Joint Optimization of Uplink Power and Computational Resources in Mobile Edge Computing-Enabled Cell-Free Massive MIMO

Giovanni Interdonato and Stefano Buzzi

***Abstract*—The coupling of cell-free massive MIMO (CF-mMIMO) with Mobile Edge Computing (MEC) is investigated in this paper. A MEC-enabled CF-mMIMO architecture implementing a distributed user-centric approach both from the radio and the computational resource allocation perspective is proposed. A multi-objective optimization problem (MOOP) for the joint allocation of radio and remote computational resources is formulated, aimed at striking an optimal balance between total uplink power minimization and sum spectral efficiency maximization, under resource budget and latency constraints. In order to solve such a challenging non-convex problem, we convert the MOOP to an equivalent single-objective optimization problem (SOOP) through the weighted sum method and propose an iterative algorithm based on alternating optimization and sequential convex programming, along with an alternative heuristic resource allocation for distributed networks. Finally, we provide a detailed performance comparison between the proposed MEC-enabled CF-mMIMO architecture with its co-located counterpart, and its small-cell implementation. Numerical results reveal the effectiveness of the proposed resource allocation scheme, under different access point selection strategies, and the natural suitability of CF-mMIMO in supporting computation-offloading applications with benefits over users' transmit power and energy consumption, the effective latency experienced, and the computation offloading efficiency.**

***Index Terms*—Cell-free massive MIMO, computation offloading, mobile edge computing, sequential optimization.**

I. Introduction

THE RECENT evolution of wireless networks has been characterized by an impressive growth not only of the amount of conveyed data traffic, but also of computationally-intensive applications with strict latency requirements for mobile devices. Applications such as online gaming, augmented reality and video image processing not only request extreme broadband connections, but also a considerable amount of computational power at the mobile devices. A possible approach to indirectly increase the computing capabilities of the devices and prolong their battery lifetime is to (either fully or partially) delegate their computational tasks to the network, specifically to network entities known as *network edge servers*[1], in charge of collecting, processing and feeding data back to the users in a centralized fashion. This approach is known as *mobile edge computing* (MEC) or mobile-edge computation offloading [2]–[6].

Cell-free massive multiple-input multiple-output (CF-mMIMO) is the ultimate embodiment of network MIMO [7]–[9]. CF-mMIMO is a technology based on the use of several distributed low-complexity access points (APs) that jointly serve the active users in their coverage area. It inherits all the outstanding features of co-located massive MIMO [10], [11], such as nearly-optimal linear signal processing, predictable accurate performance, and simplified resource allocation and channel estimation, while providing additional key ingredients to theoretically achieve unprecedented levels of uniform data rates and ubiquitous connectivity: macro-diversity gain, inter-cell interference mitigation, and user proximity (see e.g. [12, Chapter 3.2] and references therein). Moreover, it is also amenable to scalable user-centric implementations [13]–[15].

In this paper, we investigate the promising marriage between CF-mMIMO and MEC which share the same principle of bringing the resources (radio and computing, respectively) closer to the user. CF-mMIMO, thanks to its dense distributed topology and user-centric architecture, may greatly facilitate the computation offloading by enabling mobile devices to delegate either all or part of their computational tasks to multiple APs, each of which may be equipped with an edge server. Moreover, the central processing unit (CPU) of a CF-mMIMO system, which is generally equipped with a more powerful server, may serve as a backup edge computing to give, in turn, computation offloading support to the APs. User proximity and the macro-diversity may significantly shorten the delay due to the computation offloading, thereby supporting stricter latency requirements, and reduce user's power consumption. Moreover, the user-centric approach ensures more uniform spectral efficiency (SE) and thereby the access to the remote computational resources may be indiscriminately granted to every user. The ability of the network to accomplish users' computation offloading depends on how the radio and remote computational resources are allocated. This coupling calls for a joint optimization which is the main subject of this study.

This paper was supported by the Italian Ministry of Education University and Research (MIUR) Project "Dipartimenti di Eccellenza 2018-2022" and by the MIUR PRIN 2017 Project "LiquidEdge". An excerpt of this article has been published in the proceedings of the 2022 IEEE International Conference on Communications (ICC) [1]. The authors are with the Department of Electrical and Information Engineering (DIEI) of the University of Cassino and Southern Lazio, 03043 Cassino, Italy (e-mail: giovanni.interdonato@unicas.it, buzzi@unicas.it), and with the Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 43124, Parma, Italy. S. Buzzi is also affiliated with Politecnico di Milano, Milano, Italy, and his work was also supported by the European Union under the *Italian National Recovery and Resilience Plan* (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", Structural Project 6GWINET).

[1]The edge servers are network entities figuratively placed at the *edge* of the cellular access network, that is between the radio access network and the core network.

Related Works. Many works on MEC optimize the interplay between the amount of computational tasks to offload, the latency due to the offloading process, and the energy consumption of mobile devices. First studies on MEC assumed simplified system models by considering either single-user systems [16]–[18] or interference-free multi-user systems [19], [20], focusing either on minimizing the energy consumption under latency constraints [20] or the delay due to the computation offloading under energy consumption constraints [21]. Some of these works consider a *binary computation offloading* model, wherein each device executes its computational tasks either remotely or locally [16], [20]. Other studies assumed a more general *partial computation offloading* [17], [19], [22] with only a fraction of computational tasks executed remotely. An integrated framework for computation offloading and interference management in cellular networks was proposed in [23]. All these works assumed single-antenna base stations (BSs).

More recently, MEC has been studied in conjunction with MIMO technologies. As an example, [4] considers a multi-cell MIMO system served by an edge server in a centralized fashion, and formulates a total energy minimization problem under latency and minimum rate constraints. Similarly, the MEC solution proposed in [24] focuses on minimizing the maximum latency of all the devices in a cloud-radio access network (C-RAN) with MIMO technology, while [25] addresses the energy minimization problem accounting for imperfect channel state information (CSI) in a single-cell MIMO system. In [26] an optimal association of mobile users to MEC resources is devised for a multi-user MIMO system with C-RAN architecture. In [27] a successive inner convexification framework to minimize the total transmit power of the devices under latency constraints is proposed. Several authors have also examined the coupling between MEC and massive MIMO. The work [28] proposes a low-complexity algorithm to jointly optimize the radio and computing resources for a massive MIMO-enabled heterogeneous network with MEC. Similarly, [29], [30] presented the effectiveness of employing massive MIMO for MEC, under zero-forcing combining, aiming at minimizing the maximum delay for offloading and computing among the devices. The paper [31], instead, considers a massive MIMO system operating at the millimeter-wave (mmWave) frequency bands underlying traditional wireless local area networks with MEC. A dynamic computation offloading in MEC for ultra-reliable and low-latency communications at the mmWave frequency bands is proposed in [32]. The main common conclusion of these works is that multiple users can simultaneously offload their computational tasks by leveraging the additional degrees of freedom provided by massive MIMO, and the offloading efficiency, as well as the energy saving of the mobile devices, grows with the number of antennas of the massive MIMO BS.

Finally, the performance of CF-mMIMO with MEC has been recently explored in [33] and [34]. The former proposes a joint optimization of the partial offloading ratio per task and the resulting computational resources allocated at a single MEC server to minimize either the aggregated latency or the energy consumption at each user. The author in [34] investigate the successful edge computing probability (SECP) for a target computation latency by using queuing theory and stochastic geometry, and by considering a random computation latency model. The system model in [34] consists of APs equipped with independent MEC servers, a CPU with a central MEC server (CS), and devices performing offloading either to the CS or to one of their serving APs, with some successful computing probability. The joint decoding of the offloaded user data at one of the serving APs/CS is, however, not recommended due to extra delays caused by the fronthaul communications. In fact, the considered architecture implements a specific instance of cell-free massive MIMO, namely a small-cells network. Moreover, in [34] both uplink transmit powers and allocated computational resources are fixed rather than optimized. Following on this track, in this paper we explore the potential benefits of jointly optimizing radio and computational resources in a MEC-enabled CF-mMIMO system.

Our problem formulation is characterized by a multi-objective function and aims at striking an optimal balance between the minimization of the total uplink transmit power and the maximization of the sum uplink SE. Multi-objective optimization (MOO) is a mathematical framework to deal with optimization problems with multiple conflicting objective functions [35]–[37]. A survey of MOO applied to signal processing in wireless networks, with emphasis on massive MIMO systems and conflicting metrics such as SE, energy efficiency, coverage and total transmit power, was given in [38]. A conventional optimization approach consists in converting some of the objectives into constraints, whereas the fundamental approach of MOO consists in considering multiple objectives at once. There are two main methods: (i) computing the sample points on the *Pareto frontier* upon which making subjective decisions a posteriori, or (ii) a priori converting the MOO problem (MOOP) to a single-objective optimization problem (SOOP) by combining the objectives into a suitable goal function (the most common is the *weighted sum*) which reflects a subjective trade-off between metrics of interest. In the latter, the objectives are conventionally combined by using coefficients whose values reflect the subjective weight given to each metric.

Contributions: Our technical contribution can be summarized as follows.

- We propose a MEC-enabled CF-mMIMO architecture implementing a user-centric approach both from the radio and the computational resource allocation perspective. Unlike prior studies investigating computation-offloading implementations in distributed networks [27], [28], [33], [34], our model considers that users' computational tasks can be divided into independent subtasks which can be remotely executed in a distributed fashion and in parallel at the MEC servers of properly selected APs and at the MEC server of the CPU.
- We formulate an optimization problem for jointly allocating users' transmit powers and the remote computational resources for offloading. Unlike prior works [27], [28], [30], [34] we formulate a MOOP that optimizes the trade-off between total uplink transmit power minimization and sum SE maximization, under latency and resource budget

constraints.

- For efficiently solving the non-convex MOOP, we formulate an equivalent SOOP by using the weighted sum method and devise a framework including alternating optimization and successive convex approximation (SCA) which, unlike prior works, accounts for: (i) the user-centric cooperation clustering framework; (ii) a distributed allocation of the computational resources; (iii) a general formulation for any combining scheme and arbitrary correlated fading channels. As there is no unique solution for this SOOP, we provide its sub-optimal *Pareto frontier*, namely the set of objectives corresponding to the Pareto sub-optimal solutions obtained by iteratively solving the SOOP for several values of the weights. Finally, we show how the weights in the SOOP indirectly determine the effective latency of the offloading process experienced by the users.
- We propose an alternative low-complexity approach to the proposed joint resource allocation, which consists in heuristically allocating the MEC server computational resources to the users, and then optimizing with respect to the uplink powers.
- Since the final solution achieved by SCA-based methods may depend on the feasible solution initialization, we present a method to properly initialize the proposed iterative optimization algorithm, and to provide a rigorous assessment on the problem feasibility.
- For benchmarking purposes, we extend our joint optimization strategy to a multi-cell co-located massive MIMO system, and to a small-cell implementation of CF-mMIMO. The latter constitutes a deterministic variant of the framework described in [34] wherein the task offloading model hinges on the knowledge of the users' computational demands and MEC servers' available computing resources.
- We provide a comprehensive simulation campaign to highlight the improvements introduced by the proposed MEC-enabled CF-mMIMO system in terms of: (i) users' transmit power and energy consumption, (ii) offloading latency, (iii) amount of allocated remote computational resources, and (iv) computation offloading efficiency. We also study its performance under different strategies of AP selection for providing the communication service.
- A further insight about the effectiveness of the proposed joint uplink power and computational resource allocation (JPCA) scheme is provided by evaluating the interplay between energy consumption, allocated remote computational resources and offloading latency.

II. System Model

We consider a CF-mMIMO system operating in time-division duplexing (TDD) mode and at sub-6 GHz frequency bands. A set of L APs, equipped with M antennas each, are geographically distributed and connected through a fronthaul network to a CPU. The APs coherently serve K single-antenna users in the same time-frequency resources, with $LM \gg K$. The conventional block-fading channel model is considered, and let τ_c denote the channel coherence block length. In TDD mode, each coherence block accommodates uplink training, uplink and downlink data transmission, such that $\tau_{\mathrm{c}}=\tau_{\mathrm{p}}+\tau_{\mathrm{u}}+\tau_{\mathrm{d}}$, where τ_{p}, τ_{u} and τ_{d} are the training duration, the uplink and the downlink data transmission duration, respectively.

Borrowing the notation of [15], the channel between the k-th user and the l-th AP is denoted by the M-dimensional vector $\mathbf{h}_{lk}$, with $\mathbf{h}_{lk} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{lk})$, and $\mathbf{R}_{lk} \in \mathbb{C}^{M\times M}$ being the spatial correlation matrix. The corresponding large-scale fading coefficient is defined as $\beta_{lk} = \mathrm{tr}(\mathbf{R}_{lk})/M$. The channel between the k-th user and all the APs in the system is obtained by stacking the channel vectors $\mathbf{h}_{lk}$, $\forall l$ as $\mathbf{h}_k = [\mathbf{h}_{1k}^{\mathrm{T}} \cdots \mathbf{h}_{Lk}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{ML}$. The channel vectors of different APs are reasonably assumed to be independently distributed. As a consequence, we have $\mathbf{h}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_k)$, where $\mathbf{R}_k = \mathrm{blkdiag}(\mathbf{R}_{1k}, \ldots, \mathbf{R}_{Lk}) \in \mathbb{C}^{ML\times ML}$ is user's k block-diagonal spatial correlation matrix.

