Distributed Resource Management in Downlink Cache-enabled Multi-cloud Radio Access Networks

Alaa Alameer Ahmad, *Student Member, IEEE*, Robert-Jeron Reifert, *Student Member, IEEE*, Hayssam Dahrouj, *Senior Member, IEEE*, Anas Chaaban, *Senior Member, IEEE*, Aydin Sezgin, *Senior Member, IEEE*, Tareq Y. Al-Naffouri, *Senior Member, IEEE*, and Mohamed-Slim Alouini, *Fellow, IEEE*

Abstract

In the scope of beyond fifth generation (B5G) networks and the massive increase of data-hungry systems, the need of extending conventional single-cloud radio access networks (C-RAN) arises. A compound of several clouds, jointly managing inter-cloud and intra-cloud interference, constitutes a practical solution to cope with requirements of B5G networks. This paper considers a multi-cloud radio access network model (MC-RAN) where each cloud is connected to a distinct set of base stations (BSs) via limited capacity fronthaul links. The BSs are equipped with local cache storage and base-band processing capabilities, as a means to alleviate the fronthaul congestion problem. The paper then investigates the problem of jointly assigning users to clouds and determining their beamforming vectors so as to maximize the network-wide energy efficiency (EE) subject to fronthaul capacity, and transmit power constraints. This paper solves such a mixed discrete-continuous, non-convex optimization problem using fractional programming (FP) and successive inner-convex approximation (SICA) techniques to deal with the non-convexity of the continuous part of the problem, and l_0-norm approximation to account for the binary association part. A highlight of the proposed algorithm is its capability of being implemented in a distributed fashion across the network multiple clouds through a reasonable amount of information exchange. The numerical simulations illustrate the pronounced role the proposed algorithm plays in alleviating the interference of large-scale MC-RANs, especially in dense networks.

Part of this paper was presented at the IEEE International Conference on Communications Workshops, June 2020 [1].

Alaa Alameer Ahmad, Robert-Jeron Reifert and Aydin Sezgin are with Digital Communication Systems, Ruhr-Universität Bochum, Bochum, Germany. (Email: {alaa.alameerahmad,robert-.reifert,aydin.sezgin}@rub.de)

Hayssam Dahrouj, Tareq Y. Al-Naffouri and Mohamed-Slim Alouini are with King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. (Email: {hayssam.dahrouj,tareq.alnaffouri,slim.alouini}@kaust.edu.sa)

Anas Chaaban is with the School of Engineering, The University of British Columbia, Kelowna, Canada. (Email: anas.chaaban@ubc.ca)

I. Introduction

A. Overview

Beyond fifth generation (B5G) wireless communication networks are expected to enable ultra-connectivity through the empowerment of Internet of Things (IoT) systems [2]. IoT systems introduce unprecedented amounts of data traffic, thanks to the tremendous increase in the number of efficient mobile communication devices such as smartphones and tablets and the extreme popularity of content-provider social media platforms such as YouTube and Netflix [2]–[4]. Video data-traffic leads to an exponential increase in mobile data traffic. The video data usage is anticipated to increase from 63% of the total data-traffic of 38 exabytes (EB) per month in 2019 to 76% of the total data-traffic of 160 EB per month in 2025 [5]. While the data traffic exponentially increases and the requirements for modern communication systems introduce new challenges, restraining the network's total energy consumption is vital. In recent years, cloud radio access networks (C-RANs) have emerged as promising network architecture to accommodate the requirements of B5G wireless networks. In C-RAN, a large set of geographically distributed base stations are connected to a central processor (CP) at the cloud via high-speed digital fronthaul links [6]. Also, edge caching in wireless networks is proposed as an efficient and promising technique to reduce the congestion in the network and the content delivery time during peak-traffic communication [7]. This is achieved by storing the popular content at the BSs closer to end-users. Such caching approach can further improve the content delivery rate and reduce the communication latency via alleviating the communication load on the fronthaul links, which represents the bottleneck in achieving high data-rates in C-RANs. The majority of works on C-RAN consider a single-cloud scenario, where a single CP in the cloud is responsible for coordinating the operation of the well-spread multi-cell networks, containing a large number of BSs and users (see [8], [9] and references therein). However, the plurality and widespread of devices in next-generation systems would necessitate the deployment of multiple CPs, each responsible for managing a distinct set of BSs [10]–[12]. Each CP at the cloud coordinates the data processing and beamforming vectors of the set of BSs associated with it. Such coordination between CPs, however, needs not to exacerbate the communication backbones, and is rather limited to message passing among the different clouds; hence the need to distributively manage their underling infrastructures on a message passing level, which this paper tackles in details. We refer to a C-RAN with multiple CPs as multicloud-radio access network (MC-RAN) to distinguish it from the classical single CP C-RAN. In MC-RAN, the *inter-cloud* interference becomes an additional performance barrier metric, especially given the limited communication between distributed CPs; thus managing both the inter-cloud and the intra-cloud interference for a true assessment of MC-RAN performance. In this paper, we consider a content-based MC-RAN, where each cloud coordinates the operation of a set of BSs. Each BS is equipped with a local memory that has a certain storage capacity to cache the most popular files. This paper accounts for the challenges that arise by the multi-objective nature of the resource allocation as well as the importance of minimizing the power consumption in future wireless networks and adopts the problem of maximizing the energy efficiency (EE) metric, which is defined as the rate-to-power ratio. In MC-RAN with edge caching capabilities, the system performance becomes a function of the user-to-cloud association and caching strategies, as well as the beamforming vector of each user which constitutes a challenging non-convex optimization problem. The paper tackles this problem and devises an efficient algorithm that can be implemented in a distributed fashion across the multiple CPs.

B. Related Works

The MC-RAN resource management problem considered in this paper is related to recent works on wireless edge caching, cloud-radio access network, and distributed resource allocation. To assist the performance of C-RANs, many works have focused on cooperative beamforming transmission to maximize the network-wide sum-rate of the users as the wireless network evolution was mainly driven by a need for higher data-rates [13]. However, the emergence of IoT has introduced new challenges to wireless networks. Hence, B5G networks need to support many heterogeneous services with different, and often, conflicting requirements like different throughput and latency requirements. These services can be classified into three main categories: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low latency communications (URLLC) services. Wireless edge caching has received a lot of attention in research society recently. The seminal paper [7] highlights the benefits of caching in reducing the end-to-end transmission delay and alleviating the bottleneck of fronthaul capacity in wireless communication. In [14], the authors investigate both coded and uncoded caching strategies and analyze their impact on the EE of the system. Cooperative caching and optimizing transmission schemes jointly in small cell networks are studied in [15]. The seminal work [16] considers a cache-enabled single C-RAN. The authors in [16] investigate the dynamic content-centric BS clustering and multicast beamforming design and formulate the problem of minimizing a weighted sum of fronthaul cost and transmit power under the quality-of-service (QoS) constraints. The authors in [17] study a coded-caching strategy in C-RAN and use semi-definite relaxations (SDP) to optimize the beamforming vectors from BSs to users. Other works have studied edge caching in C-RAN with different objectives. In [18], the authors study the impact of caching on balancing the outage probability against fronthaul usage in a single C-

RAN. The paper [18] suggests a caching strategy that jointly optimizes the cell average outage probability and fronthaul usage. The paper [19] studies the joint design of cloud and edge processing, where the edge nodes, i.e., the BSs, are equipped with local caches. All these works [16]–[19], however, consider a single cloud RAN model. Moreover, apart of [19], these works consider a conventional C-RAN model in which the CP is responsible for performing most tasks in the baseband processing protocol, while radio transmission is done by the BSs. In [19], the necessity of the existence of baseband processing capabilities at the BSs is pointed out. Such capabilities are necessary when the BSs are equipped with a local cache which enables them to send the content directly to the end-user without the need for CP interaction, thus reducing the usage of fronthaul capacity. Promising results show that edge computing architecture has huge potentials for meeting B5G system requirements. In [20], the authors propose a computational cost model, which directly links the resource blocks reserved for a certain service with the computational capacity required for performing the processing tasks. Moreover, it is known [21], [20] that the required resource blocks mainly depend on the type of service requested by the users. In [22], the authors utilize power and subchannel allocation schemes as well as edge caching. Methods to optimize the resource allocation in mobile edge computing networks are proposed in [23]. A hybrid resource allocation algorithm for mobile edge computing networks is proposed in [24]; results show that this technology is well suited for future IoT applications. Recently, MC-RAN systems have been studied in references [10] and [11]. In [10], the authors study an MC-RAN problem in which each CP adopts a compression-based transmission strategy. In the current article, however, we focus on the data-sharing strategy, since it is shown to achieve better performance in terms of sum-rate [25]. The authors in [12] consider the user-to-CP association problem in a multi-cloud setup and assume fixed beamforming and an infinite fronthaul capacity. The authors in [26] partially overcome this issue by assuming a discrete set of fixed resources associated with each cluster of BSs connected to a specific CP. The impact of finite fronthaul links is further considered in [11].

Departing from previous works, which mainly focused on single-cloud architecture and either consider the edge caching problem or the local processing power, but not the connection of both facets. In this article, we consider the downlink of an MC-RAN in which the BSs are equipped with local caches and baseband processing capabilities. The performance in such a system becomes therefore a function of the user-to-cloud association and the baseband functional split between the CPs and the local BSs. As the caches require processing power, additional energy consumption at the BSs has to be considered [27]. We propose a transmission scheme in which the content requested by each user can be served directly from the BS, if it is stored in the cache, or can be retrieved from the CP in case local processing is not affordable. To the best of our knowledge this the first work which investigates the connection between edge caching and functional split in MC-RAN. Thus, we focus on jointly determining the user-to-cloud association and the users' beamforming vectors by maximizing the EE subject to exclusive local or global processing constraints, per-BS power, and per-BS fronthaul constraints. To tackle such a difficult mixed discrete-continuous non-convex optimization problem, we propose using centralized and distributed iterative algorithms, each requiring different cooperation levels and hence different computation/communication overhead levels between the clouds. The solution is based on a fractional programming and successive inner-convex approximations (SICA) framework for the continuous variables and a l_0-norm heuristic approximation for the discrete (binary) variables. A highlight of the proposed algorithm is its ability to determine the user-to-cloud association and beamforming vectors in a distributed fashion across the multiple clouds, which makes it amenable to practical implementation. Through extensive numerical simulations, we further show that the performance of our distributed approach significantly outperforms state-of-the-art schemes.

C. Contributions

In our conference version [1] we studied the capabilities of an MC-RAN in terms of sum-rate maximization. Herein, we tackle a more complex objective, namely the EE of an MC-RAN, which strikes a trade-off between achieving a reasonably high sum-rate for a relatively low power consumption. We consider a practical system model in which multiple CPs are responsible to manage a dense set of BSs, each equipped with local cache storage and base-band processing capabilities, as a means to alleviate the fronthaul links congestion across the multiple clouds of the network. The major contributions of this paper are then given as follows

1) *Hybrid transmission scheme:* In the studied system model we propose a flexible functional split between the CPs at clouds and the BSs. That is, if a BS caches the requested content, the baseband processing functions can be performed either locally at the BS, bypassing the interaction with the CP and the corresponding load on the fronthaul links, or centrally at the CP. Each functional split option determines a trade-off between the computation and fronthaul communication costs and results with different EE values. The resulting optimization problem is NP-hard and difficult to solve in general.
2) *Optimization framework:* We develop a general solution to the problem which is based on detaching the user-to-cloud association for complexity reasons, relaxing the binary variables using l_0-norm approximation, and then solving the continuous non-convex optimization problem using Dinkelbach-transform with a successive inner convex approximations framework.

3) *Numerical Simulations:* We perform extensive numerical simulations to evaluate the performance of distributed and centralized implementations of the proposed scheme. We also compare these implementations against state-of-the-art schemes. In particular, we investigate different aspects, i.e., cache size, processing costs, and convergence behavior. We analyze the influence of fronthaul capacity and number of users on the EE of the considered MC-RAN.

D. Notations

Throughout the paper, boldface lower-case and capital letters (e.g. $\mathbf{h}$, $\mathbf{H}$) denote vectors and matrices, respectively. Calligraphic letters (e.g. $\mathcal{H}$) represent sets. A column vector consisting of all the elements in set $\mathcal{H}$ is defined as $\text{vec}\{\mathcal{H}\}$. If $\mathcal{H} = \{h_1, \cdots, h_N\}$, then $\text{vec}(\mathcal{H}) \equiv [h_1, \cdots, h_N]^T$. If $\mathcal{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_N\}$, then $\text{vec}(\mathcal{H}) \equiv [\mathbf{h}_1^T, \cdots, \mathbf{h}_N^T]^T$. $\mathbf{0}_N$ is a vector of length N with all elements set to zero. The real and complex field are noted as $\mathbb{R}$ and $\mathbb{C}$, respectively, while the real part of complex numbers is given by $\Re\{\cdot\}$. Finally $(\cdot)^\dagger$ denotes the hermitian transpose and $(\cdot)^T$ the transpose operator, also $|\cdot|$ is the absolute value and $\left\|\cdot\right\|_p$ the l_p-norm.

II. System Model

In this section, we describe the overall system model for the considered MC-RAN. We explain the received signal and cache models, and the two considered cost models.

A. Received Signal Model

Consider the downlink of an MC-RAN, consisting of C CPs, where each cloud coordinates a certain number of BSs B_c, each equipped with L antennas, over a network comprising K single-antenna users. We allow for a limited cooperation between the clouds on a control level. Each BS is also equipped with a local cache memory assumed to cache a total of $F_b \leq F$ local files, where F are files assumed in the library. Each BS is connected to one (and only one) CP via a digital fronthaul link with finite capacity.

