a maximum of 15 minutes for an algorithm converge to a solution, since solutions taking longer times are unlikely to be usable.

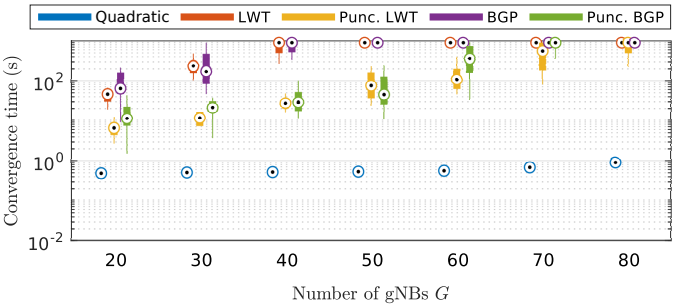


Figure 5: Time of convergence of the MILP reformulations.

In Fig. 5 we show the distributions of the convergence time of the five MILP reformulations (quadratic, LWT, BGP, and punctured LWT and BGP formulations) as a function of the number of gNBs. The solver is configured for a maximum running time of 15 minutes with a relative gap tolerance of 0.01% and each boxplot represents at least 100 runs. We observe that the convergence times of unpunctured LWT and BGP formulations reach the 15-minutes limit with only 40 gNBs, and with 50 gNBs all instances take longer or equal than this limit. Assuming a cell density of 115 gNBs/km^2, this implies that they may be suitable only for areas smaller than 0.4 km^2. The running time of the punctured LWT and BGP formulations is noticeably smaller, although the limit is once again reached with only 60 or 70 gNBs, corresponding to an area of ca. 0.6 km^2.

In contrast, the convergence time of the quadratic approach remains below 1 s for areas with less than 80 gNBs. In Fig. 6 we show an expanded range of gNBs and include Heuristics 1 and 2 in the comparison. We can see that the instances of the quadratic approach do not reach the 15-minutes limit until $G = 350$, and by $G = 400$ almost all instances hit this limit. This translates into an area of 3 to 3.5 km^2 and 3500 to 4000 simultaneously active UEs, which is enough to cover the densely populated centers of most cities. We can also observe that the convergence time of Heuristic 2 is approximately twice that of the quadratic approach, which is also expected. Finally, all instances of Heuristic 1 converge in less than 1 s for the whole shown range.

B. Performance evaluation

After evaluating the running time of the approaches, we measure their performance in terms of the achieved geometric mean of the spectral efficiency $\widetilde{\eta}(\mathbf{x})$, as it is defined in (6). We first evaluate $\widetilde{\eta}(\mathbf{x})$ for all considered approaches over a set of extreme cases so as to compare them. Then, we analyze the

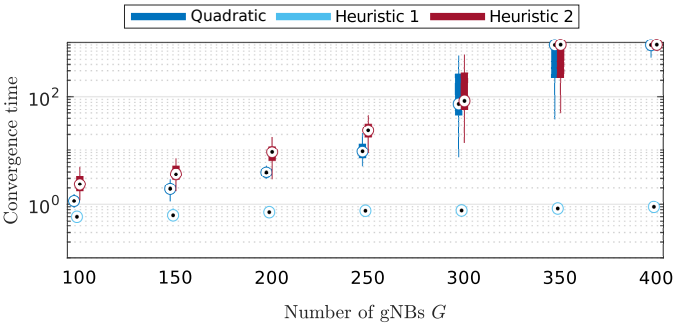
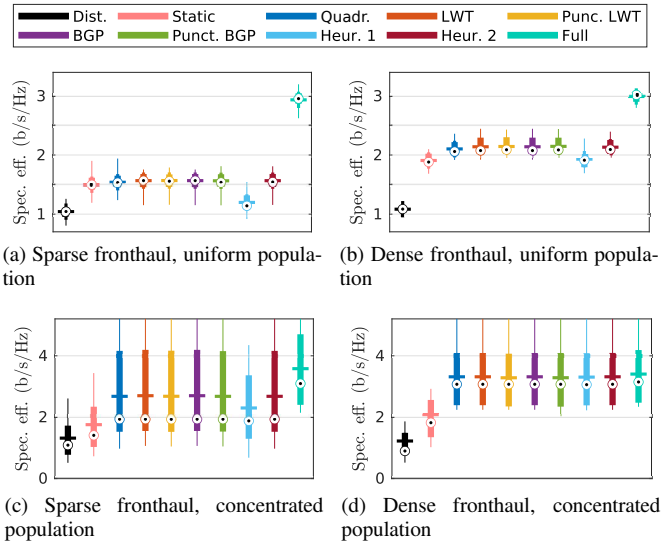


Figure 6: Time of convergence of the quadratic reformulation and Heuristics 1 and 2.



(a) Sparse fronthaul, uniform population

(b) Dense fronthaul, uniform population

(c) Sparse fronthaul, concentrated population

(d) Dense fronthaul, concentrated population

Figure 7: Distributions of the mean spectral efficiency $\widetilde{\eta}(\mathbf{x})$ achieved by the proposed approaches, a fully distributed approach, a static approach, and a fully centralized approach in four extreme scenarios.

impact of the configuration of the fronthaul network and the user concentration.

1) Approach comparison: In order to compare the performance of all proposed approaches, we evaluate their spectral efficiency $\widetilde{\eta}(\mathbf{x})$ on the same scenario. Since the LWT and BGP formulations are only applicable to small networks, we choose $G = 50$ gNBs for this comparison, corresponding to $U = 500$ UEs and an area of approximately 0.4 km^2. We now generate four types of scenarios to cover a wide range of interference cases: (i) a sparse fronthaul ($\psi = 2$) with uniform population ($\theta = 0.5$); (ii) a dense fronthaul ($\psi = 5$), with uniform population ($\theta = 0.5$); (iii) a sparse fronthaul ($\psi = 2$), with concentrated population ($\theta = 0.95$); and (iv) a dense fronthaul ($\psi = 5$), with concentrated population ($\theta = 0.95$).

The simulation results after 200 runs are shown in Fig. 7, with scenarios (i)–(iv) being used in Fig. 7a–7d, respectively. Apart from the proposed approaches, we also include in the comparison the spectral efficiencies of a fully distributed solution ($x_g = 0 \ \forall g \in \mathbb{G}$), a static solution (in which the optimal x is precomputed for a uniform population), and a fully centralized solution ($x_g = Q \ \forall g \in \mathbb{G}$). The fully distributed solution represents a network in which all processing is distributed, thus providing a lower bound to all solutions. The static solution is the one proposed in works such as [15], [17], in which the functional split of every gNB is calculated for the average traffic and not adapted to the instantaneous interference situation. The fully centralized solution, in which all gNBs are centralized, is infeasible in all cases, but it serves as a upper bound for the other approaches.