A. Centralized Uplink Training

During the uplink training all the K users synchronously send a pre-determined pilot sequence of τ_{p} samples. The pilot sequences are drawn by a set of τ_{p} orthonormal vectors. Specifically, $\sqrt{\tau_{\mathrm{p}}}\boldsymbol{\varphi}_k \in \mathbb{C}^{\tau_{\mathrm{p}}}$ denotes the pilot sent by the k-th user, with $\|\boldsymbol{\varphi}_k\| = 1$. Whenever K is larger than τ_{p}, the same pilot must be assigned to more than one user, causing pilot contamination. The pilot signal observed by AP l is $\mathbf{Y}_{\mathrm{p},l} = \sum_{j=1}^{K} \sqrt{\tau_{\mathrm{p}} p_{\mathrm{p},j}}\ \mathbf{h}_{lj}\boldsymbol{\varphi}_j^{\mathrm{T}} + \boldsymbol{\Omega}_{\mathrm{p},l} \in \mathbb{C}^{M\times\tau_{\mathrm{p}}}$, where $p_{\mathrm{p},j}$ is the transmit power of the uplink pilot symbol, and $\boldsymbol{\Omega}_{\mathrm{p},l}$ is a matrix of additive noise whose elements are independently distributed as $\mathcal{CN}(0, \sigma^2)$. For any user k, the l-th AP projects $\mathbf{Y}_{\mathrm{p},l}$ along the k-th pilot sequence, which yields:

$$\begin{aligned}\mathbf{y}_{lk}^{\mathrm{p}} &= \mathbf{Y}_{\mathrm{p},l}\boldsymbol{\varphi}_k^* \\ &= \sqrt{\tau_{\mathrm{p}} p_{\mathrm{p},k}}\mathbf{h}_{lk} + \sum_{j\neq k}^{K} \sqrt{\tau_{\mathrm{p}} p_{\mathrm{p},j}}\mathbf{h}_{lj}\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_k^* + \boldsymbol{\Omega}_{\mathrm{p},l}\boldsymbol{\varphi}_k^* \in \mathbb{C}^M, \quad (1)\end{aligned}$$

where the second term captures the interference due to pilot contamination. Assuming that the channel estimation is performed by the CPU, in each coherence interval, each AP needs to send the vector $\mathbf{y}_{lk}^{\mathrm{p}}$ to the CPU. Upon a prior knowledge of the channel correlation matrices, the CPU performs linear minimum-mean square error (MMSE) estimation of the k-th user channel $\mathbf{h}_{lk}$ as $\hat{\mathbf{h}}_{lk} = \sqrt{\tau_{\mathrm{p}} p_{\mathrm{p},k}}\mathbf{R}_{lk}\boldsymbol{\Psi}_{lk}^{-1}\mathbf{y}_{lk}^{\mathrm{p}}$, where $\boldsymbol{\Psi}_{lk} = \mathsf{E}\left\{\mathbf{y}_{lk}^{\mathrm{p}}(\mathbf{y}_{lk}^{\mathrm{p}})^{\mathrm{H}}\right\} = \tau_{\mathrm{p}}\sum_{j=1}^{K} p_{\mathrm{p},j}\mathbf{R}_{lj}\ |\boldsymbol{\varphi}_k^{\mathrm{H}}\boldsymbol{\varphi}_j|^2 + \sigma^2\mathbf{I}_M$. The estimation error is independent of the estimate, and given by $\tilde{\mathbf{h}}_{lk} = \mathbf{h}_{lk} - \hat{\mathbf{h}}_{lk}$. It is distributed as $\tilde{\mathbf{h}}_{lk} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_{lk})$, with $\mathbf{C}_{lk} = \mathsf{E}\left\{\tilde{\mathbf{h}}_{lk}\tilde{\mathbf{h}}_{lk}^{\mathrm{H}}\right\} = \mathbf{R}_{lk} - \tau_{\mathrm{p}} p_{\mathrm{p},k}\mathbf{R}_{lk}\boldsymbol{\Psi}_{lk}^{-1}\mathbf{R}_{lk}$. Collecting all the channel estimates of user k in a vector, we have $\hat{\mathbf{h}}_k = [\hat{\mathbf{h}}_{1k}^{\mathrm{T}} \cdots \hat{\mathbf{h}}_{Lk}^{\mathrm{T}}]^{\mathrm{T}}$. Accordingly, it holds $\tilde{\mathbf{h}}_k = \mathbf{h}_k - \hat{\mathbf{h}}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_k)$, with $\mathbf{C}_k = \mathrm{diag}(\mathbf{C}_{1k}, \ldots, \mathbf{C}_{Lk})$.

B. User-Centric Uplink Data Transmission

The uplink data signal received by AP l is $\mathbf{y}_l = \sum_{i=1}^{K} \mathbf{h}_{li}s_i + \mathbf{n}_l$, with s_i being the data symbol transmitted by user i, $\mathsf{E}\left\{|s_i|^2\right\} = p_i, \forall i$, and $\mathbf{n}_l \sim \mathcal{CN}(0, \sigma^2\mathbf{I}_M)$ being the additive

noise vector. In a practical and scalable user-centric implementation each user is served by a subset of APs selected among those ensuring the best channel conditions. Let $\mathcal{M}_k$ be the set of the indices of the APs serving user k, and $\mathbf{D}_{lk} \in \mathbb{C}^{M\times M}, \forall l, \forall k$ be a diagonal matrix such that $\mathbf{D}_{lk} = \mathbf{I}_M$, if $l \in \mathcal{M}_k$, $\mathbf{D}_{lk} = \mathbf{0}_M$, otherwise. Under centralized uplink operation, the CPU computes the estimate of the transmitted data symbol s_k as

$$\hat{s}_k = \sum_{l=1}^{L} \hat{s}_{lk} = \sum_{l=1}^{L} \mathbf{v}_{lk}^{\mathrm{H}} \mathbf{D}_{lk} \mathbf{y}_l = \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \mathbf{y}, \tag{2}$$

where $\mathbf{v}_{lk} \in \mathbb{C}^M$ is the receive combining vector for the pair AP l-user k, $\mathbf{v}_k = [\mathbf{v}_{1k}^{\mathrm{T}} \cdots \mathbf{v}_{Lk}^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{y} = [\mathbf{y}_1^{\mathrm{T}} \cdots \mathbf{y}_L^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{ML}$, and $\mathbf{D}_k = \mathrm{diag}(\mathbf{D}_{1k}, \ldots, \mathbf{D}_{Lk})$. Eq. (2) can be rewritten as

$$\hat{s}_k = \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \hat{\mathbf{h}}_k s_k + \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \tilde{\mathbf{h}}_k s_k + \sum_{i\neq k}^{K} \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \hat{\mathbf{h}}_i s_i + \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \mathbf{n}, \tag{3}$$

with $\mathbf{n} = [\mathbf{n}_1^{\mathrm{T}} \cdots \mathbf{n}_L^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{ML}$ being the collective noise vector. The first term in (3) is the desired signal over the known partially estimated channel, the second term is the self-interference due to the (unknown) estimation error, the third term is the multi-user interference, and, lastly, the fourth term is the noise. An achievable uplink SE (bit/s/Hz) for user k, with centralized operation, is obtained by treating the last three terms as uncorrelated noise at the receiver:

$$\overline{\mathrm{SE}}_k = \frac{\tau_{\mathrm{u}}}{\tau_{\mathrm{C}}} \mathsf{E}\left\{\log_2(1 + \mathrm{SINR}_k)\right\}, \quad \text{where} \tag{4}$$

$$\mathrm{SINR}_k = \frac{p_k |\mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \hat{\mathbf{h}}_k|^2}{\sum_{i\neq k}^{K} p_i |\mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \hat{\mathbf{h}}_i|^2 + \mathbf{v}_k^{\mathrm{H}} \mathbf{Z}_k \mathbf{v}_k + \sigma^2 \|\mathbf{D}_k \mathbf{v}_k\|^2}, \tag{5}$$

with $\mathbf{Z}_k = \sum_{i=1}^{K} p_i \mathbf{D}_k \mathbf{C}_i \mathbf{D}_k$. This achievable SE holds for any combining scheme and arbitrary correlated fading channels, and accounts for user-centric data detection, channel estimation error, pilot contamination and estimation overhead. Hereafter, we consider the so-called Partial MMSE (P-MMSE) combining [15] which guarantees scalability and an excellent trade-off between performance and computational complexity. For an arbitrary user k, P-MMSE suppresses only the strongest interference contributions which are caused by the users whose indices are in the set $\mathcal{S}_k = \{i : \mathbf{D}_k \mathbf{D}_i \neq \mathbf{0}_{LM}\}$. The P-MMSE collective combining vector is given by

$$\mathbf{v}_k^{\mathsf{P-MMSE}} = p_k \left(\sum_{i\in\mathcal{S}_k} p_i \mathbf{D}_k \hat{\mathbf{h}}_i \hat{\mathbf{h}}_i^{\mathrm{H}} \mathbf{D}_k + \mathbf{Z}_{\mathcal{S}_k} + \sigma^2 \mathbf{I}_{LM} \right)^{-1} \mathbf{D}_k \hat{\mathbf{h}}_k, \tag{6}$$

where $\mathbf{Z}_{\mathcal{S}_k} = \sum_{i\in\mathcal{S}_k} p_i \mathbf{D}_k \mathbf{C}_i \mathbf{D}_k$.

III. Computation-Offloading and Latency Model

We assume that both the APs and the CPU[2] offer computational facility to the users. Each user has a set of computational tasks to offload to multiple distributed MEC servers on some APs and/or to the MEC server at the CPU. In particular, we denote by $\mathcal{G}_k$ the set of MEC servers at the APs and CPU that can provide computational offloading service to user k. Let $\mathcal{O}_k \subseteq \mathcal{G}_k$ be the set of APs where user k's computational tasks can be offloaded[3]. In a small system with one CPU, as it is the case for this paper, it is also reasonable that $\mathcal{O}_k$ coincides with the set of all the APs in the system. We assume that user k needs to execute one or more computational tasks within a maximum tolerable latency $\mathcal{L}_k$. All the relevant information on these computational tasks can be encoded into b_k bits, and their execution requires a total of w_k computation cycles, that can be decomposed in T_k computational subtasks, consisting of $w_{k,1}, w_{k,2}, \ldots, w_{k,T_k}$ computation cycles, with $\sum_{t=1}^{T_k} w_{k,t} = w_k$. AP l has a computational capability of f_l^{AP} computation cycles per second (*computational rate*). While, the CPU can execute up to f^{CPU} computation cycles per second. The fractions of computational resources assigned to subtask i of user k by the generic AP l and by the CPU are denoted by $f_{l,k}^{\mathsf{AP}}(i)$ and $f_k^{\mathsf{CPU}}(i)$, respectively[4]. Accordingly, it holds

$$\sum_{k=1}^{K}\sum_{i=1}^{T_k} f_k^{\mathsf{CPU}}(i) \leq f^{\mathsf{CPU}}, \text{ and } \sum_{k=1}^{K}\sum_{i=1}^{T_k} f_{l,k}^{\mathsf{AP}}(i) \leq f_l^{\mathsf{AP}}, \ \forall l.$$

The total amount of remote computational resources assigned to subtask i of user k is given by $f_k(i) = f_k^{\mathsf{CPU}}(i) + \sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(i)$. Then, $w_{k,i}/f_k(i)$ represents the computational time needed to execute $w_{k,i}$ cycles (*computational latency*). Since the T_k computational subtasks for user k are executed in parallel, the resulting computational latency for the task of user k is $\max_{i=1,\ldots,T_k} w_{k,i}/f_k(i)$.

While, the amount b_k/R_k is the time needed to transmit b_k bits to the APs (*transmission latency*) over the wireless channel supporting a rate $R_k = B \times \mathrm{SE}_k$, with B being the transmission bandwidth and SE_k being the instantaneous SE, i.e., the value attained by (4) with no expectation. Lastly, an additional latency contribution (*fronthaul latency*) is due to the forwarding of the b_k bits from all the APs in the

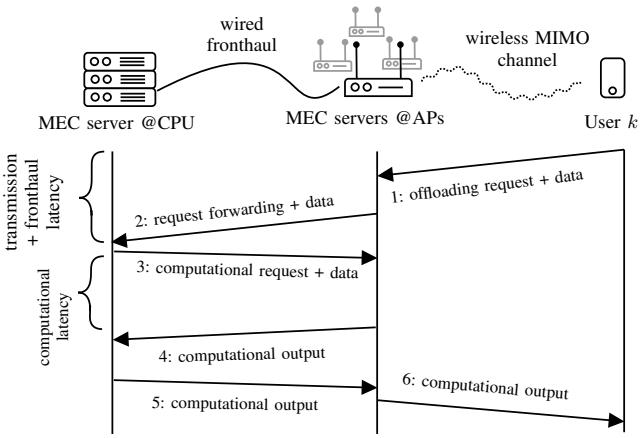

Fig. 1. Signalling diagram describing the computational offloading process for an arbitrary user k. The computational offloading may occur at the MEC servers of both the CPU and the APs.

[2]The CPU is either a physical or a logical entity, an edge-cloud processor located in the same geographical area as the APs.

[3]Notice that there is no implicit relation between $\mathcal{O}_k$ and $\mathcal{M}_k$, even though it is reasonable to expect that $\mathcal{M}_k \subseteq \mathcal{O}_k$.

[4]The optimization problem that will be considered in this paper will enforce the constraint that each subtask is entirely executed at only one AP or at the CPU, i.e., subtasks cannot be further divided into parallel sub-subtasks to be executed in different MEC servers.

set $\mathcal{M}_k$ to the CPU over the fronthaul network, which, assuming synchronous transmission across the APs, amounts to $2b_kM\xi/C_{\mathsf{FH}}$, where ξ denotes the number of bits used to quantize both real and imaginary parts of the uplink data signal $\mathbf{y}$, and C_{FH} is the fronthaul capacity of the link between any AP and the CPU, expressed in bit/s. Hence, the computational offloading must fulfill the following latency constraint [27]

$$\frac{b_k}{R_k} + \max_{i=1,\ldots,T_k} \frac{w_{k,i}}{f_k(i)} + \frac{2b_kM\xi}{C_{\mathsf{FH}}} \leq \mathcal{L}_k, \forall k, \tag{7}$$

where we assume that $\mathcal{L}_k$ includes any delay related to the signalling between AP and CPU, and the time needed to send the computational output back to the user. The latency constraint in (7) clearly couples radio and computational resources. Fig. 1 illustrates the signalling diagram of the computational offloading process for an arbitrary user k. As a first step, user k sends on the air interface a *computational offloading request* followed by the symbols encoding the data to be processed, the program to be executed remotely, and, also, the details on the subtasks which the main computational task is composed of. It is assumed that the overall computational offloading message has a length of b_k bits. These b_k bits are treated as normal information data, so they are sent from the user k during the uplink data transmission phase; the corresponding received signals are locally processed at the APs serving user k (i.e., the APs in the set $\mathcal{M}_k$), and sent to the CPU over the fronthaul network for the centralized receive combining and the data decoding. The CPU, based on the received computational requests from all the users in the network, and on the knowledge of the estimated uplink channels, and of the available computing power at the CPU itself and at the APs, allocates each *batch job*, represented by an entry of the set $\{w_{k,i} : k = 1, \ldots, K; i = 1, \ldots, T_k\}$, either to itself or to one AP. The allocation of the computational resources between CPU and APs is subject to optimization as detailed in the next subsection. Once received the computational output of each batch job from the APs, the CPU sends the combined computational output back to the APs in the set $\mathcal{M}_k$, which finally perform a joint downlink data transmission to user k.