Now, let $\mathcal{C} = \{1, \cdots, C\}$ be the set of CPs and $\mathcal{B} = \{1, \cdots, B\}$ be the set of BSs in the network, where $B = \sum_{c \in \mathcal{C}} B_c$. Furthermore, let $\mathcal{K} = \{1, \cdots, K\}$ be the set of users and $\mathcal{F} = \{1, \cdots, F\}$ be the set of all files. We assume that each user $k \in \mathcal{K}$ can be assigned to one and only one CP $c \in \mathcal{C}$. Furthermore, we assume that every CP $c \in \mathcal{C}$ is connected to a cluster of BSs denoted by $\mathcal{B}_c = \{1, \cdots, B_c\}$. The networks clusters $\mathcal{B}_c$, $c \in \mathcal{C}$ are assumed to be disjoint, i.e., $\cup_{c \in \mathcal{C}} \mathcal{B}_c = \mathcal{B}$, $\mathcal{B}_c \cap \mathcal{B}_{c'} = \varnothing, \forall c \neq c'$, and B_c is the total number of BSs in the cluster connected to CP c. An example of such a system is given in Fig. 1.

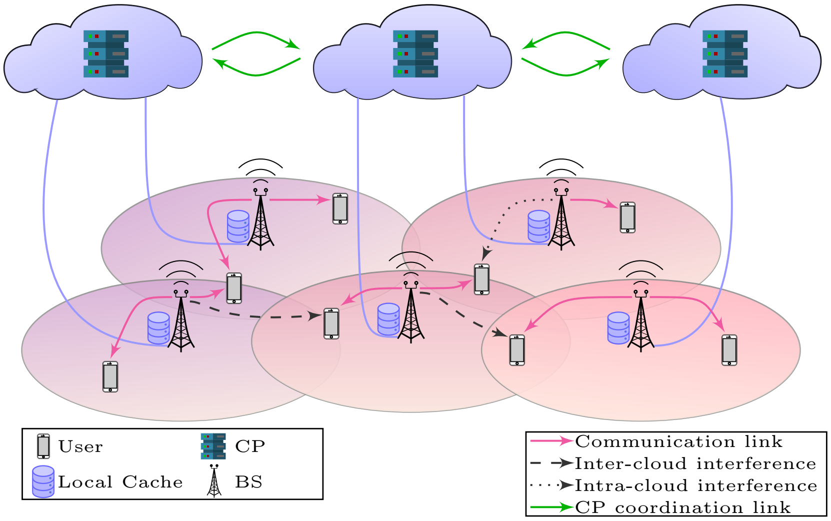


Fig. 1: System model of an MC-RAN consisting of three clouds, 5 BSs and 8 users. Examples for inter- and intra-cloud interference, as well as generic communication links are provided.

Let $\mathbf{h}_{c,b,k} \in \mathbb{C}^L$ be the channel vector from the b-th BS of the c-th cloud to the k-th user, and let $\mathbf{h}_{c,k} \in \mathbb{C}^{B_c L \times 1}$ be the aggregated channel vector from the c-th cloud to the k-th user. This can be expressed as $\mathbf{h}_{c,k} \triangleq [\mathbf{h}_{c,1,k}^T, \cdots, \mathbf{h}_{c,B_c,k}^T]^T$. To simplify our discussion and make the problem mathematically tractable, we assume that each CP has access to the full channel state information (CSI), the cached content of BSs in $\mathcal{B}_c$ and the demands (requested files) of all the users in the network. To deliver the requested files, we adopt a time-slotted block-based transmission model where each transmission block consists of several time slots. The channel fading coefficients remain constant within one block and may vary independently from one block to another. We focus on optimizing the EE of the cache-aided MC-RAN within one transmission block. Without loss of generality, we consider that the CPs divide each requested file into several data chunks, so that the transmission of each file may take place on several consecutive transmission blocks and the number of required transmission blocks to transmit each file may be different from other files. Moreover, the size of a data chunk in a single transmission block for each user depends on its achievable rate and the transmission resource blocks reserved for this user. Hence, the transmission resource blocks and the size of the data chunk depend on the service required by a certain user.

B. Cache Model

In content delivery networks (CDN), edge caching is employed to bring the content closer to users. In general, we can distinguish between two phases in content delivery process to mobile users, namely *cache placement* and *cache delivery* phases [28], [29]. Therefore, the recent works studying cache-aided wireless networks can be divided into two main categories: 1) optimizing the cached content delivery process for a given cache placement to get the best possible performance [30]; 2) improve the content delivery process through efficient design of cache placement strategies [18]. Essentially, in cache placement phase, the popular content is stored at edge-network nodes, i.e., at BSs, with the sole purpose of improving the content-delivery phase, especially in peak-traffic times. Hence, the cache placement phase takes place over a much longer time-scale as that required in cache-delivery phase, since the popularity of the content changes much slower than the time required to deliver requested content to the users.

In this article, we focus on the optimization of the content-delivery phase, while the cache placement phase is considered to be performed a priori. The cache content at BSs and the whole library of files are assumed to be known at the clouds. Let $\mathbf{C} \in \{0,1\}^{F\times B}$ be the binary cache placement matrix where $[\mathbf{C}]_{f,b} = c_{f,b}$ is the element in the f-th row and the b-th column. Now, let $f_k \in \mathcal{F}$ be the requested file of user k, then $c_{f_k,b} = 1$, a *cache hit*, means that f_k is cached at BS b and $c_{f_k,b} = 0$, a *cache miss*, means it is not. Define the set of *cache hit* users as $\mathcal{K}_1 \triangleq \{k \in \mathcal{K} | \, \exists (c,b) \in \mathcal{C} \times \mathcal{B} : c_{f_k,b} = 1\}$. Hence, the set $\mathcal{K}_1$ contains all users whose requested files are cached locally at the BSs. On the other hand, we define the set of *cache-miss* users $\mathcal{K}_2$ as the set of users whose requested files are not cached the BSs, i.e., $\mathcal{K}_2 \triangleq \{k \in \mathcal{K} | \, \forall (c,b) \in \mathcal{C} \times \mathcal{B} : c_{f_k,b} = 0\}$. Note that in the special case where $\mathbf{C} = \mathbf{0}_{F\times B}$, no files are stored in BSs caches, i.e., $\mathcal{K}_1 = \varnothing$ and $\mathcal{K}_2 = \mathcal{K}$. On the other hand, when $\sum_{b\in\mathcal{B}} c_{f,b} \geq 1 \; \forall f \in \mathcal{F}$, each file is cached at least one BS, i.e., $\mathcal{K}_2 = \varnothing$ and $\mathcal{K}_1 = \mathcal{K}$.

C. Baseband Processing and Fronthaul Communication Cost Models

In RAN communication, traditional network functions (NF) for the physical and MAC layer constitute a series of baseband processing tasks such as coding, modulation, and FFT [21], [20], [31]. Several functional split options for performing the baseband processing tasks are discussed in the literature. Each function split results in a trade-off between the required fronthaul links and computational loads. Due to local caching capabilities, in this work we consider two functional split options, i.e., either the CP or the BS performs encoding of the data. In particular, for the requested content which is not cached at the BSs, the corresponding CP performs encoding for the requested file. Encoding is the most intensive processing task and accounts for most of the computational load in physical layer NF processing chain [32]. On the other hand, if the BSs cache the requested file, the encoding task can be either performed at the corresponding CP or the local processing unit of BS. Hence, performing the processing task at the CP saves computational costs as the encoding is performed centrally and the encoded symbols are then shared over the fronthaul links with a set of BSs. However, this comes at the cost of increasing the load on the scarce resources of the limited-capacity fronthaul links. The majority of works on wireless caching consider the fronthaul communication and transmit costs, but ignore the computation cost required to process the requested file before transmitting it to the users. Different from previous works, in this paper we account for all these factors while optimizing the delivery phase strategy such that the EE of the MC-RAN is maximized. We consider a simple computational cost model, in which the processing cost associated with each requested content is fixed and depends on the service class requested by the user and the number of resource blocks served for delivering the requested content. We denote this cost as $\{P_k^{\text{proc}}, \quad \forall k \in \mathcal{K}\}$. More complicated computational cost models are considered in [33], where a normalized cubic function of the rate is assumed, a quadratic model is used in [34] and a linear model in [35]. Investigating various computational cost models for evaluating the overall performance is out of this work's scope and left for future investigations on this topic. Next, we describe different transmitting strategies considered in the cache-aided MC-RAN. Each uses the processing resources, fronthaul links, and transmit resources differently.

III. Transmit Scheme and Functional Split

In this section, we describe our proposed hybrid transmission scheme. Therein, we differentiate between two schemes, the encoding can be done at either at the cloud or at the BSs. To this end, different terms, e.g., achievable rate, and required fronthaul capacity, are defined, also we elaborate on the EE in our MC-RAN.

A. Design Transmit Signals at the CP

In this paper we focus on the *data sharing* transmission strategy. In this strategy, the CP performs joint encoding of users' data. In more details, CP c encodes v_k, the data chuck of the file f_k requested by user k into s_k. Here s_k denotes the symbol of the encoded data at CP c to be transmitted to user k at the current time-slot. We assume that s_k is chosen independently from a complex Gaussian distribution with zero-mean and unit variance. After that, the CP forwards s_k, the encoded data chunks, through limited capacity fronthaul links to the cluster of BSs serving user k. The BSs then cooperate to transmit the signal to user k using a joint beamforming vector. Although the beamforming vector coefficients are designed at the CP, the modulation and precoding tasks are performed at the BSs. We assume that the rate required to transmit beamforming vector coefficients over the fronthaul links is negligible compared to that required for transmitting the coded symbols of the users [8]. Let $\mathbf{w}_{c,k} \in \mathbb{C}^{B_c L \times 1} = \left[\mathbf{w}_{c,1,k}^T, \ldots, \mathbf{w}_{c,B_c,k}^T\right]^T$ be the aggregate beamforming vector of user k when associated with CP c. Note that if BS $b \in \mathcal{B}_c$ is not in the BSs' cluster serving user k, then $\mathbf{w}_{c,b,k} = \mathbf{0}_L$, and the CP in this case does not share any data of user k with BS b. Thus, the aggregate beamforming vector $\mathbf{w}_{c,k}$ is a group-sparse vector by construction.

B. Design Transmit Signals Locally at BSs

The advantage of the *data sharing* strategy is that the complicated encoding processing task is done jointly for the users centrally at the CP. On one hand, the encoding process for each user k is done once for all the BSs in the serving cluster, which significantly saves processing power required to encode the users' data. Further, the CP is assumed to profit from cloud computing infrastructure, which enables scaling the required processing resources, i.e., virtual machines (VMs), up and down as needed; thereby consuming less power as compared to stand-alone hardware local processing unit at the BSs. On the other hand, since the encoded data must be delivered to BSs through finite capacity fronthaul links connecting each BS to the CP, the BS cluster size and accordingly the achievable rates may be significantly limited by the fronthaul links' capacity. Besides, as the data is not processed in the proximity of users (not processed at the edge) we encounter an additional processing and communication delay which may limit the support for delay-sensitive applications. Caching the most popular files locally at the BSs overcomes the disadvantages of processing the data at the cloud and significantly reduces the load on the fronthaul links. Hence, the usage of the fronthaul link boils down to the exchange of essential control information between CPs and BSs (e.g., beamforming coefficients and scheduling information). However, despite the advantages of caching the content locally at BSs in terms of reducing latency and saving the fronthaul link bandwidth, this comes at the cost of increasing the processing cost at the BSs. Hence, we assume that the BSs cache the uncoded data locally and therefore encoding the data before transmission is done at the local processing unit at the BSs. That is, we assume the baseband processing tasks can be completely done at the BSs serving user k when the BSs cache the required file of user k. To this end, let $\tilde{\mathbf{w}}_{c,b,k} \in \mathbb{C}^{L \times 1}$ refer to the beamforming vector explicitly used at BS $b \in \mathcal{B}_c$ for user k when the baseband processing tasks are performed at the BS. Let $\tilde{\mathbf{w}}_{c,k} \in \mathbb{C}^{B_c L \times 1} = \left[\tilde{\mathbf{w}}_{c,1,k}^T, \ldots, \tilde{\mathbf{w}}_{c,B_c,k}^T\right]^T$ be the aggregate beamforming vector at BSs in cluster $\mathcal{B}_c$ which cache the requested file from user k. Note that the BS b can encode the data locally and independently of the CP connected to it when it caches the requested file from user k. The control information needed for transmitting the signal is, however, assumed to be provided from the cloud. Hence, partial cooperation between CPs on the control level is assumed to be possible in this work.

