

Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing

Stefania Sardellitti, Gesualdo Scutari, and Sergio Barbarossa

Abstract—Migrating computational intensive tasks from mobile devices to more resourceful cloud servers is a promising technique to increase the computational capacity of mobile devices while saving their battery energy. In this paper, we consider an MIMO multicell system where multiple mobile users (MUs) ask for computation offloading to a common cloud server. We formulate the offloading problem as the *joint* optimization of the radio resources—the transmit precoding matrices of the MUs—and the computational resources—the CPU cycles/second assigned by the cloud to each MU—in order to minimize the overall users' energy consumption, while meeting latency constraints. The resulting optimization problem is nonconvex (in the objective function and constraints). Nevertheless, in the single-user case, we are able to compute the global optimal solution in closed form. In the more challenging multiuser scenario, we propose an iterative algorithm, based on a novel successive convex approximation technique, converging to a local optimal solution of the original nonconvex problem. We then show that the proposed algorithmic framework naturally leads to a distributed and parallel implementation across the radio access points, requiring only a limited coordination/signaling with the cloud. Numerical results show that the proposed schemes outperform disjoint optimization algorithms.

Index Terms—Mobile cloud computing, computation offloading, energy minimization, resources allocation, small cells.

I. INTRODUCTION

MOBILE terminals, such as smartphones, tablets and notebooks, are increasingly penetrating into our everyday lives as convenient tools for communication, entertainment, business, social networking, news, etc. Current predictions foresee a doubling of mobile data traffic every year. However such a growth in mobile wireless traffic is not matched with an equally fast improvement on mobile handsets' batteries, as testified in [3]. The limited battery lifetime is then going to represent the stumbling block to the deployment of computation-intensive applications for mobile devices. At the same time,

Manuscript received December 27, 2014; revised May 09, 2015; accepted June 15, 2015. Date of publication June 22, 2015; date of current version August 31, 2015. The work of S. Sardellitti and S. Barbarossa was supported by the European Community 7th Framework Programme Project ICT-TROPIC, under Grant 318784. The work of G. Scutari was supported by the USA NSF under Grant CMS 1218717 and Grant CAREER Award no. 1254739. Part of this work was presented at IEEE SPAWC 2014 [1] and at IEEE CloudNet 2014 [2]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ying-Chang Liang.

S. Sardellitti and S. Barbarossa are with the Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome 00184, Italy (e-mail: stefania.sardellitti@uniroma1.it; sergio.barbarossa@uniroma1.it).

G. Scutari is with the Department of Industrial Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: gscutari@purdue.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSIPN.2015.2448520

in the Internet-of-Things (IoT) paradigm, a myriad of heterogeneous devices, with a wide range of computational capabilities, are going to be interconnected. For many of them, the local computation resources are insufficient to run sophisticated applications. In all these cases, a possible strategy to overcome the above energy/computation bottleneck consists in enabling resource-constrained mobile devices to offload their most energy-consuming tasks to nearby more resourceful servers. This strategy has a long history and is reported in the literature under different names, such as *cyber foraging* [4], or *computation offloading* [5]. In recent years, cloud computing (CC) has provided a strong impulse to computation offloading through virtualization, which decouples the application environment from the underlying hardware resources and thus enables an efficient usage of available computing resources. In particular, Mobile Cloud Computing (MCC) [6] makes possible for mobile users to access cloud resources, such as infrastructures, platforms, and software, on-demand. Several works addressed mobile computation offloading, such as [7]–[16]. Recent surveys are [6], [17], and [18]. Some works addressed the problem of program partitioning and offloading the most demanding program tasks, as e.g. in [7]–[10]. Specific examples of mobile computation offloading techniques are: *MAUI* [19], *ThinkAir* [20], and *Phone2Cloud* [21]. The trade-off between the energy spent for computation and communication was studied in [12]–[14], [22]. A dynamic formulation of computation offloading was proposed in [15]. These works optimized offloading strategies, assuming a given radio access, and concentrated on single-user scenarios. In [23], it was proposed a *joint* optimization of radio and computational resources, for the single user case. The joint optimization was then extended to the multiuser case in [24]; see also [25] for a recent survey on joint optimization for computation offloading in a 5G perspective. In the above works, the allocation of radio and computing resources is managed at the cloud in a centralized manner. A decentralized solution, based on a game-theoretic formulation of the problem, was recently proposed in [26], [11].

In current cellular networks, the major obstacles limiting an effective deployment of MCC strategies are: i) the energy spent by mobile terminals, especially cell edge users, for radio access; and ii) the latency experienced in reaching the (remote) cloud server through a wide area network (WAN). Indeed, in macro-cellular systems, the transmit power necessary for cell edge users to access a remote base station might null all potential benefits coming from offloading. Moreover, in many real-time mobile applications (e.g., online games, speech recognition, Facetime) the user Quality of Experience (QoE) is strongly

affected by the system response time. Since controlling latency over a WAN might be very difficult, in many circumstances the QoE associated to MCC could be poor. A possible way to tackle these challenges is to bring *both* radio access and computational resources closer to MUs. This idea was suggested in [17], [27], with the introduction of *cloudlets*, providing proximity radio access to fixed servers through Wi-Fi. However, the lack of available fixed servers could limit the applicability of cloudlets. The European project TROPIC [28] suggested to endow small cell LTE base stations with, albeit limited, cloud functionalities. In this way, one can exploit the potential dense deployment of small cell base stations to facilitate proximity access to computing resources and have advantages over Wi-Fi access in terms of Quality-of-Service guarantee and a single technology system (no need for the MUs to switch between cellular and Wi-Fi standards). Very recently, the European Telecommunications Standards Institute (ETSI) launched a new standardization group on the so called *Mobile-Edge Computing* (MEC), whose aim is to provide information technology and cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscribers in order to offer a service environment characterized by proximity, low latency, and high rate access [29].

Merging MEC with the dense deployment of (small cell) Base Stations (BSs), as foreseen in the 5G standardization roadmap, makes possible a real proximity, ultra-low latency access to cloud functionalities. However, in a dense deployment scenario, offloading becomes much more complicated because of intercell interference. The goal of this paper is to tackle this challenge by proposing a *joint* optimization of radio and computational resources for computation offloading in a dense deployment scenario, *in the presence of intercell interference*. More specifically, the offloading problem is formulated as the minimization of the overall energy consumption, at the mobile terminals' side, under transmit power and latency constraints. The optimization variables are the mobile radio resources—the covariance matrices of the mobile MIMO transmitters—and the computational resources—the CPU cycles/second assigned by the cloud to each MU. The latency constraint is what couples computation and communication optimization variables. This problem is much more challenging than the (special) cases studied in the literature because of the presence of intercell interference, which introduces a coupling among the precoding matrices of all MUs, while making the optimization problem nonconvex. In this context, the main contributions of the paper are the following: i) in the single-user case, we first establish the equivalence between the original nonconvex problem and a *convex one*, and then derive the *closed form* of its (global optimal) solution; ii) in the multi-cell case, hinging on recent Successive Convex Approximation (SCA) techniques [30], [31], we devise an iterative algorithm that is proved to converge to local optimal solutions of the original nonconvex problem; and iii) we propose alternative decomposition algorithms to solve the original centralized problem in a distributed form, requiring limited signaling among BSs and cloud; the algorithms differ for convergence speed, computational effort, communication overhead, and a-priori knowledge of system parameters, but they are all convergent under a unified set

of conditions. Numerical results show that all the proposed schemes converge quite fast to “good” solutions, yielding a significant energy saving with respect to disjoint optimization procedures, for applications requiring intensive computations and limited exchange of data to enable offloading.

This paper is organized as follows. In Section II we introduce the system model; Section III formulates the offloading optimization problem in the single user case, whereas Section IV focuses on the multi-cell scenario along with the proposed SCA algorithmic framework. The decentralized implementation is discussed in Section V.

II. COMPUTATION OFFLOADING

Let us consider a network composed of N_c cells; in each cell $n = 1, \dots, N_c$, there is one Small Cell enhanced Node B (SCeNB in LTE terminology) serving K_n MUs. We denote by i_n the i -th user in the cell n , and by $\mathcal{I} \triangleq \{i_n : i = 1, \dots, K_n, n = 1, \dots, N_c\}$ the set of all the users. Each MU i_n and SCeNB n are equipped with $n_{T_{i_n}}$ transmit and n_{R_n} receive antennas, respectively. The SCeNB's are all connected to a common cloud provider, able to serve multiple users concurrently. We assume that MUs in the same cell transmit over orthogonal channels, whereas users of different cells may interfere against each other.

MUs can run their application locally or remotely depending on the available energy and computational needs. For example, users with low battery levels may be more in need of computation offloading than users with full battery. Conversely, mobile devices with limited computational capabilities, like smart sensors in an Internet of Things (IoT) scenario, might be more willing to offload computations. Different priorities could also be associated to different users' requests and different applications. Based on these considerations, we partition the MUs in two subsets: the subset \mathcal{I}_o of users asking for computation offloading and the subset \mathcal{I}_{no} of users not involved in any computation offloading, but accessing the radio channel for communication purposes. Note that both kind of users compete over the radio resources, but only the users in \mathcal{I}_o compete for the computational resources. The overall set of users is then $\mathcal{I} = \mathcal{I}_o \cup \mathcal{I}_{no}$. In this work, we assume that the partition \mathcal{I}_o and \mathcal{I}_{no} is given; for instance, the decision on where to run the applications (or part of them) is taken by the cloud manager, after collecting all users' requests and exchanging information with the associated base stations about channel state and interference level.

In this scenario, all MUs aim to minimize their energy consumption. MUs $i_n \in \mathcal{I}_o$ wish to run an application within a given maximum time T_{i_n} . To offload their computations, the users need to send all the information necessary to transfer the execution of the program (or part of it) to the server. The program can be split in modules. Each module to be executed is characterized by: the number w_{i_n} of CPU cycles necessary to run the module itself; the number b_{i_n} of input bits necessary to transfer the program execution from local to remote sides; and the number $b_{i_n}^o$ of output bits encoding the result of the computation, to be sent back from remote to local sides. In case of offloading, the latency incorporates the time to transmit the

input bits to the server necessary to enable the execution transfer, the time necessary for the server to execute the instructions, and the time to send the result back to the MU. More specifically, the overall latency experienced by each MU $i_n \in \mathcal{I}_o$ can be written as

$$\Delta_{i_n} = \Delta_{i_n}^t + \Delta_{i_n}^{\text{exe}} + \Delta_{i_n}^{\text{tx/rx}} \quad (1)$$

where $\Delta_{i_n}^t$ is the time necessary for the MU i_n to transfer the input bits b_{i_n} to its SCeNB; $\Delta_{i_n}^{\text{exe}}$ is the time for the server to execute w_{i_n} CPU cycles; and $\Delta_{i_n}^{\text{tx/rx}}$ is the time necessary for SCeNB n to send the b_{i_n} bits to the cloud through the backhaul link plus the time necessary to send back the result (encoded in $b_{i_n}^o$ bits) from the server to MU i_n . We derive next an explicit expression of $\Delta_{i_n}^t$ and $\Delta_{i_n}^{\text{exe}}$ as a function of the radio and computational resources.

A. Radio Resources

The optimization variables at radio level are the users' transmit covariance matrices $\mathbf{Q} \triangleq (\mathbf{Q}_{i_n})_{i_n \in \mathcal{I}}$, subject to standard power budget constraints, i.e.,

$$\mathcal{Q}_{i_n} \triangleq \{ \mathbf{Q}_{i_n} \in \mathbb{C}^{n_{T_{i_n}} \times n_{T_{i_n}}} : \mathbf{Q}_{i_n} \succeq \mathbf{0}, \text{tr}(\mathbf{Q}_{i_n}) \leq P_{i_n} \}, \quad (2)$$

where P_{i_n} is the average transmit power of user i_n . We will denote by \mathcal{Q} the joint set $\mathcal{Q} \triangleq \prod_{i_n \in \mathcal{I}} \mathcal{Q}_{i_n}$.

For any given profile $\mathbf{Q} \triangleq (\mathbf{Q}_{i_n})_{i_n \in \mathcal{I}}$, the maximum achievable rate of MU i_n is:

$$r_{i_n}(\mathbf{Q}) = \log_2 \det(\mathbf{I} + \mathbf{H}_{i_n n}^H \mathbf{R}_n (\mathbf{Q}_{-n})^{-1} \mathbf{H}_{i_n n} \mathbf{Q}_{i_n}) \quad (3)$$

where

$$\mathbf{R}_n(\mathbf{Q}_{-n}) \triangleq \mathbf{R}_w + \sum_{j_m \in \mathcal{I}, m \neq n} \mathbf{H}_{j_m n} \mathbf{Q}_{j_m} \mathbf{H}_{j_m n}^H, \quad (4)$$

is the covariance matrix of the noise $\mathbf{R}_w \triangleq \sigma_w^2 \mathbf{I}$ (assumed to be diagonal w.l.o.g, otherwise one can always pre-whitening the channel matrices) plus the inter-cell interference at the SCeNB n (treated as additive noise); $\mathbf{H}_{i_n n}$ is the channel matrix of the uplink i in the cell n , whereas $\mathbf{H}_{j_m n}$ is the cross-channel matrix between the interferer MU j in the cell m and the SCeNB of cell n ; and $\mathbf{Q}_{-n} \triangleq ((\mathbf{Q}_{j_m})_{j=1}^{K_m})_{m \neq n=1}^{N_c}$ denotes the tuple of the covariance matrices of all users interfering with the SCeNB n .

Given each $r_{i_n}(\mathbf{Q})$, the time $\Delta_{i_n}^t$ necessary for user i in cell n to transmit the input bits b_{i_n} of duration $T_{b_{i_n}}$ to its SCeNB can be written as

$$\Delta_{i_n}^t = \Delta_{i_n}^t(\mathbf{Q}) = \frac{c_{i_n}}{r_{i_n}(\mathbf{Q})} \quad (5)$$

where $c_{i_n} = b_{i_n} T_{b_{i_n}}$. The energy consumption of each MU due to the transmission is then

$$E_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-n}) = \text{tr}(\mathbf{Q}_{i_n}) \cdot \Delta_{i_n}^t(\mathbf{Q}), \quad (6)$$

which depends also on the covariance matrices \mathbf{Q}_{-n} of the users in the other cells, because of the intercell interference. Note that, for the MUs $i_n \in \mathcal{I}_o$, E_{i_n} represents the energy consumption due to offloading, with b_{i_n} being the number of transmitted bits necessary to transfer the program execution to the server.