A. Joint Power and Computational Resource Allocation(JPCA)

We jointly allocate the users' transmit powers and the remote computational resources assigned to the users aiming at simultaneously minimizing the total uplink transmit power and maximizing the sum SE. These are two conflicting objective functions of a MOOP which will be treated through the classical scalarization technique. In particular, the MOOP is converted into a SOOP designing a single goal function that reflects a pre-determined trade-off between the objectives. To this end, let us first introduce the vector of the uplink powers $\mathbf{p} = [p_1 \cdots p_K]^{\mathrm{T}}$ and the set $\mathcal{F}$ containing the allocated computational resources, that is

$$\mathcal{F} = \{f_k^{\mathsf{CPU}}(i),\ f_{l,k}^{\mathsf{AP}}(i) : l = 1, \ldots, L;\ k = 1, \ldots, K;\ i = 1, \ldots, T_k\}. \tag{8}$$

The SOOP for the proposed JPCA can be formulated as

$$\min_{\mathbf{p},\boldsymbol{\nu},\mathcal{F}} \ \varpi_{\mathrm{p}} \mathbf{1}_K^{\mathrm{T}} \mathbf{p} - \varpi_{\mathrm{se}} \mathbf{1}_K^{\mathrm{T}} \boldsymbol{\nu} \tag{9a}$$

$$\text{s.t. } \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})} + \max_{j=1,\ldots,T_k} \frac{w_{k,j}}{f_k^{\mathsf{CPU}}(j) + \sum\limits_{l \in \mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j)} \leq \widetilde{\mathcal{L}}_k, \forall k, \tag{9b}$$

$$\mathrm{SE}_k(\mathbf{p}) \geq \nu_k,\ \forall k, \tag{9c}$$

$$\sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{T_k} f_k^{\mathsf{CPU}}(i) \leq f^{\mathsf{CPU}}, \tag{9d}$$

$$\sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{T_k} f_{l,k}^{\mathsf{AP}}(i) \leq f_l^{\mathsf{AP}},\ \forall l, \tag{9e}$$

$$u(f_k^{\mathsf{CPU}}(i)) + \sum\nolimits_{l \in \mathcal{O}_k} u(f_{l,k}^{\mathsf{AP}}(i)) = 1,\ \forall k, \forall i, \tag{9f}$$

$$\mathbf{0}_K \preceq \mathbf{p} \preceq p_{\max} \cdot \mathbf{1}_K, \tag{9g}$$

$$f \in \mathbb{R}_{\geq 0},\ \forall f \in \mathcal{F}. \tag{9h}$$

In the above problem, $\widetilde{\mathcal{L}}_k = \mathcal{L}_k - (2b_kM\xi/C_{\mathsf{FH}})$, $p_{\max}$ is the maximum transmit power per user, ν_k represents the minimum instantaneous SE target for user k and $\boldsymbol{\nu} = [\nu_1 \cdots \nu_K]^{\mathrm{T}}$. Constraints (9d) and (9e) ensure that the allocate computational cycles do not exceed the computational capacity of the CPU and the APs, respectively, while constraint (9f), where $u(\cdot)$ denotes the unit-step function, ensures that each computational subtask is executed in one processing unit only, i.e. either at the CPU or in one of the APs. Constraint (9h) ensures that the allocated cycles are positive real numbers, and possible non-integer optimal solutions are then rounded (i.e., continuous relaxation). Moreover,

$$\varpi_{\mathrm{p}} = \frac{\omega_{\mathrm{p}}}{K p_{\max}}, \quad \varpi_{\mathrm{se}} = \frac{\omega_{\mathrm{se}}}{K \max_k \mathrm{SE}_k^{(0)}}, \tag{10}$$

where $\{\mathrm{SE}_k^{(0)}\}$ are constants denoting reference instantaneous SEs attained by a pre-determined uplink power allocation (e.g., setting $p_k = p_{\max},\ \forall k$) and $\omega_{\mathrm{p}},\ \omega_{\mathrm{se}} \in [0, 1]$ so that each term of the objective function (9a) is dimensionless and takes on values in the interval $[0, 1]$. These weights determine a trade-off between the minimization of the total trasmit power and the maximization of the sum SE, hence they indirectly act over the effective offloading latency, which is given by the l.h.s. of the constraint (9b). Problem (9) is clearly non-convex with respect to $\mathbf{p}$ due to the non-convexity (non-concavity) of the latency constraint (9b) and the minimum SE constraint (9c). Moreover, the non-linear constraint (9f) makes the optimization problem a mixed integer one. To solve the above problem, we resort to the alternating optimization approach, i.e. we first set some initialization values for $\mathbf{p}$ and $\boldsymbol{\nu}$ and solve the problem with respect to $\mathcal{F}$, and then for the obtained value of $\mathcal{F}$, we solve the problem with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$. The process is iterated until the value attained by the objective function converges and/or a maximum number of iteration has been reached. Notice that, since at each step the value of the optimized variable is updated only if it leads to a smaller value of the objective function, and since the objective function is bounded from below, the procedure provably converges.

1) Optimization with respect to $\mathcal{F}$: We focus on the problem of determining the frequencies in $\mathcal{F}$ for fixed values of $\mathbf{p}$ and $\boldsymbol{\nu}$. First of all, notice that the objective function in (9) does not depend on $\mathcal{F}$. The purpose of the optimization with respect to the computational cycles is thus to make the

second term in (9b) as small as possible so that to make the latency constraint as loose as possible, and the subsequent optimization with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$ can be done on a wider search domain. Based on the above reasoning, we consider the following problem:

$$\underset{k=1,\ldots,K,\ \mathcal{F}}{\text{minimize}} \quad \max_{j=1,\ldots,T_k} \frac{w_{k,j}}{f_k^{\mathsf{CPU}}(j)+\sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j)} \tag{11a}$$

$$\text{s.t.} \quad f_k^{\mathsf{CPU}}(j)+\sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j) \geq \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}}, \quad \forall k,\ \forall j=1,\ldots,T_k, \tag{11b}$$

$$(9d), (9e), (9f), (9h)\ . \tag{11c}$$

In the above formulation, constraint (11b) descends from, and is equivalent to, constraint (9b). An equivalent formulation for problem (11) is the following:

$$\underset{\mathcal{F}}{\text{minimize}} \quad t \tag{12a}$$

$$\text{s.t.} \quad f_k^{\mathsf{CPU}}(j)+\sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j) \geq \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}}, \quad \forall k,\ \forall j=1,\ldots,T_k, \tag{12b}$$

$$f_k^{\mathsf{CPU}}(j)+\sum\nolimits_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j) \geq \frac{w_{k,j}}{t}, \quad \forall k,\ \forall j=1,\ldots,T_k, \tag{12c}$$

$$(9d), (9e), (9f), (9h)\ . \tag{12d}$$

Problem (12) is a feasibility program; specifically, the goal is to find the optimal frequencies in $\mathcal{F}$ that minimize the value of t and such that the problem (12) admits a non-empty feasible set. This can be accomplished through the *bisection method* [39]. In particular, feasibility should be first checked assuming that t is unboundedly large, i.e., which makes constraints (12b) inactive. If the problem reveals to be feasible with $t \to +\infty$, then the bisection method can be applied; specifically, elaborating on the constraints (12b) and (12c), the following values can be used as the start and the end of the initial search interval on t:

$$\begin{aligned} t_0 &= 0.9\frac{\max_{k,j} w_{k,j}}{\max\{f^{\mathsf{CPU}}, f_1^{\mathsf{AP}}, \ldots, f_L^{\mathsf{AP}}\}}, \\ t_1 &= 1.1 \max_k \left[\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}\right]. \end{aligned} \tag{13}$$

Notice that without constraint (9f), (12) would be a simple linear program which could be solved with any off-the-shelf optimization routine. The presence of (9f), instead, requires further efforts. For a preassigned value of t, namely for each iteration of the bisection algorithm, we consider the following associated problem to ascertain the feasibility of the constraints in (12):

$$\max_{\mathcal{F}} \sum_{k=1}^{K}\sum_{j=1}^{T_k}\left(f_k^{\mathsf{CPU}}(j)+\sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j)\right) \tag{14a}$$

$$\text{s.t. } f_k^{\mathsf{CPU}}(j)+\sum_{l\in\mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j) \geq \max\left\{\frac{w_{k,j}}{t}, \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}}\right\}, \quad \forall k,\ \forall j=1,\ldots,T_k, \tag{14b}$$

$$(9d), (9e), (9f), (9h)\ . \tag{14c}$$

Problem (14) aims at finding both the optimal user-to-MEC-server association for the computation offloading and the optimal amount of remote computational resources to be assigned. This problem can be shown to be cast as a *multiple knapsack problem*. Firstly, we introduce $\dot{f}_{\ell,k}(j),\ \ell\in\mathcal{G}_k$, to denote the computational resources that can be allocated to user k for offloading its subtask j on MEC server ℓ. Hence, $\dot{f}_{\ell,k}(j)$ is equal to either $f_k^{\mathsf{CPU}}(j)$ or $f_{l,k}^{\mathsf{AP}}(j)$, with $l\in\mathcal{O}_k$. Then, we introduce $\mathcal{X} = \{x_{\ell,k,j}\in\{0,1\} : k=1,\ldots,K;\ j=1,\ldots,T_k;\ \ell\in\mathcal{G}_k\}$ to denote the set of the binary variates $\{x_{\ell,k,j}\}$ mathematically handling the user-to-MEC-server association. Hence, the amount of computational resources effectively allocated to user k for its subtask j is given by $\sum_{\ell\in\mathcal{G}_k}\dot{f}_{\ell,k}(j)x_{\ell,k,j}$, with $x_{\ell,k,j}\in\{0,\ 1\}$ and $\sum_{\ell\in\mathcal{G}_k} x_{\ell,k,j}=1$, since each subtask is executed either at the CPU or in one of the APs. Therefore, problem (14) can be rewritten as

$$\max_{\mathcal{F},\ \mathcal{X}} \sum_{k=1}^{K}\sum_{j=1}^{T_k}\sum_{\ell\in\mathcal{G}_k}\dot{f}_{\ell,k}(j)x_{\ell,k,j} \tag{15a}$$

$$\text{s.t.} \sum_{\ell\in\mathcal{G}_k}\dot{f}_{\ell,k}(j)x_{\ell,k,j} \geq \max\left\{\frac{w_{k,j}}{t}, \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}}\right\}, \quad \forall k,\ \forall j=1,\ldots,T_k, \tag{15b}$$

$$(9d), (9e), (9f), (9h)\ , \tag{15c}$$

and $\mathcal{F}$ is redefined as

$$\mathcal{F} = \left\{\dot{f}_{\ell,k}(j)\ :\ k=1,\ldots,K;\ \ell\in\mathcal{G}_k;\ j=1,\ldots,T_k\right\}.$$

Secondly, we avoid optimizing the amount of computational resources in (15) by setting $\dot{f}_{\ell,k}(j)$ to a constant as

$$\xi_{k,j}(t,\mathbf{p}) \triangleq \max\left\{\frac{w_{k,j}}{t}, \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{B\ \mathrm{SE}_k(\mathbf{p})}}\right\}, \forall\, \ell\in\mathcal{G}_k.$$

By doing so, constraint (15b) becomes inactive, and the amount of computational resources to assign to each user will be eventually determined at the end of the bisection algorithm, being possibly dependent on the smallest value of t. While, for each iteration of the bisection algorithm, we only optimize the user-to-MEC-server offloading association. Hence, problem (15) can be reformulated as the following multiple knapsack problem:

$$\underset{\mathcal{X}}{\text{maximize}} \quad \sum_{k=1}^{K}\sum_{j=1}^{T_k}\sum_{\ell\in\mathcal{G}_k}\xi_{k,j}(t,\mathbf{p})x_{\ell,k,j} \tag{16a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\sum_{j=1}^{T_k}\xi_{k,j}(t,\mathbf{p})x_{\ell,k,j} \leq \dot{f}_\ell\ , \quad \forall\ell, \tag{16b}$$

$$\sum_{\ell\in\mathcal{G}_k} x_{\ell,k,j} \leq 1\ , \quad \forall k,\forall j=1,\ldots,T_k\ , \tag{16c}$$

$$x_{\ell,k,j}\in\{0,\ 1\} \quad \forall\ell,\ \forall k,\forall j=1,\ldots,T_k\ , \tag{16d}$$

where $\dot{f}_\ell$ denotes the computational capacity of the MEC server ℓ, and constraint (16c), originally with equality, has been relaxed. In the above problem, the *knapsacks* are the MEC servers at the APs and at the CPU, the knapsacks' capacity is the number of CPU cycles available at each MEC server, the objects to be put in the knapsacks are

the computational loads to be offloaded, and, finally, the *weight* (and *profit*) of the generic computational task is given by $\xi_{k,j}(t,\mathbf{p})$. Problem (16) can be solved using some of the methods available in the literature, see for instance [40, Chapter 6]. Once the problem has been solved, the values of the output variables in $\mathcal{X}$ provide the sought allocation of the CPU cycles to the tasks. For instance, if, for certain values $t=t^*$, $\ell=\ell^*$, $k=k^*$ and $j=j^*$ we have $x_{\ell^*,k^*,j^*}=1$, this means that subtask j^* of user k^* is to be executed at the MEC server ℓ^*, using $\xi_{k^*,j^*}(t^*,\mathbf{p})$ cycles per second. Hence, the optimal $\mathcal{F}$ is obtained upon the optimal $\mathcal{X}$ and value of t at the last iteration of the bisection method as $\mathcal{F}=\{\xi_{k,j}(t,\mathbf{p})\,x_{\ell,k,j}:\,k=1,\ldots,K;\,\ell\in\mathcal{G}_k;\,j=1,\ldots,T_k\}$.