C. Hybrid Transmit Strategy

In this paper, we are interested in a hybrid transmit strategy. Under such a strategy, for each BS there is a flexible decision between three possibilities on serving a user k (for each user $k \in \mathcal{K}$). Either it participates in transmitting data to k following the CP processing strategy, i.e., the CP performs encoding, or it processes the data locally. Otherwise a BS may not transmit to user k at all. The baseband transmit signal at BS b from the cluster $\mathcal{B}_c$, $\mathbf{x}_{c,b} \in \mathbb{C}^{L\times 1}$ can thus be written as follows

$$\mathbf{x}_{c,b} = \sum_{k\in\mathcal{K}} \left(\mathbf{w}_{c,b,k} + \tilde{\mathbf{w}}_{c,b,k}\right) s_k. \tag{1}$$

The encoding process can be either done at the cloud or locally at the BS but not in both at the same time. Also, the BS can perform the processing locally only in case it caches the requested file. Therefor, equation (1) is accompanied with the following two conditions on the beamforming vectors

$$\mathbb{1}\left\{\left\|\mathbf{w}_{c,b,k}\right\|_2^2\right\} + \mathbb{1}\left\{\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right\} \leq 1, \quad \forall k \in \mathcal{K} \text{ and } \forall b \in \mathcal{B}_c \tag{2}$$

$$\tilde{\mathbf{w}}_{c,b,k} = \mathbf{0}_L \quad \forall k \in \mathcal{K}_2 \text{ and } \forall b \in \mathcal{B}_c \tag{3}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function such that $\mathbb{1}\left\{\left\|\mathbf{w}_{c,b,k}\right\|_2^2\right\} = 1$ if $\left\|\mathbf{w}_{c,b,k}\right\|_2^2 > 0$, and 0 otherwise, similarly $\mathbb{1}\left\{\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right\} = 1$ if $\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2 > 0$, and 0 otherwise. Equations (1), (2), and (3) can be interpreted as follows: if file f_k is not cached at BS b then BS $b \in \mathcal{B}_c$ transmits to user k only if $\mathbb{1}\left\{\left\|\mathbf{w}_{c,b,k}\right\|_2^2\right\} = 1$, since in this case $\tilde{\mathbf{w}}_{c,b,k} = \mathbf{0}_L$. In case file f_k is cached at BS b then whether the data is encoded at the CP or locally at the BS is solely decided by equation (2). Moreover, by construction, if user k is not associated with CP c then $\mathbf{w}_{c,b,k} = \mathbf{0}, \forall b \in \mathcal{B}_c$. The specific design of beamforming vectors $\mathbf{w}_{c,b,k}$ or $\tilde{\mathbf{w}}_{c,b,k}$ in this case is based on solving our optimization problem, as discussed in details in section IV. After forming the transmit signal as in [1], BS b sends $\mathbf{x}_{c,b}$ subject to the following maximum transmit power constraint:

$$\mathbb{E}\left\{\mathbf{x}_{c,b}^{\dagger}\mathbf{x}_{c,b}\right\} \leq P_b^{\text{Max}}. \tag{4}$$

D. Achievable Rates and Fronthaul Constraint

A user k can be served from any subset of a BS-cluster connected to CP c with an aggregate beamforming vector $\mathbf{w}_{c,k}$ if and only if user k is associated with CP c. Therefor, we define the user-to-cloud association as a binary variable $z_{c,k}$, i.e., $z_{c,k} = 1$ if user k is associated to cloud

c. We further assume that each user can be associated to one and only one CP since, otherwise, a signal-level coordination would be required between the clouds, rather than a control-level coordination. To this end, we can write the signal to interference plus noise ratio (SINR) of user k when associated with CP c as follows

$$\mathrm{SINR}_{c,k} = \frac{\left|\mathbf{h}^{\dagger}_{c,k}\left(\mathbf{w}_{c,k} + \tilde{\mathbf{w}}_{c,k}\right)\right|^2}{\sum_{(c',k')\neq(c,k)}\left|\mathbf{h}^{\dagger}_{c',k}\left(\mathbf{w}_{c',k'} + \tilde{\mathbf{w}}_{c',k'}\right)\right|^2 + \sigma^2}, \tag{5}$$

and the achievable rate of user k associated to cloud c is bounded as

$$R_{c,k} \leq W\log_2(1 + \mathrm{SINR_{c,k}}). \tag{6}$$

The transmit power per-BS can be expressed as

$$P_b\left(\mathbf{w}, \tilde{\mathbf{w}}\right) = \frac{1}{\eta_b}\sum_{k\in\mathcal{K}}\left(\left\|\mathbf{w}_{c,b,k}\right\|_2^2 + \left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right) \tag{7}$$

where $\eta_b < 1$ is the efficiency of transmit amplifier at BS b. The required fronthaul capacity at BS b is given as

$$C_b\left(\mathbf{w}, \tilde{\mathbf{w}}\right) = \sum_{k\in\mathcal{K}}\left(\mathbb{1}\left\{\left\|\mathbf{w}_{c,b,k}\right\|_2^2\right\} + (1 - c_{f_k,b})\mathbb{1}\left\{\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right\}\right)R_{c,k}, \tag{8}$$

where $\mathbf{w} \triangleq \mathrm{vec}(\{\mathbf{w}_{c,k}|\forall(c,k)\in\mathcal{C}\times\mathcal{K}\})$, $\tilde{\mathbf{w}} \triangleq \mathrm{vec}(\{\tilde{\mathbf{w}}_{c,k}|\forall(c,k)\in\mathcal{C}\times\mathcal{K}\})$. It is obvious from equation (8) that if BS b caches file f_k requested by user k, i.e., $c_{f_k,b} = 1$ then user k does not add to the burden of the fronthaul link of BS b, especially if $\mathbf{w}_{c,b,k} = \mathbf{0}_L$.

E. Energy Efficiency at the Cloud

In the context of our paper, the EE metric of each cloud c is defined as the sum-rate of all users associated with c divided by the power consumption required to serve these users. This work takes into account the transmit power, processing power, fronthaul power consumption, and operational fixed power consumption. The latter does not depend on the number of users and is defined as P_c^{Pr}. Such operational fixed power allocation include, but not limited to, required cooling and circuitry power resources for the functionality of the C-RAN. Mathematically we define the energy efficiency at the cloud c as follows

$$f_{\mathrm{EE}}(c) \triangleq \frac{\sum_{k\in\mathcal{K}} R_{c,k}}{P_c^{\mathrm{Tx}} + g_{\mathrm{EE}}(c) + P_c^{\mathrm{Pr}}}, \tag{9}$$

where P_c^{Tx} is the total transmit power consumed by the BSs of cluster $\mathcal{B}_c$ defined as

$$P_c^{\mathrm{Tx}} = \sum_{b\in\mathcal{B}_c} P_b\left(\mathbf{w}, \tilde{\mathbf{w}}\right), \tag{10}$$

where the processing power of the CP, fronthaul and BSs, can be written as

$$g_{\mathrm{EE}}(c) = \underbrace{\sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} \mathbb{1}\{\|\mathbf{w}_{c,b,k}\|_2^2\}P_b^{\mathrm{fthl}}}_{\text{Fronthaul processing cost}} + \underbrace{\sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} \mathbb{1}\{\|\tilde{\mathbf{w}}_{c,b,k}\|_2^2\}P_k^{\mathrm{proc}}}_{\text{Processing at BS}} + \underbrace{\sum_{k\in\mathcal{K}} \mathbb{1}\{\|\mathbf{w}_{c,k}\|_2^2\}P_k^{\mathrm{proc}}}_{\text{Processing at CP}}. \tag{11}$$

Interestingly, (9) captures the trade-off between the local processing of cached files at the BSs and the fronthaul usage when the files are processed at the CP. From a computational cost perspective, local cache processing is more expensive than CP processing. The reason is that the processing is only done at the corresponding BS, i.e., the same processing process needs to be repeated at each BS in the cluster serving the user which requests locally cached content. On the other hand, the CP processes the data centrally and shares the processed data with the serving cluster. The drawback is that the CP needs to use the fronthaul links to share the processed data. Hence, in limited capacity fronthaul link regimes, local caching can significantly improve the EE performance despite the extra computational cost occurring when processing the data locally. Next, we describe the optimization problem which maximizes the sum of EE of all the clouds in an MC-RAN so as to determine the user-to-cloud assignment variables, the processing power decision variables, user-to-BS association variables, and the joint transmit beamforming optimization for all users across the network.

IV. Distributed Resource Management and Algorithms

In this section, after describing the EE maximization problem, we elaborate on the algorithmic framework. Several reformulations result in an efficient iterative algorithm that can be implemented distributively.

A. General Problem

In the context of distributed EE across the MC-RAN, we seek to jointly optimize the functional split mode for each BS, the beamforming vectors and user-to-cloud association of all users in the network subject to per BS maximum transmit power and maximum fronthaul capacity constraints. Also, we consider different constraints for the user-to-cloud association binary variables. The optimization problem under consideration can be mathematically written as:

$$\underset{\mathbf{w},\tilde{\mathbf{w}},\mathbf{z},\mathbf{r}}{\text{maximize}} \quad \sum_{c\in\mathcal{C}} f_{\text{EE}}(c) \tag{12a}$$

$$\text{subject to} \quad (2),(3),$$

$$P_b\left(\mathbf{w},\tilde{\mathbf{w}}\right) \leq P_b^{\text{Max}} \qquad \forall b\in\mathcal{B}_c, \forall c\in\mathcal{C}, \tag{12b}$$

$$C_b\left(\mathbf{w},\tilde{\mathbf{w}}\right) \leq F_{b,c} \qquad \forall b\in\mathcal{B}_c, \forall c\in\mathcal{C}, \tag{12c}$$

$$\mathrm{SINR_{c,k}} \geq 2^{\mathrm{R_{c,k}/W}} - 1 \qquad \forall k\in\mathcal{K}, \forall c\in\mathcal{C}, \tag{12d}$$

$$\sum_{c\in\mathcal{C}} z_{c,k} = 1 \qquad \forall k\in\mathcal{K}, \tag{12e}$$

$$z_{c,k}\in\{0,1\} \qquad \forall k\in\mathcal{K}, \forall c\in\mathcal{C}, \tag{12f}$$

$$\left\|\mathbf{w}_{c,b,k}\right\|_2^2 \leq M z_{c,k} \qquad \forall k\in\mathcal{K}, \forall b\in\mathcal{B}_c, \forall c\in\mathcal{C}, \tag{12g}$$

$$\sum_{k\in\mathcal{K}} z_{c,k} \leq K_c^{\text{Max}} \qquad \forall c\in\mathcal{C}, \tag{12h}$$

where $\mathbf{z} = \mathrm{vec}(\{z_{c,k}|\forall\,(c,k)\in\mathcal{C}\times\mathcal{K}\})$, $\mathbf{r} = \mathrm{vec}(\{R_{c,k}|\forall\,(c,k)\in\mathcal{C}\times\mathcal{K}\})$ and M is a big positive integer $M\in\mathbb{R}_{++}$, e.g., $M = 2\sum_{(c,b)\in\mathcal{C}\times\mathcal{B}_c} P_b^{\max}$. $P_b^{\max}$ and $F_{b,c}$ are the maximum transmit power and the fronthaul capacity of BS b in cloud c, respectively. K_c^{Max} is the maximum number of users that can connect to cloud c. Constraint (12b) represents the maximum transmit power available to BS b, and constraint (12c) represents the available fronthaul capacity of BS b connected to CP c. Constraint (12d) gives an upper bound on the maximum achievable rate of user k when assigned to cloud c.

Constraints (12e)-(12f) assure that each user can be associated with one and only one CP. Constraints (12g) represent the big-M constraints and can be read as follows: if the CP c is associated with user k, then constraint (12g) is deactivated [36]. Otherwise, k is not associated with c, (12g) forces the corresponding beamforming coefficients in $\mathbf{w}_{c,b,k}$ to zero. The number of associated users to cloud c does not exceed a given maximum number of users, which is ensured by (12h). The constraints in (2), (3) make sure that the data of each user can be either processed locally at the BS or at the CP but not both at the same time. Constraint (14b) makes sure that the number of associated users to cloud c does not exceed a given maximum number of users.

The above optimization (12) is over binary variables $\mathbf{z}$, continuous beamforming vectors $\mathbf{w}$ and $\tilde{\mathbf{w}}$, and rates $\mathbf{r}$. Problem (12) is challenging to solve due to the non-convexity of the objective function and constraints (12c)-(12f), besides the discrete nature of variables $\mathbf{z}$.

B. Algorithmic Framework

The optimization of the association variables and beamforming vectors in (12) is hard to tackle jointly and may be computationally prohibitive to solve globally. Therefore, our paper proposes adopting a two-step optimization approach. In the first step, we adopt an auxiliary network utility function that represents the benefit of associating a user k to cloud c. Then we formulate a generalized assignment problem to compute the optimal user-to-cloud association for a given utility function. Afterwards, in the second stage, given the user-to-cloud assignment solution, we solve the optimization problem (12) using a l_0-norm relaxation followed by a successive inner-convex approximation approach. We start by discussing the generalized assignment formulation to solve the user-to-cloud association problem.

C. Generalized Assignment Problem

Since (12) is a too complicated problem to solve directly, we propose an ad-hoc solution to find the user-to-cloud association as given to the problem. Let $\mathcal{U}(c,k)$ be a utility function to measure the benefit of associating user k with cloud c. A reasonable choice of $\mathcal{U}(c,k)$ is the following EE-like function:

$$\mathcal{U}(c,k) = \frac{R_{c,k}}{\sum_{b\in\mathcal{B}_c} \frac{1}{\eta_b}\left(\left\|\mathbf{w}_{c,b,k}\right\|_2^2 + \left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right) + P_k^{\text{proc}}}. \tag{13}$$

The intuition behind such choice is two-fold. First, the utility function in (13) defines the benefit of associating user k with cloud c as the fraction between the achievable rate for such an association and the processing and transmit power costs. Such a utility helps mimicking a reasonable energy efficiency of the MC-RAN. It is clear from eq. (13) that the utility function depends mainly on the aggregate beamforming vector from cloud c to user k. Second, such choice helps formulating a generalized assigned problems, which allows us to derive efficient algorithms to find the association variables $\mathbf{z}$; thereby alleviating the complexity of the solution of the complex problem (12). We start first by assuming a practical choice where the beamforming vectors have maximum ratio transmitter (MRT) structures. Specifically, the beamforming vector from cloud c to user k is defined as $\left\{\mathbf{w}_{c,k} \in \mathbb{C}^{B_cL\times 1} = \frac{\mathbf{h}_{c,k}}{\|\mathbf{h}_{c,k}\|_2^2}, \quad \forall k \in \mathcal{K}\right\}$. Now we can formulate our generalized assignment problem as follows

$$\underset{\mathbf{z}}{\text{maximize}} \quad \sum_{(c,k)\in\mathcal{C}\times\mathcal{K}} z_{c,k}\ \mathcal{U}(c,k) \tag{14a}$$

subject to

$$\sum_{k\in\mathcal{K}} z_{c,k} \leq K_c^{\text{Max}} \qquad \forall c \in \mathcal{C}, \tag{14b}$$

$$\sum_{c\in\mathcal{C}} z_{c,k} \leq 1 \qquad \forall k \in \mathcal{K}, \tag{14c}$$

$$z_{c,k} \in \{0,1\} \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C}. \tag{14d}$$

The optimization is carried over the binary association variables $\mathbf{z}$. Constraint (14b) makes sure that the number of associated users to cloud c does not exceed a given maximum number of users, i.e., K_c^{Max}. That is, each cloud can handle a maximum number of users. This constant, for instance can be chosen according to the number of total transmit antennas controlled by cloud c. Note that the constraint (14b) also balances the load (number of assigned users) between the clouds which can help significantly improving the total EE in MC-RANs. Problem (14) is known to be a NP-hard problem [37]. In this work, we use global optimization methods such as the branch and cut algorithm for problem (14). To implement the generalized assignment problem (14) in a distributed manner, the authors in [12] and [37] propose using an auction-based iterative algorithm, where only reasonable information exchange between the clouds is required. A similar algorithm is used to tackle a multi-robot assignment problem in [38].