B. Computational Resources

The cloud provider is able to serve multiple users concurrently. The computational resources made available by the cloud and to be shared among the users are quantified by the computational rate f_T , expressed in terms of number of CPU cycles/second. This quantity typically takes into account the number of machines composing the cloud, the computational capability of each machine, and the tasks running in the background. We denote by $f_{i_n} \geq 0$ the fraction of f_T to be assigned to each user $i_n \in \mathcal{I}_o$. Note that no resources are assigned to the users $i_n \in \mathcal{I}_{no}$. The rates f_{i_n} , with $i_n \in \mathcal{I}_o$, are subject to the computational budget constraint $\sum_{i_n \in \mathcal{I}_o} f_{i_n} \leq f_T$. Given the resource assignment f_{i_n} , the time $\Delta_{i_n}^{\text{exe}}$ needed to run w_{i_n} CPU cycles of user i_n 's instructions remotely is then

$$\Delta_{i_n}^{\text{exe}} = \Delta_{i_n}^{\text{exe}}(f_{i_n}) = w_{i_n} / f_{i_n}, \quad i_n \in \mathcal{I}_o. \quad (7)$$

The expression of the overall latency Δ_{i_n} [cf. (1), (5), and (7)] clearly shows the interplay between radio access and computational aspects, which motivates a *joint* optimization of the radio resources, the transmit covariance matrices $\mathbf{Q} \triangleq (\mathbf{Q}_{i_n})_{i_n \in \mathcal{I}}$ of the MUs, and the computational resources, the computational rate allocation $\mathbf{f} \triangleq (f_{i_n})_{i_n \in \mathcal{I}_o}$.

We are now ready to formulate the offloading problem rigorously. We focus first on the single-user scenario (cf. Sec. III); this will allow us to shed light on the special structure of the optimal solution. Then, we will extend the formulation to the multiple-cells case (cf. Sec. IV).

III. THE SINGLE-USER CASE

In the single-user case, there is only one active MU having access to the cloud. In such interference-free scenario, the maximum achievable rate on the MU and energy consumption due to offloading reduce to [cf. (3) and (6)]

$$r(\mathbf{Q}) = \log_2 \det(\mathbf{I} + \mathbf{H} \mathbf{Q} \mathbf{H}^H \mathbf{R}_w^{-1}) \quad (8)$$

and

$$E(\mathbf{Q}) = c \cdot \frac{\text{tr}(\mathbf{Q})}{r(\mathbf{Q})}, \quad (9)$$

respectively, with $c = b \cdot T_b$ (for notational simplicity, we omit the user index; \mathbf{Q} denotes now the covariance matrix of the MU).

We formulate the offloading problem as the minimization of the energy spent by the MU to run its application remotely, subject to latency and transmit power constraints, as follows:

$$\begin{aligned} \min_{\mathbf{Q}, f} \quad & E(\mathbf{Q}) \\ \text{s.t.} \quad & \left. \begin{aligned} \text{a) } \frac{c}{r(\mathbf{Q})} + \frac{w}{f} - \tilde{T} &\leq 0 \\ \text{b) } 0 &\leq f \leq f_T \\ \text{c) } \text{tr}(\mathbf{Q}) &\leq P_T, \quad \mathbf{Q} \succeq \mathbf{0} \end{aligned} \right\} \triangleq \mathcal{X}_s \end{aligned} \quad (\mathcal{P}_s)$$

where a) reflects the user latency constraint $\Delta \leq T$ [cf. (1)], with \tilde{T} capturing all the constant terms, i.e., $\tilde{T} \triangleq T - \Delta^{\text{tx/rx}}$; b) imposes a limit on the cloud computational resources made available to the users; and c) is the power budget constraint on the radio resources.

A. Feasibility

Depending on the system parameters, problem \mathcal{P}_s may be feasible or not. In the latter case, offloading is not possible and thus the MU will perform its computations locally. It is not difficult to prove that the following condition is *necessary* and *sufficient* for \mathcal{X}_s to be nonempty and thus for offloading to be feasible:

$$\frac{c}{r^{\max}} + \frac{w}{f_T} - \tilde{T} \leq 0 \quad (10)$$

where r^{\max} is the capacity of the MIMO link of the MU, i.e.,

$$r^{\max} = \max_{\mathbf{Q} \succeq \mathbf{0} : \text{tr}(\mathbf{Q}) \leq P_T} r(\mathbf{Q}). \quad (11)$$

The unique (closed-form) solution of (11) is the well-known MIMO water-filling. Note that condition (10) has an interesting physical interpretation: offloading is feasible if and only if $\tilde{T} > 0$, i.e., the delay on the wired network $\Delta^{\text{tx/rx}}$ is less than the maximum tolerable delay, and the overall latency constraint is met (at least) when the wireless and computational resources are fully utilized (i.e., $r(\mathbf{Q}) = r^{\max}$, and $f = f_T$). It is not difficult to check that this worst-case scenario is in fact achieved when (10) is satisfied with equality; in such a case, the (globally optimal) solution (\mathbf{Q}^*, f^*) to \mathcal{P}_s is trivially given by $(\mathbf{Q}^*, f^*) = (\mathbf{Q}^{\text{wf}}, f_T)$, where \mathbf{Q}^{wf} is the waterfilling solution to (11). Therefore in the following we will focus w.l.o.g. on \mathcal{P}_s under the tacit assumption of *strict* feasibility [i.e., the inequality in (10) is tight].

B. Solution Analysis

Problem \mathcal{P}_s is nonconvex due to the non-convexity of the energy function. A major contribution of this section is to i) cast \mathcal{P}_s into a convex equivalent problem, and ii) compute its global optimal solution (and thus optimal also to \mathcal{P}_s) in closed form. To do so, we introduce first some preliminary definitions.

Let \mathcal{Q}_s be the following auxiliary *convex* problem

$$\left. \begin{array}{l} \min_{\mathbf{Q}, f} \text{tr}(\mathbf{Q}) \\ \text{s.t. a) } \frac{c}{r(\mathbf{Q})} + \frac{w}{f} - \tilde{T} \leq 0 \\ \text{b) } 0 \leq f \leq f_T \\ \text{c) } \text{tr}(\mathbf{Q}) \leq P_T, \quad \mathbf{Q} \succeq \mathbf{0} \end{array} \right\} = \mathcal{X}_s \quad (\mathcal{Q}_s)$$

which corresponds to minimizing the transmit power of the MU under the same latency and power constraints as in \mathcal{P}_s . Also, let $\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H} = \mathbf{U} \mathbf{D} \mathbf{U}^H$ be the (reduced) eigenvalue decomposition of $\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H}$, with $r \triangleq \text{rank}(\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H}) = \text{rank}(\mathbf{H})$, where $\mathbf{U} \in \mathbb{C}^{n_T \times r}$ is the (semi-)unitary matrix whose columns are the eigenvectors associated with the r positive eigenvalues of $\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H}$, and $\mathbb{R}_{++}^{r \times r} \ni \mathbf{D} \triangleq \text{diag}\{(d_i)_{i=1}^r\}$ is the diagonal matrix, whose diagonal entries are the eigenvalues arranged in decreasing order. We are now ready to establish the connection between \mathcal{P}_s and \mathcal{Q}_s .

Theorem 1: Given problems \mathcal{P}_s and \mathcal{Q}_s under strict feasibility, the following hold.

(a) \mathcal{P}_s and \mathcal{Q}_s are equivalent;

(b) \mathcal{Q}_s (and \mathcal{P}_s) has a unique solution (\mathbf{Q}^*, f^*) , given by

$$f^* = f_T, \quad \text{and} \quad \mathbf{Q}^* = \mathbf{U} (\alpha \mathbf{I} - \mathbf{D}^{-1})^+ \mathbf{U}^H, \quad (12)$$

where $\alpha > 0$ must be chosen so that the latency constraint (a) in \mathcal{X}_s is satisfied with equality at (\mathbf{Q}^*, f^*) , and $(\mathbf{x})^+ \triangleq \max(\mathbf{0}, \mathbf{x})$ (intended component-wise).

The water-level $\alpha > 0$ can be efficiently computed using the hypothesis-testing-based algorithm described in Algorithm 1.

Proof: See Appendix -A. ■

Algorithm 1. Efficient computation of α in (12)

Data: $(d_i)_{i=1}^r > 0$ (arranged in decreasing order), $r = \text{rank}(\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H})$, and $L \triangleq \tilde{T} - w/f_T > 0$;

(S.0): Set $r_e = r$;

(S.1): Repeat

$$\begin{array}{l} \text{(a): Set } \alpha = 2^{\frac{c}{r_e L} - \frac{1}{r_e} \sum_{i=1}^{r_e} \log_2(d_i)}; \\ \text{(b): If } p_i \triangleq (\alpha - 1/d_i) \geq 0, \forall i = 1, \dots, r_e, \\ \quad \text{and } \sum_{i=1}^{r_e} p_i \leq P_T, \\ \quad \text{then STOP;} \\ \quad \text{else } r_e = r_e - 1; \\ \text{until } r_e \geq 1. \end{array}$$

Theorem 1 states that, in the single-user case, the latency constraint has to be met with equality and then the offloading strategy minimizing energy consumption coincides with the one minimizing the transmit power. Note that \mathbf{Q}^* has a water-filling-like structure: the optimal transmit “directions” are aligned with the eigenvectors \mathbf{U} of the equivalent channel $\mathbf{H}^H \mathbf{R}_w^{-1} \mathbf{H}$. However, differently from the classical waterfilling solution \mathbf{Q}^{wf} [cf. (11)], the waterlevel α is now computed to meet the latency constraint with equality. This means that a transmit strategy using the full power P_T (like \mathbf{Q}^{wf}) is no longer optimal. The only case in which $\mathbf{Q}^* \equiv \mathbf{Q}^{\text{wf}}$ is the case where the feasibility condition (10) is satisfied with equality. Note that the water-level α depends now on *both* communication and computational parameters (maximum tolerable delay, size of the program state, CPU cycle budget, etc.).

IV. COMPUTATION OFFLOADING OVER MULTIPLE-CELLS

In this section we consider the more general multi-cell scenario described in Sec. II. The weighted sum of the overall energies spent by the MUs offloading computations and the MUs simply accessing the network for communication, is now given by

$$E(\mathbf{Q}) \triangleq \sum_{i_n \in \mathcal{I}} a_{i_n} E_{i_n}(\mathbf{Q}), \quad (13)$$

where $E_{i_n}(\mathbf{Q})$ is defined in (6), and $(a_{i_n})_{i_n \in \mathcal{I}}$ is a set of fixed positive coefficients, chosen to assign (possibly) different priorities among the users’ requests. For instance, higher coefficients a_{i_n} may be assigned to users with lower battery levels asking for the execution of computationally intensive applications having a high priority level. As a case-study, in the following,

we will focus on the minimization of the weighted sum-energy $E(\mathbf{Q})$, but the proposed algorithmic framework can be readily applied to other functions of the users' energies $E_{i_n}(\mathbf{Q})$, such as the geometric mean, the alpha-fairness, etc.

Each MU i_n is subject to the power budget constraint (2) and, in case of offloading, to an overall latency given by

$$g_{i_n}(\mathbf{Q}, f_{i_n}) \triangleq \frac{c_{i_n}}{r_{i_n}(\mathbf{Q})} + \frac{w_{i_n}}{f_{i_n}} - \tilde{T}_{i_n} \leq 0, \quad i_n \in \mathcal{I}_0. \quad (14)$$

MUs $i_n \in \mathcal{I}_{\text{no}}$ that do not perform any offloading are instead subject to the rate constraint

$$r_{i_n}(\mathbf{Q}) \geq R_{i_n}^{\min}, \quad \forall i_n \in \mathcal{I}_{\text{no}} \quad (15)$$

where $R_{i_n}^{\min}$ is the minimum rate required to meet the desired QoS. The offloading problem in the multi-cell scenario is then formulated as follows:

$$\begin{aligned} \min_{\mathbf{Q}, \mathbf{f}} \quad & E(\mathbf{Q}) \\ \text{s.t.} \quad & \left. \begin{aligned} \text{a) } g_{i_n}(\mathbf{Q}, f_{i_n}) &\leq 0, \quad \forall i_n \in \mathcal{I}_0, \\ \text{b) } r_{i_n}(\mathbf{Q}) &\geq R_{i_n}^{\min}, \quad \forall i_n \in \mathcal{I}_{\text{no}}, \\ \text{c) } \sum_{i_n \in \mathcal{I}_0} f_{i_n} &\leq f_T, \quad f_{i_n} \geq 0, \quad \forall i_n \in \mathcal{I}_0, \\ \text{d) } \mathbf{Q}_{i_n} &\in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I}, \end{aligned} \right\} \triangleq \mathcal{X} \quad (\mathcal{P}) \end{aligned}$$

where a) represent the latency constraints $\Delta_{i_n} \leq T_{i_n}$ with $\tilde{T}_{i_n} \triangleq T_{i_n} - \Delta_{i_n}^{\text{tx/rx}}$ of the users requesting computation offloading; b) are the rate constraints associated to non-offloading users; and c) is due to the limited cloud computational resources.

A. Feasibility

In general, depending on the users' requests, channel status and cloud computational capability, not all requests can be accommodated. The following conditions are sufficient for \mathcal{X} to be nonempty and thus for offloading to be feasible: $\tilde{T}_{i_n} > 0$ for all $i_n \in \mathcal{I}_0$, and there exists a $\bar{\mathbf{Q}} \triangleq (\bar{\mathbf{Q}}_{i_n})_{i_n \in \mathcal{I}} \in \mathcal{Q}$ such that

$$\begin{aligned} \tilde{T}_{i_n} &> \frac{c_{i_n}}{r_{i_n}(\bar{\mathbf{Q}})}, \quad \forall i_n \in \mathcal{I}_0, \quad r_{i_n}(\bar{\mathbf{Q}}) \geq R_{i_n}^{\min}, \quad \forall i_n \in \mathcal{I}_{\text{no}} \\ \text{and} \quad &\sum_{i_n \in \mathcal{I}_0} \frac{w_{i_n}}{\tilde{T}_{i_n} - \frac{c_{i_n}}{r_{i_n}(\bar{\mathbf{Q}})}} \leq f_T. \end{aligned} \quad (16)$$

Hereafter, we will assume that problem \mathcal{P} is feasible. For example, conditions (16) can be enforced by a proper admission control strategy.

Problem \mathcal{P} is nonconvex, due to the nonconvexity of the objective function and the constraints a), b). In what follows we exploit the structure of \mathcal{P} and, building on some recent Successive Convex Approximation (SCA) techniques proposed in [30], [31], we develop a fairly general class of efficient approximation algorithms, all converging to a local optimal solution of \mathcal{P} . The numerical results will show that the proposed

algorithms converge in a few iterations to “good” locally optimal solutions of \mathcal{P} (that turn out to be quite insensitive to the initialization). The main algorithmic framework, along with its convergence properties, is introduced in Sec. IV-B; alternative distributed implementations are studied in Sec. V.