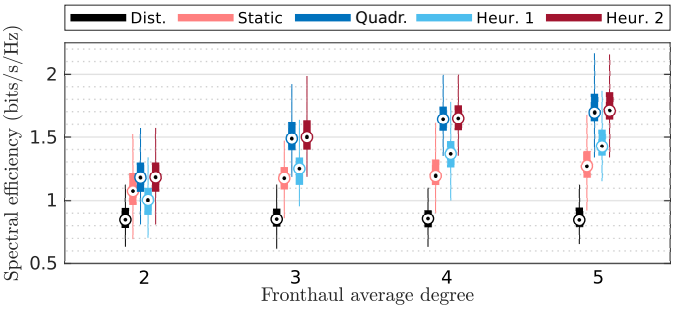


Figure 8: Average spectral efficiency achieved by the quadratic approach, Heuristics 1 and 2, a static and a fully distributed network for $G = 300$ and a UE concentration index of 0.75 as the fronthaul average degree varies.

From these results we can observe two general trends. First, the denser the fronthaul network, the higher the spectral efficiency of all solutions, but also the better the performance of our proposed solutions with respect to the static solutions. This trend is discussed in more detail in Section VI-B2. Second, the more concentrated the population, the higher the variance and the mean spectral efficiency achieved by our solutions with respect to the static solutions. This phenomenon is explored in more detail in Section VI-B3. As a result, when the fronthaul is dense and the population is concentrated, our approaches achieve similar spectral efficiencies to that achieved by full centralization.

Apart from these general trends, we can compare the performance of our proposed approaches. We can conclude that, with the exception of Heuristic 1, all these approaches perform very similarly in all cases. Under close examination, we can see that the best performance is achieved by LWT and BGP transformations and Heuristic 2, followed closely by the quadratic approach and the punctured LWT and BGP transformations. Nonetheless, the maximum difference in their average spectral efficiency is less than 2% in all scenarios.

Given their good-quality solutions and their fast convergence time, we conclude that the quadratic approach and Heuristic 2 are the most efficient approaches, and hence the most suitable for actual deployments. Heuristic 1 may be still adequate for large networks or medium-sized networks in which convergence time needs to be very short. Similarly, the punctured and unpunctured LWT and BGP transformations may be applicable to small networks in which maximizing the spectral efficiency is of utmost importance, such as industrial or ultra-reliable networks, or for medium-size networks with slow adaptation rates.

2) Impact of the fronthaul network: In Fig. 7 we can see how the density of the fronthaul network affects the quality

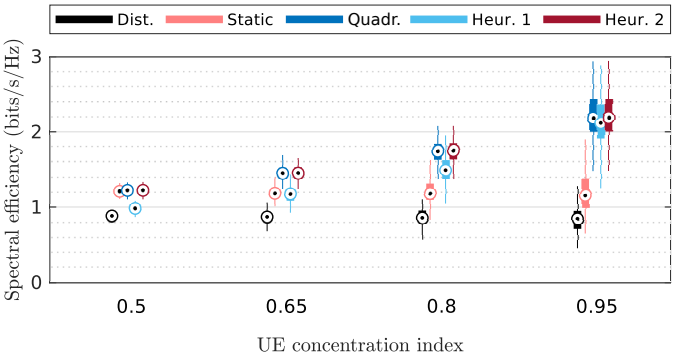
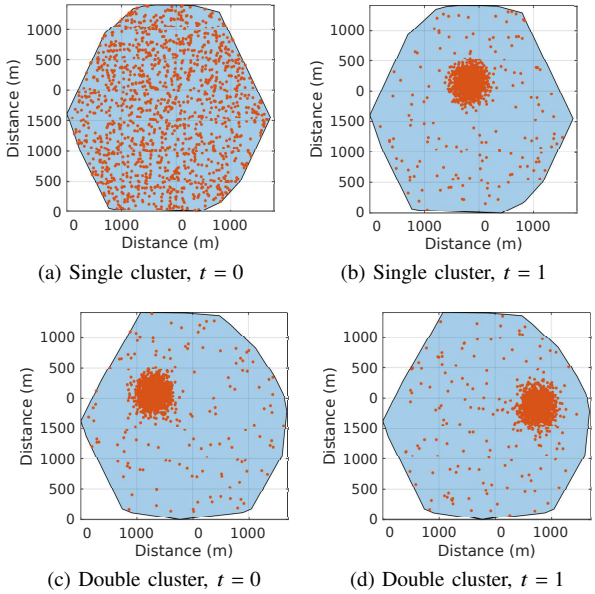


Figure 9: Average spectral efficiency achieved by the quadratic approach, Heuristics 1 and 2, a static and a fully distributed network for $G = 300$ and a fronthaul average degree of 3.5 the fronthaul average degree varies as the UE concentration index varies.



(a) Single cluster, $t = 0$ (b) Single cluster, $t = 1$

(c) Double cluster, $t = 0$ (d) Double cluster, $t = 1$

Figure 10: Examples of UEs positions for the single-cluster scenario (a and b) and double-cluster scenario (c and d) at normalized times $t = 0$ and 1.

of the solutions achieved by all approaches for a deployment with $G = 50$ gNBs. When using a sparse fronthaul network of $\psi = 2$, few centralization vectors are feasible because of the limited number of paths, which leads to mean optimal spectral efficiencies of around 1.57 b/s/Hz for dispersed UEs and 2.14 b/s/Hz for concentrated UEs. With a denser fronthaul network of $\psi = 5$, the average spectral efficiency grows up to around 2.7 b/s/Hz for dispersed UEs and 3.31 b/s/Hz for concentrated UEs.

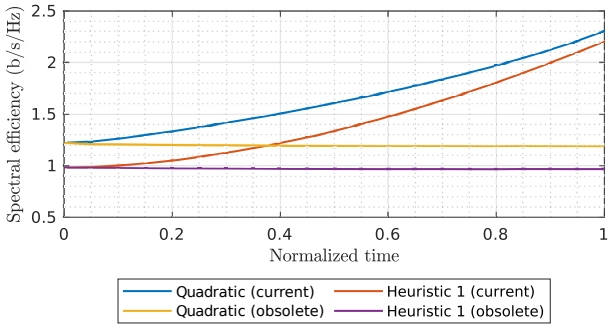
With the intention of observing this trend more clearly, we perform another experiment on a larger network ($G = 300$, 2.6 km^2), with partially concentrated UEs ($\theta = 0.75$) and let the density of the fronthaul vary from $\psi = 2$ to $\psi = 5$. The results are shown in Fig. 8. We conclude that with sparse, tree networks ($\psi = 2$) the benefits of implementing an adaptive solution are marginal, improving only from 1.07 b/s/Hz (static solution) to 1.18 b/s/Hz (Heuristic 2), a 10% improvement. With an average degree of $\psi = 3$ this improvement increases up to 28% (Heuristic 2), and with $\psi = 5$ it reaches 36% (Heuristic 2).