D. l_0-relaxation

After determining the association variables as described above, the problem of determining the optimal joint beamforming vectors that maximize the sum of EE can be expressed as follows:

$$\underset{\mathbf{w},\tilde{\mathbf{w}},\mathbf{r}}{\text{maximize}} \quad \sum_{c\in\mathcal{C}} f_{\text{EE}}(c) \tag{15a}$$

$$\text{subject to} \quad (2), (3), (12\text{b}), (12\text{c}), (12\text{d}), (12\text{g}).$$

Note that this problem is now of simpler form compared to the general problem (12) due to the lack of the discrete association variables $\mathbf{z}$. However, this problem is still challenging due to the non-convexity of the objective function and feasible set. Thus we next tackle this by reformulating the non-convex constraints (12c) and (12d). We begin by reformulating problem (12) as follows:

$$\underset{\mathbf{w},\tilde{\mathbf{w}},\boldsymbol{\gamma},\mathbf{r}}{\text{maximize}} \quad \sum_{c\in\mathcal{C}} f_{\text{EE}}(c) \tag{16a}$$

$$\text{subject to} \quad (2), (3), (12\text{b}), (12\text{c}), (12\text{g}),$$

$$R_{c,k} \leq W \log_2\left(1+\gamma_{c,k}\right) \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \tag{16b}$$

$$\sigma^2 + \sum_{(c',k')\neq(c,k)} \left|\mathbf{h}_{c',k}^{\dagger}\left(\mathbf{w}_{c',k'}+\tilde{\mathbf{w}}_{c',k'}\right)\right|^2 - \frac{\left|\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}+\tilde{\mathbf{w}}_{c,k}\right)\right|^2}{\gamma_{c,k}} \leq 0 \quad \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \tag{16c}$$

where we introduce the variables $\boldsymbol{\gamma} = \text{vec}(\{\gamma_{c,k} | \forall\,(c,k) \in \mathcal{C}\times\mathcal{K}\})$ to reformulate the maximum achievable rate constraint (12d) into the non-convex constraints (16b)-(16c). (16c) is now in the form of difference of convex (DC) functions which can be tackled using an efficient SICA approach. Moreover, the indicator functions $\mathbb{1}\{\|\tilde{\mathbf{w}}_{c,b,k}\|_2^2\}$, $\mathbb{1}\{\|\mathbf{w}_{c,k}\|_2^2\}$, and $\mathbb{1}\{\|\mathbf{w}_{c,b,k}\|_2^2\}$ which decide if the data of user k is processed locally at BS b, if the CP c processes the data of user k and whether the BS b is in the serving cluster of user k or not, respectively, present additional hurdles within the framework of the challenging problem (16). We note that the benefit of using indicator functions is to determine the decision variables exclusively based on beamforming vectors. To deal with the challenging non-convex discrete indicator functions, we assort to the l_0-norm relaxation as described next. First, we note that the indicator function in the objective (16a) and the fronthaul constraint (12c) can be equivalently expressed as a l_0-norm of the beamforming vectors. We can write $\mathbb{1}\{\|\tilde{\mathbf{w}}_{c,b,k}\|_2^2\} \triangleq \|\|\tilde{\mathbf{w}}_{c,b,k}\|_2^2\|_0$, $\mathbb{1}\{\|\mathbf{w}_{c,b,k}\|_2^2\} \triangleq \|\|\mathbf{w}_{c,b,k}\|_2^2\|_0$, and $\mathbb{1}\{\|\mathbf{w}_{c,k}\|_2^2\} \triangleq \|\|\mathbf{w}_{c,k}\|_2^2\|_0$. This equivalence is important since the l_0-norm function, which is a discrete function can be approximated with a weighted l_1-norm convex function [8]. To enable the use of such approximation in the context of our paper, we write the function $\|\|\mathbf{w}_{c,b,k}\|_2^2\|_0$ as a reweighed l_1-norm as follows:

$$\|\|\mathbf{w}_{c,b,k}\|_2^2\|_0 = \beta_{c,b,k}\|\mathbf{w}_{c,b,k}\|_2^2. \tag{17}$$

Here, $\beta_{c,b,k}$ is a constant weight associated with BS b in $\mathcal{B}_c$ and user k and is defined in this work as

$$\beta_{c,b,k} = \frac{1}{\delta + \|\mathbf{w}_{c,b,k}\|_2^2}, \tag{18}$$

where $\delta > 0$ is a regularization constant[1]. In a similar manner we define $\tilde{\beta}_{c,b,k}$ and $\beta_{c,k}$. Since the l_1-norm is applied to a quadratic function of the beamforming vectors, the resulting approximation

[1] The regularization parameter can be chosen very small to make the approximation error arbitrary small. In the simulations we choose $\delta = 10^{-12}$ and we set the beamforming vector $\mathbf{w}_{c,b,k} = \mathbf{0}$ in iteration t if $\|\mathbf{w}_{c,b,k}\|_2^2 \leq \delta$. This results in negligible error on the achievable data rate of user k.

is a smooth continuous function which is easier to optimize as compared to a non-smooth l_0-norm. The weights in (18) are chosen since BSs with a small transmit power allocated to user k get higher weights $\beta_{c,b,k}$, and eventually drop out of the cluster of BSs sharing the message of user k. Only those BSs which have non-negligible transmit power allocated to user k participate in the transmission to user k. The reformulated objective now reads as

$$f_{2,\text{EE}}(c) \triangleq \frac{\sum_{k\in\mathcal{K}} R_{c,k}}{P_c^{\text{Tx}} + p_{2,\text{EE}}(c) + P_c^{\text{Pr}}}, \tag{19}$$

where

$$p_{2,\text{EE}}(c) = \underbrace{\sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} \beta_{c,b,k}\left\|\mathbf{w}_{c,b,k}\right\|_2^2 P_b^{\text{fthl}}}_{\text{Fronthaul processing cost}} + \underbrace{\sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} \tilde{\beta}_{c,b,k}\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2 P_k^{\text{proc}}}_{\text{Processing at BS}} + \underbrace{\sum_{k\in\mathcal{K}} \beta_{c,k}\left\|\mathbf{w}_{c,k}\right\|_2^2 P_k^{\text{proc}}}_{\text{Processing at CP}}. \tag{20}$$

Note that the function $p_{2,\text{EE}}(c)$ is a l_0-norm relaxed formulation of (11). The reformulated problem (16) can now be written as

$$\underset{\mathbf{w},\tilde{\mathbf{w}},\boldsymbol{\gamma},\mathbf{r}}{\text{maximize}} \quad \sum_{c\in\mathcal{C}} f_{2,\text{EE}}(c) \tag{21a}$$

$$\text{subject to} \quad (3), (12\text{b}), (12\text{c}), (12\text{g}), (16\text{b}), (16\text{c}),$$

$$\beta_{c,b,k}\left\|\mathbf{w}_{c,b,k}\right\|_2^2 + \tilde{\beta}_{c,b,k}\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2 \leq 1 \qquad \forall k\in\mathcal{K}, \forall b\in\mathcal{B}_c, \forall c\in\mathcal{C}, \tag{21b}$$

$$C_{2,b}\left(\mathbf{w}\right) \leq F_{b,c} \qquad \forall b\in\mathcal{B}_c, \forall c\in\mathcal{C}. \tag{21c}$$

Note that constraints (2) and (12c) are now replaced by (21b) and (21c), respectively. Before reformulating the fronthaul capacity constraint, we will first elaborate on the second term in (8), namely $(1-c_{f_k,b})\mathbb{1}\left\{\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2\right\}$. If a BS b caches file f_k then $c_{f_k,b}=1$, which means the fronthaul link of BS b is not affected by user k. Otherwise, if the BS b does not cache the requested file by user k, we know from previous definitions and (3) that $\tilde{\mathbf{w}}_{c,b,k} = \mathbf{0}_L$. Based on these observations, we can conclude that the second term in (8) does not influence the fronthaul link and can thus be ignored. The reformulated fronthaul term is now

$$C_{2,b}\left(\mathbf{w}\right) = \sum_{k\in\mathcal{K}} \beta_{c,b,k}\left\|\mathbf{w}_{c,b,k}\right\|_2^2 R_{c,k}. \tag{22}$$

Through these reformulations, we overcame the discrete nature of the original problem (16). However, the non-convex problem (21) remains difficult to solve from an optimization perspective, and is rather tackled next using fractional programming and success inner-convex approximation.

E. Fractional Programming and Successive Inner-Convex Approximations

Note that the objective function in (21a) is a fraction of linear and convex functions. Hence, it is appealing to apply Dinkelbach algorithm to solve problem (21) [39]. However, considering the non-convex feasible set of problem (21), a direct application of the algorithm would be inefficient [40]. Hence, in each iteration of the Dinkelbach algorithm, we need to solve a non-convex problem and obtain a stationary solution, which is computationally prohibitive, especially when the problem becomes larger. The non-convexity stems from constraints (21c), (16b) and (16c), i.e., being the available fronthaul capacity constraints and the reformulated achievable rate constraints, respectively. To overcome this difficulty, we use a SICA approach combined with the Dinkelbach algorithm for obtaining a stationary solution. Our algorithm guarantees convergence to a KKT point of problem (21). For a more general overview regarding fractional programming for EE maximization, especially a general formulation of Dinkelbach's algorithm, please refer to the work [39]. A more detailed description of sequential optimization can be found in [40] and [41]. We start by reformulating problem (21) to get a problem formulation that is amenable to apply SICA techniques.

F. Convexification of Problem (21)

First we tackle constraint (21c) by introducing slack variables $\mathbf{t} = \mathrm{vec}(\{t_{k,b}|\forall\,(k,b) \in \mathcal{K} \times \mathcal{B}\})$, $\tilde{\mathbf{t}} = \mathrm{vec}(\{\tilde{t}_{k,b}|\forall\,(k,b) \in \mathcal{K} \times \mathcal{B}\})$, and $\mathbf{u} = \mathrm{vec}(\{u_{c,k}|\forall\,(c,k) \in \mathcal{C} \times \mathcal{K}\})$. Then, for all k, b, and c, define the following auxiliary constraints

$$\beta_{c,b,k}\left\|\mathbf{w}_{c,b,k}\right\|_2^2 \leq t_{k,b}, \tag{23}$$

$$\tilde{\beta}_{c,b,k}\left\|\tilde{\mathbf{w}}_{c,b,k}\right\|_2^2 \leq \tilde{t}_{k,b}, \tag{24}$$

$$\sum_{b\in\mathcal{B}_c} \beta_{c,b,k}\left\|\mathbf{w}_{c,b,k}\right\|_2^2 \leq u_{c,k}. \tag{25}$$

The fronthaul capacity constraint (21c) is reformulated using slack variable $\mathbf{t}$ as follows

$$\sum_{k\in\mathcal{K}} t_{k,b}R_{c,k} \leq F_{b,c} \qquad \forall b \in \mathcal{B}_c, \forall c \in \mathcal{C}. \tag{26}$$

This function is non-convex as it is bilinear in the optimization variables. However, using some algebraic transformations the term can be equivalently written as

$$\sum_{k\in\mathcal{K}} \frac{1}{4}\Big(\underbrace{\left(t_{k,b} + R_{c,k}\right)^2}_{\text{convex}} - \underbrace{\left(t_{k,b} - R_{c,k}\right)^2}_{\text{convex}}\Big) \leq F_{b,c}. \tag{27}$$

This formulation is now in the form of DC functions (convex plus concave function). Even though function (27) is still non-convex, it is in the form of DC, which allows for applying SICA methods. The idea of SICA is to find a convex surrogate upper-bound to the non-convex function (27). This can be done by keeping the convex part and linearizing the concave one using the first-order Taylor expansion. We define the function $g_1(\mathbf{t}, \mathbf{r}, \mathbf{t}', \mathbf{r}')$ as

$$g_1(\mathbf{t}, \mathbf{r}, \mathbf{t}', \mathbf{r}') \triangleq \sum_{k \in \mathcal{K}} \Big((t_{k,b} + R_{c,k})^2 - 2\left(t'_{k,b} - R'_{c,k}\right)\left(t_{k,b} - R_{c,k}\right) + \left(t'_{k,b} - R'_{c,k}\right)^2 \Big) - 4F_{b,c}. \quad (28)$$

Here $\mathbf{t}' = \mathrm{vec}(\{t'_{k,b} | \forall\,(k,b) \in \mathcal{K} \times \mathcal{B}\})$ and $\mathbf{r}' = \mathrm{vec}(\{R'_{c,k} | \forall\,(c,k) \in \mathcal{C} \times \mathcal{K}\})$ are feasible fixed values, which satisfy the previously defined constraints (23) and (26). These feasible fixed values will be updated iteratively, such that the feasible set is refined in every iteration of the SICA.