2) Optimization with respect to $\mathbf{p}$ *and* $\boldsymbol{\nu}$*:* Let us assume now that the elements in $\mathcal{F}$ are fixed and let us solve problem (9) with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$. Basically, we have to minimize the objective function in (9) with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$ and with constraints (9b), (9c) and (9g). The problem is not convex due to the spectral efficiency expression $\mathrm{SE}_k(\mathbf{p})$ in constraints (9b) and (9c). We thus resort to *sequential convex programming*, that is an iterative optimization framework wherein in each iteration we optimize a related convex approximation of the original problem. Notice that the uplink instantaneous SE for user k can be expressed as

$$\begin{aligned}\mathrm{SE}_k(\mathbf{p}) &= \frac{\tau_\mathrm{u}}{\tau_\mathrm{c}}\log_2\left(1+\frac{\mathsf{num}_k(\mathbf{p})}{\mathsf{den}_k(\mathbf{p})}\right)\\ &= \frac{\tau_\mathrm{u}}{\tau_\mathrm{c}}[\log_2(\mathsf{num}_k(\mathbf{p})+\mathsf{den}_k(\mathbf{p}))-\log_2(\mathsf{den}_k(\mathbf{p}))], \quad (17)\end{aligned}$$

where $\mathsf{num}_k(\mathbf{p})$ and $\mathsf{den}_k(\mathbf{p})$ describe the numerator and the denominator of (5), respectively, which are functions of the power coefficients. As $\log_2(\cdot)$ is increasing and the summation preserves concavity, the r.h.s. of (17) is the difference of two concave functions. Recalling that any concave function is upper-bounded by its Taylor expansion around any given point $\mathbf{p}^{(0)}$, a concave lower-bound of $\mathrm{SE}_k(\mathbf{p})$ is obtained as

$$\begin{aligned}\mathrm{SE}_k(\mathbf{p}) &\geq \frac{\tau_\mathrm{u}}{\tau_\mathrm{c}}\Bigg[\log_2(\mathsf{num}_k(\mathbf{p})+\mathsf{den}_k(\mathbf{p}))-\log_2(\mathsf{den}_k(\mathbf{p}^{(0)}))\\ &\qquad -\nabla_{\mathbf{p}}^{\mathrm{T}}\log_2(\mathsf{den}_k(\mathbf{p}))\Big|_{\mathbf{p}=\mathbf{p}^{(0)}}(\mathbf{p}-\mathbf{p}^{(0)})\Bigg]\\ &= \widetilde{\mathrm{SE}}_k(\mathbf{p},\mathbf{p}^{(0)})\,. \quad (18)\end{aligned}$$

Hence, constraints (9b) and (9c) can be approximated and convexified by taking

$$\begin{aligned}\frac{b_k}{B\,\widetilde{\mathrm{SE}}_k(\mathbf{p},\mathbf{p}^{(0)})}+\max_{j=1,\ldots,T_k}\frac{w_{k,j}}{f_k(j)} &\leq \widetilde{\mathcal{L}}_k\,,\ \forall k,\\ \widetilde{\mathrm{SE}}_k(\mathbf{p},\mathbf{p}^{(0)}) &\geq \nu_k\,,\ \forall k,\end{aligned}$$

for any feasible choice of $\mathbf{p}^{(0)}$. The arguments above hold if the receive combining vector is independent of the uplink powers. This is not true in general, as for the P-MMSE combining scheme in (6). In this case, $\widetilde{\mathrm{SE}}_k(\mathbf{p},\mathbf{p}^{(0)})$ is still a non-linear function of the uplink powers. This issue can be tackled by treating the combining vectors at the n-th iteration of the SCA optimization framework as constant with respect to the current uplink transmit powers $\mathbf{p}^{(n)}$, and being exclusively function of $\mathbf{p}^{(n-1)}$. Hence, the problem to be solved at the n-th iteration of the proposed SCA method can be formulated as

$$\begin{aligned}\underset{\mathbf{p}^{(n)},\,\boldsymbol{\nu}^{(n)}}{\text{minimize}}\quad & \varpi_\mathrm{p}\mathbf{1}_K^{\mathrm{T}}\mathbf{p}^{(n)}-\varpi_\mathrm{se}\mathbf{1}_K^{\mathrm{T}}\boldsymbol{\nu}^{(n)} && (19a)\\ \text{s.t.}\quad & \frac{b_k}{B\,\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n)},\mathbf{p}^{(n-1)})\big|_{\mathbf{v}_k^{(n)}(\mathbf{p}^{(n-1)})}}\\ &\qquad +\max_{j=1,\ldots,T_k}\frac{w_{k,j}}{f_k(j)}\leq\widetilde{\mathcal{L}}_k,\forall k, && (19b)\\ & \widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n)},\mathbf{p}^{(n-1)})\big|_{\mathbf{v}_k^{(n)}(\mathbf{p}^{(n-1)})}\geq\nu_k^{(n)},\ \forall k, && (19c)\\ & \mathbf{0}_K\preceq\mathbf{p}^{(n)}\preceq p_\mathrm{max}\cdot\mathbf{1}_K, && (19d)\end{aligned}$$

where ϖ_p and ϖ_se are set as in (10) with $\mathrm{SE}_k^{(0)}=\mathrm{SE}_k(\mathbf{p}^{(0)})$, $\forall k$. Notice that the notation $\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n)},\mathbf{p}^{(n-1)})\big|_{\mathbf{v}_k^{(n)}(\mathbf{p}^{(n-1)})}$ emphasizes that the receive combining vectors, involved in the expression of the SE, are constant with respect to the current transmit powers $\mathbf{p}^{(n)}$ (i.e., the optimization variables), and are computed upon the optimal values of the transmit powers at the previous iteration of the SCA algorithm, namely $\mathbf{p}^{(n-1)}$. For any iteration n of the SCA method, $\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n)},\mathbf{p}^{(n-1)})\big|_{\mathbf{v}_k^{(n)}(\mathbf{p}^{(n-1)})}$ is a suitable convex approximation of $\mathrm{SE}_k(\mathbf{p}^{(n)})$, as the following properties are fulfilled [41]:

$$\begin{aligned}&\mathrm{SE}_k(\mathbf{p}^{(n)})\geq\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n)},\mathbf{p}^{(n-1)})\big|_{\mathbf{v}_k^{(n)}(\mathbf{p}^{(n-1)})},\ \forall n,\ \forall k, && (20a)\\ &\mathrm{SE}_k(\mathbf{p}^{(n-1)})=\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n-1)},\mathbf{p}^{(n-1)}),\ \forall n,\ \forall k, && (20b)\\ &\nabla_{\mathbf{p}}\mathrm{SE}_k(\mathbf{p}^{(n-1)})=\nabla_{\mathbf{p}}\widetilde{\mathrm{SE}}_k(\mathbf{p}^{(n-1)},\mathbf{p}^{(n-1)}),\ \forall n,\ \forall k. && (20c)\end{aligned}$$

According to the theory in [41], by virtue of the properties (20a), (20b), the sequence of the values attained by the objective function (9a) at the optimal points of each iteration of the SCA algorithm is monotonically decreasing with the increase of the iteration number and converges to a finite limit. Moreover, due to the property (20c), the optimal solution of the SCA algorithm at convergence satisfies the Karush-Kuhn-Tucker (KKT) conditions of problem (9).

Algorithm 1, to be run at the CPU, summarizes the proposed alternate maximization strategy for sub-optimally solving problem (9).

B. Algorithm initialization

We now discuss on how to initialize with a suitable power vector $\mathbf{p}^{(0)}$ the alternating optimization procedure in Algorithm 1. The procedure that we adopt is the following: first of all, we start by considering an allocation of the computational resources in $\mathcal{F}$ minimizing the maximum SE requirement, so that the load of the latency constraint on the air interface of the system is minimized; next, we compute the transmit powers needed to achieve the found minimum SE requirement for all the users. The obtained values of the transmit power will be used to initialize the proposed JCPA algorithm. To begin with, we notice that constraint (9b) can be written as

$$\mathrm{SE}_k(\mathbf{p})\geq\frac{b_k/B}{\widetilde{\mathcal{L}}_k-\max\limits_{j=1,\ldots,T_k}\dfrac{w_{k,j}}{f_k(j)}}\,,\quad\forall k\,. \quad (21)$$

Algorithm 1 Alternating optimization for problem (9)

Input: Any choice of feasible transmit powers $\mathbf{p}^{(0)}$, $M_{\max}$, ϵ;

Compute the SE values: $\boldsymbol{\nu}^{(0)} \leftarrow \left[\mathrm{SE}_1(\mathbf{p}^{(0)}), \ldots, \mathrm{SE}_K(\mathbf{p}^{(0)})\right]^{\mathrm{T}}$;
Evaluate the objective function in (9);
Initialize $m \leftarrow 1$;
repeat
  %% Updating $\mathcal{F}$
  Check feasibility of (16) with $t = +\infty$;
  if (16) is unfeasible **then**
    Exit procedure and declare the problem unfeasible;
  else
    Set t_0 and t_1 as in (13);
    repeat %% Bisection algorithm
      $t \leftarrow (t_0 + t_1)/2$;
      Solve problem (16) with current value of t;
      if (16) is unfeasible **then** $t_0 \leftarrow t$; **else** $t_1 \leftarrow t$;
    until $|(t_1 - t_0)/t_1| \leq \epsilon$
    Set $\mathcal{F}$ as resulting from the solution of (16) with $t = t_1$;
    %% Updating $\mathbf{p}$ and $\boldsymbol{\nu}$
    Initialize $n \leftarrow 1$; $\widetilde{\mathbf{p}}^{(0)} \leftarrow \mathbf{p}^{(n)}$;
    Initialize $\widetilde{\boldsymbol{\nu}}^{(0)} \leftarrow [\mathrm{SE}_1(\widetilde{\mathbf{p}}^{(0)}), \ldots, \mathrm{SE}_K(\widetilde{\mathbf{p}}^{(0)})]^{\mathrm{T}}$;
    repeat %% SCA algorithm
      Let $\mathbf{p}^\star, \boldsymbol{\nu}^\star$ be the optimal solutions of problem (19);
      $\widetilde{\mathbf{p}}^{(n)} \leftarrow \mathbf{p}^\star$; $\widetilde{\boldsymbol{\nu}}^{(n)} \leftarrow \boldsymbol{\nu}^\star$; $n \leftarrow n + 1$;
    until convergence
  end if
  $m \leftarrow m + 1$;
  Evaluate the new value of the objective function in (9);
until the objective function in (9) converges or $m == M_{\max}$

Output: $\mathbf{p}, \mathcal{F}, \boldsymbol{\nu}$;

We thus seek for a set of computational rates $\mathcal{F}$ minimizing the maximum SE requirement, i.e. we focus on the optimization problem

$$\begin{aligned} \underset{\mathcal{F}}{\text{minimize}} \quad & \max_k \frac{b_k/B}{\widetilde{\mathcal{L}}_k - \max\limits_{j=1,\ldots,T_k} \frac{w_{k,j}}{f_k(j)}} \\ \text{s.t.} \quad & (9\text{d}), (9\text{e}), (9\text{f}), (9\text{h}) \,, \end{aligned} \tag{22}$$

which can be written in epigraph form as

$$\min_{\mathcal{F}} \; t \tag{23a}$$

$$\text{s.t.} \;\; f_k^{\mathsf{CPU}}(j) + \sum_{l \in \mathcal{O}_k} f_{l,k}^{\mathsf{AP}}(j) \geq \frac{w_{k,j}}{\widetilde{\mathcal{L}}_k - \frac{b_k}{t\,B}}, \forall k, \tag{23b}$$

$$(9\text{d}), (9\text{e}), (9\text{f}), (9\text{h}) \,. \tag{23c}$$

The above problem has the same structure as (12) and can be solved following the same procedure outlined in the previous subsection. Let us denote by $\mathcal{F}^\star$ the solution to problem (23). We are now ready to compute the transmit vectors that fulfill the minimum SE requirement. Assuming that the computational rates in (7) are those corresponding to $\mathcal{F}^\star$, the latency constraint requires that

$$R_k(\mathbf{p}) \geq \frac{b_k}{\widetilde{\mathcal{L}}_k - \max\limits_{j=1,\ldots,T_k} \frac{w_{k,j}}{f_k^\star(j)}}, \; \forall k, \tag{24}$$

Algorithm 2 *Standard* power control assuming P-MMSE

Input: $\{\boldsymbol{\Upsilon}\}$, $\{\mathbf{C}_k\}$, $\{\mathbf{D}_k\}$, $\{\hat{\mathbf{h}}_k\}$;

Initialize $n = 0$, $\chi = 1$; $\mathbf{p}^{(0)}$; $\mathbf{v}(\mathbf{p}^{(0)})$;
while $\chi > 0.005$ **do**
  Compute $\mathbf{G}(\mathbf{p}^{(n)})$, $\mathbf{Z}(\mathbf{p}^{(n)})$ according to (27);
  Compute $\rho(\boldsymbol{\Upsilon}\mathbf{G}^{-1}\mathbf{Z})$;
  if $\mathbf{G}$ is non-negative **and** $\rho < 1$ **then**
    $n \leftarrow n + 1$;
    $\mathbf{p}^{(n)} \leftarrow \mathbf{I}(\mathbf{p}^{(n-1)})$;
    Compute $\mathbf{v}_k(\mathbf{p}^{(n)})$ according to (6), $\forall k$;
    $\chi \leftarrow \max\limits_k \left| \frac{p_k^{(n)} - p_k^{(n-1)}}{p_k^{(n-1)}} \right|$;
  else Exit procedure and declare the problem unfeasible;
  end if
end while

Output: $\mathbf{p}^\star \leftarrow \mathbf{p}^{(n)}$;

which represents a QoS requirement. The above inequality translates to the following instantaneous SINR requirement

$$\frac{p_k g_{kk}}{\sum\limits_{i \neq k}^{K} p_i \left(g_{ki} + c_{ki}\right) + p_k c_{kk} + \sigma^2 \left\|\mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k\right\|^2} \geq \underbrace{2^{z_k} - 1}_{\widetilde{\gamma}_k}, \; \forall k, \tag{25}$$

where $g_{ki} = |\mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \hat{\mathbf{h}}_i|^2$, $c_{ki} = \mathbf{v}_k^{\mathrm{H}} \mathbf{D}_k \mathbf{C}_i \mathbf{D}_k \mathbf{v}_k$, and

$$z_k = \frac{b_k \tau_{\mathrm{c}}}{B \tau_{\mathrm{u}}} \left(\widetilde{\mathcal{L}}_k - \max_{j=1,\ldots,T_k} \frac{w_{k,j}}{f_k^\star(j)} \right)^{-1}.$$

Hence, the instantaneous SINR requirement in (25) can be rewritten as the vector inequality

$$\mathbf{p} \succeq \boldsymbol{\Upsilon} \mathbf{G}^{-1} (\mathbf{Z}\mathbf{p} + \sigma^2 \mathbf{u}), \tag{26}$$

where $\boldsymbol{\Upsilon} = \mathrm{diag}(\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_K)$, $\mathbf{u} = \left[\left\|\mathbf{v}_1^{\mathrm{H}} \mathbf{D}_1\right\|^2 \ldots \left\|\mathbf{v}_K^{\mathrm{H}} \mathbf{D}_K\right\|^2 \right]^{\mathrm{T}}$, and

$$[\mathbf{G}]_{ki} = \begin{cases} g_{kk} - c_{kk}\widetilde{\gamma}_k, & \text{if } k = i, \\ 0, & \text{otherwise,} \end{cases}$$
$$[\mathbf{Z}]_{ki} = \begin{cases} 0, & \text{if } k = i, \\ g_{ki} + c_{ki}, & \text{otherwise.} \end{cases} \tag{27}$$

Hence, a set of nonnegative uplink powers can be determined capitalizing on the requirement in (26). The set of SINR targets $\{\widetilde{\gamma}_k\}$ is feasible if and only if all the diagonal elements of $\mathbf{G}$ are nonnegative[5], and the Perron-Frobenius eigenvalue of the matrix $\boldsymbol{\Upsilon}\mathbf{G}^{-1}\mathbf{Z}$, denoted by ρ, is real and nonnegative, and $\rho < 1$ [42]. If these conditions are satisfied, then $\mathbf{I}(\mathbf{p}) = \boldsymbol{\Upsilon}\mathbf{G}^{-1}(\mathbf{Z}\mathbf{p} + \sigma^2\mathbf{u})$ is a *standard interference function* [43], and an optimal solution for the uplink transmit powers, is obtained iteratively through the *standard power control algorithm* [43] as $\mathbf{p}^{(n)} = \mathbf{I}(\mathbf{p}^{(n-1)})$, for any given initial choice $\mathbf{p}^{(0)}$.[6] Algorithm 2 specifies the steps of the *standard power control algorithm* based on sequential optimization, and assuming P-MMSE. If Algorithm 2 converges to an optimal solution, say $\mathbf{p}^\star$, then this solution can be used as initial feasible choice for Algorithm 1, that is $\mathbf{p}^{(0)} = \mathbf{p}^\star$. Whenever a set of feasible uplink transmit powers is found (i.e., line 7 of

[5]Notice that it is not guaranteed that $g_{kk} - c_{kk}\widetilde{\gamma}_k \geq 0, \forall k$.