3) Impact of UE concentration: Implementing an adaptive functional split is specially beneficial when the UEs tend to be clustered in time-varying clusters. This can be observed once again in the experiment shown in Fig. 7: as the UE concentration index changes from $\theta = 0.5$ to $\theta = 0.95$, the average spectral efficiency increases from 1.57 b/s/Hz to 2.7 b/s/Hz for $\psi = 2$, and from 2.14 b/s/Hz to 3.31 b/s/Hz for $\psi = 5$. Conversely, the average spectral efficiency of the static solution barely changes. This is due to the fact that, when UEs are concentrated around the same spots, an adaptive network can mitigate their interference more efficiently than when they are spread apart.

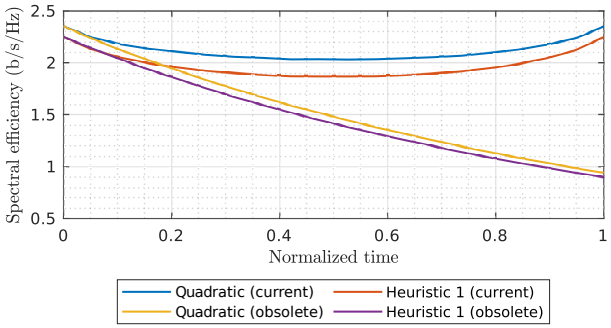
In order to evaluate the effect of UE concentration in more detail, we perform another experiment on a larger network ($G = 300$, 2.6 km^2) with a constant fronthaul average degree of $\psi = 3.5$ and let the UE concentration index vary from $\theta = 0.5$ to $\theta = 0.95$. The results are shown in Fig. 9. We conclude that an adaptive approach may achieve substantially better spectral efficiency when UEs are concentrated with respect to a static solution. Indeed, when $\theta = 0.8$, the mean spectral efficiency can be improved from 1.18 b/s/Hz to 1.75 b/s/Hz, a 48% improvement. For $\theta = 0.95$, this improvement reaches almost 90%. Interestingly, the mean spectral efficiency of a static solution is barely affected by the UE concentration, although its variance does increase. This is because clusters form at any point of the covered area with equal probability, combined with the fact that the static solution explicitly optimizes for the average UE position, regardless of the instantaneous UE concentration. Finally, note also that even if Heuristic 1 is used, a precomputed static solution can be always available and used instead if its performance is better. Thus no performance degradation should be expected from implementing an adaptive approach.

C. Dynamic adaptation strategy

Thus far we have presented and evaluated the performance of multiple approaches to select the optimal functional split of a RAN, emphasizing on the adaptive nature of the problem being solved. However, it is not stated how frequently the adaptation algorithm should run so as to adapt to a changing interference situation. There are two naive approaches to decide on this update frequency. One possibility is to run the adaptation algorithm continuously, that is, restart it with updated information immediately after convergence. This would guarantee that the network always operates with the best known centralization vector, but it may be costly for the operator, as the CPU utilization of the optimization servers would always be close to 100%. Another option is to run the adaptation algorithm periodically, based on known traffic patterns, as those shown in [45], [46]. This is a more resource-efficient approach, but it partially defeats the purpose of implementing an adaptive system. That is, unpredicted UE



(a) Single-cluster scenario



(b) Double-cluster scenario

Figure 11: Average spectral efficiency achieved by current and obsolete (initial) solutions of the quadratic approach and Heuristic 1 for the single- and double-cluster scenarios with $G = 300$ and average fronthaul degree 3.5

concentrations or unusual patterns would be ignored, even though the network has the ability to adapt.

We propose a better alternative to both methods that exploits the similarities between Heuristic 1 and the quadratic approach. It is observed that, although Heuristic 1 often yields poor-quality solutions with respect to the other algorithms, its behavior (regarding when and how solutions change) remains similar to any of the other algorithms. In order to show this, we present two additional experiments representing two extreme scenarios. In the first, single-cluster experiment, we vary the UE concentration between uniform and clustered as (normalized) time passes from $t = 0$ to $t = 1$, as shown in Fig. 10a and 10b. In the second, double-cluster experiment, we use vary the UE population between an initial cluster and a final cluster, as shown in Fig. 10c and 10d. For both experiments, the performance of the optimal solutions provided by the quadratic formulation and Heuristic 1 is recorded for each time increment. Since these are the best solutions that each approach can provide for the current scenario, we refer to them as *current* solutions. In addition, we store the solutions provided by both approaches when $t = 0$ and record their performance for every new UE position between $t = 0$ and $t = 1$, as if the network did not adapt during the experiments. We refer to these initial, outdated solutions as *obsolete* solutions.

In Fig. 11, we show the geometric mean of the spectral efficiencies achieved by the current and obsolete solutions, after 500 runs of the single-cluster and double-cluster experiments. We observe that, although there is always a performance gap between the quadratic formulation and Heuristic 1, their solutions corresponding evolve similarly. In the single-cluster experiment (Fig. 11a), current solutions tend to yield better performance in a similar fashion, whereas obsolete solutions remain almost constant for both approaches. The latter behavior is consistent with the performance of a static solution shown in Fig. 9, as the formation of a cluster of UEs brings an opportunity for enhanced interference mitigation, which a static approach misses. In the double-cluster experiment (Fig. 11b), obsolete solutions optimize for a disappearing cluster while ignoring the one appearing, leading to a steep performance degradation. Also in this case current and obsolete solutions behave similarly for both approaches. As a consequence, we can exploit the low running time of Heuristic 1 and run it frequently to serve as a predictor of a network change by comparing it with the performance of previous solutions. Once a solution change has been detected, a better adaptive algorithm, such as the quadratic approach, can be used to decide on the optimal functional split.

VII. Cost analysis

The main objective of this work is to find a fast and accurate optimization approach to select a throughput-maximizing functional split, so that it can be used in a dynamically-adapting RAN. Nonetheless, the viability of such a RAN is unclear unless the cost of finding, applying, and operating the selected functional splits is taken into account. In this section, we assess the deployment and operating costs of implementing a throughput-maximizing approach and compare it to alternative approaches, such as a fully distributed RAN, a fully centralized RAN or a static RAN in which the operating cost is minimized. Moreover, we briefly discuss the cost implications of a dynamic adaptation of the functional split.