Lemma 1. *For all feasible values* $(\mathbf{t}', \mathbf{r}')$ *and all* $(c,b) \in (\mathcal{C}, \mathcal{B}_c)$ *the function* $g_1(\mathbf{t}, \mathbf{r}, \mathbf{t}', \mathbf{r}')$ *satisfies*

$$g_1(\mathbf{t}, \mathbf{r}, \mathbf{t}', \mathbf{r}') \geq \sum_{k \in \mathcal{K}} t_{k,b} R_{c,k} - F_{b,c}. \quad (29)$$

Proof. Please refer to Appendix A. □

Now, the fronthaul capacity constraint (21c) is a convex function of the optimization variables. As a next step, we shift our focus to the non-convex constraint (16b). We define $g_2(\boldsymbol{\gamma}, \mathbf{r})$ as follows

$$g_2(\boldsymbol{\gamma}, \mathbf{r}) = W \log_2\left(1 + \gamma_{c,k}\right) - R_{c,k} \geq 0. \quad (30)$$

The function in (30) is non-convex in $\gamma_{c,k}$; however, it is amenable for applying SICA methods. We approximate the non-convex set by linearizing the concave part, namely $\log_2\left(1 + \gamma_{c,k}\right)$, around $\boldsymbol{\gamma}'$ using the first-order Taylor expansion. The convex upper-bound to $g_2(\boldsymbol{\gamma}, \mathbf{r})$ is defined as

$$g_2(\boldsymbol{\gamma}, \mathbf{r}, \boldsymbol{\gamma}') \triangleq \log_2\left(1 + \gamma'_{c,k}\right) + \frac{1}{\ln(2)\left(1 + \gamma'_{c,k}\right)}\left(\gamma_{c,k} - \gamma'_{c,k}\right) - \frac{R_{c,k}}{W} \geq 0. \quad (31)$$

Variables $\boldsymbol{\gamma}' = \mathrm{vec}(\{\gamma'_{c,k} | \forall\,(c,k) \in \mathcal{C} \times \mathcal{K}\})$ are feasible fixed values as discussed next. Since the achievable rate constraint is now in a convex form, we tackle the next constraint (16c) defining

$$\zeta^{+}(\mathbf{w}, \tilde{\mathbf{w}}) - \zeta^{-}(\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\gamma}) \leq 0, \quad (32)$$

where

$$\zeta^{+}(\mathbf{w}, \tilde{\mathbf{w}}) = \sigma^2 + \sum_{(c',k') \neq (c,k)} \left|\mathbf{h}^{\dagger}_{c',k}\left(\mathbf{w}_{c',k'} + \tilde{\mathbf{w}}_{c',k'}\right)\right|^2, \quad (33)$$

and

$$\zeta^{-}(\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\gamma}) = \frac{\left|\mathbf{h}^{\dagger}_{c,k}\left(\mathbf{w}_{c,k} + \tilde{\mathbf{w}}_{c,k}\right)\right|^2}{\gamma_{c,k}}. \quad (34)$$

The formulation in (32) is in a form of DC functions as ζ^{+} is quadratic in $\mathbf{w}$ and $\tilde{\mathbf{w}}$, and ζ^{-} is a rational function with quadratic numerator and non-negative linear denominator, which is known to be convex [42]. The following lemma states a viable first-order approximation of such functions.

Lemma 2. *The first-order approximation of a function in the form of $\zeta(\mathbf{x},\xi) = \frac{|\mathbf{x}|^2}{\xi}$, where $\mathbf{x} \in \mathbb{C}^P$ and $\xi > 0$, around the feasible point $(\mathbf{x}',\xi')$ satisfies*

$$\zeta(\mathbf{x},\xi) \geq \tilde{\zeta}(\mathbf{x},\xi,\mathbf{x}',\xi') = \frac{2\Re\{\left(\mathbf{x}'\right)^{\dagger}\mathbf{x}\}}{\xi'} - \frac{\xi}{\left(\xi'\right)^2}\left|\mathbf{x}'\right|^2. \tag{35}$$

Proof. Please refer to Appendix B. □

In order to obtain a convex formulation, we linearize the function ζ^{-} around the feasible point $(\mathbf{w}',\tilde{\mathbf{w}}',\boldsymbol{\gamma}')$ and according to Lemma 2 to get:

$$\begin{aligned}\frac{\left|\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}+\tilde{\mathbf{w}}_{c,k}\right)\right|^2}{\gamma_{c,k}} \geq \frac{2}{\gamma_{c,k}'}\Re\left\{\left(\mathbf{w}_{c,k}'+\tilde{\mathbf{w}}_{c,k}'\right)^{\dagger}\mathbf{h}_{c,k}\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}+\tilde{\mathbf{w}}_{c,k}\right)\right\} \\ -\frac{\gamma_{c,k}}{\left(\gamma_{c,k}'\right)^2}\left|\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}'+\tilde{\mathbf{w}}_{c,k}'\right)\right|^2.\end{aligned} \tag{36}$$

Inserting this convex upper-bound into the non-convex formulation (32) we get the inner convex approximation denoted as

$$\begin{aligned}g_3(\mathbf{w},\tilde{\mathbf{w}},\boldsymbol{\gamma},\mathbf{w}',\tilde{\mathbf{w}}',\boldsymbol{\gamma}') \triangleq \sigma^2 + \sum_{(c',k')\neq(c,k)}\left|\mathbf{h}_{c',k}^{\dagger}\left(\mathbf{w}_{c',k'}+\tilde{\mathbf{w}}_{c',k'}\right)\right|^2 \\ -\sum_{c\in\mathcal{C}}\left[\frac{2}{\gamma_{c,k}'}\Re\left\{\left(\mathbf{w}_{c,k}'+\tilde{\mathbf{w}}_{c,k}'\right)^{\dagger}\mathbf{h}_{c,k}\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}+\tilde{\mathbf{w}}_{c,k}\right)\right\}+\frac{\gamma_{c,k}}{\left(\gamma_{c,k}'\right)^2}\left|\mathbf{h}_{c,k}^{\dagger}\left(\mathbf{w}_{c,k}'+\tilde{\mathbf{w}}_{c,k}'\right)\right|^2\right].\end{aligned} \tag{37}$$

Note that $\mathbf{w}' = \mathrm{vec}(\{w_{c,k}'|\forall(c,k)\in\mathcal{C}\times\mathcal{K}\})$, $\tilde{\mathbf{w}}' = \mathrm{vec}(\{\tilde{w}_{c,k}'|\forall(c,k)\in\mathcal{C}\times\mathcal{K}\})$ and the previously defined $\boldsymbol{\gamma}'$ are feasible fixed values fulfilling constraints (12b), (12g), (16c), (21b), (21c), and (30).

As a last convexification step, the objective function (21a) can be written in a more compact form

$$f_{3,\mathrm{EE}}(c) \triangleq \frac{\sum_{k\in\mathcal{K}}R_{c,k}}{\sum\limits_{(b,k)\in\mathcal{B}_c\times\mathcal{K}}t_{k,b}P_b^{\mathrm{fthl}} + \sum\limits_{(b,k)\in\mathcal{B}_c\times\mathcal{K}}\tilde{t}_{k,b}P_k^{\mathrm{proc}} + \sum\limits_{k\in\mathcal{K}}u_{c,k}P_k^{\mathrm{proc}} + P_c^{\mathrm{Tx}} + P_c^{\mathrm{Pr}}}. \tag{38}$$

As shown implicitly from equation (38), the slack variables $t_{k,b}$, $\tilde{t}_{k,b}$, and $u_{c,k}$ denote at which point of the system the data from user k is processed and distributed. To be more precise, the variables $u_{c,k}$ and $t_{k,b}$ refer to processing the data of user k at cloud c. The variable $u_{c,k}$ refers to the data from user k being processed at cloud c and then being forwarded to all BSs participating in serving k, i.e., $t_{k,b}$ at all BSs $b \in \mathcal{B}_c$. In that case user k does not only burden the fronthaul link of all BSs with P_b^{fthl} but also causes the allocation of processing power P_k^{proc} at cloud c. In contrast, $\tilde{t}_{k,b}$ refers to processing the data of user k locally at BS b in case the requested file for user k is cached there, i.e., $c_{f_k,b} = 1$. The BS then has to allocate P_k^{proc}, which in comparison is better from an EE perspective. Note that data can only be processed in one place and not be split, hence, both cases exclude each other.

Combining the previous results into one optimization problem, we now have the functions (37),(31), and (28) as well as the constraints (3), (12b), (21b), (23), (24), and (25) defining a convex feasible set, which is an inner-approximation of the non-convex feasible set of problem (21). The formulation of such approximated problem becomes

$$\begin{aligned}
&\underset{\mathbf{y}}{\text{maximize}} \quad \sum_{c\in\mathcal{C}} f_{3,\text{EE}}(c) && && \text{(39a)}\\
&\text{subject to} \quad (3), (12\text{b}), (21\text{b}), (23), (24), (25), && && \\
&g_1(\mathbf{t}, \mathbf{R}; \mathbf{t}', \mathbf{R}') \leq 0 && \forall b \in \mathcal{B}_c, \forall c \in \mathcal{C}, && \text{(39b)}\\
&g_2(\boldsymbol{\gamma}, \mathbf{R}; \boldsymbol{\gamma}') \geq 0 && \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, && \text{(39c)}\\
&g_3(\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\gamma}; \mathbf{w}', \tilde{\mathbf{w}}', \boldsymbol{\gamma}') \leq 0 && \forall k \in \mathcal{K}, \forall c \in \mathcal{C}. && \text{(39d)}
\end{aligned}$$

With $\mathbf{y} = \left[\mathbf{w}^T, \tilde{\mathbf{w}}^T, \mathbf{t}^T, \tilde{\mathbf{t}}^T, \mathbf{u}^T, \boldsymbol{\gamma}^T, \mathbf{r}^T\right]^T$ a vector containing all optimization variables and $\mathbf{y}' = \left[\mathbf{w}'^T, \tilde{\mathbf{w}}'^T, \mathbf{t}'^T, \tilde{\mathbf{t}}'^T, \mathbf{u}'^T, \boldsymbol{\gamma}'^T, \mathbf{r}'^T\right]^T \in \mathcal{Y}$ is a vector containing all fixed values, where $\mathcal{Y}$ is the convex feasible set defined by the constraints (3), (12b), (21b), and (23)-(25).

G. Iterative Algorithm

Problem (39) is the inner convex approximation of problem (21) and can be solved iteratively using a combined SICA and Dinkelbach algorithm. In order to apply Dinkelbach algorithm, we first define $g_4(c)$ and $g_5(c)$ as the numerator and denominator of $f_{3,\text{EE}}(c)$, respectively

$$g_4(c) \triangleq \sum\nolimits_{k\in\mathcal{K}} R_{c,k}, \tag{40}$$

and

$$g_5(c) \triangleq \sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} t_{k,b} P_b^{\text{fthl}} + \sum_{(b,k)\in\mathcal{B}_c\times\mathcal{K}} \tilde{t}_{k,b} P_k^{\text{proc}} + \sum_{k\in\mathcal{K}} u_{c,k} P_k^{\text{proc}} + P_c^{\text{Tx}} + P_c^{\text{Pr}}. \tag{41}$$

To solve the fractional problem (39) we iteratively search for a unique solution to an easier auxiliary problem, thereby we define the auxiliary problem as

$$F(c; \lambda_j(c)) = \max_{\mathbf{y}\in\mathcal{Y}} \left\{ g_4(c) - \lambda_j(c) g_5(c) \right\}, \tag{42}$$

where we update $\lambda_j(c)$ after each iteration according to

$$\lambda_{j+1}(c) = \frac{g_4(c)}{g_5(c)}. \tag{43}$$

To solve problem (39), we distinguish between an outer and an inner loop. In the outer loop we update the feasible fixed values for SICA, initialize λ_0 for the inner loop, and check for convergence. In the inner loop we use the Dinkelbach algorithm, solving $F(c; \lambda_j(c))$ iteratively. In the end, this produces the global optimal solution to the underlying fractional program (39) with optimal values $\hat{\mathbf{y}}_\nu$. At iteration ν of the outer loop, we refine the feasible set $\mathbf{y}'$, using the optimal values $\hat{\mathbf{y}}_\nu$ as fixed values for the next iteration. The algorithm stops when it converges to a stationary solution, hence, we compare the objective $f^{\nu}_{3,\text{EE}}(c)$ at iteration ν to the previous objective $f^{\nu-1}_{3,\text{EE}}(c)$. The detailed steps are summarized in Algorithm 1.

Theorem 1. *Algorithm 1 converges to a stationary point of the relaxed problem* (21).