[6]This optimal solution satisfies all the inequalities in (26) with equality, and minimizes the sum of the transmitted powers [43].

Algorithm 2), the receive combining vectors $\{\mathbf{v}_k\}$ must be updated accordingly, such that the matrix $\boldsymbol{\Upsilon}\mathbf{G}^{-1}\mathbf{Z}$ at the next iteration is properly computed. If this two-stage procedure fails to find a feasible set of uplink powers and computational rates, then a non-empty feasible set for problem (19) can anyhow be enforced by a proper admission control [44], [45].

IV. Benchmarks

A. Co-located Massive MIMO System Model and Resource Allocation

A co-located massive MIMO system can be seen as a special case of a CF-mMIMO wherein each user is served by only one of the few deployed base stations (BSs), each of which is equipped with many antennas. An achievable uplink SE for user k served by BS l, is given by

$$\overline{\mathrm{SE}}_{lk}^{(\mathrm{cell})} = \frac{\tau_{\mathrm{u}}}{\tau_{\mathrm{c}}} \mathsf{E}\left\{\log_2(1+\mathrm{SINR}_{lk}^{(\mathrm{cell})})\right\}, \quad \text{where} \tag{28}$$

$$\mathrm{SINR}_{lk}^{(\mathrm{cell})} = \frac{p_k |\mathbf{v}_{lk}^{\mathrm{H}} \hat{\mathbf{h}}_{lk}|^2}{\sum\limits_{i\neq k}^{K} p_i |\mathbf{v}_{lk}^{\mathrm{H}} \hat{\mathbf{h}}_{li}|^2 + \mathbf{v}_{lk}^{\mathrm{H}} \left(\sum\limits_{i=1}^{K} p_i \mathbf{C}_{li}\right) \mathbf{v}_{lk} + \sigma^2 \|\mathbf{v}_{lk}\|^2}, \tag{29}$$

for an arbitrary receive combining vector $\mathbf{v}_{lk} \in \mathbb{C}^M$. The effective uplink SINR, $\mathrm{SINR}_{lk}^{(\mathrm{cell})}$, is maximized by using the *Local*-MMSE[7] (L-MMSE) receive combining, which is

$$\mathbf{v}_{lk}^{\mathsf{L-MMSE}} = p_k \left(\sum_{i=1}^{K} p_i \left(\hat{\mathbf{h}}_{li}\hat{\mathbf{h}}_{li}^{\mathrm{H}} + \mathbf{C}_{li}\right) + \sigma^2 \mathbf{I}_M\right)^{-1} \hat{\mathbf{h}}_{lk}. \tag{30}$$

L-MMSE is not scalable but can be used as a benchmark, since it constitutes the optimal combining scheme for cellular networks [9]. For co-located massive MIMO, we can reasonably assume that only the serving BS offers computational facility to its user terminals. Hence, using the same notation as in (9), we can formulate the JPCA problem for cellular networks as

$$\underset{\mathbf{p},\ \boldsymbol{\zeta}\in\mathbb{R}_{>0}^{K},\ \boldsymbol{\nu}}{\text{minimize}} \quad \varpi_{\mathrm{p}} \mathbf{1}_K^{\mathrm{T}} \mathbf{p} - \varpi_{\mathrm{se}} \mathbf{1}_K^{\mathrm{T}} \boldsymbol{\nu} \tag{31a}$$

$$\text{s.t.} \quad \frac{b_k}{B\,\mathrm{SE}_{lk}(\mathbf{p})} + \frac{w_k}{\boldsymbol{\zeta}(k)} \leq \mathcal{L}_k^{\mathrm{cell}},\ \forall k,\ l \in \mathcal{M}_k, \tag{31b}$$

$$\mathrm{SE}_{lk}(\mathbf{p}) \geq \nu_k,\ \forall k,\ l \in \mathcal{M}_k, \tag{31c}$$

$$\tilde{\mathbf{c}}_l^{\mathrm{T}} \boldsymbol{\zeta} \leq f_l^{\mathsf{BS}}, \forall l, \tag{31d}$$

$$\mathbf{0}_K \preceq \mathbf{p} \preceq p_{\max} \cdot \mathbf{1}_K, \tag{31e}$$

where SE_{lk} represents the instantaneous SE, that is the value attained by (28) without expectation and $\mathcal{L}_k^{\mathrm{cell}}$ denotes the maximum tolerable latency for user k in the cellular setup. This latency value is presumably larger than its cell-free counterpart as it does not include the delay produced by the fronthaul signalling (i.e., steps 2–5 in Fig. 1). Moreover, in (31), f_l^{BS} indicates the computational capability of BS l, $\boldsymbol{\zeta}$ denotes a $K\times 1$ vector of optimization variables, where its k-th element, $\boldsymbol{\zeta}(k)$, represents the amount of computational resources allocated to user k by its serving BS l, that is $\boldsymbol{\zeta}(k) = f_{l,k}^{\mathsf{BS}},\ l \in \mathcal{M}_k$, with $|\mathcal{M}_k| = 1$. Moreover, $\tilde{\mathbf{c}}_l \in \{0,1\}^K$ is an auxiliary binary vector, where the k-th element is 1 if BS l serves user k, and 0 otherwise. Lastly, ν_k represents the minimum instantaneous SE for user k. Problem (31) can be convexified via sequential optimization, using a similar methodology as in (18). Hence, the optimization problem at the n-th iteration of the SCA method can be formulated as

$$\underset{\substack{\mathbf{p}^{(n)},\boldsymbol{\nu}^{(n)}\\ \boldsymbol{\zeta}^{(n)}\in\mathbb{R}_{>0}^{K}}}{\text{minimize}} \quad \varpi_{\mathrm{p}} \mathbf{1}_K^{\mathrm{T}} \mathbf{p}^{(n)} - \varpi_{\mathrm{se}} \mathbf{1}_K^{\mathrm{T}} \boldsymbol{\nu}^{(n)} \tag{32a}$$

$$\text{s.t.} \quad \frac{b_k}{B\ \widetilde{\mathrm{SE}}_{lk}(\mathbf{p}^{(n)}, \mathbf{p}^{(n-1)})\big|_{\mathbf{v}_{lk}^{(n)}(\mathbf{p}^{(n-1)})}} + \frac{w_k}{\boldsymbol{\zeta}^{(n)}(k)} \leq \mathcal{L}_k^{\mathrm{cell}},\ \forall k, \forall l \in \mathcal{M}_k, \tag{32b}$$

$$\widetilde{\mathrm{SE}}_{lk}(\mathbf{p}^{(n)}, \mathbf{p}^{(n-1)})\big|_{\mathbf{v}_{lk}^{(n)}(\mathbf{p}^{(n-1)})} \geq \nu_k^{(n)},\ \forall k, \forall l \in \mathcal{M}_k, \tag{32c}$$

$$\tilde{\mathbf{c}}_l^{\mathrm{T}} \boldsymbol{\zeta}^{(n)} \leq f_l^{\mathsf{BS}}, \forall l, \tag{32d}$$

$$\mathbf{0}_K \preceq \mathbf{p}^{(n)} \preceq p_{\max} \cdot \mathbf{1}_K, \tag{32e}$$

where ϖ_{p} is set as in (10), $\varpi_{\mathrm{se}} = \omega_{\mathrm{se}}/[K \max\limits_{l,k} \mathrm{SE}_{lk}(\mathbf{p}^{(0)})]$, and $\widetilde{\mathrm{SE}}_{lk}(\mathbf{p}^{(n)}, \mathbf{p}^{(n-1)})$ is a concave lower-bound of $\mathrm{SE}_{lk}(\mathbf{p}^{(n)})$ around the point $\mathbf{p}^{(n-1)}$, obtained by using the same methodology as in (18). The SCA algorithm is run in a centralized fashion by a network entity, e.g., one of the BSs, an its convergence is guaranteed, as $\widetilde{\mathrm{SE}}_{lk}(\mathbf{p}^{(n)}, \mathbf{p}^{(n-1)})$ is a suitable convex approximation of $\mathrm{SE}_{lk}(\mathbf{p}^{(n)})$. As per the feasibility, problem (32) admits a non-empty feasible set if

$$R_k > \frac{b_k}{\mathcal{L}_k^{\mathrm{cell}}},\ \forall k, \text{ and } \sum_{k\in\mathcal{K}_l} \frac{w_k}{\mathcal{L}_k^{\mathrm{cell}} - b_k/R_{lk}} < f_l^{\mathsf{BS}},\ \forall l, \tag{33}$$

where $R_{lk} = B \times \mathrm{SE}_{lk}$, is the uplink instantaneous rate of user k served by BS l, and $\mathcal{K}_l$ is the set of the users served by BS l. The conditions in (33) are necessary but not sufficient due to the interference-limited scenario that makes simultaneously maximizing the per-user SEs intractable.

B. Heuristic Resource Allocation for Distributed Network Topology

In this section, we propose an alternative approach to the JPCA for cell-free massive MIMO which consists in heuristically allocating the MEC server computational resources to the users according to a pre-determined metric, and then optimizing with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$ as described in Section III-A. Hence, unlike the JPCA, such a heuristic resource allocation does not jointly optimize the uplink power consumption and the allocated computational resources. However, it represents a low-complexity solution with respect to the JPCA which requires solving a multiple knapsack problem—an *NP-hard* problem in strong sense [40]. The psuedo-code of the proposed heuristic allocation of uplink powers and computational resources is reported in Algorithm 3. As for this heuristic scheme, we assume that a subtask offloaded by any user can either be processed at the CPU MEC server or at one of the AP MEC servers. Firstly, the subtasks are sorted in descending order by the metric $\mu_{k,j} = w_{k,j}\left(\widetilde{\mathcal{L}}_k - b_k/R_k\right)^{-1}$ [cycles/s], $k = 1,\ldots,K; i = 1,\ldots,T_k$, which represents the computational demand of user

[7]The AP in each cell performs MMSE combining on its own, only relying upon the local channel estimates.

Algorithm 3 Heuristic allocation of uplink powers and computational resources

Input: $\{\mu_{k,j}\}$, $\{\dot{f}_\ell\}$;
1: Initialize $\dot{f}_{\ell,k}(j)=0,\ \forall \ell,\ \forall k,\ \forall j$;
2: **for** each task in descending order by $\mu_{k,j}$ **do**
3: $\bar{f}_\ell = \dot{f}_\ell - \sum_{k=1}^{K}\sum_{j=1}^{T_k}\dot{f}_{\ell,k}(j),\ \ell\in\mathcal{G}_k$;
4: $\ell^* = \arg\max_\ell \bar{f}_\ell$;
5: **if** $\mu_{k,j}\le \bar{f}_{\ell^*}$ **then** $\dot{f}_{\ell^*,k}(j)=\mu_{k,j}$;
6: **else** Exit procedure and declare the problem unfeasible;
7: **end if**
8: **end for**
9: $\vartheta_\ell = \dot{f}_\ell / \sum_{k=1}^{K}\sum_{j=1}^{T_k}\dot{f}_{\ell,k}(j),\ \forall \ell:\ \exists\ \ell,k,j.\ \dot{f}_{\ell,k}(j)\neq 0$;
10: $\dot{f}_{\ell,k}(j)=\vartheta_\ell \dot{f}_{\ell,k}(j),\ \forall \ell\in\mathcal{G}_k$;
11: Initialize $n\leftarrow 1$; $\widetilde{\mathbf{p}}^{(0)}\leftarrow \mathbf{p}^{(n)}$;
12: Initialize $\widetilde{\boldsymbol{\nu}}^{(0)}\leftarrow [\mathrm{SE}_1(\widetilde{\mathbf{p}}^{(0)}),\ldots,\mathrm{SE}_K(\widetilde{\mathbf{p}}^{(0)})]^{\mathrm{T}}$;
13: **repeat** `%% SCA algorithm`
14: Let $\mathbf{p}^\star,\boldsymbol{\nu}^\star$ be the optimal solutions of problem (19);
15: $\widetilde{\mathbf{p}}^{(n)}\leftarrow \mathbf{p}^\star$; $\widetilde{\boldsymbol{\nu}}^{(n)}\leftarrow \boldsymbol{\nu}^\star$; $n\leftarrow n+1$;
16: **until** convergence
Output: $\mathbf{p},\mathcal{F},\boldsymbol{\nu}$;

k for subtask j related to its latency requirements and uplink rate, conditioned to a pre-determined power allocation. Then, each task is offloaded at the MEC server with more available computing resources, either at one of the APs or at the CPU. The fractions of computational resources assigned to each user's subtask are further scaled so as to saturate the computational capabilities of those APs and (possibly) the CPU involved in the offloading process. Once the set $\mathcal{F}$ is determined, Algorithm 3 concludes by solving problem (19) as described in Section III-A. Hence, this scheme heuristically allocates the remote computational resources and only optimizes with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$, unless the computational capabilities at the MEC servers are insufficient, i.e., there exists at least a subtask j of user k such that $\mu_{k,j} > \max_{\ell\in\mathcal{G}_k}\bar{f}_\ell$, with $\bar{f}_\ell$ being the "online" available computing resources at the CPU and at each of the AP MEC server. Notice that if the computational resource allocation problem in Algorithm 3 is feasible, then the necessary but not sufficient condition for the feasibility of problem (19) at the first iteration is $R_k > b_k/\widetilde{\mathcal{L}}_k, \forall k$.

C. Small-Cell Implementation and Resource Allocation

With the terminology *small-cell* we indicate an instance of cell-free network where each user receives communication service from only one of the APs and computational offloading service from either one AP or the CPU. With respect to an arbitrary user k, it holds $|\mathcal{M}_k|=1$ and $|\mathcal{G}_k|=1$, and not necessarily $\mathcal{M}_k$ coincides with $\mathcal{G}_k$. A similar MEC-enabled architecture was advocated in [34] whose task offloading model is random and arbitrarily hinges on an offloading probability. Moreover, in [34] both uplink transmit powers and allocated computational resources are fixed rather than optimized. Conversely, we herein consider a deterministic, heuristic task offloading model which accounts for the user computational demands and available remote computing resources. Specifically, the set $\mathcal{M}_k$ comprises the AP with the best average channel gain towards user k. While, the set $\mathcal{G}_k$ comprises the MEC server with more available computing resources according to Algorithm 3 (up to line 15). Importantly, since each user receives computational offloading service from only one MEC server, the offloading process is carried out on a task basis rather than on a subtask basis, namely for an arbitrary user k it holds $T_k=1$. As for the small-cell implementation, an achievable uplink SE for user k served by AP l is given by equations (28)-(29), which is maximized by the LMMSE combining scheme in (30). Since data decoding is performed locally at the AP, there is no need for the AP to forward the uplink data signal $\mathbf{y}$ to the CPU through the fronthaul network, but it can transmit the b_k bits directly to the MEC server in charge of the offloading process. As $b_k/C_{\mathsf{FH}} \ll 2b_kM\xi/C_{\mathsf{FH}}$, and the former is very small, we assume that $\mathcal{L}_k=\widetilde{\mathcal{L}}_k$ for the small-cell implementation.