B. Algorithmic Design

To solve the non-convex problem \mathcal{P} efficiently, we develop a SCA-based method where \mathcal{P} is replaced by a sequence of *strongly convex* problems. At the basis of the proposed technique, there is a suitable *convex* approximation of the nonconvex objective function $E(\mathbf{Q})$ and the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$, $r_{i_n}(\mathbf{Q}) \geq R_{i_n}^{\min}$ around the iterates of the algorithm, which are preliminarily discussed next.

1) *Approximant of $E(\mathbf{Q})$* : Let $\mathbf{Z} \triangleq (\mathbf{Q}, \mathbf{f})$ and $\mathbf{Z}^\nu \triangleq (\mathbf{Q}^\nu, \mathbf{f}^\nu)$, with ν being the current iterate index, $\mathbf{f} \triangleq (f_{i_n})_{i_n \in \mathcal{I}_0}$, and $\mathbf{f}^\nu \triangleq (f_{i_n}^\nu)_{i_n \in \mathcal{I}_0}$. Let $\mathcal{E} \supseteq \mathcal{X}$ be any closed convex set containing \mathcal{X} such that $E(\mathbf{Q})$ is well-defined on it. Note that such a set exists. For instance, noting that at every (feasible) $(\mathbf{Q}, \mathbf{f}) \in \mathcal{X}$, it must be $r_{i_n}(\mathbf{Q}) > 0$, for all i and n , and $f_{i_n} > w_{i_n}/\tilde{T}_{i_n}$, $\forall i_n \in \mathcal{I}_0$. Hence, condition $g_{i_n}(\mathbf{Q}, f_{i_n}) \leq 0$ in \mathcal{P} can be equivalently rewritten as

$$r_{i_n}(\mathbf{Q}) \geq \alpha_{i_n}(f_{i_n}) \triangleq \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} > 0, \quad \forall i_n \in \mathcal{I}_0$$

so that one can choose $\mathcal{E} \triangleq \{(\mathbf{Q}, \mathbf{f}) : \text{c), d) hold, } r_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-i_n} = \mathbf{0}) \geq \alpha_{i_n}(f_{i_n}), \forall i_n \in \mathcal{I}_0, r_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-i_n} = \mathbf{0}) \geq R_{i_n}^{\min}, \forall i_n \in \mathcal{I}_{\text{no}}\}$.

Following [30], [31], our goal is to build, at each iteration ν , an approximant, say $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$,¹ of the nonconvex (nonseparable) $E(\mathbf{Q})$ around the current (feasible) iterate $\mathbf{Z}^\nu \in \mathcal{X}$ that enjoys the following key properties:

P1: $\tilde{E}(\bullet; \mathbf{Z}^\nu)$ is uniformly *strongly convex* on \mathcal{E} ;

P2: $\nabla_{\mathbf{Q}^*} \tilde{E}(\mathbf{Z}^\nu; \mathbf{Z}^\nu) = \nabla_{\mathbf{Q}^*} E(\mathbf{Q}^\nu)$, $\forall \mathbf{Z}^\nu \in \mathcal{X}$;

P3: $\nabla_{\mathbf{Z}^*} \tilde{E}(\bullet; \bullet)$ is Lipschitz continuous on $\mathcal{E} \times \mathcal{X}$;

where $\nabla_{\mathbf{Z}^*} \tilde{E}(\mathbf{Z}^\nu; \mathbf{Z}^\nu)$ denotes the conjugate gradient [32] of $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ with respect to the first argument \mathbf{Z} , evaluated at \mathbf{Z}^ν . Conditions P1-P2 just guarantee that the candidate approximation $\tilde{E}(\bullet; \mathbf{Z}^\nu)$ is strongly convex while preserving the same first order behaviour of $E(\mathbf{Q})$ at any iterate \mathbf{Q}^ν ; P3 is a standard continuity requirement.

We build next a $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ satisfying P1-P3. Observe that i) for any given $\mathbf{Q}_{-n} = \mathbf{Q}_{-n}^\nu$, each term $E_{i_n}(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu) = \text{tr}(\mathbf{Q}_{i_n}) \cdot \Delta_{i_n}^t(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$ of the sum in $E(\mathbf{Q})$ [cf. (13)] is the product of two convex functions in \mathbf{Q}_{i_n} [cf. (6)], namely: $\text{tr}(\mathbf{Q}_{i_n})$ and $\Delta_{i_n}^t(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$; and ii) the other terms of the sum— $\sum_{j_m \in \mathcal{I}, m \neq n} E_{j_m}(\mathbf{Q}_{i_n}; \mathbf{Q}_{-i_n, j_m}^\nu)$ with $\mathbf{Q}_{-i_n, j_m}^\nu \triangleq (\mathbf{Q}_{i_q}^\nu, (\mathbf{Q}_{i_q}^\nu)_{\forall l, q \neq m, l_q \neq i_n})$ —are not convex in \mathbf{Q}_{i_n} . Exploiting such a structure, a convex approximation of $E(\mathbf{Q})$ can be obtained for each MU i_n by convexifying the term $\text{tr}(\mathbf{Q}_{i_n}) \cdot \Delta_{i_n}^t(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$ and linearizing the nonconvex part $\sum_{j_m \in \mathcal{I}, m \neq n} E_{j_m}(\mathbf{Q}_{i_n}; \mathbf{Q}_{-i_n, j_m}^\nu)$. More formally, denoting

¹Note that $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ is a function of \mathbf{Z} , given the current iterate \mathbf{Z}^ν .

$\mathbf{Z}_{i_n} \triangleq (\mathbf{Q}_{i_n}, f_{i_n})$, for each i_n , let us introduce the ‘‘approximation’’ function $\tilde{E}_{i_n}(\mathbf{Z}_{i_n}; \mathbf{Z}^\nu)$:

$$\begin{aligned} \tilde{E}_{i_n}(\mathbf{Z}_{i_n}; \mathbf{Z}^\nu) &\triangleq \frac{c_{i_n} \cdot \text{tr}(\mathbf{Q}_{i_n})}{r_{i_n}(\mathbf{Q}_{i_n}^\nu, \mathbf{Q}_{-n}^\nu)} + \frac{c_{i_n} \cdot \text{tr}(\mathbf{Q}_{i_n}^\nu)}{r_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-n}^\nu)} \\ &+ \sum_{j_m \in \mathcal{I}, m \neq n} \left\langle \nabla_{\mathbf{Q}_{i_n}^*} E_{j_m}(\mathbf{Q}^\nu), \mathbf{Q}_{i_n} - \mathbf{Q}_{i_n}^\nu \right\rangle \\ &+ \tau_{i_n} \|\mathbf{Q}_{i_n} - \mathbf{Q}_{i_n}^\nu\|^2 + \frac{c_{f_{i_n}}}{2} (f_{i_n} - f_{i_n}^\nu)^2 \end{aligned} \quad (17)$$

where: the first two terms on the right-hand side are the aforementioned convexification of $\text{tr}(\mathbf{Q}_{i_n}) \cdot \Delta_{i_n}^t(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$; the third term comes from the linearization of $\sum_{j_m \in \mathcal{I}, m \neq n} E_{j_m}(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$, with $\langle \mathbf{A}, \mathbf{B} \rangle \triangleq \text{Re}\{\text{tr}(\mathbf{A}^H \mathbf{B})\}$ and $\nabla_{\mathbf{Q}_{i_n}^*} E_{j_m}(\mathbf{Q}^\nu)$ denoting the conjugate gradient of $E_{j_m}(\mathbf{Q})$ with respect to \mathbf{Q}_{i_n} evaluated at \mathbf{Q}^ν , and given by

$$\begin{aligned} \nabla_{\mathbf{Q}_{i_n}^*} E_{j_m}(\mathbf{Q}^\nu) &= \frac{\text{tr}(\mathbf{Q}_{j_m}^\nu) \Delta_{j_m}^t(\mathbf{Q}^\nu)}{\log(2) r_{j_m}(\mathbf{Q}^\nu)} \cdot [\mathbf{H}_{i_n m}^H (\mathbf{R}_m(\mathbf{Q}_{-m}^\nu)^{-1} \\ &- (\mathbf{R}_m(\mathbf{Q}_{-m}^\nu) + \mathbf{H}_{j_m m} \mathbf{Q}_{j_m}^\nu \mathbf{H}_{j_m m}^H)^{-1}) \mathbf{H}_{i_n m}]; \end{aligned} \quad (18)$$

and the two last terms in (17) are quadratic regularization terms added to make $\tilde{E}_{i_n}(\bullet; \mathbf{Z}^\nu)$ uniformly strongly convex on \mathcal{E} , (note that $c_{f_{i_n}} = 0$, for all $i_n \in \mathcal{I}_{\text{no}}$).

Based on each $\tilde{E}_{i_n}(\mathbf{Z}_{i_n}; \mathbf{Z}^\nu)$, we can now define the candidate sum-energy approximation $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ as: given $\mathbf{Z}^\nu \in \mathcal{X}$,

$$\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu) \triangleq \sum_{i_n \in \mathcal{I}} \tilde{E}_{i_n}(\mathbf{Z}_{i_n}; \mathbf{Z}^\nu). \quad (19)$$

It is not difficult to check that $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ satisfies P1-P3; in particular it is strongly convex on \mathcal{E} , with constant $c_{\tilde{E}} \geq \min(\min_{i_n \in \mathcal{I}_o} (\tau_{i_n}, c_{f_{i_n}}), \min_{i_n \in \mathcal{I}_{\text{no}}} (\tau_{i_n})) > 0$. Note that $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ is also separable in the users variables \mathbf{Z}_{i_n} , which is instrumental to obtain distributed algorithms, see Sec. V.

2) *Inner convexification of the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$* : We aim at introducing an inner convex approximation, say $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu)$, of the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$ around $\mathbf{Z}^\nu \in \mathcal{X}$, satisfying the following key properties [30], [31]:

- C1: $\tilde{g}_{i_n}(\bullet; \mathbf{Z}^\nu)$ is uniformly convex on \mathcal{E} ;
- C2: $\nabla_{\mathbf{Z}^*} \tilde{g}_{i_n}(\mathbf{Q}^\nu, f_{i_n}^\nu; \mathbf{Z}^\nu) = \nabla_{\mathbf{Z}^*} g_{i_n}(\mathbf{Q}^\nu, f_{i_n}^\nu), \forall \mathbf{Z}^\nu \in \mathcal{X}$;
- C3: $\nabla_{\mathbf{Z}^*} \tilde{g}_{i_n}(\bullet; \bullet)$ is continuous on $\mathcal{E} \times \mathcal{X}$;
- C4: $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu) \geq g_{i_n}(\mathbf{Q}, f_{i_n}), \quad \forall (\mathbf{Q}, f_{i_n}) \in \mathcal{E} \quad \text{and} \quad \forall \mathbf{Z}^\nu \in \mathcal{X}$;
- C5: $\tilde{g}_{i_n}(\mathbf{Q}^\nu, f_{i_n}^\nu; \mathbf{Z}^\nu) = g_{i_n}(\mathbf{Q}^\nu, f_{i_n}^\nu), \forall \mathbf{Z}^\nu \in \mathcal{X}$;
- C6: $\tilde{g}_{i_n}(\bullet; \bullet)$ is Lipschitz continuous on $\mathcal{E} \times \mathcal{X}$.

Conditions C1-C3 are the counterparts of P1-P3 on \tilde{g}_{i_n} ; the extra conditions C4-C5 guarantee that \tilde{g}_{i_n} is an inner approximation of g_{i_n} , implying that any (\mathbf{Q}, f_{i_n}) satisfying $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu) \leq 0$ is feasible also for the original nonconvex problem \mathcal{P} .

To build a \tilde{g}_{i_n} satisfying C1-C6, let us exploit first the concave-convex structure of the rate functions $r_{i_n}(\mathbf{Q})$ [cf. (3)]:

$$r_{i_n}(\mathbf{Q}) = r_{i_n}^+(\mathbf{Q}) + r_{i_n}^-(\mathbf{Q}_{-n}), \quad (20)$$

where

$$\begin{aligned} r_{i_n}^+(\mathbf{Q}) &\triangleq \log_2 \det (\mathbf{R}_n(\mathbf{Q}_{-n}) + \mathbf{H}_{i_n n} \mathbf{Q}_{i_n} \mathbf{H}_{i_n n}^H) \\ r_{i_n}^-(\mathbf{Q}_{-n}) &\triangleq -\log_2 \det (\mathbf{R}_n(\mathbf{Q}_{-n})) \end{aligned} \quad (21)$$

with $\mathbf{R}_n(\mathbf{Q}_{-n})$ defined in (4). Note that $r_{i_n}^+(\bullet)$ and $r_{i_n}^-(\bullet)$ are concave on \mathcal{Q} and convex on $\mathcal{Q}_{-n} \triangleq \prod_{m \neq n} \mathcal{Q}_m$, respectively. Using (20), and observing that at any (feasible) $(\mathbf{Q}, \mathbf{f}) \in \mathcal{X}$, it must be $r_{i_n}(\mathbf{Q}) > 0$ for all i_n , and $f_{i_n} > w_{i_n}/\tilde{T}_{i_n}$ for all $i_n \in \mathcal{I}_o$, the constraints $g_{i_n}(\mathbf{Q}, f_{i_n}) \leq 0$ in \mathcal{P} can be equivalently rewritten as

$$g_{i_n}(\mathbf{Q}, f_{i_n}) = -r_{i_n}^+(\mathbf{Q}) - r_{i_n}^-(\mathbf{Q}_{-n}) + \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} \leq 0, \quad (22)$$

where with a slight abuse of notation we used the same symbol $g_{i_n}(\mathbf{Q}, f_{i_n})$ to denote the constraint in the equivalent form.

The desired inner convex approximation $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu)$ is obtained from $g_{i_n}(\mathbf{Q}, f_{i_n})$ by retaining the convex part in (22) and linearizing the concave term $-r_{i_n}^-(\mathbf{Q}_{-n})$, resulting in:

$$\begin{aligned} \tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu) &\triangleq -r_{i_n}^+(\mathbf{Q}) + \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} \\ &- r_{i_n}^-(\mathbf{Q}_{-n}^\nu) - \sum_{j_m \in \mathcal{I}} \langle \Pi_{j_m, n}^-(\mathbf{Q}^\nu), \mathbf{Q}_{j_m} - \mathbf{Q}_{j_m}^\nu \rangle \end{aligned} \quad (23)$$

where each $\Pi_{j_m, n}^-(\mathbf{Q}^\nu)$ is defined as

$$\Pi_{j_m, n}^-(\mathbf{Q}^\nu) \triangleq \begin{cases} \nabla_{\mathbf{Q}_{j_m}^*} r_{i_n}^-(\mathbf{Q}_{-n}^\nu), & \text{if } m \neq n; \\ 0, & \text{otherwise;} \end{cases} \quad (24)$$

and $\nabla_{\mathbf{Q}_{j_m}^*} r_{i_n}^-(\mathbf{Q}_{-n}^\nu) = -\mathbf{H}_{j_m n}^H \mathbf{R}_n(\mathbf{Q}_{-n}^\nu)^{-1} \mathbf{H}_{j_m n}$.