Proof. Please refer to Appendix C. □

Algorithm 1 Combined SICA and Dinkelbach algorithm.

```
$\nu = 0;\ \mathbf{y}' \in \mathcal{Y};\ f^{-1}_{3,\text{EE}}(c) = 0;$
while $\left| f^{\nu}_{3,\text{EE}}(c) - f^{\nu-1}_{3,\text{EE}}(c) \right| > \epsilon$ do
    $j = 0, \lambda_j(c)$ with $F(c; \lambda_j(c)) \geq 0;$
    while $F(c; \lambda_j(c)) > \epsilon$ do
        $\hat{\mathbf{y}}_\nu = \arg\max_{\mathbf{y} \in \mathcal{Y}} \left\{ \sum_{c \in \mathcal{C}} g_4(c) - \lambda_j(c) g_5(c) \right\};$
        $\lambda_{j+1}(c) = \frac{g_4(c)}{g_5(c)};$
        $j = j + 1;$
    end while
    $\mathbf{y}' = \hat{\mathbf{y}}_\nu;$
    $\nu = \nu + 1;$
end while
```

To start the algorithm, the fixed values $\mathbf{y}'$ are computed. First, the beamforming vectors are initialized with feasible MRC beamformers [43]. Please note that in order to determine $\tilde{\mathbf{w}}'$, the cache placement and the user requests have to be known. Based on these beamformers, the variables $\boldsymbol{\gamma}', \mathbf{r}'$ can be computed using equations (16b) and (16c). At last, we compute $\mathbf{t}', \tilde{\mathbf{t}}', \mathbf{u}'$ replacing the inequalities in (23)-(25) with equalities. To initialize $\lambda_0(c)$ with $F(c; \lambda_0(c)) \geq 0$ we use the feasible fixed values $\mathbf{y}'$ and compute

$$\lambda_0(c) = \frac{g_4(c)}{g_5(c)}. \tag{44}$$

H. Beamforming and Fixed Clusters

Through previously described methods, we are able to find a stationary solution to problem (21). In particular, we find optimal association variables $\mathbf{t}, \tilde{\mathbf{t}}, \mathbf{u}$ that define the serving clusters. We now fix these association clusters and focus on finding optimal beamforming vectors by revisiting problem (39). The optimization variables now become group sparse variables $\mathbf{y}_2 = \left[\mathbf{w}^T, \tilde{\mathbf{w}}^T, \boldsymbol{\gamma}^T, \mathbf{r}^T\right]^T$. The fixed feasible variables are $\mathbf{y}_2' = \left[\mathbf{w}'^T, \tilde{\mathbf{w}}'^T, \boldsymbol{\gamma}'^T, \mathbf{r}'^T\right]^T$. The optimization problem with fixed clusters can be written as

$$\underset{\mathbf{y}_2}{\text{maximize}} \quad \sum_{c \in \mathcal{C}} f_{3,\text{EE}}(c), \tag{45a}$$

$$\text{subject to} \quad (12\text{b}), (26)$$

$$g_2(\boldsymbol{\gamma}, \mathbf{r}; \boldsymbol{\gamma}', \mathbf{r}') \geq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C}, \tag{45b}$$

$$g_3(\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\gamma}; \mathbf{w}', \tilde{\mathbf{w}}', \boldsymbol{\gamma}') \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C}. \tag{45c}$$

We can use a slightly simpler version of Algorithm 1 to solve (45), where the set of optimization variables is reduced to $\{\mathbf{w}, \tilde{\mathbf{w}}, \boldsymbol{\gamma}, \mathbf{r}\}$.

I. Distributed Implementation

Algorithm 1 can be implemented in a distributed manner, so that the beamforming design is done at the individual clouds for their assigned users. Using this method, the clouds only need to exchange certain information between each other. A cloud only needs interference information like $\sum_{(c',k') \neq (c,k)} \left|\mathbf{h}_{c',k}^{\dagger} \left(\mathbf{w}_{c',k'} + \tilde{\mathbf{w}}_{c',k'}\right)\right|^2$ from all other clouds $c \neq c'$ to solve the problem locally. With predetermined user-to-cloud association, a local implementation of problem (39) boils down to a simpler problem, whereas only a subset of $\mathbf{Z}$ is utilized at the respective CPs. We let $\mathcal{K}_c$ be the set of users served by CP c, i.e., $\mathcal{K}_c \triangleq \{k \in \mathcal{K} | \, \exists k \in \mathcal{K} : z_{c,k} = 1\}$. The beamforming vectors become $\mathbf{w}_c = \text{vec}(\{\mathbf{w}_{c,k} | \forall k \in \mathcal{K}_c\})$, $\tilde{\mathbf{w}}_c = \text{vec}(\{\tilde{\mathbf{w}}_{c,k} | k \in \mathcal{K}_c\})$, the serving clusters effectively reduce to $\mathbf{t}_c = \text{vec}(\{t_{k,b} | \, (k,b) \in \mathcal{K}_c \times \mathcal{B}_c\})$, $\tilde{\mathbf{t}}_c = \text{vec}(\left\{\tilde{t}_{k,b} | \, (k,b) \in \mathcal{K}_c \times \mathcal{B}_c\right\})$, and $\mathbf{u}_c = \text{vec}(\{u_{c,k} | k \in \mathcal{K}_c\})$. Equally $\boldsymbol{\gamma}_c = \text{vec}(\{\gamma_{c,k} | k \in \mathcal{K}_c\})$ and $\mathbf{R}_c = \text{vec}(\{R_{c,k} | k \in \mathcal{K}_c\})$ are reduced. Note that these variables are independent from each other, thus the use of distributed computing saves computing overhead as compared to the centralized implementation. Extending Algorithm 1, the CPs exchange interference information in every iteration of the outer loop as an additional step, i.e., between steps 8 and 9.

J. Complexity Analysis

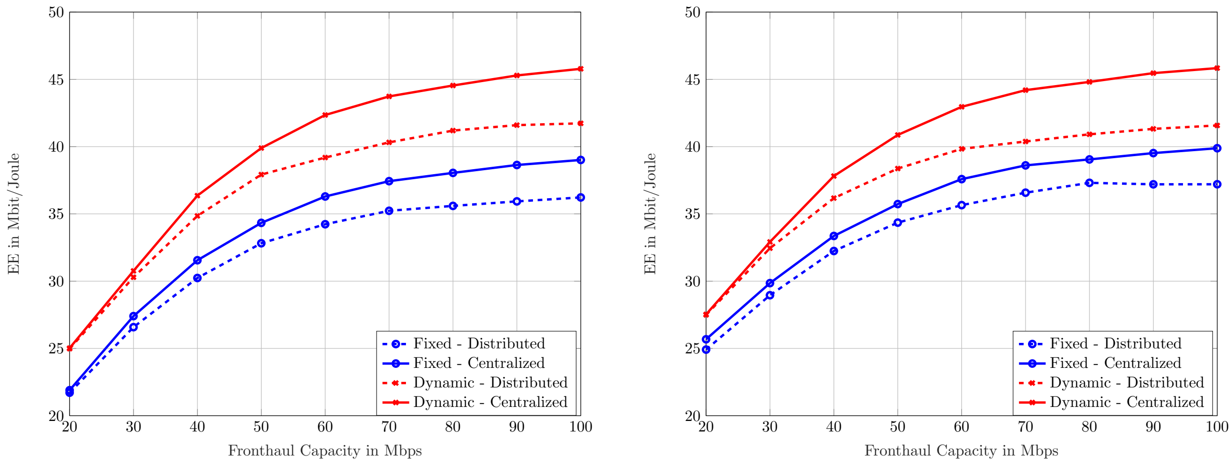
Now we focus on the overall computational complexity of our proposed method. Starting with the inner loop that utilizes Dinkelbach algorithm, the overall complexity depends on each subproblems' complexity as well as the convergence rate of the auxiliary problem series. Each subproblem (39) has linear and quadratic terms as objective, hence we have a quadratic convex problem that can be cast as a second order cone program (SOCP) [44]. Such problems can be solved using interior-point methods. The total number of variables for each subproblem is given as $d_1 = (\mathrm{K}(2\mathrm{B}(\mathrm{L}+1)+3))$ and thus the complexity metric becomes $\mathcal{O}((d_1)^{3.5})$. We let $\mathrm{V}_{1,\max}$ be the worst-case fixed number of iterations for convergence of the Dinkelbach algorithm. Since no optimization problem is solved in the outer loop, we can define $\mathrm{V}_{2,\max}$ as the worst-case fixed number of iterations needed for it to converge. To this end we can state the overall complexity of Algorithm 1 as $\mathcal{O}(\mathrm{V}_{1,\max}\mathrm{V}_{2,\max}(d_1)^{3.5})$, which is an upper bound on the complexity metric. Note that our proposed method consists of two instances of Algorithm 1, one for determining the serving clusters and one for finding a high-quality solution of beamforming vectors. The second instance of Algorithm 1 operates on the sparse optimization problem (45) with even fever optimization variables, which typically requires fewer iterations.

V. Numerical Simulations

In this section, we present numerical simulations that illustrate the performance of proposed algorithms. Considering a multi-cloud network scenario occupying a square area of [-400 400] × [-400 400] m^2. The BSs and the users are randomly placed in the studied MC-RAN. The distribution is uniform. Each BS is equipped with $L = 2$ transmit antennas and all BSs share the same fronthaul capacity constraint. The maximum transmit power is set to 32 dBm for each BS. The channel model used for our simulations consists of the following components:
1) the path-loss

$$\mathrm{PL}_{b,k} = 128.1 + 37.6\log_{10}(d_{b,k}), \tag{46}$$

where $d_{b,k}$ is the distance in km between BS b and user k;
2) the log-normal shadowing with 8dB standard deviation;
3) the Rayleigh channel fading with zero mean and unit variance.
The noise power σ^2 can be modeled as $-102 + 10\log_{10}(W) + n_f$ dBm, where the channel bandwidth is set to $W = 10$ MHz and the noise figure to $n_f = 15$ dBm. Maximum ratio combiners (MRC) are chosen for the beamforming vectors. The number of total files for caching is $F_b = 100$, we adopt the *popularity aware* cache placement scheme from [45]. The local memory size at each BS is considered to be 10 files, unless otherwise mentioned. As for the popularity of the files, we use the Zipf distribution [45] with parameter $a = 0.15$. Regarding the costs of the EE metric, P_b^{fthl} is chosen to be 40% of the processing power, $P_k^{\mathrm{proc}} = 20$ dBm, and $P_c^{\mathrm{Pr}} = 38$ dBm unless specified otherwise [46]. At last we define the number of users to be 28 and the number of BSs to be 10, unless mentioned otherwise.



(a) Cache holds up to 10 files.

(b) Cache holds up to 20 files.

Fig. 2: EE as a function of fronthaul capacity for different cache sizes.

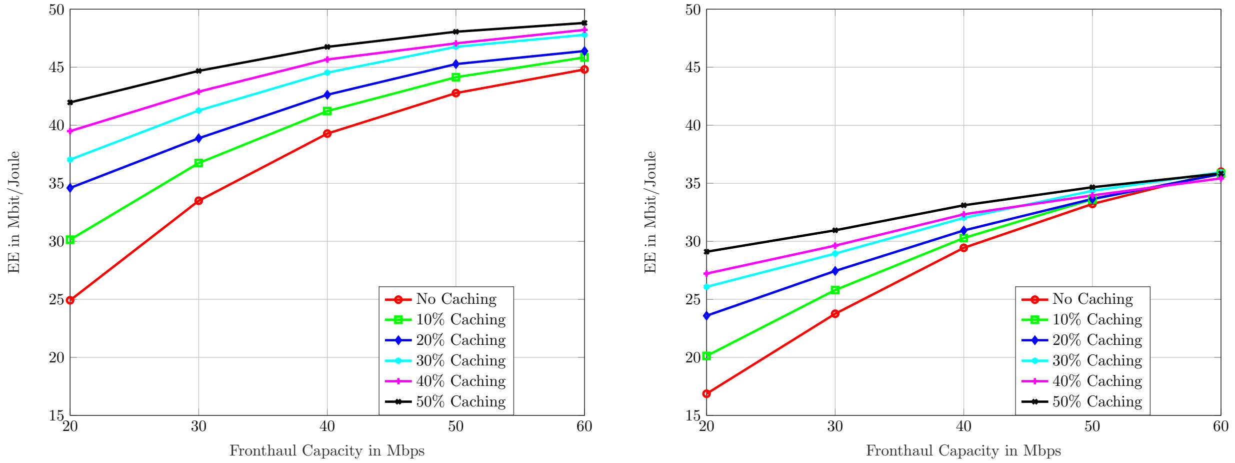
We propose optimizing beamforming vectors and serving clusters jointly in a dynamic clustering scheme. To best benchmark our methods, we use a static clustering scheme as a baseline to our proposed algorithm, where predetermined fixed clusters are used instead and the optimization is carried out on the beamforming vectors only. To determine such clusters, we use a load balancing algorithm, applied in [8] for the case of a single cloud. This benchmark will be referred to as fixed clustering. Both schemes can be implemented either in a centralized or distributed fashion. The former algorithm is implemented at one CP, processing data from all clouds, while the latter algorithm is implemented at every CP, managing their respective computations solitary. The second method requires less communication overhead between the CPs, as only inter-cloud interference information has to be exchanged.

A. *Impact of Fronthaul Capacity*

First, we evaluate the performance of the two schemes, static and dynamic clustering as centralized and distributed procedures. Fig. 2 shows the EE as a function of the fronthaul capacity for two different cache sizes, i.e., 10 and 20.

It can be observed that dynamic clustering outperforms the fixed clustering approaches regardless of being implemented distributively or centrally as shown in Fig. 2a. Particularly, while BSs might drop out of serving clusters due to overloading or power constraints, the need for dynamic clustering emerges, as such situations cannot be compensated by a fixed cluster.

Further, the centralized implementation outperforms the distributed implementation for both schemes in Fig. 2a and 2b. Note that the gain of using a centralized instead of distributed implementation increases jointly with fronthaul capacity, i.e., the gap widens. In a low-fronthaul



(a) $P_k^{\text{proc}} = 10$ dBm.

(b) $P_k^{\text{proc}} = 25$ dBm.

Fig. 3: EE as a function of fronthaul capacity, where different processing powers are utilized.

regime, the difference of both iterations is visibly insignificant, which highlights the role of our proposed distributed algorithm in limited fronthaul-capacity regimes, i.e., in cases where alleviating the fronthaul congestion is mostly required, as the performances of distributed and centralized algorithms become relatively similar.

A general observation from comparing Fig. 2a and 2b is an EE gain when bigger cache sizes at the BSs are employed. For the dynamic centralized scheme there is a 10% gain at 20 Mbps and 0.11% thus, limited gain at 100 Mbps fronthaul capacity. Intuitively, the EE metric benefits from cache hits since a user can be served without utilizing the fronthaul link only requiring processing costs at the respective BS. Especially at lower capacity regimes, fronthaul capacity is a scarce resource, thus the activation of a fronthaul link is a sensible decision, and the EE metric gains a lot from cache hits. As bigger caches are generally accompanied by more cache hits, it becomes clear why such results are obtained.