V. Simulation Results

We consider a coverage area of 1 km^2 served by a total number of antennas $N=LM=400$. For the co-located massive MIMO setup, we choose $L=4$ BSs, equipped with $M=100$ antennas each, and deployed as a regular grid with intersite distance equal to 500 m. For the CF-mMIMO setup, we select $L=100$ APs, equipped with $M=4$ antennas each, and deployed as a regular grid with intersite distance equal to 100 m. For all the setups a wrap-around simulation technique is used to remove the edge effects of the (nominal) coverage area. All the systems operate at 2 GHz carrier frequency, over a communication bandwidth $B=20$ MHz. The receiver noise power is conventionally set to -94 dBm, while the maximum transmit power per user is $p_{\max}=100$ mW. The TDD coherence block is $\tau_{\mathrm{c}}=200$ samples long, $\tau_{\mathrm{d}}=0$, and $\tau_{\mathrm{p}}=5$ samples is the uplink training duration. All the setups serve the same set of $K=20$ users, that are uniformly distributed at random over the coverage area. A random realization of users' locations defines a network snapshot, and determines a set of large-scale fading coefficients. These are computed according to the 3GPP Urban Microcell model defined in [46, Table B.1.2.1-1]. The channel correlation matrices $\{\mathbf{R}_{lk}\}$ are generated by using the popular *local scattering* [9, Sec. 2.5.3] model assuming half-wavelength spaced ULAs, and jointly Gaussian angular distributions of the multipath components around the nominal azimuth and elevation angles. The random variations in the azimuth and elevation angles are assumed to be independent, and the corresponding angular standard deviations (ASDs) are equal to 15°, which represents strong spatial channel correlation.

As $\tau_{\mathrm{p}}<K$, pilots are to be re-assigned across users. To this end, we resort to the joint pilot assignment and AP (BS)-user association described in [9, Sec. 5.4], so as to ensure that users served by the same set of APs (same BS, for the co-located setup) are given orthogonal pilots. Concerning the power control, we assume that the initial choice for the feasible transmit powers of the SCA algorithm follows, for all the setups, a fractional power control strategy given by

$$[\mathbf{p}^{(0)}]_k = p_{\max}\frac{\left(\sum_{l\in\mathcal{M}_k}\beta_{kl}\right)^{-0.5}}{\max_{i\in\mathcal{S}_k}\left(\sum_{l\in\mathcal{M}_i}\beta_{il}\right)^{-0.5}}, \forall k. \tag{34}$$

Concerning the computation-offloading and latency model, for the cell-free setup we assume $f^{\mathsf{CPU}}=10^{10}$ cycles/s,

while f_l^{AP} are uniformly distributed random integers from the interval $[2,\ 4] \times 10^9$ cycles/s. The latency requirements are $\mathcal{L}_k = 0.2$ s $\forall k$. The fronthaul capacity is $C_{\mathsf{FH}} = 10$ Gbps, and the number of bits for quantization is set as $\xi = 16$. As per the co-located setup, we select $f_l^{\mathsf{BS}} = \left\lceil \left(\sum_{1=l}^{L^{\mathsf{AP}}} f_l^{\mathsf{AP}} + f^{\mathsf{CPU}}\right)/L^{\mathsf{BS}} \right\rceil$, where L^{AP} is the number of APs in the cell-free setup, while L^{BS} is the number of BSs in the co-located setup, that is 100 and 4, respectively. This choice ensures the same amount of available computational resources over the simulation area for both the setups. Lastly, the latency requirements for the users in the co-located setup is $\mathcal{L}_k^{\text{cell}} = 0.3$ s $\forall k$. Common to all the setups, the computational bits, $\{b_k\}$, are uniformly distributed random integers from the interval $[1, 4]$ Mbits, and the number of computation cycles needed to run the task itself is set as a linear function of b_k, that is $w_k = \alpha\, b_k$, with $\alpha = 50$ cycles/bit [27]. Finally, the number of subtasks any user's task is divided into is an integer drawn uniformly at random from the interval $[1, 4]$. An instance of the multiple knapsack problem, involved in Algorithm 1, is sub-optimally solved in polynomial time (with respect to the total number of subtasks) by using the *Lagrangian Relaxation* technique [40, Section 6.2.2] combined with the *cross-entropy* optimization method [47].

A. Performance Comparison between Network Architectures

Firstly, we focus on the radiated power consumption. In Fig. 2(a), we show the cumulative distribution function (CDF), obtained over 200 network snapshots, of the uplink transmit power per user, expressed in mWatt, being the solution of the Algorithm 1 and the SCA problem (32) for cell-free and co-located massive MIMO, respectively. With the label "Cell-free, Alg. 3" we refer to the framework wherein uplink powers and computational rates are heuristically allocated according to Algorithm 3. As for the small-cell implementation, we consider two cases: (i) the label "Small-cell" indicates the framework wherein uplink powers and computational rates are not optimized, as in [34]. The uplink powers result from (34), while the computational rates are assigned according to the approach described in lines 1–15 of Algorithm 3, with $T_k = 1$; (ii) the label "Small-cell + Alg. 3" indicates the framework wherein uplink powers and computational rates are heuristically allocated according to Algorithm 3, with $T_k = 1$.In Fig. 2(a), we consider the configuration: $\omega_{\text{p}} = 1$, $\omega_{\text{se}} = 0.5$, which applies to all the scheme but "Small-cell".

Numerical results reveal a dramatic transmit power saving for the CF-mMIMO users as compared to the co-located massive MIMO and the small-cell users. Assuming $\omega_{\text{p}} = 1, \omega_{\text{se}} = 0.5$,—which is the configuration that prioritizes the power saving over the transmission latency—at high percentiles, where the transmit power consumption is more significant, we indeed observe that the CF-mMIMO users can considerably reduce their transmit power as compared to the co-located massive MIMO and small-cell users. In co-located massive MIMO, those users with worse channel conditions, presumably at the cell-edge, need to employ more power to receive the required computational offloading service. In small-cell implementations instead, the users would benefit from a power optimization rather than employing a fixed power control strategy, such as fractional power control. To achieve excellent performance in small-cell implementations, the proposed Algorithm 3 should be employed for a proper resource allocation. Fig. 2(a) also highlights that the proposed heuristic allocation in Algorithm 3 performs as well as the proposed JPCA in Algorithm 1, in terms of power consumption. At low percentiles, presumably corresponding to the users with better channel conditions and exiguous computational demands, the performance gap between co-located massive MIMO and CF-mMIMO (including its instance "Small-cell + Alg. 3") reduces. Importantly, our JPCA scheme in CF-mMIMO is able to guarantee fairness among the users in terms of transmit power consumption. In Fig. 2(b) we consider configurations giving equal and more weight to the SE with respect to the power consumption, with $\omega_{\text{p}} = \omega_{\text{se}} = 1$, and (iii) $\omega_{\text{p}} = 0.5$, $\omega_{\text{se}} = 1$, respectively. The levels of transmit power are larger than those attained by the previous configuration to guarantee larger SEs and thereby reducing the latency of the offloading process. Interestingly, the performance gap between CF-mMIMO and co-located massive MIMO increases as a higher SE is required. The macro-diversity gain enables CF-mMIMO to provide comparable SE levels to those of co-located massive MIMO, yet with lower uplink power consumption.

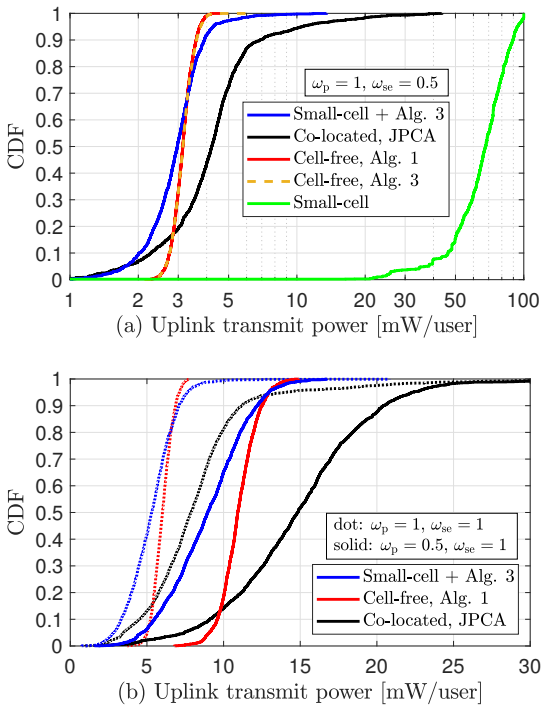

Fig. 2. CDF of the per-user uplink transmit power for cell-free and co-located massive MIMO, assuming three configurations of $\{\omega_{\text{p}}, \omega_{\text{se}}\}$. $K = 20$, and $p_{\text{max}} = 100$ mW for all the users. The x-axis in Fig. 2(a) is in logarithmic scale.

To better motivate the previous performance, we now focus on the amount of computational resources allocated to the users, that is $\{f_k(i)\}$. In Fig. 3(a), we show the CDF of the computational resources allocated to the single user, expressed in GHz ($10^9 \times$cycles/s). While, Fig. 3(b) shows

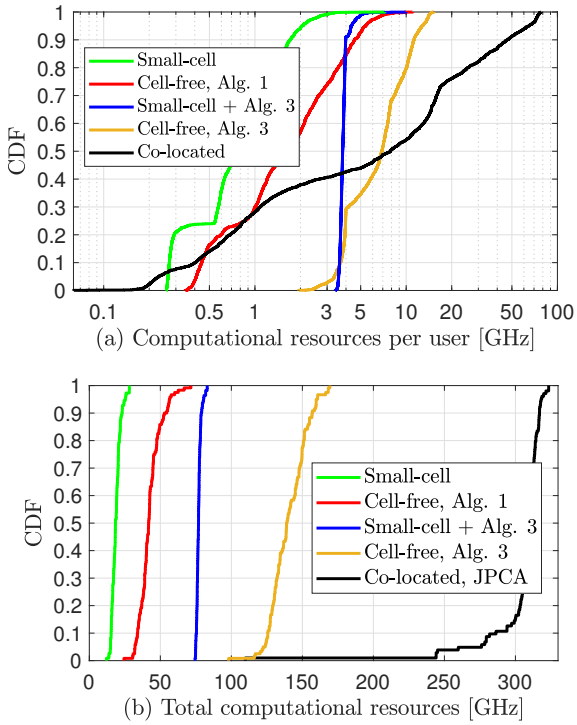

Fig. 3. CDF of the per-user (a) and total (b) computational resources allocated remotely, for cell-free, co-located massive MIMO and small-cell implementations. $f_l^{\mathsf{BS}} = \left\lceil \left(\sum_{l=1}^{L^{\mathsf{AP}}} f_l^{\mathsf{AP}} + f^{\mathsf{CPU}}\right)/L^{\mathsf{BS}} \right\rceil$, $f^{\mathsf{CPU}} = 10$ GHz, $f_l^{\mathsf{AP}} \sim \mathcal{U}(2, 4)$ GHz, $\mathcal{L}_k = 200$ ms and $\mathcal{L}_k^{\text{cell}} = 300$ ms, $\forall k$. The x-axis in Fig. 3(a) is in logarithmic scale.

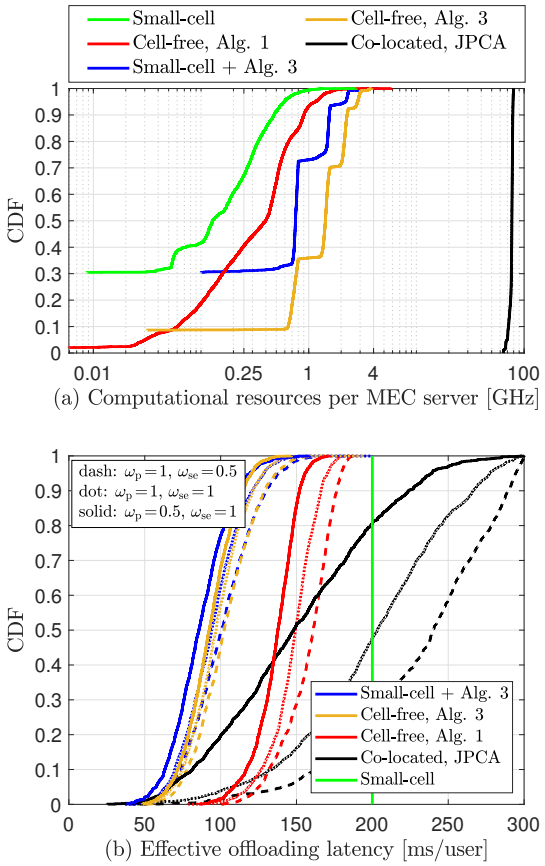

Fig. 4. (a) Computational resources allocated per MEC server. (b) Effective offloading latency per user.

the CDF of the total allocated computational resources, computed as $\sum_{k=1}^{K}\sum_{i=1}^{T_k} f_k(i)$. As we experienced negligible performance differences between the three considered weight configurations, we only report the results achieved with $\omega_{\text{p}} = \omega_{\text{se}} = 1$. The amount of computing resources allocated by Algorithm 3 is significantly larger than that allocated by Algorithm 1 for CF-mMIMO and small-cell implementations. Indeed, the fine-tuning of the allocated computational rates described by lines 16–19 of Algorithm 3 is carried out to utilize all the residual available resources after a first, conservative, feasible allocation based on the metric $\mu_{k,j}$ (the latter constitutes the "Small-cell" resource allocation approach). Allocating more computing resources entails reducing the computational latency and enables to increase the transmission latency as a result of lowering the uplink powers, while meeting the latency constraint. This motivates why "Small-cell + Alg. 3" allows higher levels of power saving than "Cell-free, Alg. 1", and how the performance gap, in terms of power consumption, between the nearly-optimal Algorithm 1 and the heuristic Algorithm 3 is filled. Moreover, we recall that the effective latency constraint for the CF-mMIMO setup is stricter than that of its small-cell counterpart due to the fronthaul latency contribution, and this entails a higher power consumption in CF-mMIMO to further reduce the transmission latency. In co-located massive MIMO, the MEC servers at the BSs offer huge computational power which is fully exploited by the users, especially those with higher computational demands and poor channel conditions, for which drastically reducing the computational latency is the only way to fulfill the latency requirement. Changing perspective, Fig. 4(a) shows the amount of computing resources allocated per MEC server, including APs and CPU. As already mentioned earlier, the "Small-cell" resource allocation is conservative and leads to a misuse of the computational resources, which in turn results to an uplink power waste. Notice that the "Small-cell" approach attains the minimum energy consumption at the MEC servers—which is proportional to the cube of the computational rates—in line with the objective in [34]. The resource allocation for CF-mMIMO via Algorithm 1 leads to excellent uplink power savings with a relative small amount of allocated computational resources per MEC server. On the other hand, the uplink power savings achieved by Algorithm 3, both for CF-mMIMO and small-cells, can only be obtained by increasing the computational rates, hence the energy consumption, at the MEC servers. As per the co-located setup, the MEC servers basically work at full processing capacity to guarantee the latency requirements.