The inner convex approximation of the rate constraints $r_{i_n}(\mathbf{Q}) \geq R_{i_n}^{\min}$ in \mathcal{P} can be easily obtained by equation (23) as

$$\tilde{g}_{i_n}(\mathbf{Q}, 0; \mathbf{Z}^\nu) + R_{i_n}^{\min} \leq 0, \quad \forall i_n \in \mathcal{I}_{\text{no}}. \quad (25)$$

3) *Inner SCA algorithm-centralized implementation*: We are now ready to introduce the proposed inner convex approximation of the nonconvex problem \mathcal{P} , which consists in replacing the nonconvex objective function $E(\mathbf{Q})$ and constraints $g_{i_n}(\mathbf{Q}, f_{i_n}) \leq 0$, and $r_{i_n}(\mathbf{Q}) \geq R_{i_n}^{\min}$ in \mathcal{P} with the approximations $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$, $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu) \leq 0$ and $\tilde{g}_{i_n}(\mathbf{Q}, 0; \mathbf{Z}^\nu) + R_{i_n}^{\min} \leq 0$, respectively. More formally, given the feasible point \mathbf{Z}^ν , we have

$$\begin{aligned} \hat{\mathbf{Z}}(\mathbf{Z}^\nu) &\triangleq \underset{\mathbf{Q}, \mathbf{f}}{\text{argmin}} \tilde{E}(\mathbf{Q}; \mathbf{Q}^\nu) \\ \text{s.t.} \quad & \begin{aligned} &a) \tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Z}^\nu) \leq 0, \quad \forall i_n \in \mathcal{I}_o, \\ &b) \tilde{g}_{i_n}(\mathbf{Q}, 0; \mathbf{Z}^\nu) + R_{i_n}^{\min} \leq 0, \quad \forall i_n \in \mathcal{I}_{\text{no}}, \\ &c) \sum_{i_n \in \mathcal{I}_o} f_{i_n} \leq f_T, \quad f_{i_n} \geq 0, \quad \forall i_n \in \mathcal{I}_o, \\ &d) \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I}, \end{aligned} \end{aligned} \quad (\mathcal{P}^\nu)$$

where we denoted by $\hat{\mathbf{Z}}(\mathbf{Z}^\nu) \triangleq (\hat{\mathbf{Q}}(\mathbf{Z}^\nu), \hat{\mathbf{f}}(\mathbf{Z}^\nu))$ the unique solution of the strongly convex optimization problem.

The proposed solution consists in solving the sequence of problems \mathcal{P}^ν , starting from a feasible $\mathbf{Z}^0 \triangleq (\mathbf{Q}^0, \mathbf{f}^0)$. The formal description of the method along with its convergence properties are given in Algorithm 2 and Theorem 2, respectively. Note that the optimal solution $\hat{\mathbf{Z}}$ of \mathcal{P}^ν computed in Step 2 of the algorithm is used in Step 3 to set the next iterate $\mathbf{Z}^{\nu+1} \triangleq (\mathbf{Q}^{\nu+1}, \mathbf{f}^{\nu+1})$ by including a step-size in the updating rule. A practical termination criterion in Step 1 is $|E(\mathbf{Q}^{\nu+1}) - E(\mathbf{Q}^\nu)| \leq \delta$, where $\delta > 0$ is the prescribed accuracy. The proof of Theorem 2 consists in showing that [31, Theorem 1] holds true, and is omitted because of space limitation.

Algorithm 2. Inner SCA Algorithm for \mathcal{P}

Initial data: $\mathbf{Z}^0 \triangleq (\mathbf{Q}^0, \mathbf{f}^0) \in \mathcal{X}$; $\{\gamma^\nu\}_\nu \in (0, 1]$;
 (S.1): If \mathbf{Z}^ν satisfies a suitable termination criterion, STOP
 (S.2): Compute $\hat{\mathbf{Z}}(\mathbf{Z}^\nu) \triangleq (\hat{\mathbf{Q}}(\mathbf{Z}^\nu), \hat{\mathbf{f}}(\mathbf{Z}^\nu))$ [cf. \mathcal{P}^ν];
 (S.3): Set $\mathbf{Z}^{\nu+1} = \mathbf{Z}^\nu + \gamma^\nu (\hat{\mathbf{Z}}(\mathbf{Z}^\nu) - \mathbf{Z}^\nu)$;
 (S.4): $\nu \leftarrow \nu + 1$ and go to (S.1).

Theorem 2: Given the nonconvex problem \mathcal{P} , choose $c_{\tilde{E}} > 0$ and $\{\gamma^\nu\}_\nu$ such that

$$(0, 1] \ni \gamma^\nu \rightarrow 0, \forall \nu \geq 0, \quad \text{and} \quad \sum_\nu \gamma^\nu = +\infty. \quad (26)$$

Then every limit point of $\{\mathbf{Z}^\nu\}$ (at least one of such points exists) is a stationary solution of \mathcal{P} . Furthermore, none of such points is a local maximum of the energy function E .

Theorem 2 offers some flexibility in the choice of the free parameters $c_{\tilde{E}}$ and $\{\gamma^\nu\}_\nu$ while guaranteeing convergence of Algorithm 2. For instance, $c_{\tilde{E}}$ is positive if all τ_{i_n} and $(c_{f_{i_n}})_{i_n \in \mathcal{I}_o}$ are positive (but arbitrary); in the case of full-column rank matrices $\mathbf{H}_{i_n n}$, one can also set $\tau_{i_n} = 0$ (still resulting in $c_{\tilde{E}} > 0$). Many choices are possible for the step-size γ^ν ; a practical rule satisfying (26) that we found effective in our experiments is [33]:

$$\gamma^{\nu+1} = \gamma^\nu (1 - \bar{\alpha} \gamma^\nu), \quad \gamma^0 \in (0, 1], \quad (27)$$

with $\bar{\alpha} \in (0, 1/\gamma^0)$.

4) *On the implementation of Algorithm 2:* Since the base stations are connected to the cloud throughout high speed wired links, a good candidate place to run Algorithm 2 is the cloud itself: The cloud collects first all system parameters needed to run the algorithm from the SCAEsNBs (MUs' channel state information, maximum tolerable latency, etc.); then, if the feasibility conditions (16) are satisfied, the cloud solves the strongly convex problems \mathcal{P}^ν (using any standard nonlinear programming solver), and sends the solutions \mathbf{Q}_n back to the corresponding SCAEsNBs; finally, each SCAEsNB communicates the optimal transmit parameters to the MUs it is serving.

5) *Related works:* Algorithm 2 hinges on the idea of successive convex programming, which aims at computing stationary solutions of some classes of nonconvex problems by solving a sequence of convexified subproblems. Some relevant instances of this method that have attracted significant interest in recent years are: i) the basic DCA (Difference-of-Convex Algorithm) [34], [35]; ii) the M(ajorization)-M(inimization)

algorithm [36], [37]; iii) alternating/successive minimization methods [38]–[40]; and iv) partial linearization methods [33], [41], [42]. The aforementioned methods identify classes of “favorable” nonconvex functions, for which a suitable convex approximation can be obtained and convergence of the associated sequential convex programming method can be proved. However, the sum-energy function $E(\mathbf{Q})$ in (13) and the resulting nonconvex optimization problem \mathcal{P} do not belong to any of the above classes. More specifically, what makes current algorithms not readily applicable to problem \mathcal{P} is the lack in the objective function $E(\mathbf{Q})$ of a(n additively) separable convex and nonconvex part [each $E_{i_n}(\mathbf{Q})$ in (13) is in fact the product of two functions, $\text{tr}(\mathbf{Q}_{i_n})$ and $\Delta_{i_n}^\dagger(\mathbf{Q}_{i_n}; \mathbf{Q}_{-n}^\nu)$, of the same set of variables]. Therefore, the proposed approximation function $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$, along with the resulting SCA-algorithm, i.e., Algorithm 2, are an innovative contribution of this work.

V. DISTRIBUTED IMPLEMENTATION

To alleviate the communication overhead of a centralized implementation (Algorithm 2), in this section we devise *distributed* algorithms converging to local optimal solutions of \mathcal{P} . Following [31], the main idea is to choose the approximation functions \tilde{E} , \tilde{g}_{i_n} so that (on top of satisfying conditions P.1-P.3 and C.1-C.6, needed for convergence) the resulting convexified problems \mathcal{P}^ν can be decomposed into (smaller) subproblems solvable in parallel across the SCAEsNBs, with limited signaling between the SCAEsNBs and the cloud.

Since the approximation function \tilde{E} introduced in (19) is (sum) separable in the optimization variables of the MUs in each cell, any choice of \tilde{g}_{i_n} 's enjoying the same decomposability structure leads naturally to convexified problems \mathcal{P}^ν that can be readily decomposed across the SCAEsNBs by using standard primal or dual decomposition techniques.

Of course there is more than one choice of \tilde{g}_{i_n} meeting the above requirements; all of them lead to *convergent* algorithms that however differ for convergence speed, complexity, communication overhead, and a-priori knowledge of the system parameters. As case study, in the following, we consider two representative valid approximants. The first candidate \tilde{g}_{i_n} is obtained exploiting the Lipschitz property of the gradient of the rate functions r_{i_n} , whereas the second one is based on an equivalent reformulation of \mathcal{P} introducing proper slack variables. The first choice offers a lot of flexibility in the design of distributed algorithms—both primal and dual-based schemes can be invoked—but it requires knowledge of all the Lipschitz constants. The second choice does not need this knowledge, but it involves a higher computational cost at the SCAEsNBs side, due to the presence of the slack variables.

A. Per-cell Distributed Dual and Primal Decompositions

The approximation function \tilde{g}_{i_n} in (23) has the desired property of preserving the structure of the original constraint function g_{i_n} “as much as possible” by keeping the concave part $r_{i_n}^+(\mathbf{Q})$ of $r_{i_n}(\mathbf{Q})$ unaltered. Numerical results show that this choice leads to fast convergence schemes, see Sec. VI. However the structure of \tilde{g}_{i_n} prevents \mathcal{P}^ν to be decomposed across the

SCeNBs due to the *nonadditive* coupling among the variables \mathbf{Q}_n in $r_{i_n}^+(\mathbf{Q})$. To cope with this issue, we lower bound $r_{i_n}^+(\mathbf{Q})$ [and thus upper bound \tilde{g}_{i_n} in (23)], so that we obtain an alternative approximation of g_{i_n} that is *separable in all* the \mathbf{Q}_n 's, while still satisfying C.1-C.6. Invoking the Lipschitz property of the (conjugate) gradients $\nabla_{\mathbf{Q}_{j_l}^*} r_{i_n}^+(\bullet)$ on \mathcal{Q} , with constant L_{j_l, i_n} [we omit the explicit expression of L_{j_l, i_n} because of space limitations], we have

$$r_{i_n}^+(\mathbf{Q}) \geq \tilde{r}_{i_n}^+(\mathbf{Q}; \mathbf{Q}^\nu) \triangleq r_{i_n}^+(\mathbf{Q}^\nu) + \sum_{j_l \in \mathcal{I}} (\langle \Pi_{j_l, i_n}^+(\mathbf{Q}^\nu), \mathbf{Q}_{j_l} - \mathbf{Q}_{j_l}^\nu \rangle - c_{j_l, i_n} \|\mathbf{Q}_{j_l} - \mathbf{Q}_{j_l}^\nu\|^2),$$

for all $\mathbf{Q}, \mathbf{Q}^\nu \in \mathcal{Q}$, where each $\Pi_{j_l, i_n}^+(\mathbf{Q}^\nu)$ and c_{j_l, i_n} are defined respectively as

$$\Pi_{j_l, i_n}^+(\mathbf{Q}^\nu) \triangleq \begin{cases} \nabla_{\mathbf{Q}_{j_l}^*} r_{i_n}^+(\mathbf{Q}^\nu), & \text{if } l \neq n \text{ or } j_l = i_n, \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

with $\nabla_{\mathbf{Q}_{j_l}^*} r_{i_n}^+(\mathbf{Q}^\nu) = \mathbf{H}_{j_l n}^H (\mathbf{R}_n(\mathbf{Q}_{-n}^\nu) + \mathbf{H}_{i_n n} \mathbf{Q}_{i_n}^\nu \mathbf{H}_{i_n n}^H)^{-1} \cdot \mathbf{H}_{j_l n}$ and

$$c_{j_l, i_n} \triangleq \begin{cases} L_{j_l, i_n}, & \text{if } l \neq n \text{ or } j_l = i_n, \\ 0, & \text{otherwise.} \end{cases} \quad (29)$$

Note that $\tilde{r}_{i_n}^+(\mathbf{Q}; \mathbf{Q}^\nu)$ is (sum) separable in the MUs' covariance matrices \mathbf{Q}_{i_n} 's. The desired approximant of g_{i_n} can be then obtained just replacing $r_{i_n}^+(\mathbf{Q})$ in \tilde{g}_{i_n} with $\tilde{r}_{i_n}^+(\mathbf{Q}; \mathbf{Q}^\nu)$ [cf. (23)], resulting in

$$\begin{aligned} \tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu) &\triangleq -\tilde{r}_{i_n}^+(\mathbf{Q}; \mathbf{Q}^\nu) + \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} \\ &\quad - r_n^-(\mathbf{Q}_{-n}^\nu) - \sum_{j_l \in \mathcal{I}} \langle \Pi_{j_l, n}^-(\mathbf{Q}^\nu), \mathbf{Q}_{j_l} - \mathbf{Q}_{j_l}^\nu \rangle \\ &\triangleq \sum_{j_l \in \mathcal{I}} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) + \bar{q}_{i_n}(f_{i_n}; \mathbf{Q}^\nu) \end{aligned} \quad (30)$$

with $\tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu)$ and $\bar{q}_{i_n}(f_{i_n}; \mathbf{Q}^\nu)$ given by

$$\begin{aligned} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) &\triangleq c_{j_l, i_n} \|\mathbf{Q}_{j_l} - \mathbf{Q}_{j_l}^\nu\|^2 \\ &\quad - \langle \Pi_{j_l, i_n}^+(\mathbf{Q}^\nu) + \Pi_{j_l, n}^-(\mathbf{Q}^\nu), \mathbf{Q}_{j_l} - \mathbf{Q}_{j_l}^\nu \rangle, \\ \bar{q}_{i_n}(f_{i_n}; \mathbf{Q}^\nu) &\triangleq \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} - r_{i_n}(\mathbf{Q}^\nu). \end{aligned}$$

It is not difficult to check that $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu)$, on top of being separable in the MUs' covariance matrices, also satisfies the required conditions C.1-C.6. Using $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu)$ instead of $g_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu)$, the convexified subproblem replacing \mathcal{P}^ν is: given $\mathbf{Z}^\nu \in \mathcal{X}$,

$$\begin{aligned} \hat{\mathbf{Z}}(\mathbf{Z}^\nu) &\triangleq \underset{\mathbf{Q}, \mathbf{f}}{\operatorname{argmin}} \sum_{i_n \in \mathcal{I}} \tilde{E}_{i_n}(\mathbf{Z}_{i_n}; \mathbf{Z}^\nu) \\ \text{s.t.} \quad &\text{a) } \sum_{j_l \in \mathcal{I}} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) + \bar{q}_{i_n}(f_{i_n}; \mathbf{Q}^\nu) \leq 0, \\ &\quad \forall i_n \in \mathcal{I}_0, \\ &\text{b) } \sum_{j_l \in \mathcal{I}} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) - r_{i_n}(\mathbf{Q}^\nu) + R_{i_n}^{\min} \leq 0, \\ &\quad \forall i_n \in \mathcal{I}_{\text{no}}, \\ &\text{c) } \sum_{i_n \in \mathcal{I}_0} f_{i_n} \leq f_T, \quad f_{i_n} \geq 0, \quad \forall i_n \in \mathcal{I}_0, \\ &\text{d) } \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I}, \end{aligned} \quad (\mathcal{P}_d^\nu)$$

where with a slight abuse of notation we still use $\hat{\mathbf{Z}}(\mathbf{Z}^\nu) \triangleq (\hat{\mathbf{Q}}(\mathbf{Z}^\nu), \hat{\mathbf{f}}(\mathbf{Z}^\nu))$ to denote the unique solution of \mathcal{P}_d^ν .