A. Deployment cost

In order to be able to implement a dynamic functional split, both CU and DUs need to have the capacity to host those functions whose placement can be configured. As a result, there is always redundant, unused processing capacity on either the CU or the DUs for every possible configuration. Compared to fully distributed or fully centralized static configurations, this may lead to higher operating costs, which are investigated in the next section, and also to higher deployment costs. Conversely, the main advantage of a dynamic functional split is the possibility to reuse the fronthaul (or backhaul) network while still allowing for partial centralization. Since deploying a new fronthaul network can be very expensive for operators [49], the cost of a well-planed dynamic RAN may still be substantially lower than that of a fully centralized RAN, while offering higher data rates than a fully distributed RAN.

According to [50], the cost of deploying a distributed gNB is $\$50,000$ per macro cell and $\$20,000$ per micro cell, whereas deploying a centralized gNB costs $\$25,000$ for a macro cell and $\$10,000$ per micro cell. Assuming that there are three times more micro cells than micro cells (as recommended in 3GPP TS38.193 [37]), this results in an average deployment

cost of $\$27,500$ for a distributed gNB and $\$13,750$ for a centralized gNB. In addition, the estimated deployment cost of the data center to host the CU is $\$40,000$. Finally, an operator implementing a fully centralized RAN would probably need to deploy a new optical fronthaul network in order to support the large capacity requirements of C-RAN [49]. In [50], the cost of deploying such a network is estimated as $\$100,000$ per kilometer[2]. As a result, we can estimate the deployment cost $K_{\text{D-RAN}}$ of a fully distributed RAN as:

$$K_{\text{D-RAN}} = \$27,500 \cdot G, \tag{29}$$

and the deployment cost $K_{\text{C-RAN}}$ of a fully centralized RAN as:

$$K_{\text{C-RAN}} = \$40,000 + \$13,750 \cdot G + \$100,000 \cdot \zeta, \tag{30}$$

where ζ is the total length (in km) of the fronthaul network.

A RAN implementing a dynamic functional split can be regarded as a hybrid between full centralization and full distribution, thus its deployment cost can be calculated as follows. Firstly, since the DUs need to have enough capacity to host all functions, their cost should also be $\$27,500$ per gNB, as in a fully distributed RAN. Secondly, the data center hosting the CU is still needed, so that its cost has to be included. Finally, although the fronthaul network does not need to be replaced, some extensions and optimizations may be still required in order to leverage its full capacity. The estimated cost of these extensions is $\$5,000$ per kilometer, according to [50]. As a consequence, we estimate the cost $K_{\text{Dyn-RAN}}$ of deploying a dynamic RAN as:

$$K_{\text{Dyn-RAN}} = \$40,000 + \$27,500 \cdot G + \$5,000 \cdot \zeta, \tag{31}$$

We conclude from this expression that the deployment cost of a dynamic RAN is always higher than that of a fully distributed RAN, although the superior performance of the former may render it more profitable. Furthermore, since the fronthaul network can be reused, the cost of a dynamic RAN can be substantially lower than that of a centralized RAN.

In order to illustrate this, we consider three different RAN sizes, $G = \{150, 300, 450\}$, that use a fronthaul network of average degree of $\psi = 3.5$ and $\nu = 10$, that is, a medium-density fronthaul network. After simulating this scenario in the same conditions as described in Sec. V, we observe that the average taxicab lengths of the resulting fronthaul networks are $\zeta = \{31.6, 60.4, 88.9\}$ km. If we feed these input parameters into (29), (30) and (31), we obtain the values shown in Table I. We conclude that for this example scenario, the deployment cost of a dynamic RAN is around 4.5% higher than that of a distributed RAN, whereas the cost of a centralized RAN is around 25% higher. This suggests that a dynamic RAN can indeed compete with distributed and centralized options in terms of deployment cost.

G	$K_{\text{D-RAN}}$	$K_{\text{C-RAN}}$	$K_{\text{Dyn-RAN}}$
150	$4,125,000	$5,263,000	$4,323,000
300	$8,250,000	$10,205,000	$8,592,000
450	$12,375,000	$15,118,000	$12,860,000

Table I: Estimated deployment cost of a fully distributed RAN, a fully centralized RAN, and a RAN implementing a dynamic functional split.

B. Operating cost

We use the model provided in [51] for the operating cost of a RAN implementing a configurable functional split, which consists of three components: (i) the cost of instantiating mobile functions at the CU or DU, (ii) the computational costs of running these functions, and (iii) the routing costs. We refer to the first component as κ_1, which can be calculated as:

$$\kappa_1 = (\upsilon_{\text{CU}} + \upsilon_{\text{DU}}) \cdot G, \tag{32}$$

where υ_{CU} and υ_{DU} are the costs of providing resources for a single gNB at the CU and DU, respectively. Based on the data presented at [51], we assume values of $\upsilon_{\text{DU}} = 1$ ncu and $\upsilon_{\text{CU}} = 0.5$ ncu, where ncu stands for "normalized cost units". The second component, denoted by $\kappa_2(\mathbf{x})$, can be computed as:

$$\kappa_2(\mathbf{x}) = \sum_{g=1}^{G} \left(\tau_{\text{CU}}(x_g)\phi_{\text{CU}} + \tau_{\text{DU}}(x_g)\phi_{\text{DU}}\right)\mu_g, \tag{33}$$

where μ_g is the instantaneous user data rate experienced by gNB g, τ_{CU} (τ_{DU}) is the CPU cycles per Gb/s required to cope with user traffic at the CU (DU) when using split x_g, and ϕ_{CU} (ϕ_{DU}) is the relative cost per CPU cycle at the CU (DU). Values for μ_g can be estimated directly from the number of active users, as shown in [52], and values for $\tau_{\text{CU}}(x)$, $\tau_{\text{DU}}(x)$, ϕ_{CU}, and ϕ_{CU} are taken directly from [51]. Function $\tau_{\text{CU}}(x_g)$ is not linear in $\mathbf{x}$, but it can be redefined as a linear function of $\mathbf{y}_g$, with slight abuse of notation:

$$\tau_{\text{CU}}(\mathbf{y}_g) = \tau_{\text{CU}}(0) + \sum_{q=1}^{Q-1} \varepsilon_{\text{CU}}(x_g) y_g, \tag{34}$$

where $\varepsilon_{\text{CU}}(x) = \tau_{\text{CU}}(x+1) - \tau_{\text{CU}}(x)$. The same derivation applies to $\tau_{\text{DU}}(x_g)$, so that we can redefine $\kappa_2(\mathbf{x})$ as $\kappa_2(\mathbf{y})$. Finally, the routing costs can be calculated as:

$$\kappa_3(\mathbf{f}) = \sum_{g=1}^{G} \sum_{e \in \mathbb{E}^+(n_g)} \omega f_e^g, \tag{35}$$

where ω is the average normalized cost per Gb/s over all links.