B. Processing Power vs. Caching Gain

In the second set of simulations, we consider only the distributed implementation of Algorithm 1 with dynamic clustering. In Fig. 3 we compare the EE for two different processing costs, we also vary the cache sizes for fixed costs, i.e., no cache up to a cache size of 50 files.

Intuitively, a higher processing cost will reduce the EE, as this factor can not be compensated. Such behavior can be recognized comparing Fig. 3a and 3b. The EE for a cache size of 10 files at 40 Mbps fronthaul capacity decreases by 26.54 % as the processing cost increase from 10 dBm to 25 dBm.

In Fig. 4 we plot the caching gain, i.e., the EE gain of using a cache of 10 files over not

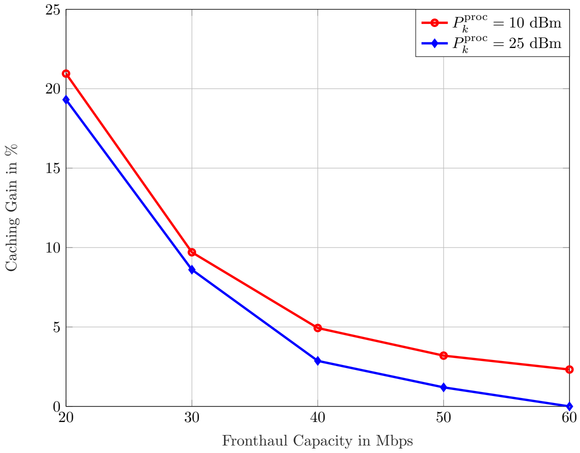


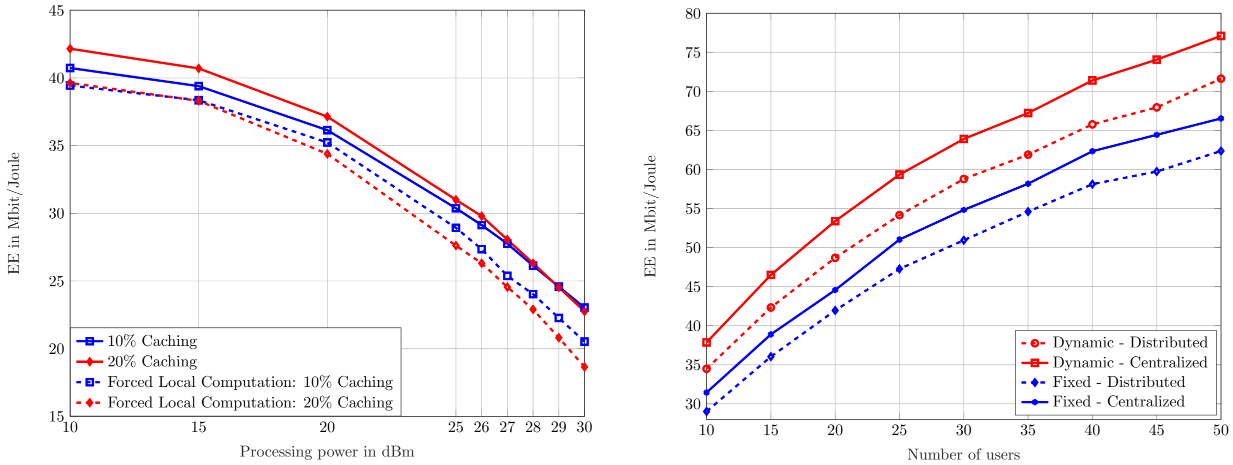
Fig. 4: Caching gain in $\%$ as a function of the fronthaul capacity for different processing powers.

using a cache, in percent over the fronthaul capacity for the two processing powers. To link the plots, 4 depicts the gain of the green line over the red line of Fig. 3a and 3b. In coincidence with previously made observations, the EE gain from a bigger cache decreases with increasing fronthaul capacity. Interestingly, the gain of utilizing a bigger cache size also decreases with higher processing costs. As visible in Fig. 3b and 4, for $P_k^{\text{proc}} = 25$ dBm at 60 Mbps fronthaul capacity there is no observable gain from utilizing caching capabilities.

C. Impact of Processing Power

Before describing the next set of simulations, we introduce a state-of-the-art scheme comparable to our proposed scheme, to set another baseline reference. In [16] the authors propose fixing the optimization variable $\tilde{\mathbf{t}}$ a priori, as cache placement and user requests are known, before jointly optimizing serving clusters and beamforming vectors. A BS that caches the requested file for user k has to serve this user and process its data locally. This method leaves no choice for a BS to leave the computation to the respective CP. This is motivated by the fact, that from an EE perspective local processing should be preferred to cloud computing since energy usage is lower in some regimes. This state-of-the-art scheme will be referred to as Forced Local Computation.

In Fig. 5a the EE as a function of processing power P_k^{proc} is shown for two different cache sizes, where the fronthaul capacity is set to 40 Mbps. In these simulations, our proposed scheme achieves better EE, when considering the required processing power for local caches. Interestingly, focusing on our proposed scheme, we notice a convergence of the EE for the two cache sizes, when P_k^{proc} increases. This matches the observations from V-B. In networks, where users require computationally intensive services, caching is not helpful in an EE metric. The



(a) A comparison with a state-of-the-art.

(b) Cache of 10 files.

Fig. 5: EE as a function of fronthaul capacity and number of users for different cache sizes and schemes.

main cause for such behavior lays in the EE metric itself since both BSs and CPs have to allocate P_k^{proc} when serving user k. As this value increases it becomes more dominant over the fronthaul processing cost P_k^{fthl}, which is only applied when a CP processes user k's data. From an EE perspective, the outsourcing of computation to the BSs becomes less significant with increasing processing costs.

D. *Impact of Network Densification*

In Fig. 5b we examine the EE as a function of the number of users. There are 14 BSs in the network, the processing power is set to $P_k^{\text{proc}} = 10$ dBm and the fronthaul capacity is 80 Mbps. As a first observation, we find, that generally with more users the EE increases. Similar to previous observations, we observe that the dynamic implementation always outperforms the fixed association implementation and for both schemes, the centralized version performs better than the distributed version from an EE perspective.

E. *Convergence Behavior*

In another set of simulations the processing power P_k^{proc} is set to 10 dBm and the BSs can cache up to 20 files. We only consider our proposed dynamic clustering scheme. To now illustrate the convergence behavior of our proposed algorithm 1, we plot the objective as a function of the number of iterations executed until converging in Fig. 6. We compare a distributed as well as a centralized implementation for different fronthaul capacities, i.e., $F_{b,c} \in \{20, 40, 60, 80\}$Mbps. In Fig. 6 the advantages of our algorithm in terms of convergence behavior and execution speed are

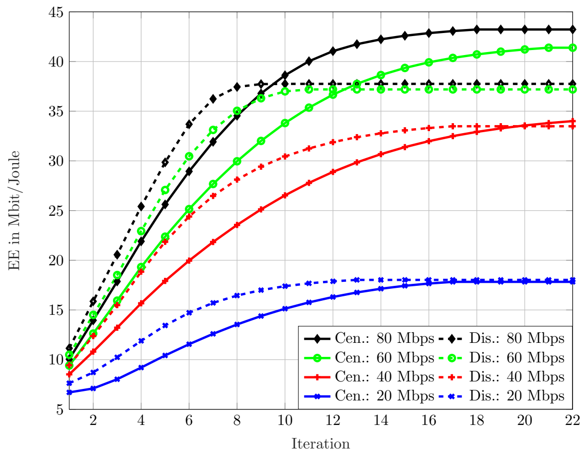


Fig. 6: Convergence behavior of the distributed and centralized implementation of our proposed algorithm.

highlighted, as the maximum iterations required for convergence are comparatively low. In all cases, the centralized implementation takes more iterations until convergence in comparison to the distributed implementation. The centralized approach jointly optimizes beamforming vectors while the distributed approach only optimizes subsets of these variables. In Fig. 6, at low fronthaul capacities, the distributed algorithm has acceptable loss in terms of EE against a centralized approach and performs better in terms of convergence. This constitutes an additional numerical feature of our proposed distributed resource allocation framework.

F. Comparison with Sum-Rate Maximization Algorithm

At last we make a connection to the weighted sum-rate maximization in MC-RAN from our work [1]. That work solves a mixed discrete-continuous non-convex optimization problem using a different fractional programming approach to tackle the non-convex part and l_0-norm approximation for the binary association part. Towards that end, [1] proposes an efficient iterative algorithm for joint association and beamformer design that can be implemented in a distributed fashion across multiple CPs. To conduct a fair comparison, we shall examine a special case of Algorithm 1, i.e., the case when there are no cache hits ($\mathcal{K}_1 = \varnothing$ and $\mathcal{K}_2 = \mathcal{K}$). This is due to the fact, that we do not consider edge intelligence and caches in [1]. We fix the processing power as $P_k^{\text{proc}} = 10$ dBm and the number of BSs as 14. To tackle the problem of comparing a maximized sum-rate to a maximized EE, we propose converting the results of the algorithm from [1] into the EE metric. Therefore we parse the final optimization variables from the sum-rate maximization into the EE formulation in (38). Such comparison is done for both centralized and distributed implementations in Fig. 7.

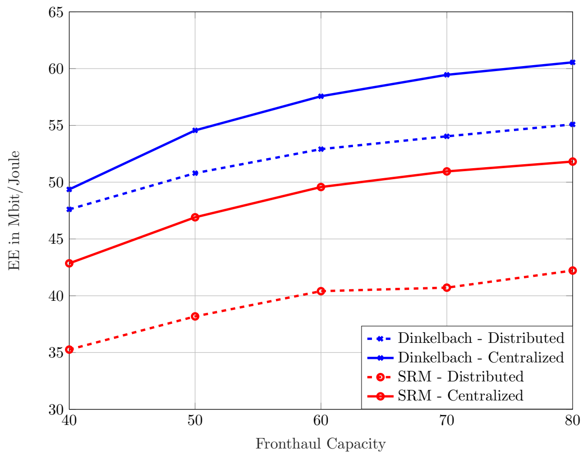


Fig. 7: EE as a function of fronthaul capacity comparing the proposed algorithm and the algorithm from [1]. We use the abbreviation SRM for sum-rate maximization.

Several observations can be made from this figure. First, we note that for every fronthaul capacity both Dinkelbach implementations, referring to Algorithm 1, outperform the sum-rate maximization (SRM) implementation. Interestingly, we see a difference in the gain of using a centralized over a distributed implementation. Different from previous observations with our proposed scheme, the loss of the distributed SRM implementation in terms of EE is vastly increased.

VI. Conclusion

While managing wireless systems with multiple CPs is a promising technique to cope with B5G network requirements, the energy efficiency metric plays key role in modern networks. This paper investigates the problem of joint user-to-cloud association and beamforming design in MC-RAN to maximize network-wide EE. An efficient algorithm based on Dinkelbach transform and SICA, which can be implemented in a distributed fashion across multiple clouds, is proposed. The numerical simulations show that our proposed distributed implementation achieves comparable performance to the centralized implementation. Comparisons with various state-of-the-art schemes are conducted, showing the pronounced role of our proposed algorithm in terms of maximizing EE.

Appendix A
Proof of lemma 1

First we revisit the function (27) and define the difference of convex functions

$$\vartheta(\mathbf{x}) \triangleq \sum_{k\in\mathcal{K}} \frac{1}{4}\Big(\underbrace{(t_{k,b}+R_{c,k})^2}_{\vartheta^+(\mathbf{x})} - \underbrace{(t_{k,b}-R_{c,k})^2}_{\vartheta^-(\mathbf{x})}\Big) - F_{b,c}, \tag{47}$$

where the functions $\vartheta^+(\mathbf{x})$ and $\vartheta^-(\mathbf{x})$ are convex and $\mathbf{x} = \left[\mathbf{t}^T, \mathbf{r}^T\right]^T$. Now we only replace the concave part $-\vartheta^-(\mathbf{x})$ by its first order approximation around point $\left(t'_{k,b}, R'_{c,k}\right)$ and thus get the convex upper approximation of function $\vartheta(\mathbf{x})$:

$$\hat{\vartheta}(\mathbf{x},\mathbf{x}') \triangleq \sum_{k\in\mathcal{K}} \frac{1}{4}\left(\vartheta^+(\mathbf{x}) - \vartheta^-(\mathbf{x}') - \nabla_{\mathbf{x}}\vartheta^-(\mathbf{x}')^T\left(\mathbf{x}-\mathbf{x}'\right)\right) - F_{b,c}. \tag{48}$$

Here we have $\mathbf{x}' = \left[\mathbf{t}'^T, \mathbf{r}'^T\right]^T$, and we can rewrite function $\hat{\vartheta}(\mathbf{x},\mathbf{x}')$ as

$$\begin{aligned}\hat{\vartheta}(\mathbf{x},\mathbf{x}') \triangleq & \sum_{k\in\mathcal{K}} \frac{1}{4}\Bigg((t_{k,b}+R_{c,k})^2 - \left(t'_{k,b}-R'_{c,k}\right)^2 \\ & - 2\left(t'_{k,b}-R'_{c,k}\right)\left(t_{k,b}-t'_{k,b}\right) + 2\left(t'_{k,b}-R'_{c,k}\right)\left(R_{k,b}-R'_{k,b}\right)\Bigg) - F_{b,c}.\end{aligned} \tag{49}$$

Since $\hat{\vartheta}(\mathbf{x},\mathbf{x}')$ is a convex upper approximation of $\vartheta(\mathbf{x})$ the following inequality is valid: $\vartheta(\mathbf{x}) \leq \hat{\vartheta}(\mathbf{x},\mathbf{x}')$. As $\hat{\vartheta}(\mathbf{x},\mathbf{x}')$ can be transformed into $g_1(\mathbf{t},\mathbf{r},\mathbf{t}',\mathbf{r}')$, and $\vartheta(\mathbf{x})$ corresponds to the right hand side of (29), this completes the proof.