Problem \mathcal{P}_d^ν is now (sum) separable in the MUs' covariance matrices; it can be solved in a distributed way using standard primal or dual decomposition techniques. We briefly show next how to customize standard dual algorithms to \mathcal{P}_d^ν .

1) *Per-cell optimization via dual decomposition:* The subproblems \mathcal{P}_d^ν can be solved in a distributed way if the side constraints $\tilde{q}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu) \leq 0$ are dualized (note that there is zero duality gap). The dual problem associated with \mathcal{P}_d^ν is: given $\mathbf{Z}^\nu \triangleq (\mathbf{Q}^\nu, \mathbf{f}^\nu) \in \mathcal{X}$,

$$\lambda \triangleq ((\lambda_{i_n})_{i_n \in \mathcal{I}}, \lambda_f) \geq \mathbf{0} \quad \max_{\lambda} D(\hat{\mathbf{Z}}(\lambda; \mathbf{Z}^\nu), \lambda; \mathbf{Z}^\nu) \quad (31)$$

with $\hat{\mathbf{Z}}(\lambda; \mathbf{Z}^\nu) \triangleq (\hat{\mathbf{Z}}_n(\lambda; \mathbf{Z}^\nu))_{n=1}^{N_c}$, $\hat{\mathbf{Z}}_n(\lambda; \mathbf{Z}^\nu) \triangleq (\hat{\mathbf{Q}}_n(\lambda; \mathbf{Z}^\nu), \hat{\mathbf{f}}_n(\lambda; \mathbf{Z}^\nu)) = ((\hat{\mathbf{Q}}_{i_n}(\lambda; \mathbf{Z}^\nu))_{i=1}^{K_n}, (\hat{f}_{i_n}(\lambda; \mathbf{Z}^\nu))_{i=1}^{K_n})$ and K_n^o denoting the number of MUs in cell n performing offloading. $\hat{\mathbf{Z}}$ is the unique minimizer of the Lagrangian function associated with \mathcal{P}_d^ν , which after reorganizing terms can be written as

$$\hat{\mathbf{Z}}(\lambda; \mathbf{Z}^\nu) \triangleq \underset{\mathbf{Q} \in \mathcal{Q}, \mathbf{f} \in \mathbb{R}_+^{|\mathcal{I}_0|}}{\operatorname{argmin}} \sum_{n=1}^{N_c} (\mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \lambda; \mathbf{Q}^\nu) + \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \lambda; \mathbf{f}_n^\nu)), \quad (32)$$

where $\mathbf{Q}_n \triangleq (\mathbf{Q}_{i_n})_{i=1}^{K_n}$, $\mathbf{f}_n \triangleq (f_{i_n})_{i=1}^{K_n^o}$ and

$$\begin{aligned} \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \lambda; \mathbf{Q}^\nu) &= \sum_{i=1}^{K_n} \left\{ \tilde{E}_{i_n}(\mathbf{Q}_{i_n}, f_{i_n}^\nu; \mathbf{Z}^\nu) + \sum_{j_l \in \mathcal{I}} \lambda_{j_l} \tilde{q}_{i_n, j_l}(\mathbf{Q}_{i_n}; \mathbf{Q}^\nu) \right\}, \\ \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \lambda; \mathbf{f}_n^\nu) &= \sum_{i=1}^{K_n^o} \left\{ \frac{c_{f_{i_n}}}{2} (f_{i_n} - f_{i_n}^\nu)^2 + \frac{\lambda_{i_n} \cdot c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \tilde{T}_{i_n} - w_{i_n}} + \lambda_f f_{i_n} \right\}. \end{aligned} \quad (33)$$

Note that, thanks to the separability structure of the Lagrangian function, the optimal solutions $\hat{\mathbf{Z}}_n(\lambda; \mathbf{Z}^\nu) = (\hat{\mathbf{Q}}_n(\lambda; \mathbf{Q}^\nu), \hat{\mathbf{f}}_n(\lambda; \mathbf{f}^\nu))$ of (32) can be computed in parallel

across the SCeNBs, solving each SCeNBs n the following strongly convex problems: given $\lambda \geq 0$,

$$\begin{aligned}\hat{\mathbf{Q}}_n(\lambda; \mathbf{Q}^\nu) &\triangleq \underset{\mathbf{Q}_n \in \Pi_{i=1}^{K_n} \mathcal{Q}_{i_n}}{\operatorname{argmin}} \{ \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \lambda; \mathbf{Q}^\nu) \} \\ \hat{\mathbf{f}}_n(\lambda; \mathbf{f}^\nu) &\triangleq \underset{\mathbf{f}_n \in \mathbb{R}_+^{K_n}}{\operatorname{argmin}} \{ \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \lambda; \mathbf{f}_n^\nu) \}.\end{aligned}\quad (34)$$

The solution of \mathcal{P}_d^ν can be then computed solving the dual problem (31). It is not difficult to prove that the dual function D is differentiable with Lipschitz gradient. One can then solve (31) using, e.g., the gradient-based algorithm with diminishing step-size described in Algorithm 3, whose convergence is stated in Theorem 3 (the proof follows standard arguments and thus is omitted, because of space limitations).

Theorem 3: Given \mathcal{P}_d^ν , choose $\{\beta_k\}$ so that $\beta_k > 0$, $\beta_k \rightarrow 0$, $\sum_k \beta_k = +\infty$, and $\sum_k (\beta_k)^2 < \infty$. Then, the sequence $\{\lambda_k\}$ generated by Algorithm 3 converges to a solution of (31). Therefore, the sequence $\{\hat{\mathbf{Z}}^k(\lambda_k; \mathbf{Z}^\nu)\}_k$ converges to the unique solution of \mathcal{P}_d^ν .

Algorithm 3. Distributed implementation of S.2 in Alg. 2.

Initial data: $\lambda^0 \geq 0$, $\mathbf{Z}^\nu = (\mathbf{Q}^\nu, \mathbf{f}^\nu)$, $\{\beta_k\} > 0$. Set $k = 0$,
(S.1): If λ^k satisfies a suitable termination criterion: STOP;
(S.2): For each SCeNB n , compute in parallel $\mathbf{Q}_n^{k+1}(\lambda^k; \mathbf{Z}^\nu)$ and $\mathbf{f}_n^{k+1}(\lambda^k; \mathbf{Z}^\nu)$ [cf. (34)];

(S.3): Update at the master node λ^{k+1} according to

$$\lambda_{i_n}^{k+1} \triangleq \left[\lambda_{i_n}^k + \beta_k \left(\sum_{j_l \in \mathcal{I}} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) + \bar{q}_{i_n}(f_{i_n}; \mathbf{Q}^\nu) \right) \right]^+, \quad \forall i_n \in \mathcal{I}_0$$

$$\lambda_{i_n}^{k+1} \triangleq \left[\lambda_{i_n}^k + \beta_k \left(\sum_{j_l \in \mathcal{I}} \tilde{q}_{j_l, i_n}(\mathbf{Q}_{j_l}; \mathbf{Q}^\nu) - r_{i_n}(\mathbf{Q}^\nu) + R_{i_n}^{\min} \right) \right]^+, \quad \forall i_n \in \mathcal{I}_{\text{no}}$$

$$\lambda_f^{k+1} \triangleq \left[\lambda_f^k + \beta_k \left(\sum_{i_n \in \mathcal{I}_0} f_{i_n}^{k+1} - f_T \right) \right]^+$$

(S.4): $k \leftarrow k + 1$ and go back to (S.1).

B. Alternative Decomposition via Slack Variables

In this section we present an alternative decomposition strategy of problem \mathcal{P} that does not require the knowledge of the Lipschitz constants L_{j_l, i_n} . At the basis of our approach there is an equivalent reformulation of \mathcal{P} based on the introduction of proper slack variables that are instrumental to decouple in each $r_{i_n}^+(\mathbf{Q})$ [cf. (21)] the covariance matrix \mathbf{Q}_{i_n} of user i_n from those of the MUs in the other cells—the interference term $\mathbf{R}_n(\mathbf{Q}_{-n})$ [cf. (4)]. More specifically, introducing the slack variables \mathbf{Y}_{i_n} , and

$$\mathbf{I}_{i_n}(\mathbf{Q}) \triangleq \sum_{j_m \in \mathcal{I}, m \neq n} \mathbf{H}_{j_m n} \mathbf{Q}_{j_m} \mathbf{H}_{j_m n}^H + \mathbf{H}_{i_n n} \mathbf{Q}_{i_n} \mathbf{H}_{i_n n}^H, \quad (35)$$

we can write

$$r_{i_n}^+(\mathbf{Q}) = \bar{r}_{i_n}^+(\mathbf{Y}), \quad (36)$$

with

$$\bar{r}_{i_n}^+(\mathbf{Y}_{i_n}) \triangleq \log_2 \det(\mathbf{R}_w + \mathbf{Y}_{i_n}) \text{ and } \mathbf{Y}_{i_n} = \mathbf{I}_{i_n}(\mathbf{Q}). \quad (37)$$

Using (36), (37), and $g_{i_n}(\mathbf{Q}, f_{i_n})$ written as in (22), the original offloading problem \mathcal{P} can be rewritten in the following equivalent form: denoting $\mathbf{Y} \triangleq (\mathbf{Y}_{i_n})_{i_n \in \mathcal{I}}$,

$$\begin{aligned}\min_{\mathbf{Q}, \mathbf{f}, \mathbf{Y}} E(\mathbf{Q}) \\ \text{s.t. a) } & -\bar{r}_{i_n}^+(\mathbf{Y}_{i_n}) - r_n^-(\mathbf{Q}_{-n}) + \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \bar{T}_{i_n} - w_{i_n}} \leq 0, \forall i_n \in \mathcal{I}_0, \\ \text{b) } & -\bar{r}_{i_n}^+(\mathbf{Y}_{i_n}) - r_n^-(\mathbf{Q}_{-n}) + R_{i_n}^{\min} \leq 0, \forall i_n \in \mathcal{I}_{\text{no}}, \\ \text{c) } & \sum_{i_n \in \mathcal{I}_0} f_{i_n} \leq f_T, \quad f_{i_n} \geq 0, \forall i_n \in \mathcal{I}_0, \\ \text{d) } & \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \forall i_n \in \mathcal{I}, \\ \text{e) } & \mathbf{0} \preceq \mathbf{Y}_{i_n} \preceq \mathbf{I}_{i_n}(\mathbf{Q}), \forall i_n \in \mathcal{I}.\end{aligned}\quad (\tilde{\mathcal{P}})$$

We denote by $\tilde{\mathcal{X}}$ the feasible set of $\tilde{\mathcal{P}}$. The equivalence between \mathcal{P} and $\tilde{\mathcal{P}}$ is stated next.

Lemma 1: Given the nonconvex problems \mathcal{P} and $\tilde{\mathcal{P}}$, the following hold:

- (a): Every feasible point of $\tilde{\mathcal{P}}$ (or \mathcal{P}) is regular (i.e., satisfies the Mangasarian-Fromovits Constraint Qualification [43]);
- (b): \mathcal{P} and $\tilde{\mathcal{P}}$ are equivalent in the following sense. If $(\bar{\mathbf{Q}}, \bar{\mathbf{f}})$ is a stationary solution of \mathcal{P} , then there exists a $\bar{\mathbf{Y}}$ such that $(\bar{\mathbf{Q}}, \bar{\mathbf{f}}, \bar{\mathbf{Y}})$ is a stationary solution of $\tilde{\mathcal{P}}$; and viceversa. \square

Condition (a) in the lemma guarantees the existence of stationary points of $\tilde{\mathcal{P}}$, whereas (b) allows us to compute (stationary) solutions of \mathcal{P} solving $\tilde{\mathcal{P}}$.