Cost components κ_1, $\kappa_2(\mathbf{y})$, and $\kappa_3(\mathbf{f})$ can be used to calculate the total operating cost $\kappa(\mathbf{y}, \mathbf{f})$ of the solutions provided by our proposed approaches as follows:

$$\kappa(\mathbf{y}, \mathbf{f}) = \kappa_1 + \kappa_2(\mathbf{y}) + \kappa_3(\mathbf{f}). \tag{36}$$

Moreover, we can use this expression to figure out a cost-optimal solution, i. e., a solution that minimizes the operating cost, by using it as the objective function of a new optimization problem:

$$\min_{\mathbf{y},\mathbf{f}} \kappa_1 + \kappa_2(\mathbf{y}) + \kappa_3(\mathbf{f}) \tag{P8a}$$

subject to (P0c)–(P0d) and (P1b)–(P1d).

Problem (P8) is equivalent to that presented in [51], using an edge formulation instead of a path formulation for flow

[2]Note that this is a rough estimate of the actual cost, since it does not take into account the network topology, legal fees, etc.

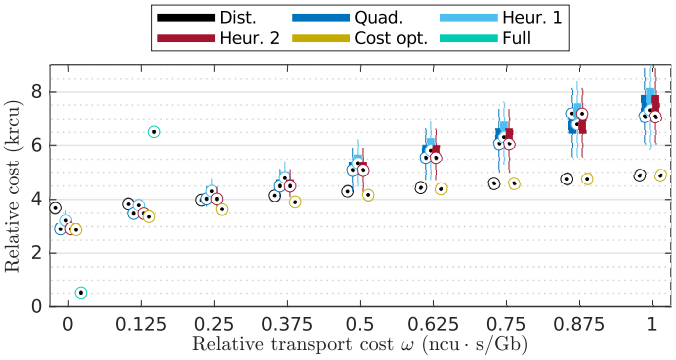


Figure 12: Relative operating cost achieved by fully distributed, fully centralized, quadratic, cost-optimal, and heuristic approaches when $G = 300$, average fronthaul degree $\psi = 3.5$, and UE concentration index $\theta = 0.75$.

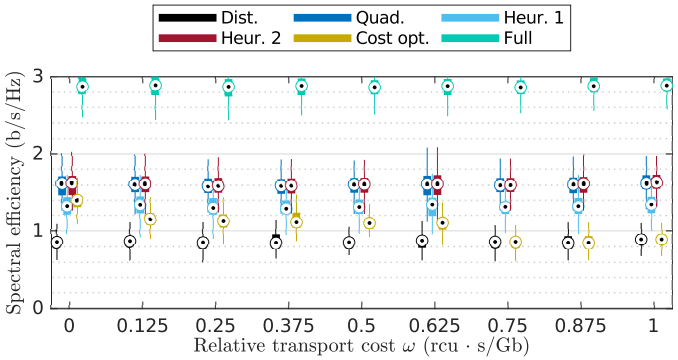


Figure 13: Average spectral efficiency achieved by fully distributed, fully centralized, quadratic, cost-optimal, and heuristic approaches when $G = 300$, average fronthaul degree $\psi = 3.5$, and UE concentration index $\theta = 0.75$.

modeling. In addition, it is an MILP of the same form as (P7), our quadratic formulation, so that it can be solved using the same methods in comparable time.

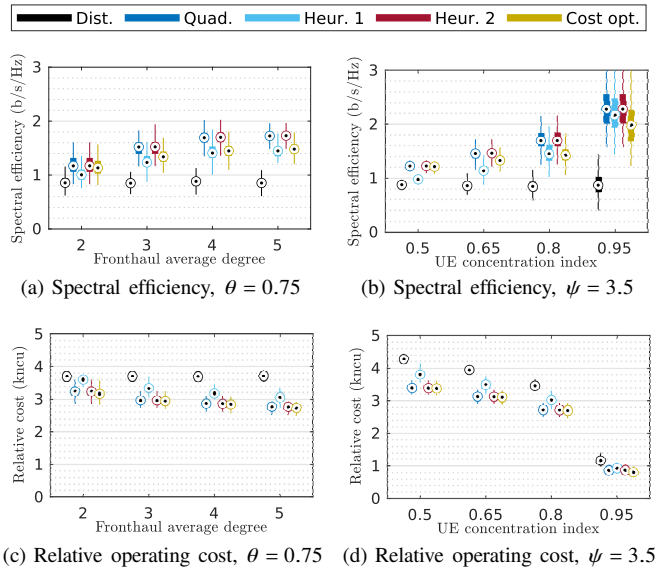
Although the FSSP and problem (P8) have different objectives, their solutions may not be radically different. Indeed, there may be a positive correlation between cost minimization and throughput maximization as a result of the influence of computational costs $\kappa_2(\mathbf{y})$. Since the cost of running mobile functions at the CU is much smaller than at the DU ($\phi_{\text{CU}} = 0.017$ ncu/cycle vs. $\phi_{\text{DU}} = 1$ ncu/cycle, according to [51]), minimizing this cost component also entails centralizing as many functions as possible, which is also desired by the throughput-maximization approach. Nonetheless, the routing costs $\kappa_3(\mathbf{f})$ have the opposite effect, since function centralization increases network usage. As a result, the value of ω greatly influences how exceedingly costly a throughput-maximizing solution is, and how much throughput is lost by a cost-optimal solution.

Fig. 12 shows the impact of ω on the operating cost of solutions provided by the quadratic approach and heuristics 1 and 2, along with that of the cost-optimal solution from (P8). For reference, we also include the cost of fully distributed and fully centralized networks, although the latter is always infeasible. We observe that when $\omega = 0$, the cost of our proposed approaches is very similar to that of the cost-optimal approach. In fact, the average cost difference between the cost-optimal and quadratic approaches is less than 1%. As ω grows, the cost-optimal solution converges to the distributed solution, since centralization incurs in high routing costs. Conversely, the cost of our proposed approaches increases linearly with ω, as this parameter is not taken into account for solution selection. As a consequence, at $\omega = 0.5$, the average cost difference between the cost-optimal and quadratic approaches is 22%, and if $\omega = 1$ this difference increases up to 45%.