Appendix B
Proof of lemma 2

As the functions $\nabla_{\mathbf{x}}\zeta(\mathbf{x},\xi) = \frac{2}{\xi}(\mathbf{x})^\dagger$ and $\nabla_{\xi}\zeta(\mathbf{x},\xi) = -\frac{1}{(\xi)^2}|\mathbf{x}|^2$ are partial derivatives of $\zeta(\mathbf{x},\xi)$, the first-order Taylor expansion is

$$\begin{aligned}\tilde{\zeta}(\mathbf{x},\xi,\mathbf{x}',\xi') &= \zeta(\mathbf{x}',\xi') + \nabla_{\mathbf{x}}\zeta(\mathbf{x}',\xi')(\mathbf{x}-\mathbf{x}') + \nabla_{\xi}\zeta(\mathbf{x}',\xi')(\xi-\xi') \\ &= \frac{|\mathbf{x}'|^2}{\xi'} + \frac{2}{\xi'}(\mathbf{x}')^\dagger(\mathbf{x}-\mathbf{x}') - \frac{1}{(\xi')^2}|\mathbf{x}'|^2(\xi-\xi') \\ &= \frac{|\mathbf{x}'|^2}{\xi'} + \frac{2}{\xi'}(\mathbf{x}')^\dagger\mathbf{x} - \frac{2}{\xi'}(\mathbf{x}')^\dagger\mathbf{x}' - \frac{\xi}{(\xi')^2}|\mathbf{x}'|^2 + \frac{\xi'}{(\xi')^2}|\mathbf{x}'|^2 \\ &= \frac{1}{\xi'}|\mathbf{x}'|^2 + \frac{1}{\xi'}|\mathbf{x}'|^2 - \frac{2}{\xi'}|\mathbf{x}'|^2 + \frac{2}{\xi'}(\mathbf{x}')^\dagger\mathbf{x} - \frac{\xi}{(\xi')^2}|\mathbf{x}'|^2 \\ &= \frac{2}{\xi'}\Re\{\,(\mathbf{x}')^\dagger\,\mathbf{x}\} - \frac{\xi}{(\xi')^2}|\mathbf{x}'|^2.\end{aligned} \tag{50}$$

This completes the proof.

Appendix C
Proof of theorem 1

The approximations in (28), (31) and (37) satisfy the conditions in [40, Section III.c, Assumptions 1-3]. To prove this, we shift our focus on (28). We define

$$\tilde{g}_1(\mathbf{x}) \triangleq \sum_{k\in\mathcal{K}} t_{k,b}R_{c,k} - F_{b,c}, \tag{51}$$

and $g_1(\mathbf{x},\mathbf{x}') \triangleq g_1(\mathbf{t},\mathbf{r},\mathbf{t}',\mathbf{r}')$, where $\mathbf{x} = [\mathbf{t}^T,\mathbf{r}^T]^T$ and $\mathbf{x}' = [\mathbf{t}'^T,\mathbf{r}'^T]^T$. Towards this end, we show that following properties are satisfied:

T1) $g_1(\mathbf{x}',\mathbf{x}') = \tilde{g}_1(\mathbf{x}')$.

T2) $g_1(\mathbf{x},\mathbf{x}') \geq \tilde{g}_1(\mathbf{x}), \qquad \forall \mathbf{x}' \in \mathcal{Z}$.

T3) $g_1(\bullet,\mathbf{x}')$ is a convex function, $\qquad \forall \mathbf{x}' \in \mathcal{Z}$.

T4) $g_1(\bullet,\bullet)$ is a continuous function on the feasible set.

T5) $\nabla_{\mathbf{x}} g_1(\mathbf{x}',\mathbf{x}') = \nabla_{\mathbf{x}}\tilde{g}_1(\mathbf{x}')$.

T6) The function $\nabla_{\mathbf{x}} g_1(\bullet,\bullet)$ is continuous on the feasible set.

Injecting $\mathbf{x}'$ into (28) and (51) ensures the equality in T1

$$\begin{aligned} g_1(\mathbf{x}',\mathbf{x}') &= \tilde{g}_1(\mathbf{x}') \\ \sum_{k\in\mathcal{K}} \Big(\left(t'_{k,b} + R'_{c,k}\right)^2 - 2\left(t'_{k,b} - R'_{c,k}\right)\left(t'_{k,b} - R'_{c,k}\right) + \\ \left(t'_{k,b} - R'_{c,k}\right)^2 \Big) - 4F_{b,c} &= \sum_{k\in\mathcal{K}} t'_{k,b}R'_{c,k} - F_{b,c} \\ \frac{1}{4}\left(\left(t'_{k,b} + R'_{c,k}\right)^2 - \left(t'_{k,b} - R'_{c,k}\right)^2\right) &= t'_{k,b}R'_{c,k} \\ \frac{1}{4}\left(4t'_{k,b}R'_{c,k}\right) &= t'_{k,b}R'_{c,k}. \end{aligned}$$

T2 follows directly from Lemma 1. To verify T3 and T4, we take a closer look at the structure of $g_1(\bullet,\mathbf{x}')$ with

$$g_1(\mathbf{x},\mathbf{x}') = \sum_{k\in\mathcal{K}} \Big(\underbrace{\left(t_{k,b} + R_{c,k}\right)^2}_{\text{convex}} - \underbrace{2\left(t'_{k,b} - R'_{c,k}\right)\left(t_{k,b} - R_{c,k}\right)}_{\text{linear}} + \left(t'_{k,b} - R'_{c,k}\right)^2 \Big) - 4F_{b,c}. \tag{52}$$

As $g_1(\bullet,\mathbf{x}')$, with fixed $\mathbf{x}'$, consists of a convex quadratic function subtracted by a linear function, which is convex, T3 holds. $g_1(\bullet,\bullet)$ has convex quadratic and bilinear terms, thus T4 is satisfied. For T5 and T6 we take the partial derivatives as follows

$$\nabla_{\mathbf{x}}\tilde{g}_1(\mathbf{x}) \begin{cases} \frac{\partial \tilde{g}_1(\mathbf{x})}{\partial \mathbf{t}} = \sum_{k\in\mathcal{K}} R_{c,k} \\ \frac{\partial \tilde{g}_1(\mathbf{x})}{\partial \mathbf{R}} = \sum_{k\in\mathcal{K}} t_{k,b} \end{cases} ; \tag{53}$$

$$\nabla_{\mathbf{x}} g_1(\mathbf{x},\mathbf{x}') \begin{cases} \frac{\partial g_1(\mathbf{x},\mathbf{x}')}{\partial \mathbf{t}} = \sum_{k\in\mathcal{K}} \frac{1}{4}\left(2\left(t_{k,b}+R_{c,k}\right)-2\left(t'_{k,b}-R'_{c,k}\right)\right) \\ \frac{\partial g_1(\mathbf{x},\mathbf{x}')}{\partial \mathbf{R}} = \sum_{k\in\mathcal{K}} \frac{1}{4}\left(2\left(t_{k,b}+R_{c,k}\right)+2\left(t'_{k,b}-R'_{c,k}\right)\right) \end{cases} . \tag{54}$$

The substitution of $\mathbf{x}$ with $\mathbf{x}'$ in (53) and (54) is straightforward and yields the equality T5. Since $\nabla_{\mathbf{x}} g_1(\bullet,\bullet)$ consists of linear terms, T6 holds. Similarly, these steps can be followed to verify T1-T6 for the remaining two reformulations in (31) and (37). This completes the proof.

REFERENCES

[1] A. A. Ahmad, H. Dahrouj, A. Chaaban, A. Sezgin, T. Y. Al-Naffouri, and M. Alouini, "Distributed cloud association and beamforming in downlink multi-cloud radio access networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[2] L. Zhang, Y. Liang, and D. Niyato, " 6G Visions: Mobile ultra-broadband, super internet-of-things, and artificial intelligence," *China Communications*, vol. 16, no. 8, pp. 1–14, Aug. 2019.

[3] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, 2016.

[4] L. Zhang, M. Xiao, G. Wu, M. Alam, Y. Liang, and S. Li, "A survey of advanced techniques for spectrum sharing in 5G networks," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 44–51, 2017.

[5] "Ericsson mobility report november 2019," Ericson, Tech. Rep. MSU-CSE-06-2, Nov. 2019. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2019

[6] T. Quek, M. Peng, O. Simone, and W. Yu, *Cloud Radio Access Networks: Principles, Technologies, and Applications*. Cambridge University Press, 2017.

[7] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[8] B. Dai and W. Yu, "Sparse Beamforming and User-Centric Clustering for Downlink Cloud Radio Access Network," *IEEE Access*, vol. 2, pp. 1326–1339, 2014.

[9] J. Tang, W. P. Tay, T. Q. S. Quek, and B. Liang, "System cost minimization in cloud RAN with limited fronthaul capacity," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3371–3384, May 2017.

[10] S. Park, O. Simeone, O. Sahin, and S. Shamai, "Inter-cluster design of precoding and fronthaul compression for cloud radio access networks," *IEEE Wireless Commun. Lett.*, vol. 3, no. 4, pp. 369–372, Aug. 2014.

[11] O. Dhifallah, H. Dahrouj, T. Y. Al-Naffouri, and M. S. Alouini, "Distributed robust power minimization for the downlink of multi-cloud radio access networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.

[12] H. Dahrouj, T. Y. Al-Naffouri, and M. S. Alouini, "Distributed cloud association in downlink multicloud radio access networks," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2015, pp. 1–3.

[13] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[14] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2827–2839, 2018.

[15] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3401–3415, 2017.

[16] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6118–6131, 2016.

[17] Y. Ugur, Z. H. Awan, and A. Sezgin, “Cloud radio access networks with coded caching,” *CoRR*, vol. abs/1512.02385, 2015. [Online]. Available: http://arxiv.org/abs/1512.02385

[18] Z. Ye, C. Pan, H. Zhu, and J. Wang, “Tradeoff caching strategy of the outage probability and fronthaul usage in a cloud-RAN,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6383–6397, 2018.

[19] S. Park, O. Simeone, and S. Shamai Shitz, “Joint optimization of cloud and edge processing for fog radio access networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7621–7632, 2016.

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

[21] “NR; physical layer procedures for data, v15.3.0,” document 3GPP TSGRAN, TS 38.214, Sep. 2018.

[22] H. Zhang, Y. Qiu, K. Long, G. K. Karagiannidis, X. Wang, and A. Nallanathan, “Resource allocation in NOMA-based fog radio access networks,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 110–115, 2018.

[23] I. Randrianantenaina, M. Kaneko, H. Dahrouj, H. ElSawy, and M. S. Alouini, “Interference management in NOMA-based fog-radio access networks via scheduling and power allocation,” *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 5056–5071, 2020.

[24] N. Pontois, M. Kaneko, T. H. L. Dinh, and L. Boukhatem, “User pre-scheduling and beamforming with outdated CSI in 5G fog radio access networks,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[25] L. Liu and W. Yu, “Cross-layer design for downlink multihop cloud radio access networks with network coding,” *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1728–1740, Apr. 2017.

[26] A. Douik, H. Dahrouj, T. Y. Al-Naffouri, and M. S. Alouini, “Distributed hybrid scheduling in multi-cloud networks using conflict graphs,” *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 209–224, Jan. 2018.

[27] D. Liu and C. Yang, “Will caching at base station improve energy efficiency of downlink transmission?” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 173–177.

[28] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, “Cache placement in fog-RANs: From centralized to distributed algorithms,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7039–7051, 2017.

[29] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[30] T. Han, N. Ansari, M. Wu, and H. Yu, “On accelerating content delivery in mobile networks,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1314–1333, 2013.

[31] A. Alameer and A. Sezgin, “Optimization framework for baseband functionality splitting in C-RAN,” in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1–5.

[32] P. Rost, C. J. Bernardos, A. D. Domenico, M. D. Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wübben, “Cloud technologies for flexible 5G radio access networks,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 68–76, 2014.

[33] J. Tang, W. P. Tay, and Y. Wen, “Dynamic request redirection and elastic service scaling in cloud-centric media networks,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1434–1445, 2014.

[34] A. De Domenico, Y. Liu, and W. Yu, “Optimal computational resource allocation and network slicing deployment in 5G hybrid C-RAN,” in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[35] A. Younis, T. X. Tran, and D. Pompili, “Fronthaul-aware resource allocation for energy efficiency maximization in C-RANs,” in *2018 IEEE International Conference on Autonomic Computing (ICAC)*, 2018, pp. 91–100.

[36] Y. Cheng, M. Pesavento, and A. Philipp, “Joint network optimization and downlink beamforming for comp transmissions using mixed integer conic programming,” *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3972–3987, 2013.

[37] L. Luo, N. Chakraborty, and K. Sycara, “Distributed algorithm design for multi-robot generalized task assignment problem,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4765–4771.

[38] ——, “Distributed algorithm design for multi-robot task assignment with deadlines for tasks,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3007–3013.

[39] A. Zappone and E. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory*, 2015.

[40] A. Zappone, E. Björnson, L. Sanguinetti, and E. Jorswieck, “Globally optimal energy-efficient power control and receiver design in wireless networks,” *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2844–2859, 2017.

[41] B. R. Marks and G. P. Wright, “A general inner approximation algorithm for nonconvex mathematical programs,” *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978. [Online]. Available: http://www.jstor.org/stable/169728

[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[43] E. Björnson and E. Jorswieck, *Optimal Resource Allocation in Coordinated Multi-Cell Systems*, 2013.

[44] M. Lobo, L. Vandenberghe, S. P. Boyd, and H. Lebret, “Applications of second-order cone programming,” *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1998.

[45] A. Alameer and A. Sezgin, “Resource cost balancing with caching in C-RAN,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[46] B. Dai and W. Yu, “Energy efficiency of downlink transmission strategies for cloud radio access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 1037–1050, 2016.