We convexify next $\tilde{\mathcal{P}}$ following the same guidelines as in Sec. IV [see P.1-P.3 and C.1-C.6]. Introducing

$$\begin{aligned}\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}, \mathbf{Y}_{i_n}; \mathbf{Q}^\nu) &\triangleq -\bar{r}_{i_n}^+(\mathbf{Y}_{i_n}) + \frac{c_{i_n} \cdot f_{i_n}}{f_{i_n} \cdot \bar{T}_{i_n} - w_{i_n}} \\ &\quad - r_n^-(\mathbf{Q}_{-n}) - \sum_{j_m \in \mathcal{I}} \langle \Pi_{j_m, n}^-(\mathbf{Q}^\nu), \mathbf{Q}_{j_m} - \mathbf{Q}_{j_m}^\nu \rangle,\end{aligned}\quad (38)$$

and using the same approximant $\tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu)$ as defined in (17), we have: given a feasible $\mathbf{W}^\nu \triangleq (\mathbf{Z}^\nu, \mathbf{Y}^\nu)$,

$$\begin{aligned}\hat{\mathbf{W}}(\mathbf{W}^\nu) &\triangleq \underset{\mathbf{Q}, \mathbf{f}, \mathbf{Y}}{\operatorname{argmin}} \tilde{E}(\mathbf{Z}; \mathbf{Z}^\nu) + \frac{c_{\mathbf{Y}}}{2} \|\mathbf{Y} - \mathbf{Y}^\nu\|^2 \\ \text{s.t. a) } & \tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}, \mathbf{Y}_{i_n}; \mathbf{Q}^\nu) \leq 0, \forall i_n \in \mathcal{I}_0, \\ \text{b) } & \tilde{g}_{i_n}(\mathbf{Q}, 0, \mathbf{Y}_{i_n}; \mathbf{Q}^\nu) + R_{i_n}^{\min} \leq 0, \forall i_n \in \mathcal{I}_{\text{no}}, \\ \text{c) } & \sum_{i_n \in \mathcal{I}_0} f_{i_n} \leq f_T, \quad f_{i_n} \geq 0, \forall i_n \in \mathcal{I}_0, \\ \text{d) } & \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I}, \\ \text{e) } & \mathbf{0} \preceq \mathbf{Y}_{i_n} \preceq \mathbf{I}_{i_n}(\mathbf{Q}), \forall i_n \in \mathcal{I}\end{aligned}\quad (\tilde{\mathcal{P}}^\nu)$$

where $\hat{\mathbf{W}}(\mathbf{W}^\nu) = (\hat{\mathbf{Q}}(\mathbf{W}^\nu), \hat{\mathbf{f}}(\mathbf{W}^\nu), \hat{\mathbf{Y}}(\mathbf{W}^\nu))$ denotes the unique solution of $\tilde{\mathcal{P}}^\nu$, and $c_{\mathbf{Y}}$ is an arbitrary positive constant.

The stationary solutions of $\tilde{\mathcal{P}}$ (and thus \mathcal{P}) can be computed solving the sequence of strongly convex problems $\tilde{\mathcal{P}}^\nu$. The formal description of the scheme is still given by Algorithm 2 wherein in Step 2, $\hat{\mathbf{Z}}(\mathbf{Z}^\nu)$ is replaced by $\hat{\mathbf{W}}(\mathbf{W}^\nu)$; convergence is guaranteed under conditions in Theorem 2.

The last thing left is showing how to solve each subproblem $\tilde{\mathcal{P}}^\nu$ in a distributed way. Problem $\tilde{\mathcal{P}}^\nu$ can be decoupled across the SCeNB's in the dual domain (note that there is zero duality gap). Indeed, denoting by $\mathbf{W} \triangleq (\mathbf{Q}, \mathbf{f}, \mathbf{Y})$, and $\boldsymbol{\lambda} \triangleq ((\lambda_{i_n})_{i_n \in \mathcal{I}}, \lambda_f)$ and $\boldsymbol{\Omega} \triangleq (\boldsymbol{\Omega}_{i_n} \succeq \mathbf{0})_{i_n \in \mathcal{I}}$ the multipliers associated with the constraints (a), (b), (c) and (e), respectively, the (partial) Lagrangian has the following *additive* structure:

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu) \triangleq \sum_{n=1}^{N_c} \{ \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu) + \mathcal{L}_{\mathbf{Y}_n}(\mathbf{Y}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu) + \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \boldsymbol{\lambda}; \mathbf{f}_n^\nu) \},$$

where

$$\begin{aligned} \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu) &= \sum_{i=1}^{K_n} \left\{ \tilde{E}_{i_n}(\mathbf{Q}_{i_n}, f_{i_n}^\nu; \mathbf{Z}^\nu) - \lambda_{i_n} r_n^-(\mathbf{Q}_{i_n}^\nu) \right. \\ &\quad - \sum_{j_m \in \mathcal{I}} \lambda_{j_m} \langle \Pi_{i_n, m}(\mathbf{Q}^\nu), \mathbf{Q}_{i_n} - \mathbf{Q}_{i_n}^\nu \rangle \\ &\quad - \sum_{j_m \in \mathcal{I}, m \neq n} \langle \boldsymbol{\Omega}_{j_m}, \mathbf{H}_{i_n m} \mathbf{Q}_{i_n} \mathbf{H}_{i_n m}^H \rangle \\ &\quad \left. - \langle \boldsymbol{\Omega}_{i_n}, \mathbf{H}_{i_n n} \mathbf{Q}_{i_n} \mathbf{H}_{i_n n}^H \rangle \right\}, \\ \mathcal{L}_{\mathbf{Y}_n}(\mathbf{Y}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu) &= \sum_{i=1}^{K_n} \left\{ -\lambda_{i_n} \bar{r}_n^+(\mathbf{Y}_{i_n}) + \langle \boldsymbol{\Omega}_{i_n}, \mathbf{Y}_{i_n} \rangle \right. \\ &\quad \left. + \frac{c_Y}{2} \|\mathbf{Y}_{i_n} - \mathbf{Y}_{i_n}^\nu\|^2 \right\}, \end{aligned}$$

and $\mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \boldsymbol{\lambda}; \mathbf{f}_n^\nu)$ is given by (33). The minimization of $\mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu)$ w.r.t. $\mathbf{W} = (\mathbf{Q}, \mathbf{f}, \mathbf{Y}) \triangleq (\mathbf{Q}_n, \mathbf{f}_n, \mathbf{Y}_n)_{n=1}^{N_c}$ becomes then

$$\begin{aligned} D(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu) &\triangleq \sum_{n=1}^{N_c} \left(\min_{\mathbf{Q}_n \in \mathcal{Q}} \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu) \right. \\ &\quad \left. + \min_{(\mathbf{Y}_{i_n} \succeq \mathbf{0})_{i_n \in \mathcal{I}}} \mathcal{L}_{\mathbf{Y}_n}(\mathbf{Y}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu) + \min_{\mathbf{f} \in \mathbb{R}_+^{|\mathcal{I}_o|}} \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \boldsymbol{\lambda}; \mathbf{f}_n^\nu) \right) \end{aligned} \quad (39)$$

whose unique solutions $\hat{\mathbf{W}}(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu) \triangleq (\hat{\mathbf{Q}}_n(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu), \hat{\mathbf{Y}}_n(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu), \hat{\mathbf{f}}_n(\boldsymbol{\lambda}; \mathbf{f}_n^\nu))_{n=1}^{N_c}$ can be computed in parallel across the SCeNBs n:

$$\hat{\mathbf{Q}}_n(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu) \triangleq \underset{\mathbf{Q}_n \in \mathcal{Q}_n}{\operatorname{argmin}} \{ \mathcal{L}_{\mathbf{Q}_n}(\mathbf{Q}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Q}^\nu) \} \quad (40)$$

$$\hat{\mathbf{Y}}_n(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu) \triangleq \underset{(\mathbf{Y}_{i_n} \succeq \mathbf{0})_{i_n \in \mathcal{I}_n}}{\operatorname{argmin}} \{ \mathcal{L}_{\mathbf{Y}_n}(\mathbf{Y}_n, \boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{Y}^\nu) \} \quad (41)$$

$$\hat{\mathbf{f}}_n(\boldsymbol{\lambda}; \mathbf{f}_n^\nu) \triangleq \underset{\mathbf{f}_n \in \mathbb{R}_+^{K_n}}{\operatorname{argmin}} \{ \mathcal{L}_{\mathbf{f}_n}(\mathbf{f}_n, \boldsymbol{\lambda}; \mathbf{f}_n^\nu) \}. \quad (42)$$

Interestingly, problem (41) admits a closed form solution.

Lemma 2: Let $\mathbf{U}_{i_n}^H \mathbf{D}_{i_n} \mathbf{U}_{i_n}$ be the eigenvalue/eigenvector decomposition of $c_Y \mathbf{Y}_{i_n}^\nu - \boldsymbol{\Omega}_{i_n}$, with $\mathbf{D}_{i_n} = \operatorname{diag}((d_{i_n, j})_{j=1}^{n_{R_n}})$. The optimal solution of problem (41) is

$$\mathbf{Y}_{i_n} = \mathbf{U}_{i_n} \mathbf{D}_{\mathbf{Y}_{i_n}} \mathbf{U}_{i_n}^H \quad (43)$$

with $\mathbf{D}_{\mathbf{Y}_{i_n}} = \operatorname{diag}((y_{i_n, j})_{j=1}^{n_{R_n}})$ given by

$$y_{i_n, j} = \left[- \left(\frac{\sigma_w^2}{2} - \frac{d_{i_n, j}}{2c_Y} \right) + \sqrt{\left(\frac{\sigma_w^2}{2} + \frac{d_{i_n, j}}{2c_Y} \right)^2 + \frac{\lambda_{i_n}}{c_Y}} \right]^+.$$

□

Given $\hat{\mathbf{W}}(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu)$, the dual problem associated with $\tilde{\mathcal{P}}^\nu$ is

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}, (\boldsymbol{\Omega}_{i_n} \succeq \mathbf{0})_{i_n \in \mathcal{I}}} D(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu), \quad (44)$$

with $D(\boldsymbol{\lambda}, \boldsymbol{\Omega}; \mathbf{W}^\nu)$ defined in (39). It can be shown that the dual function is C^2 , with Hessian Lipschitz continuous with respect to \mathbf{W}^ν on \mathcal{X} . Then, the dual problem (44) can be solved using either first or second order methods. An instance of gradient-based schemes is given in Algorithm 4, whose convergence is guaranteed under the same conditions as in the Theorem 3. In Step 3 of the algorithm, the symbol $[\mathbf{A}]_+$ denotes the Euclidean projection of the square matrix \mathbf{A} onto the convex set of positive semidefinite matrices (having the same size of \mathbf{A}).

Algorithm 4. Distributed dual scheme solving $\tilde{\mathcal{P}}^\nu$

Initial data: $\boldsymbol{\lambda}^0 \geq \mathbf{0}$, $\boldsymbol{\Omega}^0 \succeq \mathbf{0}$, $\mathbf{W}^\nu = (\mathbf{Q}^\nu, \mathbf{Y}^\nu, \mathbf{f}^\nu)$, $\{\beta_k\}_k > 0$. Set $k = 0$,
(S.1): If $\boldsymbol{\lambda}^k, \boldsymbol{\Omega}^k$ satisfy a suitable termination criterion: STOP;
(S.2): For each SCeNB n , compute in parallel $\mathbf{Q}_n^{k+1}(\boldsymbol{\lambda}^k; \boldsymbol{\Omega}^k; \mathbf{W}^\nu)$, $\mathbf{Y}_n^{k+1}(\boldsymbol{\lambda}^k; \boldsymbol{\Omega}^k; \mathbf{W}^\nu)$ and $\mathbf{f}_n^{k+1}(\boldsymbol{\lambda}^k; \mathbf{W}^\nu)$ solving (40)-(42);
(S.3): Update at the master node $\boldsymbol{\lambda}$ and $\boldsymbol{\Omega}$ according to

$$\begin{aligned} \lambda_{i_n}^{k+1} &\triangleq [\lambda_{i_n}^k + \beta_k \tilde{g}_{i_n}(\mathbf{Q}_{i_n}^{k+1}, \mathbf{Q}_{-n}^{k+1}, f_{i_n}^{k+1}; \mathbf{Q}^\nu)]^+, \forall i_n \in \mathcal{I}_o, \\ \lambda_{i_n}^{k+1} &\triangleq [\lambda_{i_n}^k + \beta_k (\tilde{g}_{i_n}(\mathbf{Q}_{i_n}^{k+1}, \mathbf{Q}_{-n}^{k+1}, 0; \mathbf{Q}^\nu) + R_{i_n}^{\min})]^+, \forall i_n \in \mathcal{I}_{no}, \\ \lambda_f^{k+1} &\triangleq \left[\lambda_f^k + \beta_k \left(\sum_{i_n \in \mathcal{I}_o} f_{i_n}^{k+1} - f_T \right) \right]^+ \\ \boldsymbol{\Omega}_{i_n}^{k+1} &\triangleq [\boldsymbol{\Omega}_{i_n}^k + \beta_k (\mathbf{Y}_{i_n}^{k+1} - \mathbf{I}_{i_n}(\mathbf{Q}^{k+1}))]_+, \forall i_n \in \mathcal{I} \end{aligned}$$

(S.4): $k \leftarrow k + 1$ and go back to (S.1).

A faster algorithm solving the dual problem can be readily obtained using second order information. It is sufficient to replace the update of the multipliers in Step 3 of Algorithm 4 with the following (convergence is still guaranteed by Theorem 3):

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \beta_k ([\hat{\boldsymbol{\lambda}}^{k+1}]^+ - \boldsymbol{\lambda}^k), \\ \boldsymbol{\Omega}^{k+1} &= \boldsymbol{\Omega}^k + \beta_k ([\hat{\boldsymbol{\Omega}}^{k+1}]_+ - \boldsymbol{\Omega}^k) \end{aligned} \quad (45)$$

where, introducing the multipliers vector $\hat{\boldsymbol{\Lambda}} \triangleq [\hat{\boldsymbol{\lambda}}^{k+1}; \operatorname{vec}(\hat{\boldsymbol{\Omega}}^{k+1})]$ with $\operatorname{vec}(\hat{\boldsymbol{\Omega}}) \triangleq (\operatorname{vec}(\hat{\boldsymbol{\Omega}}_{i_n}))_{i_n \in \mathcal{I}}$, we have

$$\begin{aligned} \hat{\boldsymbol{\Lambda}}^{k+1} &\triangleq \hat{\boldsymbol{\Lambda}}^k + (\nabla_{\boldsymbol{\lambda}, \operatorname{vec}(\boldsymbol{\Omega}^*)}^2 D(\hat{\mathbf{W}}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\Omega}^k; \mathbf{W}^\nu))^{-1} \\ &\quad \cdot \nabla_{\boldsymbol{\lambda}, \operatorname{vec}(\boldsymbol{\Omega}^*)} D(\hat{\mathbf{W}}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\Omega}^k; \mathbf{W}^\nu). \end{aligned} \quad (46)$$

The explicit expression of the Hessian matrices and gradients in (46) are omitted because of space limitations. Numerical results show that using second order information significantly enhances practical convergence speed.

VI. NUMERICAL RESULTS

In this section we present some numerical results to assess the effectiveness of the proposed joint optimization of the communication and computational resources.

The simulated scenario is the following. We consider a network composed of $N_c = 2$ cells with $K_n = 6$ active users per cell, randomly deployed. We assume that for each cell 4 mobile users are asking for computational offloading while 2 MUs are only transmitting data. The path-loss coefficient is chosen according to the small cell model in [44] with carrier frequency equal to 2 GHz and noise power equal to $N_0 = -125$ dB. In all our experiments the system parameters are set as (unless stated otherwise): $f_T = 10^{10}$ cycles/s, $\mathbf{R}_w = N_0 \mathbf{I}_{n_r}$, $P_{i_n} = P_T = 34$ dBm. This choice guarantees the nonemptiness of the feasible set \mathcal{X} ; the constant $\bar{\alpha}$ in the diminishing step-size rule (27) is chosen as $\bar{\alpha} = 1e-4$, and the termination accuracy δ is set to 10^{-3} .