In order to make a meaningful analysis, the cost difference between a cost-optimal and a throughput-maximizing approach must be evaluated against the possible additional revenue of the latter, resulting from being able to operate with higher spectral efficiency. In Fig. 13 we show the influence of the routing costs ω on the spectral efficiencies achieved by the same approaches as in Fig. 12. We observe that the cost-optimal approach always achieves noticeably lower spectral efficiencies than throughput-maximizing approaches, as we might expect. At $\omega = 0$, our quadratic approach achieves a 15% higher spectral efficiency than the cost-optimal approach, this increases to 45% at $\omega = 0.5$, and finally to 86% for $\omega \geq 0.75$, as the cost-optimal approach converges to the distributed solution. We conclude that the additional cost of throughput-maximizing approaches translates into proportionally higher improvements in the spectral efficiency. If the operator is able to profit from these improvements, then throughput-maximizing approaches may lead to a higher revenue with respect to static or cost-optimal approaches.

With the intention of providing a detailed cost analysis, we also study how the fronthaul network density and the UE concentration influence the trade off between spectral efficiency and cost in two scenarios. First, we investigate the case where routing costs are negligible ($\omega = 0$), and thus the operator is motivated to centralize as many functions as possible, either pursuing minimal cost or maximum throughput. The results for this scenario are depicted in Fig. 14. We observe that, in this case, higher fronthaul densities lead to higher spectral efficiencies (Fig. 14a) and lower costs (Fig. 14c) for all approaches, since more functions can be maximized. Nevertheless, our proposed throughput-maximizing approaches take more advantage of dense fronthaul networks, achieving better spectral efficiency than the cost-optimal approach while having comparable cost. Indeed, when $\psi = 5$, the quadratic approach achieves a 16.5% higher spectral efficiency while being only 1.3% more costly than the cost-optimal approach. A similar trend can be observed as the UEs become more concentrated: if $\theta = 0.95$, the quadratic approach achieves a 15.1% higher spectral efficiency while being only 6.3% more costly.

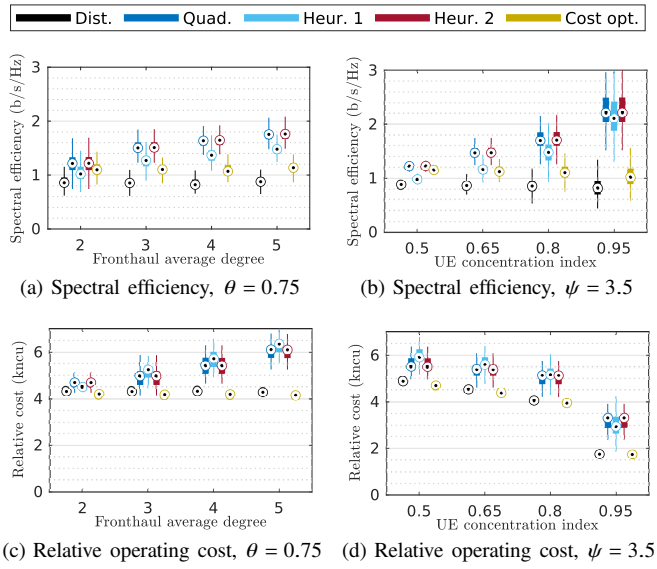
Finally, we examine a scenario where routing costs are not negligible, but still low enough so that a fully distributed configuration is not the least costly option. We set the routing cost to $\omega = 0.5$ ncu·s/Gb to illustrate this scenario, which is the midpoint between the value where the throughput-maximizing approach becomes more costly than the fully distributed configuration ($\omega \approx 0.25$ ncu·s/Gb), and the



(a) Spectral efficiency, $\theta = 0.75$ (b) Spectral efficiency, $\psi = 3.5$

(c) Relative operating cost, $\theta = 0.75$ (d) Relative operating cost, $\psi = 3.5$

Figure 14: Spectral efficiency and relative operating cost achieved by fully distributed, quadratic, cost-optimal and heuristic approaches when $G = 300$ and $\omega = 0$ ncu·s/Gb.



(a) Spectral efficiency, $\theta = 0.75$ (b) Spectral efficiency, $\psi = 3.5$

(c) Relative operating cost, $\theta = 0.75$ (d) Relative operating cost, $\psi = 3.5$

Figure 15: Spectral efficiency and relative operating cost achieved by fully distributed, quadratic, cost-optimal and heuristic approaches when $G = 300$ and $\omega = 0.5$ ncu·s/Gb.

value where the cost optimal approach converges to the fully distributed configuration ($\omega \approx 0.75$ ncu·s/Gb). The results are depicted in Fig. 15. For this scenario, we observe that the cost and spectral efficiency gaps between the cost-optimal and throughput-maximizing approaches are noticeably larger than in the previous scenario, since the former is now very limited by the routing costs. When $\psi = 5$, the quadratic approach achieves a 54.4% higher spectral efficiency (Fig. 15a) while being 47.5% more costly than the cost-optimal approach (Fig. 15c). A similar trend can be observed as the UEs become more concentrated: if $\theta = 0.95$, the quadratic approach achieves a 116.7% higher spectral efficiency (Fig. 15b) while being 90.7% more costly (Fig. 15d). These results suggest that, when the routing costs as high, throughput-maximizing approaches can only compete with the cost-optimal one as long as the higher spectral efficiency translates into higher profit.

C. Profitability in dynamic conditions

In the previous section, we show that throughput-maximizing approaches are able to deliver instantaneous network configurations that may be more profitable than other approaches, such as static or cost-optimal configurations. However, the changing nature of mobile networks still poses significant challenges at the task of evaluating the profitability of a dynamically-managed network. These challenges are discussed in [53], where the costs of operating such a flexible network are divided into two categories: *readiness cost*, which reflects the cost of operating in stable conditions, and *action cost*, which denotes the cost of sensing network changes and adapting the network configuration accordingly.

In order to carry out a meaningful cost analysis of a dynamic network, a large number of variables has to be taken into account. In this work we already consider some of them, such as network size, fronthaul density, UE distribution, and split-selection algorithm. Nonetheless, there are other aspects which still play a role, such as the type of covered area (residential, business, entertainment), UE mobility patterns, and a model of the performance degradation of the current configuration [54]. As a consequence, the complete analysis of the cost and potential profitability of a 5G network implementing a dynamic functional split is out of the scope of this paper and the matter of a separate future work.