The effective latency experienced by the users due to the offloading process is another relevant aspect to measure the effectiveness of the JPCA scheme, and it is shown in Fig. 4(b). First, we remind that the latency requirements of the cell-free users account for the delay of the data forwarding over the fronthaul network, i.e., step 2 of Fig. 1, thus are effectively stricter than those of the co-located and small-cell users. CF-mMIMO with Algorithm 1 is able to fulfill the latency requirements by a large margin compared to "Small-cell" and

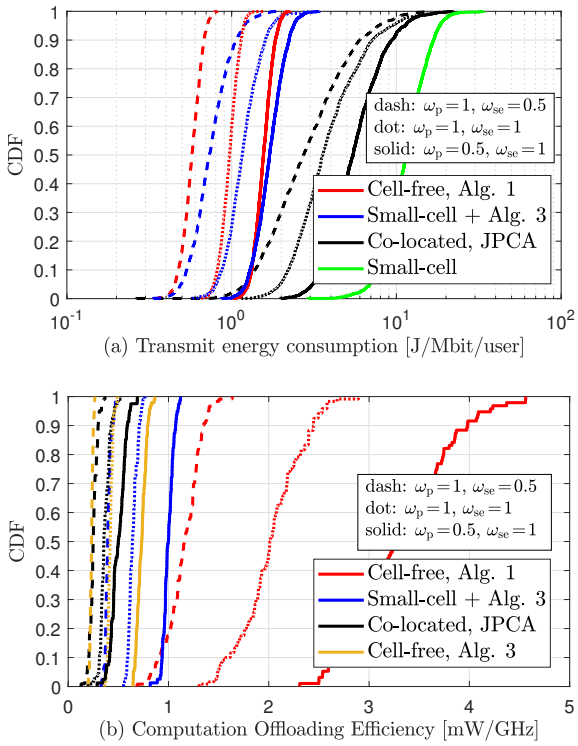

Fig. 5. (a) Transmit energy consumption in J/Mbit/user. (b) Computation offloading efficiency.

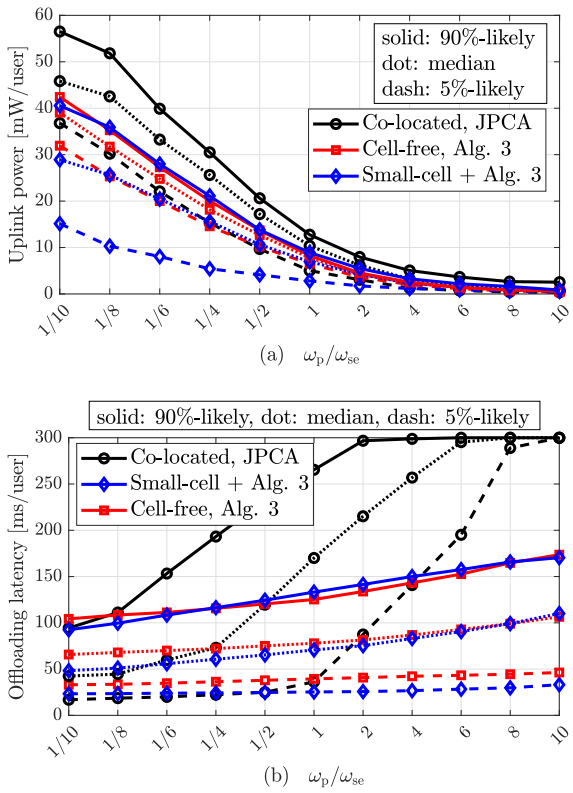

Fig. 6. (a) Average user transmit power and (b) average offloading latency per user as the ratio ω_p/ω_{se} varies.

col-located massive MIMO showing the potentiality to support even stricter requirements. On the other hand, as Algorithm 3 allocates way more computing resources to the users, the computational latency can be remarkably reduced so as to minimize the overall latency experienced by the users. In this regards, the performance of CF-mMIMO and small-cells are almost equivalent when Algorithm 3 is employed. Lastly, notice that the "Small-cell" strategy is designed upon satisfying the latency constraint with equality. The choice of the parameters $\{\omega_p, \omega_{se}\}$ clearly affects the effective offloading latency. The latency increases when the uplink power minimization is prioritized over the SE maximization. This suggests that the transmission latency is dominant over the computational latency in this scenario. Importantly, CF-mMIMO combined with Algorithm 1 can simultaneously guarantee significant transmit power saving and low offloading latency, despite the additional delay due to the transmissions over the fronthaul.

Fig. 5(a-b) show the transmit energy consumption in J/Mbit/user, given by $\mathrm{E}_k = p_k/(B \times \mathrm{SE}_k)$, and the computation offloading efficiency (OE), respectively. We define the OE as

$$\mathrm{OE} = \frac{\sum_{k=1}^{K} p_k}{\sum_{k=1}^{K}\sum_{i=1}^{T_k} f_k(i)} \cdot \frac{\sum_{k=1}^{K} \mathcal{L}_k^{\mathsf{eff}}}{\sum_{k=1}^{K} \mathcal{L}_k^{\mathsf{req}}} \quad [\mathrm{mW/GHz}], \quad (35)$$

where $\mathcal{L}_k^{\mathsf{eff}}$ denotes the effective latency experienced by user k due to the offloading process, i.e., the LHS of the latency constraint in (7) and (31b) for cell-free and co-located massive MIMO, respectively. While, $\mathcal{L}_k^{\mathsf{req}}$ denotes the latency requirement for user k, which is equal to $\mathcal{L}_k$ for CF-mMIMO and small-cells, and equal to $\mathcal{L}_k^{\mathsf{cell}}$ for co-located massive MIMO. This metric relates the optimization variables of our interest to each other, and measures the amount of uplink transmit power needed for 1 GHz of computational resources allocated at the MEC servers, also accounting for how much shorter the effective latency is as compared to the requirement. Hence, the larger this metric is, the more efficient the offloading process is. We observe that cell-free users can save a significant amount of transmit energy with respect to both the co-located and the small-cell users. This confirms the outstanding ability of CF-mMIMO (regardless of the resource allocation algorithm) of simultaneously guarantee low transmission latency and significant transmit energy consumption savings. Not least, fairness among the users is ensured unlike in small-cell and co-located massive MIMO. The energy consumption gap between CF-mMIMO and small-cell is due to the macro-diversity gain provided by the former and increases as we prioritize the power minimization over the SE maximization. Importantly, the OE attained by CF-mMIMO combined with Algorithm 1 is far superior than any other considered approach. This confirms the nearly-optimal nature of the proposed JPCA strategy over a disjoint radio and computational resource allocation (i.e., Algorithm 3) as well as over different network setups, namely small-cells and co-located massive MIMO, and despite the stricter latency requirements. Clearly, the OE increases when prioritizing the SE maximization over the uplink power minimization.

Finally, we investigate the user transmit power and the effective offloading latency as the ratio ω_p/ω_{se} varies. By increasing this ratio, the SCA approach solving the optimization with respect to $\mathbf{p}$ and $\boldsymbol{\nu}$ prioritizes the minimization of the per-user uplink power, which is clearly shown by the monotonic decreasing behaviour of the curves in Fig. 6(a). The uplink

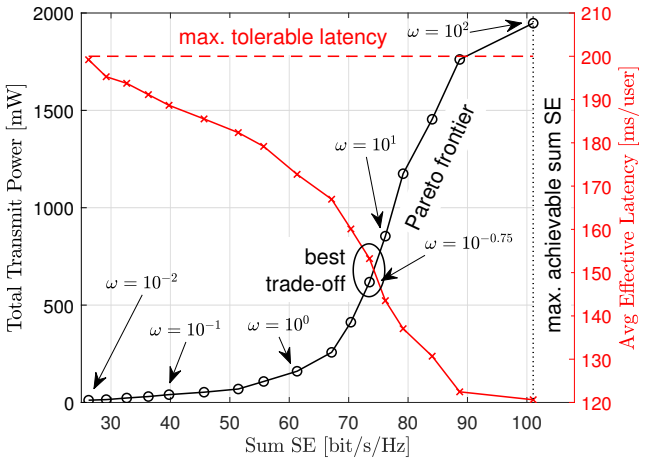

Fig. 7. Sub-optimal Pareto frontier of the JPCA problem in (9) and resulting latency per user. Results obtained by considering the setup in Section V, for one network snapshot and averaging over 200 channel realizations.

power per user achieved in co-located massive MIMO is quite sensitive to the ratio $\omega_{\mathrm{p}}/\omega_{\mathrm{se}}$ and attains values below 10 mW only for $\omega_{\mathrm{p}}/\omega_{\mathrm{se}} \geq 1$, while the power saving in CF-mMIMO and small-cell via Algorithm (3) in the region $\omega_{\mathrm{p}}/\omega_{\mathrm{se}} < 1$ is remarkable. As per the effective offloading latency experienced by the users, Fig. 6(b) shows that CF-mMIMO and small-cell implementations perform equally well. while the gap with respect to the co-located setup becomes tremendous. Co-located massive MIMO is quite sensitive to the offloading latency as $\omega_{\mathrm{p}}/\omega_{\mathrm{se}}$ increases, while for CF-mMIMO and small-cell the effective latency increases softly.

B. Pareto Frontier of the proposed MOOP

As we already mentioned in Section III-A, there is no unique solution for the SOOP in (9), but there exists a set of bounded trade-off Pareto optimal solutions, that is a Pareto optimal that enables improvements in some objectives with bounded trade-offs in others. We first reformulate the objective of problem (9) as $\varpi_{\mathrm{p}} \left(\mathbf{1}_K^{\mathrm{T}}\mathbf{p} - \mathbf{1}_K^{\mathrm{T}}\boldsymbol{\nu}\varpi_{\mathrm{se}}/\varpi_{\mathrm{p}}\right) = \varpi_{\mathrm{p}} \left(\mathbf{1}_K^{\mathrm{T}}\mathbf{p} - const \cdot \omega \mathbf{1}_K^{\mathrm{T}}\boldsymbol{\nu}\right)$, where $\omega = \omega_{\mathrm{se}}/\omega_{\mathrm{p}}$ and $const = p_{\max}/\max_k \mathrm{SE}_k^{(0)}$ are obtained from (10). Notice that the constant factor ϖ_{p} has no effects on the minimization, thus it can be removed from the objective. Finally, a sub-optimal Pareto frontier is obtained by iteratively solving the SOOP according to Algorithm 1 for several values of ω and plotting the corresponding objective values separately, as shown in Fig. 7 (black curve). The results in Fig. 7 are obtained by considering the setup in Section V, for one random realization of APs' and users' locations and averaging over two hundreds random realizations of the small-scale fading. The Pareto frontier reveals the trade-off between the total transmit power minimization and the sum SE maximization according to the selection of ω which, in turn, affects the effective latency experienced by the users (red curve). The value of the design parameter ω that provides the best trade-offs between power saving and latency can be easily identified by inspection from Fig. 7.

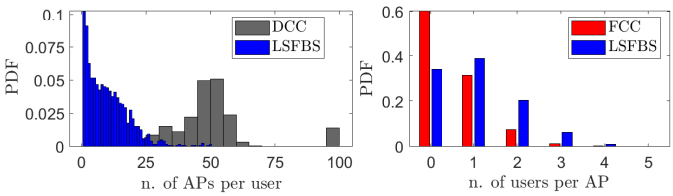

Fig. 8. PDF of the number of (a) APs per user, and (b) users per AP, for different AP selection strategies. $K = 10, \tau_{\mathrm{p}} = G = 5$.

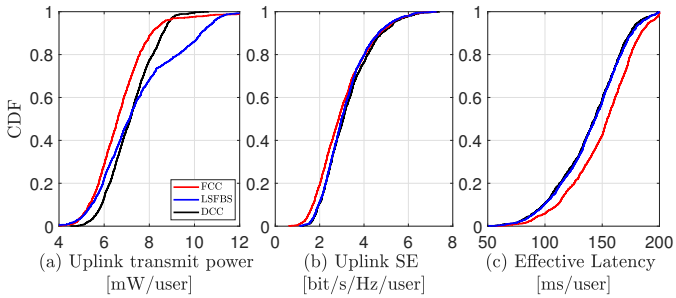

Fig. 9. CDFs of (a) uplink transmit power per user (b) uplink per-user SE (c) effective offloading latency per user, for different AP selection strategies. $K = 10, \tau_{\mathrm{p}} = G = 5$ and $\omega_{\mathrm{p}} = \omega_{\mathrm{se}} = 1$.