A. Example # 1: Joint vs. Disjoint Optimization

We start comparing the energy consumption of the proposed offloading strategy with a method where communication and computational resources are optimized separately. The benchmark used to assess the relative merits of our approach is an instance of Algorithm 2 wherein the computational rates f_{i_n} are not optimized but set proportional to the computational load of each user, while meeting the computational rate constraint f_T with equality, i.e., $f_{i_n} = w_{i_n} f_T / \sum_{i_n \in \mathcal{I}_o} w_{i_n}$ CPU cycles/second. We termed such a method *Disjoint Resource Allocation (DRA)* algorithm. Note that this algorithm is still guaranteed to converge by Theorem 2. An important parameter useful to assess the usefulness of offloading algorithms is the ratio $\eta_{i_n} := w_{i_n} / b_{i_n}$ between the computational load w_{i_n} to be transferred and the number of bits b_{i_n} enabling the transfer. Fig. 1 shows an example of overall energy consumption, assuming the same ratio $\eta_{i_n} := \eta$ for all users, obtained using Algorithm 2 and DRA algorithm. In particular, η is varied keeping a fixed work load w and changing the number b_{i_n} of bits to be sent. The radio channels are Rayleigh fading and the results are averages over 100 independent channel realizations. Fig. 1 shows a few interesting features: i) the joint optimization yields a considerable gain with respect to the disjoint optimization for applications having a low ratio η , i.e., applications with a high number of bits to be transferred, for a given computational load w ; ii) the overall energy consumption decreases for computationally intensive applications, i.e., applications characterized by a high η .

B. Example # 2: On the Convergence Speed

To test the convergence speed of Algorithm 2, Fig. 2 shows the average energy consumption $E(\mathbf{Q}^\nu)$ versus the iteration

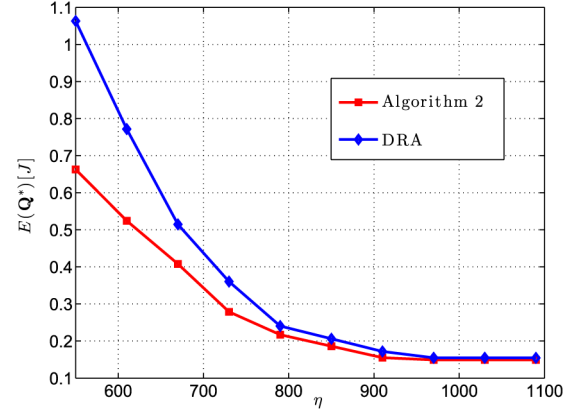


Fig. 1. Energy consumption versus $\eta = w_{i_n}/b_{i_n}$ for Algorithm 2 and for DRA.

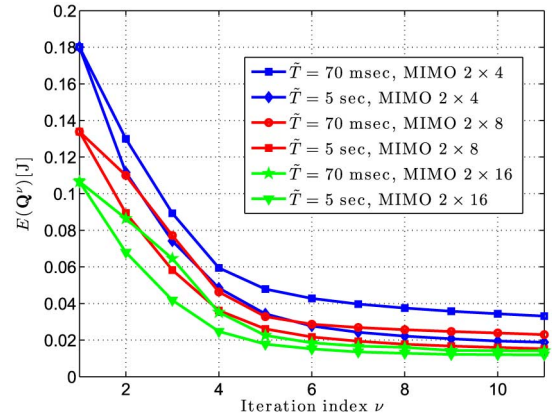


Fig. 2. Convergence speed: Optimal energy versus the iteration index for different values of \tilde{T} .

index ν , for different values of the maximum latency \tilde{T}_{i_n} (assumed to be equal for all users) and different number of receive antennas. The curves are averaged over 100 independent channel realizations. The interesting result is that the proposed algorithm converges in very few iterations. Moreover, as expected, the energy consumption increases as the delay constraint becomes more stringent because more transmit energy has to be used to respect the latency limit. Finally, it is worth noticing the gain achievable by increasing the number of receive antennas.

Since the overall optimization problem is non-convex, the proposed algorithm may fall into a local minimum. To evaluate this aspect, we ran our algorithm under 1,000 independent initializations of the initial parameter setting $\mathbf{Z}^0 = (\mathbf{Q}^0, \mathbf{f}^0) \in \mathcal{X}$ of Algorithm 2 and in Fig. 3 we plot the histogram of the energies at the initial (before running our optimization algorithm) energies $E(\mathbf{Q}^0)$ (left plot), and the histogram of the energies resulting from running Algorithm 2 (right plot). Quite interestingly, our experiments show that the proposed algorithm is robust against random initializations: Although the variance of the initial energy is quite large, the optimal final energy tends to be concentrated around a much smaller range of values (that have been observed to differ from each other on the third decimal digit).

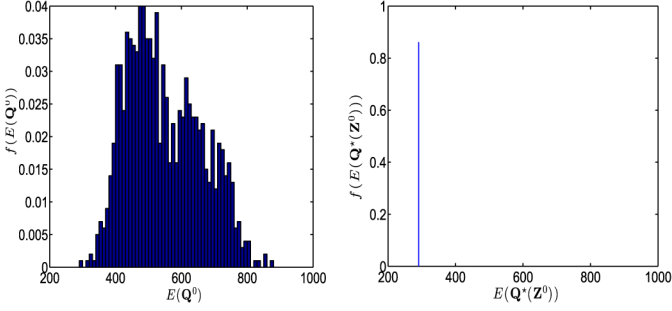


Fig. 3. Probability density function of the initial energy $E(\mathbf{Q}^0)$ (left plot) and of optimal final energy $E(\mathbf{Q}^*(\mathbf{Z}^0))$ (right plot).

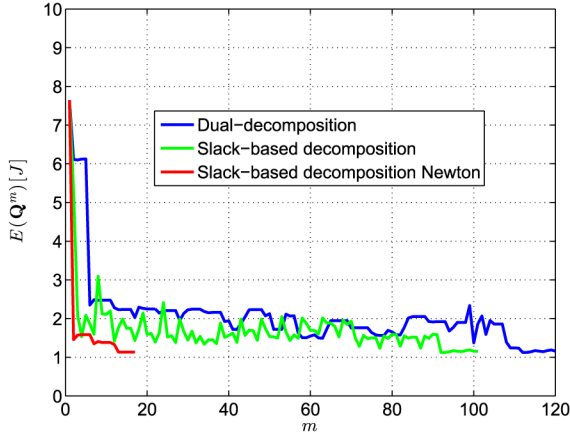


Fig. 4. Evolution of the global energy for the distributed algorithms versus the iteration index m .

C. Example # 3: Distributed Algorithms

Finally, we tested the efficiency of the distributed algorithms proposed in Section V. We assume $\bar{\alpha} = 1e-5$ and the termination accuracy δ is set to 10^{-2} . Fig. 4 shows the energy evolution versus the iteration index m , which counts the overall number of (inner and outer) iterations in Algorithm 2. More specifically, we compared three different algorithms used to run Step 2, namely: the dual-decomposition method described in Algorithm 3, the dual-scheme based on the reformulation of the nonconvex problem \mathcal{P} using slack-variables as given in Algorithm 4, and its accelerated version based on the Newton implementation (45). All implementations are quite fast. As expected, using second order information enhances convergence speed.

VII. CONCLUSION

In this paper we formulated the computation offloading problem in a multi-cell mobile edge-computing scenario, where a dense deployment of radio access points facilitates proximity high bandwidth access to computational resources, but increases also intercell interference. We formulated the resource optimization problem as the joint optimization of radio and computational resources, aimed at minimizing MUs' energy consumption, under latency and power budget constraints. In the single-user case, we computed the global optimal solution of the resulting nonconvex optimization problem in

closed form. In the more general multi-cell multi-user scenario, we developed centralized and distributed SCA-based algorithms with provable convergence to local optimal solutions of the nonconvex problem. Numerical results show that our algorithms outperform disjoint optimization schemes. Furthermore, the results show, as expected, that offloading is more convenient for applications with high computational load and small number of bits to be exchanged to enable program migration. In this paper, we focused on a static framework where a cloud manager examines a set of requests and, after exchanging information with the base stations involved in the radio access, derives the optimal allocation of radio and computational resources. An interesting extension of this approach may incorporate a dynamic setting where battery levels, application parameters, channel states evolve over time, depending on how the users are being served over time, with the goal of finding a proper scheduling mechanism. Furthermore, while we concentrated here on a single cloud, extension to multiple distributed clouds might be an important generalization.

APPENDIX

A. Proof of Theorem 1

(a) It is sufficient to prove the following two facts.

Fact 1: Any stationary point of the nonconvex problem \mathcal{P}_s is a *global* optimal solution of the problem.

Fact 2: Any stationary point of the *convex* problem \mathcal{Q}_s (and thus a globally optimal solution to \mathcal{Q}_s), is also a stationary point of \mathcal{P}_s , and viceversa.

Proof of Fact 1: Invoking [45, Theorem 3.39], it is sufficient to show that the objective function $E(\mathbf{Q})$ is a pseudo-convex function on the convex set \mathcal{X}_s , i.e., [45, Def. 3.1.3]

$$\forall \mathbf{Q}, \mathbf{Y} \in \mathcal{X}_s : E(\mathbf{Q}) < E(\mathbf{Y}) \Rightarrow \langle \nabla_{\mathbf{Q}^*} E(\mathbf{Y}), \mathbf{Q} - \mathbf{Y} \rangle < 0. \quad (47)$$

Fix $\mathbf{Y} \in \mathcal{X}_s$, and introduce the *convex* \mathcal{C}^1 function $\phi_{\mathbf{Y}} : \mathcal{X}_s \rightarrow \mathbb{R}$ defined as

$$\phi_{\mathbf{Y}}(\mathbf{Q}) \triangleq \text{tr}(\mathbf{Q}) \cdot r(\mathbf{Y}) - \text{tr}(\mathbf{Y}) \cdot r(\mathbf{Q}). \quad (48)$$

Then, for any $\mathbf{Q} \in \mathcal{X}_s$ such that $E(\mathbf{Q}) < E(\mathbf{Y})$, the following holds:

$$\begin{aligned} \langle \nabla_{\mathbf{Q}^*} E(\mathbf{Y}), \mathbf{Q} - \mathbf{Y} \rangle &\stackrel{(a)}{=} \frac{\langle \nabla_{\mathbf{Q}^*} \phi_{\mathbf{Y}}(\mathbf{Y}), \mathbf{Q} - \mathbf{Y} \rangle}{r(\mathbf{Y})^2} \\ &\stackrel{(b)}{\leq} \frac{\phi_{\mathbf{Y}}(\mathbf{Q}) - \phi_{\mathbf{Y}}(\mathbf{Y})}{r(\mathbf{Y})^2} \stackrel{(c)}{<} 0, \end{aligned} \quad (49)$$

where (a) follows from the definition of $\phi_{\mathbf{Y}}$ in (48); (b) is due to the convexity of $\phi_{\mathbf{Y}}$ on \mathcal{X}_s ; and (c) comes from $E(\mathbf{Q}) < E(\mathbf{Y}) \Rightarrow \phi_{\mathbf{Y}}(\mathbf{Q}) < \phi_{\mathbf{Y}}(\mathbf{Y})$. Since (49) holds for any given $\mathbf{Y} \in \mathcal{X}_s$, (47) holds true. \square

Proof of Fact 2: Let us prove the two directions separately. $\mathcal{Q}_s \Rightarrow \mathcal{P}_s$: Let (\mathbf{Q}^*, f^*) be the optimal solution of the convex problem \mathcal{Q}_s ; denote $\tilde{\mathbf{Q}}^* \triangleq \mathbf{U}^H \mathbf{Q}^* \mathbf{U}$. Then, there exist multipliers $\lambda_p^*, \mu_p^*, \alpha_p^*, \Phi_p^*$ such that the tuple $(\tilde{\mathbf{Q}}^*, f^*, \lambda_p^*, \mu_p^*, \alpha_p^*, \beta_p^*, \Phi_p^*)$ satisfies the KKT conditions of \mathcal{Q}_s (note that Slater's constraint qualification is satisfied):

denoting $\tilde{r}(\tilde{\mathbf{Q}}^*) \triangleq \log_2 |\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2}|$, and after some simplifications, one gets

$$\begin{aligned}
 \text{(a):} \quad & \mathbf{I} - \frac{\mu_p^*}{\log(2)} \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} \\
 & \quad \quad \quad + \lambda_p^* \mathbf{I} - \Phi_p^* = \mathbf{0} \\
 \text{(b):} \quad & \frac{\mu_p^* w c}{f^{*2} (\tilde{T} - w/f^*)^2} - \alpha_p^* = 0 \\
 \text{(c):} \quad & 0 \leq \lambda_p^* \perp (P_T - \text{tr}(\tilde{\mathbf{Q}}^*)) \geq 0 \\
 \text{(d):} \quad & 0 < \mu_p^*, \quad \frac{c}{\tilde{T} - w/f^*} - \tilde{r}(\tilde{\mathbf{Q}}^*) = 0 \\
 \text{(e):} \quad & \mathbf{0} \preceq \tilde{\mathbf{Q}}^* \perp \Phi_p^* \succeq \mathbf{0} \\
 \text{(f):} \quad & 0 \leq \alpha_p^*, \quad f^* = f_T, \quad (\text{KKT}_{\mathcal{Q}_s})
 \end{aligned}$$

where $\mathbf{A} \perp \mathbf{B}$ stands for $\langle \mathbf{A}, \mathbf{B} \rangle = 0$, and in (d) and (f) we used the fact that μ_p^* must be positive and $f^* = f_T$, respectively (otherwise $\text{KKT}_{\mathcal{Q}_s}$ cannot be satisfied). We prove next that there exist multipliers $\lambda_e^*, \mu_e^*, \alpha_e^*, \Phi_e^*$ that together with the optimal solution $(\tilde{\mathbf{Q}}^*, f^*)$ of \mathcal{Q}_s satisfy the KKT conditions of \mathcal{P}_s , i.e.,

$$\begin{aligned}
 \text{(a')} : \quad & \frac{c \cdot \mathbf{I}}{\tilde{r}(\tilde{\mathbf{Q}}^*)} - \frac{c \cdot \text{tr}(\tilde{\mathbf{Q}}^*) \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2}}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2 \log(2)} \\
 & - \frac{\mu_e^*}{\log(2)} \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} + \lambda_e^* \mathbf{I} - \Phi_e^* = \mathbf{0} \\
 \text{(b')} : \quad & \frac{\mu_e^* w c}{f^{*2} (\tilde{T} - w/f^*)^2} - \alpha_e^* = 0 \\
 \text{(c')} : \quad & 0 \leq \lambda_e^* \perp (P_T - \text{tr}(\tilde{\mathbf{Q}}^*)) \geq 0 \\
 \text{(d')} : \quad & 0 \leq \mu_e^* \perp \left(\tilde{r}(\tilde{\mathbf{Q}}^*) - \frac{c}{\tilde{T} - w/f^*} \right) \geq 0 \\
 \text{(e')} : \quad & \mathbf{0} \preceq \tilde{\mathbf{Q}} \perp \Phi_e^* \succeq \mathbf{0} \\
 \text{(f')} : \quad & 0 \leq \alpha_e^* \perp (f_T - f^*) \geq 0. \quad (\text{KKT}_{\mathcal{P}_s})
 \end{aligned}$$