VIII. Conclusion

The ambitious performance objectives of 5G regarding user data rates force operators to increase cell density. Nonetheless, increased cell density causes additional interference, which may be countered by means of centralized RAN architectures. The feasibility of these architectures is limited by the capacity of the fronthaul network, which frequently can only support partial function centralization, leading to the so-called functional split. The functional split affects the interference-mitigation capabilities of the network as well as the required fronthaul capacity, so that it has to be carefully chosen to ensure an efficient network utilization.

In this work, we tackle the problem of dynamically selecting the functional split of all gNBs in the network according to the instantaneous interference situation. We formulate it as a mixed-integer non-linear program and derive reformulations, approximations, and heuristics to enable solving it during runtime. We comprehensively evaluate each approach on operator-grade hardware using realistic network parameters obtained from 3GPP specifications and measurement traces. We also assess the impact of the fronthaul network topology and the UE concentration on the quality of the solutions. We observe that adaptive approaches perform best when the fronthaul network is dense and when the UEs are concentrated, although they outperform static alternatives in all scenarios. In conclusion, we show that adaptive architectures may yield average spectral

efficiencies up to 90% higher than those of static architectures while being comparable to static approaches in terms of cost.

Appendix A
Algorithmic description of Heuristic 1

See Algorithm 1.

Algorithm 1: Heuristic 1 (Webster/Saint-Laguë method and binary search).

Input: $X^+,\ X^-,\ \hat{I}_g\ \forall g \in \mathbb{G}$
Output: $\mathbf{x}, \mathbf{f}$

$x_g \leftarrow 0 \quad \forall g \in \mathbb{G}$
$X \leftarrow X^+$
repeat // Binary search
  repeat // W/S-L assignment
    $g^* \leftarrow \arg\max_g \{\hat{I}_g \mid g \in \mathbb{G}\}$
    $x_{g^*} \leftarrow x_{g^*} + 1$
    if $x_{g^*} < Q$ **then**
      $I_{g^*} \leftarrow \frac{I_{g^*}}{2x_{g^*}+1}$
    else
      $I_{g^*} \leftarrow -\infty$
    end
  until $\sum_{g=1}^{G} x_g = X$
  if (26) *feasible* **then** // Feasibility check
    $X^- \leftarrow X$
    $\mathbf{f} \leftarrow \mathbf{f}^*(\mathbf{x})$ as in (26).
  else
    $X^+ \leftarrow X - 1$
  end
until $X^+ = X^-$

Appendix B
Binary search for Heuristic 1

As $c(\cdot)$ is a monotonically decreasing function, it can be trivially proven that $\widetilde{\eta}(\mathbf{x}) \geq \widetilde{\eta}(\mathbf{x}')$ whenever $x_g \geq x'_g$ $\forall g \in \mathbb{G}$, and vice-versa. In words, this means that increasing (decreasing) the centralization level of any gNB can only increase (decrease), on average, the mean spectral efficiency, as intuitively expected. Similarly, given constant $I_g\ \forall g \in \mathbb{G}$ and two accumulated centralization levels X and X' such that $X > X'$, it can be easily shown that the resulting centralization vectors $\mathbf{x}$ and $\mathbf{x}'$ yielded by the W/S-L algorithm fulfill $x_g \geq x'_g$ $\forall g \in \mathbb{G}$. Finally, it is also clear that if $\mathbf{x}$ is feasible, then $\mathbf{x}'$ must be as well. We conclude that there is a single maximum value of X such that for all $X' < X$ the W/S-L method returns feasible but lower-performance solutions, and for all $X'' > X$ the W/S-L method returns infeasible solutions. In the light of the above, we can implement a binary search of the highest feasible value of X as shown in Algorithm 1.

Acknowledgment

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 647158 - FlexNets). The authors alone are responsible for the content of the paper.

References

[1] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in lte-advanced: Overview and Rel-11 enhancements," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 68–74, 2012.
[2] M. Iwamura, A. Umesh, and W. A. Hapsari, "Further enhancements of LTE –LTE release 9–," *NTT Docomo Technical Journal*, vol. 12, no. 1, pp. 45–53, 2010.
[3] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.
[4] NGMN Alliance, "5G white paper," *Next generation mobile networks, white paper*, vol. 1, 2015.
[5] S. Nagul, "A review on 5G modulation schemes and their comparisons for future wireless communications," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*. IEEE, 2018, pp. 72–76.
[6] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE access*, vol. 1, pp. 335–349, 2013.
[7] B. Soret, A. De Domenico, S. Bazzi, N. H. Mahmood, and K. I. Pedersen, "Interference coordination for 5G new radio," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 131–137, 2017.
[8] N. H. Mahmood, L. G. Uzeda Garcia, P. Popovski, and P. E. Mogensen, "On the performance of successive interference cancellation in 5g small cell networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2014, pp. 1154–1159.
[9] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Lossow, M. Sternad, R. Apelfröjd, and T. Svensson, "The role of small cells, coordinated multipoint, and massive MIMO in 5G," *IEEE communications magazine*, vol. 52, no. 5, pp. 44–51, 2014.
[10] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks–a technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405–426, 2014.
[11] A. Maeder, M. Lalam, A. De Domenico, E. Pateromichelakis, D. Wübben, J. Bartelt, R. Fritzsche, and P. Rost, "Towards a flexible functional split for cloud-RAN networks," in *2014 European Conference on Networks and Communications (EuCNC)*. IEEE, 2014, pp. 1–5.
[12] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 105–128, 2013.
[13] A. Martínez Alba, J. H. G. Velásquez, and W. Kellerer, "Traffic characterization of the MAC-PHY split in 5G networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
[14] D. Sabella, P. Rost, Y. Sheng, E. Pateromichelakis, U. Salim, P. Guitton-Ouhamou, M. Di Girolamo, and G. Giuliani, "RAN as a service: Challenges of designing a flexible ran architecture in a cloud-based heterogeneous mobile network," in *2013 Future Network & Mobile Summit*. IEEE, 2013, pp. 1–8.
[15] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC orchestration," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2366–2374.
[16] L. Diez, V. Gonzalez, and R. Agüero, "Minimizing delay in NFV 5G networks by means of flexible split selection and scheduling," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–6.
[17] D. Harutyunyan and R. Riggio, "Flex5G: Flexible functional split in 5G networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, 2018.
[18] A. Martínez Alba and W. Kellerer, "A dynamic functional split in 5G radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
[19] C.-Y. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "FlexCRAN: A flexible functional split framework over ethernet fronthaul in cloud-RAN," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.
[20] A. Martínez Alba, J. H. G. Velásquez, and W. Kellerer, "An adaptive functional split in 5G networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 410–416.
[21] 3GPP, "Study on new radio access technology: Radio access architecture and interfaces," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.801, 03 2017, version 14.0.0.