C. Impact of the AP Selection Strategy

The simulation results shown in this section aim at highlighting the impact of the AP selection strategy on the JPCA scheme in CF-mMIMO. In section Section V-A, we assumed the AP-to-user association described in [9, Sec. 5.4], so as to ensure that users served by the same set of the best APs are given orthogonal pilots. Hence, if $\tau_{\mathrm{p}} = 5$, each AP participates to the service of up to 5 users. We refer to this scheme as *dynamic cooperation clustering* (DCC). From an energy efficiency viewpoint such an AP selection strategy results to be costly as many APs, even those bringing negligible contribution to the performance, are involved and active both in the radio communication and computational offloading service of a user. We next give a qualitative study of the energy consumption at the server side, by considering alternative AP selection strategies. An AP selection strategy establishes a different fraction of APs involved in the communication service. We consider a fixed cooperation clustering (FCC) scheme, wherein each user, upon the associations established by the DCC scheme, is only served by the best (channel-wise) G APs. In addition, we consider the large-scale-fading-based AP selection (LSFBS) [48], wherein each user, upon the associations established by the DCC scheme, is only served by the APs that contribute to the 95% of its channel gain. Fig. 8 shows the probability density function (PDF) of the number of APs per user and the number of users per AP, for different AP selection strategies, assuming $K = 10, \tau_{\mathrm{p}} = G = 5$. The LSFBS involves a handful of APs per user with high probability, while the DCC scheme selects many APs per user, with a non-negligible probability of selecting all the APs. The FCC scheme always select $G = 5$ APs per user. Changing perspective, 60% and about 38% of the APs is off the communication service with FCC and LSFBS, respectively, while the DCC always selects $\tau_{\mathrm{p}} = 5$ users per AP. As we can observe in Fig. 9, selecting a fixed number of APs per

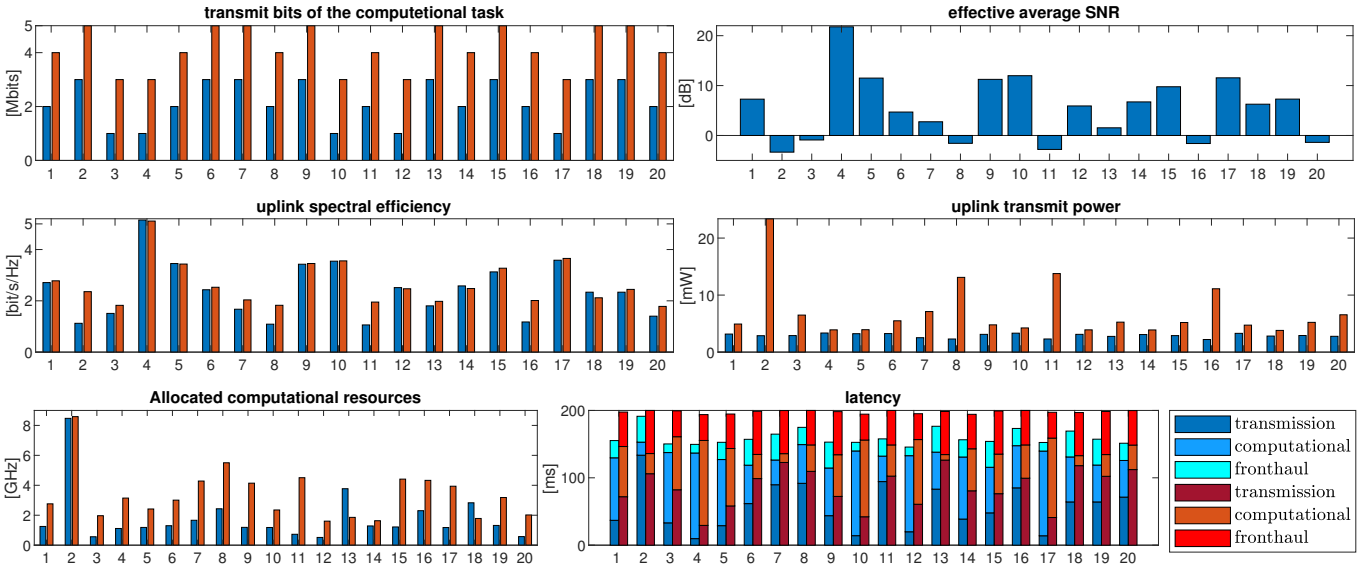

Fig. 10. Simulation results for one network snapshot, averaged over 200 channel realizations. x-axes report the user index. Blueish bars refer to the cell-free setup described in Section V, but with $\{b_k\} \sim \mathcal{U}(1,3)$. Reddish bars refer to a cell-free setup with $\{b_k\} \sim \mathcal{U}(3,5)$ Mbits. Black bars refer to both the setups.

user is not convenient, as each user needs a tailored number of cooperating APs for achieving the cell-free experience. Hence, the FCC AP selection strategy achieves an uplink SE slightly lower than DCC and LSFBS which, in turn, results in a longer effective latency. While, DCC and LSFBS strategies provide equivalent per-user SE and effective latency, although with different levels of uplink transmit power per user. To counterbalance the lack of macro-diversity gain when selecting a very few APs, LSFBS requires the users to use more transmit power than DCC, SE being equal. Conversely, when the DCC selects too many APs, a user needs to use higher uplink powers to guarantee a good receive combining at the furthest APs. Lastly, the AP selection strategy only concerns the communication service, thus it has no impact on the computing resource allocation, not shown herein for brevity.

D. JPCA: A Closer Outlook

In this section, we explore more in detail the effectiveness of the proposed JPCA, evaluating the interplay between uplink power, SE, allocated computational resources, and effective offloading latency. In Fig. 10, we present the simulation results of one network snapshot, where the final values are averaged over 200 channel realizations. For all the users, whose indices appear on the x-axes, we report: computational task size b_k in Mbits; effective average SNR computed as $\sum_{l \in \mathcal{M}_k} \beta_{lk}/\sigma^2$ and converted to dB; uplink SE in bit/s/Hz; uplink transmit power in mW; the allocated computational resources per user, namely $\{\sum_{i=1}^{T_k} f_k(i)\}$ in GHz; and the effective offloading latency consisting of transmission, computational and fronthaul latency. We consider two simulation setups: blueish bars refer to the cell-free setup described in Section V but with $\{b_k\} \sim \mathcal{U}(1,3)$, which we call "loose" setup for brevity; reddish bars refer to a "strict" cell-free setup with higher user computational demands, that is $\{b_k\} \sim \mathcal{U}(3,5)$ Mbits. The latter is of particular interest because highlights how the JPCA operates under stricter constraints. The propagation scenario is in common to both the setups, as we fixed the simulation seed in order to obtain the same channel conditions (black bars). By inspecting Fig. 10, we observe that user 2 is in the adverse conditions of poor SNR and high computational demand. The JPCA naturally needs to allocate more power and computational resources to this user than others in order to reduce its transmission latency (by increasing its SE) and computational latency. As a comparison, user 9 has the same computational demand but better SNR, hence its latency requirements can be more easily fulfilled by solely reducing its transmission latency through allocating slightly more uplink power. In the "strict" setup (reddish bars) the fronthaul latency is longer as it is proportional to the user's task size, and the overall effective latency almost equals the user's requirements of 200 ms. For the "loose" setup (blueish bars) we obtained different but equally interesting results. The fronthaul latency is less pronounced due to the lower user's computational demands. Importantly, we observe a more uniform allocation of the uplink power over the users as compared to the "strict" setup case. The computational latency is dominant over the transmission latency for those users experiencing good channel conditions as high SEs can be achieved with small amount of transmit powers. Conversely, the transmission latency is dominant for those users experiencing bad channel quality. Interestingly, we observe that the effective offloading latency is far below the latency requirement of 200 ms, which results from the choice of achieving a fair balance between transmit power saving and offloading latency by setting $\omega_{\mathrm{p}} = \omega_{\mathrm{se}} = 1$.

VI. Conclusion

The problem of jointly allocating the uplink powers and network computational resources subject to latency constraints in a MEC-enabled CF-mMIMO system was considered in this paper, with the aim of minimizing the total transmit power and simultaneously maximizing the uplink sum SE, and

thereby providing an excellent trade-off between user power consumption and effective offloading latency. For efficiently solving such a non-convex problem, a framework based on alternating optimization and successive convex approximation along with an alternative low-complexity heuristic approach were proposed. A detailed performance comparison between the proposed MEC-enabled CF-mMIMO architecture, its co-located and small-cell counterparts was also provided. Simulation results revealed that CF-mMIMO provides far superior computation offloading efficiency than other network architectures, and constitutes a promising candidate to suitably and flexibly support MEC applications. The proposed joint resource allocation strategy is effective in simultaneously guaranteeing to the users low offloading latency, fairness and significant transmit power saving by distributing the computational workload over multiple MEC servers. Devising a low-complexity JPCA algorithm based on learning [49], [50] and/or non-convex optimization (e.g., *differential evolution* [51]) is an appealing research direction for future works, as well as extending this study to a partial computational offloading model.

REFERENCES

[1] G. Interdonato and S. Buzzi, "The promising marriage of mobile edge computing and cell-free massive MIMO," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, May 2022, pp. 243–248.

[2] K. Yang, S. Ou, and H.-H. Chen, "On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications," *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 56–63, Jan. 2008.

[3] L. Y. Kumar K., Liu J. and B. B., "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, p. 129–140, Apr. 2013.

[4] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[5] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart. 2017.

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

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[8] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. and Netw.*, vol. 2019, no. 1, p. 197, 2019.

[9] Özlem Tugfe Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends® in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.

[10] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge Univ. Press, 2016.

[11] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.

[12] G. Interdonato, "Cell-free massive MIMO: Scalability, signal processing and power control," Ph.D. dissertation, Linköping University Electronic Press, 2020.

[13] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1250–1264, Feb. 2020.

[14] G. Interdonato, P. Frenger, and E. G. Larsson, "Scalability aspects of cell-free massive MIMO," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, May 2019, pp. 1–6.

[15] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[16] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[17] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[18] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[19] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[20] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[21] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[22] L. Zhang, Y. Sun, Z. Chen, and S. Roy, "Communications-caching-computing resource allocation for bidirectional data computation in mobile edge networks," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1496–1509, 2020.

[23] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[24] Q. Li, J. Lei, and J. Lin, "Min-Max latency optimization for multiuser computation offloading in fog-radio access networks," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2018, pp. 3754–3758.

[25] T. T. Nguyen, L. B. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. on Services Computing*, vol. 14, no. 6, pp. 2011–2025, Nov 2021.

[26] S. Sardellitti, M. Merluzzi, and S. Barbarossa, "Optimal association of mobile users to multi-access edge computing resources," in *Proc. IEEE Int. Conf. on Commun. Workshops (ICC Wkshps)*, May 2018, pp. 1–6.

[27] C. Pradhan, A. Li, C. She, Y. Li, and B. Vucetic, "Computation offloading for IoT in C-RAN: Optimization and deep learning," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4565–4579, Jul. 2020.

[28] Y. Hao, Q. Ni, H. Li, and S. Hou, "Energy-efficient multi-user mobile-edge computation offloading in massive MIMO enabled HetNets," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, May 2019, pp. 1–6.

[29] M. Zeng, W. Hao, O. A. Dobre, Z. Ding, and H. V. Poor, "Massive MIMO-assisted mobile edge computing: Exciting possibilities for computation offloading," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 31–38, Jun. 2020.

[30] M. Zeng, W. Hao, O. A. Dobre, and H. V. Poor, "Delay minimization for massive MIMO assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6788–6792, Jun. 2020.

[31] Y. Zhao, X. Xu, Y. Su, L. Huang, X. Du, and N. Guizani, "Multi-user MAC protocol for WLANs in mmWave massive MIMO systems with mobile edge computing," *IEEE Access*, vol. 7, pp. 181 242–181 256, Nov. 2019.

[32] M. Merluzzi, P. D. Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Trans. Signal and Inf. Process. over Netw.*, vol. 6, pp. 342–356, 2020.

[33] G. Femenias and F. Riera-Palou, "Mobile edge computing aided cell-free massive MIMO networks," *IEEE Trans. on Mobile Computing*, pp. 1–16, Dec 2022.

[34] S. Mukherjee and J. Lee, "Edge computing-enabled cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2884–2899, Apr. 2020.

[35] L. Zadeh, "Optimality and non-scalar-valued performance criteria," *IEEE Trans. Autom. Control*, vol. 8, no. 1, pp. 59–60, Jan. 1963.

[36] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Struct Multidisc Optim*, vol. 26, no. 6, pp. 369–395, 2004.

[37] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer Berlin, 2008.

[38] E. Björnson, E. A. Jorswieck, M. Debbah, and B. Ottersten, "Multiobjective signal processing optimization: The way to balance conflicting

metrics in 5G systems," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 14–23, Nov. 2014.
[39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
[40] S. Martello and P. Toth, *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.
[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, Aug. 1978.
[42] S. Ulukus and R. Yates, "Stochastic power control for cellular radio systems," *IEEE Trans. Commun.*, vol. 46, no. 6, pp. 784–798, Jun. 1998.
[43] R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.
[44] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 726–738, Sep. 2019.
[45] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. on Mobile Computing*, vol. 18, no. 4, pp. 771–786, Apr. 2019.
[46] 3GPP, *Further advancements for E-UTRA physical layer aspects (Release 9)*. 3GPP TS 36.814, Mar. 2017.
[47] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, pp. 127–190, 1999.
[48] H. Q. Ngo, L. N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Trans. Green Commun. and Netw.*, vol. 2, no. 1, pp. 25–39, Mar. 2018.
[49] Y. Guo, R. Zhao, S. Lai, L. Fan, X. Lei, and G. K. Karagiannidis, "Distributed machine learning for multiuser mobile edge computing systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 460–473, Apr. 2022.
[50] S. Tang, L. Chen, K. He, J. Xia, L. Fan, and A. Nallanathan, "Computational intelligence and deep learning for next-generation edge-enabled industrial IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2881–2893, Sep. 2023.
[51] Z. Yu and G. Fan, "Joint differential evolution and successive convex approximation in UAV-enabled mobile edge computing," *IEEE Access*, vol. 10, pp. 57 413–57 426, May 2022.

Giovanni Interdonato (Member, IEEE) received the M.Sc. degree in computer and telecommunication systems engineering from the Mediterranea University of Reggio Calabria, Italy, in 2015, and the Ph.D. degree in electrical engineering with a specialization in communication systems from Linköping University, Sweden, in 2020. From October 2015 to October 2018, he was a researcher at the Radio Network Department of Ericsson Research, Linköping, and a Marie Sklodowska-Curie research fellow of the EU-H2020 ITN project "5Gwireless".

Dr. Interdonato is currently an assistant professor at the Department of Electrical and Information Engineering (DIEI), University of Cassino and Southern Lazio, Italy. His main research interests lie in the field of wireless communications and signal processing, with focus on beyond-5G physical layer technologies, radio resource management and communication protocols. He is the co-inventor of about twenty granted patent applications on massive MIMO and cell-free massive MIMO systems. Dr. Interdonato serves as an associate editor for IEEE COMMUNICATIONS LETTERS and IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY.

He has been awarded IEEE TRANSACTIONS ON COMMUNICATIONS Exemplary Reviewer for 2021 and IEEE COMMUNICATIONS LETTERS Exemplary Editor for 2022, and he was a recipient of a research grant from the Ericsson Research Foundation in 2019.

Stefano Buzzi (Senior Member, IEEE) joined the University of Cassino and Lazio Meridionale, Italy in 2000, first as an Assistant Professor, then as an Associate Professor (since 2002) and, finally, since 2018, as a Full Professor. He received the M.Sc. degree (summa cum laude) in Electronic Engineering in 1994, and the Ph.D. degree in Electrical and Computer Engineering in 1999, both from the University of Naples "Federico II". He has had short-term research appointments at Princeton University, Princeton (NJ), USA in 1999, 2000, 2001 and 2006. He is a former Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and of the IEEE COMMUNICATIONS LETTERS, has been the guest editor of four IEEE JSAC special issues, and from 2014 to 2020 he has been an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Currently, Prof. Buzzi is Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He also serves regularly as TPC member of several international conferences. Dr. Buzzi's research interests are in the broad field of communications and signal processing, with emphasis on wireless communications and beyond-5G systems. He is currently the General Coordinator of the EU-funded Innovative Training Network project METAWIRELESS, on the application of metasurfaces to wireless communications, and of the EU-funded Doctoral Network ISLANDS, on Integrated Sensing and Communications for the Vehicular Environment. He has co-authored about 180 technical peer-reviewed journal and conference papers, and, among these, the highly cited paper "What will 5G be?", IEEE JSAC, June 2014.