[22] C.-H. Fang, P.-R. Li, and K.-T. Feng, "Joint interference cancellation and resource allocation for full-duplex cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3019–3033, 2019.

[23] 3GPP, "NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, 01 2021, version 16.4.0.

[24] G. Nardini, G. Stea, A. Virdis, A. Frangioni, L. Galli, D. Sabella, and G. M. Dell'Aera, "Practical feasibility, scalability and effectiveness of coordinated scheduling algorithms in cellular networks towards 5G," *Journal of Network and Computer Applications*, vol. 106, pp. 1–16, 2018.

[25] Y. Zhang, J. Ding, M.-W. Kwan, J. Ni, E. K. Tsang, Y.-N. R. Li, and J. Li, "Measurement and evaluations of coherent joint transmission for 5G networks," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 2017, pp. 1–5.

[26] H. Paixão Martins, "Analysis of CoMP for the management of interference in LTE," *Master Thesis*, 2017.

[27] Y. He, F. R. Yu, N. Zhao, V. C. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, 2017.

[28] J. Branke and H. Schmeck, "Designing evolutionary algorithms for dynamic optimization problems," in *Advances in evolutionary computing*. Springer, 2003, pp. 239–262.

[29] M. Koivisto, A. Hakkarainen, M. Costa, P. Kela, K. Leppanen, and M. Valkama, "High-efficiency device positioning and location-aware communications in dense 5G networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 188–195, 2017.

[30] K. Govindaraj and A. Artemenko, "Container live migration for latency critical industrial applications on edge computing," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 83–90.

[31] G. Song and Y. Li, "Cross-layer optimization for ofdm wireless networks-part i: theoretical framework," *IEEE transactions on wireless communications*, vol. 4, no. 2, pp. 614–624, 2005.

[32] F. Glover and E. Woolsey, "Converting the 0–1 polynomial programming problem to a 0–1 linear program," *Operations research*, vol. 22, no. 1, pp. 180–182, 1974.

[33] R. Horst, "A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization," *Journal of Optimization Theory and Applications*, vol. 51, no. 2, pp. 271–291, 1986.

[34] H.-L. Li, "A global approach for general 0–1 fractional programming," *European Journal of Operational Research*, vol. 73, no. 3, pp. 590–596, 1994.

[35] T.-H. Wu, "A note on a global approach for general 0–1 fractional programming," *European Journal of Operational Research*, vol. 101, no. 1, pp. 220–223, 1997.

[36] M. Tawarmalani, S. Ahmed, and N. V. Sahinidis, "Global optimization of 0–1 hyperbolic programs," *Journal of Global Optimization*, vol. 24, no. 4, pp. 385–416, 2002.

[37] 3GPP, "Study on scenarios and requirements for next generation access technologies," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.913, 07 2018, version 15.0.0.

[38] J. S. Borrero, C. Gillen, and O. A. Prokopyev, "A simple technique to improve linearized reformulations of fractional (hyperbolic) 0–1 programming problems," *Operations Research Letters*, vol. 44, no. 4, pp. 479–486, 2016.

[39] F. Glover, "Improved linear integer programming formulations of nonlinear integer problems," *Management Science*, vol. 22, no. 4, pp. 455–460, 1975.

[40] M. Gallagher, "Proportionality, disproportionality and electoral system," *Electoral studies*, vol. 10, no. 1, pp. 33–51, 1991.

[41] K. Schuster, F. Pukelsheim, M. Drton, and N. R. Draper, "Seat biases of apportionment methods for proportional representation," *Electoral Studies*, vol. 22, no. 4, pp. 651–676, 2003.

[42] P. K. Sharma and R. Singh, "Comparative analysis of propagation path loss models with field measured data," *International Journal of Engineering Science and Technology*, vol. 2, no. 6, pp. 2008–2013, 2010.

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

[44] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[45] H. D. Trinh, L. Giupponi, and P. Dini, "Urban anomaly detection by processing mobile traffic traces with LSTM neural networks," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–8.

[46] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM transactions on networking*, vol. 25, no. 2, pp. 1147–1161, 2016.

[47] A. Martínez Alba and W. Kellerer, "Large-and small-scale modeling of user traffic in 5G networks," in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–5.

[48] R. K. Polaganga and Q. Liang, "Self-similarity and modeling of LTE/LTE-A data traffic," *Measurement*, vol. 75, pp. 218–229, 2015.

[49] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, "Evaluating c-ran fronthaul functional splits in terms of network level energy and cost savings," *Journal of Communications and Networks*, vol. 18, no. 2, pp. 162–172, 2016.

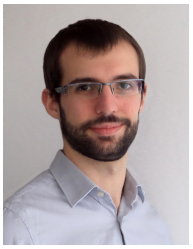
[50] V. Suryaprakash, P. Rost, and G. Fettweis, "Are heterogeneous cloud-based radio access networks cost effective?" *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2239–2251, 2015.

[51] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

[52] J.-G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 766–778, 2007.

[53] A. Martínez Alba, P. Babarczi, A. Blenk, M. He, P. Krämer, J. Zerwas, and W. Kellerer, "Modeling the cost of flexibility in communication networks," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021.

[54] A. Martínez Alba, S. Janardhanan, and W. Kellerer, "Dynamics of the flexible functional split selection in 5g networks," in *2020 IEEE Global Communications Conference: Mobile and Wireless Networks*, 2020.



Alberto Martínez Alba received his Bachelor's and Master's degrees in Telecommunication Engineering from the Technical University of Madrid, Spain. He is currently pursuing the Ph.D. degree with the Chair of Communication Networks, Technical University of Munich, Germany. His current research interests include the design, optimization, and implementation of flexible next-generation mobile networks, adaptive radio access networks, and software-defined mobile networks.

Shakthivelu Janardhanan received his Bachelor's degree in Electronics and Communication in 2019, from Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, India. He is currently pursuing his Master's degree in Communication Engineering at the Technical University of Munich, Germany. His research interests include Computer Networks, Wireless sensor networks, 5G - New Radio and Functional split of Cloud RAN architecture.

Wolfgang Kellerer (M'96, SM'11) is a Full Professor with the Technical University of Munich (TUM), heading the Chair of Communication Networks at the Department of Electrical and Computer Engineering. Before, he was for over ten years with NTT DOCOMO's European Research Laboratories. He currently serves as an associate editor for IEEE Transactions on Network and Service Management and as the area editor for Network Virtualization for IEEE Communications Surveys and Tutorials.