Plugging (a) of $(\text{KKT}_{\mathcal{Q}_s})$ in (a') of $(\text{KKT}_{\mathcal{P}_s})$ and using the fact that $\mu_p^* > 0$, we obtain:

$$\begin{aligned}
 \lambda_e^* \mathbf{I} = & -\frac{c \mathbf{I}}{\tilde{r}(\tilde{\mathbf{Q}}^*)} + \frac{(1 + \lambda_p^*)}{\mu_p^*} \left(\frac{c \text{tr}(\tilde{\mathbf{Q}}^*)}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} + \mu_e^* \right) \cdot \mathbf{I} \\
 & + \Phi_e^* - \frac{1}{\mu_p^*} \left(\frac{c \text{tr}(\tilde{\mathbf{Q}}^*)}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} + \mu_e^* \right) \cdot \Phi_p^*, \quad (50)
 \end{aligned}$$

which is satisfied if one set Φ_e^*, λ_e^* , and μ_e^* to

$$\begin{aligned}
 \Phi_e^* & \triangleq \frac{1}{\mu_p^*} \left(\frac{c \text{tr}(\tilde{\mathbf{Q}}^*)}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} + \mu_e^* \right) \cdot \Phi_p^* \\
 \mu_e^* & \triangleq \frac{c \mu_p^*}{\tilde{r}(\tilde{\mathbf{Q}}^*) (1 + \lambda_p^*)} - \frac{c \text{tr}(\tilde{\mathbf{Q}}^*)}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} \\
 \lambda_e^* & \triangleq 0. \quad (51)
 \end{aligned}$$

By (b') it must be

$$\alpha_e^* = \frac{\mu_e^* w c}{f^{*2} (\tilde{T} - w/f^*)^2}. \quad (52)$$

Note that, to be a valid candidate solution of $\text{KKT}_{\mathcal{P}_s}$, μ_e^* must be nonnegative [cf. (d')], which by (51), is equivalent to

$$\frac{1 + \lambda_p^*}{\mu_p^*} \cdot \text{tr}(\tilde{\mathbf{Q}}^*) \leq \tilde{r}(\tilde{\mathbf{Q}}^*). \quad (53)$$

We show next that (53) holds true. By multiplying both sides of (a) by $\tilde{\mathbf{Q}}^*$ and using the complementarity condition $\langle \Phi_p^*, \tilde{\mathbf{Q}}^* \rangle = 0$ [cf. (e)] we get

$$\begin{aligned}
 \frac{1 + \lambda_p^*}{\mu_p^*} \cdot \text{tr}(\tilde{\mathbf{Q}}^*) & = \frac{1}{\log(2)} \langle \tilde{\mathbf{Q}}^*, \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} \rangle \\
 & = \langle \nabla_{\mathbf{Q}^*} \tilde{r}(\tilde{\mathbf{Q}}^*), \tilde{\mathbf{Q}}^* \rangle \leq \tilde{r}(\tilde{\mathbf{Q}}^*), \quad (54)
 \end{aligned}$$

where in the last inequality we used the concavity of the rate function $\tilde{r}(\bullet)$, i.e.,

$$\tilde{r}(\mathbf{Y}) \leq \tilde{r}(\mathbf{W}) + \langle \nabla_{\mathbf{Q}^*} \tilde{r}(\mathbf{W}), \mathbf{Y} - \mathbf{W} \rangle, \quad \forall \mathbf{Y}, \mathbf{W} \succeq \mathbf{0} \quad (55)$$

evaluated at $\mathbf{Y} = \mathbf{0}$ and $\mathbf{W} = \tilde{\mathbf{Q}}^*$. The desired result, $\mu_e^* \geq 0$, follows readily combining (53) and (54).

We show now that the obtained tuple $(\tilde{\mathbf{Q}}^*, f^*, \lambda_e^*, \mu_e^*, \alpha_e^*, \Phi_e^*)$ satisfies $\text{KKT}_{\mathcal{P}_s}$. Indeed, (a') follows from (51); given $\mu_e^* \geq 0$, (b') is satisfied by α_e^* as in (52); (c') follows from $P_T - \text{tr}(\tilde{\mathbf{Q}}^*) \geq 0$ [cf. (c)] and $\lambda_e^* = 0$; (d') follows from $\mu_e^* \geq 0$ and the second equality in (d). Finally, it is not difficult to see that Φ_e^* given by (51) satisfies (e'); and finally (f') is trivially met by $\alpha_e^* \geq 0$ in (52). This completes the first part of the proof.

$\mathcal{P}_s \Rightarrow \mathcal{Q}_s$: the proof follows the same idea as for $\mathcal{Q}_s \Rightarrow \mathcal{P}_s$; we then only sketch the main steps. Let $(\tilde{\mathbf{Q}}^*, f^*, \lambda_e^*, \mu_e^*, \alpha_e^*, \Phi_e^*)$ be a tuple satisfying $\text{KKT}_{\mathcal{P}_s}$ (whose existence is guaranteed by the Slater's constraint qualification). We prove next that there exist multipliers $(\lambda_p^*, \mu_p^*, \alpha_p^*, \Phi_p^*)$ such that $(\tilde{\mathbf{Q}}^*, f^*, \lambda_p^*, \mu_p^*, \alpha_p^*, \Phi_p^*)$ satisfies $\text{KKT}_{\mathcal{Q}_s}$. Define

$$\kappa_e = \mu_e^* + \frac{c \text{tr}(\tilde{\mathbf{Q}}^*)}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} > 0.$$

Given (a'), it can be easily seen that (a) is satisfied if Φ_p^*, λ_p^* , and μ_p^* are chosen as

$$\Phi_p^* = \frac{\mu_p^*}{\kappa_e} \Phi_e^*, \quad \mu_p^* = \frac{\kappa_e}{\lambda_e^* + \frac{c}{\tilde{r}(\tilde{\mathbf{Q}}^*)}}, \quad \text{and} \quad \lambda_p^* = 0. \quad (56)$$

From (b) it must also be

$$\alpha_p^* = \frac{\mu_p^* w c}{f^{*2} (\tilde{T} - w/f^*)^2}. \quad (57)$$

It is not difficult to check that the obtained tuple $(\tilde{\mathbf{Q}}^*, f^*, \lambda_p^*, \mu_p^*, \alpha_p^*, \Phi_p^*)$ satisfies (a), (b), (c), (e), and (f) of $\text{KKT}_{\mathcal{Q}_s}$; the only condition that needs a proof is the equality constraint in (d), as given next.

Suppose by contradiction that $\tilde{r}(\tilde{\mathbf{Q}}^*) - \frac{c}{\tilde{T} - w/f^*} > 0$. Then, it follows from (d') that $\mu_e^* = 0$, and (a') reduces to

$$\frac{c\mathbf{I}}{\tilde{r}(\tilde{\mathbf{Q}}^*)} - \frac{c \operatorname{tr}(\tilde{\mathbf{Q}}^*) \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2}}{\log(2) \tilde{r}(\tilde{\mathbf{Q}}^*)^2} = -\lambda_e^* \mathbf{I} + \Phi_e^*.$$

Multiplying the above equation by $\tilde{\mathbf{Q}}^*$ and using the complementary condition (e'), we get

$$\lambda_e^* = \frac{c}{\tilde{r}(\tilde{\mathbf{Q}}^*)^2} \left(\langle \nabla_{\mathbf{Q}^*} \tilde{r}(\tilde{\mathbf{Q}}^*), \tilde{\mathbf{Q}}^* \rangle - r(\tilde{\mathbf{Q}}^*) \right), \quad (58)$$

which, given $\lambda_e^* \geq 0$ [cf. (c')] and $\langle \nabla_{\mathbf{Q}^*} \tilde{r}(\tilde{\mathbf{Q}}^*), \tilde{\mathbf{Q}}^* \rangle \leq \tilde{r}(\tilde{\mathbf{Q}}^*)$ [due to (55)], can be satisfied only if $\langle \nabla_{\mathbf{Q}^*} \tilde{r}(\tilde{\mathbf{Q}}^*), \tilde{\mathbf{Q}}^* \rangle = r(\tilde{\mathbf{Q}}^*)$, i.e.,

$$\log_2 \det(\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2}) = \operatorname{tr} \left(\tilde{\mathbf{Q}}^* \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \cdot \mathbf{D}^{1/2} \right) \cdot \frac{1}{\log(2)}.$$

Denoting by $(\sigma_i = \sigma_i(\mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2}))_{i=1}^r \geq 0$ the non-negative eigenvalues of $\mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2}$, the above equality can be rewritten as

$$\sum_{i=1}^r \log(1 + \sigma_i) = \sum_{i=1}^r \frac{\sigma_i}{1 + \sigma_i},$$

which can be true only if $\sigma_i = 0$ for all $i = 1, \dots, r$, and thus $\tilde{\mathbf{Q}}^* = \mathbf{0}$ (note that $\mathbf{D} \neq \mathbf{0}$). This however is in contradiction with the fact that \mathbf{Q}^* is an optimal solution of \mathcal{Q}_s . (b): Invoking part (a) of the theorem, the solution (\mathbf{Q}^*, f^*) of \mathcal{Q}_s (and thus \mathcal{P}_s) can be computed solving KKT $_{\mathcal{Q}_s}$. Denote $\tilde{\mathbf{Q}}^* \triangleq \mathbf{U}^H \mathbf{Q}^* \mathbf{U}$. Multiplying (a) of KKT $_{\mathcal{Q}_s}$ by $\tilde{\mathbf{Q}}^*$ and using (e), we get

$$\mathbf{I} - \alpha \mathbf{D}^{1/2} (\mathbf{I} + \mathbf{D}^{1/2} \tilde{\mathbf{Q}}^* \mathbf{D}^{1/2})^{-1} \mathbf{D}^{1/2} = \mathbf{0} \quad (59)$$

with $\alpha \triangleq \mu_p^* / \log(2)$ (recall that one can set $\lambda_p^* = 0$). By solving (59) and using $\tilde{\mathbf{Q}}^* \triangleq \mathbf{U}^H \mathbf{Q}^* \mathbf{U}$ one obtains the desired expression of \mathbf{Q}^* as in (12). Moreover, it follows from (f) that $f^* = f_T$. The only thing left to show is how to compute α (and thus μ_p^*) efficiently. Using the optimal structure of \mathbf{Q}^* and denoting $r_e \triangleq \operatorname{rank}(\mathbf{Q}^*)$, conditions (c) and (d) reduce respectively to

$$\alpha = 2 \frac{c}{r_e L} - \frac{1}{r_e} \sum_{i=1}^{r_e} \log_2(d_i) \quad \text{and} \quad \sum_{i=1}^{r_e} \left(\alpha - \frac{1}{d_i} \right) \leq P_T, \quad (60)$$

with $L = \tilde{T} - \frac{w}{f_T}$. Note that Slater's constraint qualification guarantees that there exist α and r_e satisfying (60). Moreover, it is not difficult to check that they can be efficiently computed using the procedure described in Algorithm 1.

REFERENCES

- [1] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell MIMO mobile cloud computing," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC '14)*, Toronto, ON, Canada, Jun. 22–25, 2014, pp. 354–358.
- [2] S. Sardellitti, G. Scutari, and S. Barbarossa, "Distributed joint optimization of radio and computational resources for mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Netw. (Cloudnet'14)*, Luxembourg, Oct. 8–10, 2014, pp. 211–216.
- [3] M. Palacin, "Recent advances in rechargeable battery materials: A chemists perspective," *Chem. Soc. Rev.*, vol. 38, no. 9, pp. 2565–2575, Sep. 2009.
- [4] M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1232–1243, Fourth Quarter 2012.
- [5] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.
- [6] N. Fernando, S. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, pp. 84–106, Jan. 2013.
- [7] K. Yang, S. Ou, and 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.
- [8] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, Apr. 2008, pp. 1–8.
- [9] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *J. Mobile Netw. Appl.*, vol. 16, no. 3, pp. 270–284, Jun. 2011.
- [10] N. Kaushi and J. Kumar, "A computation offloading framework to optimize energy utilisation in mobile cloud computing environment," *Int. J. Comput. Appl. Inf. Technol.*, vol. 5, no. 2, pp. 61–69, Apr/May 2014.
- [11] V. Cardellini *et al.*, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, pp. 1–29, Apr. 2015, ISSN 0025-5610.
- [12] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *IEEE Comput. Soc.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.
- [13] Y. Cui, X. Ma, H. Wang, I. Stojmenovic, and J. Liu, "A survey of energy efficient wireless transmission and modeling in mobile cloud computing," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 148–155, Feb. 2013.
- [14] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, Jun. 2010, p. 4.
- [15] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Apr. 2012.
- [16] B. K. Yi-Hsuan Kao, "Optimizing mobile computational offloading with delay constraints," in *Proc. Global Commun. Conf. (GLOBECOM'14)*, Austin, TX, USA, Dec. 8–12, 2014, pp. 49–62.
- [17] F. Liu *et al.*, "Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 14–22, Jun. 2013.
- [18] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 369–392, Feb. 2014.
- [19] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM Int. Conf. Mobile Syst. Appl. Serv.*, San Francisco, CA, USA, Jun. 15–18, 2010, pp. 49–62.
- [20] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.
- [21] F. Xia, F. Ding, J. Li, X. Kong, L. Yang, and J. Ma, "Phone2cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing," *Inf. Syst. Front.*, vol. 16, pp. 95–111, Mar. 2014.
- [22] 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. 2716–2720, Sep. 2013.
- [23] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Computation offloading for mobile cloud computing based on wide cross-layer optimization," in *Proc. Future Netw. Mobile Summit (FuNeMS'13)*, Lisboa, Portugal, Jul. 3–5, 2013, pp. 1–10.
- [24] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC'13)*, Darmstadt, Germany, Jun. 16–19, 2013, pp. 26–30.

- [25] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.
- [26] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
- [27] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
- [28] FP7 European Project. (2012). Distributed Computing, Storage and Radio Resource Allocation over Cooperative Femtocells (TROPIC) [Online]. Available: <http://www.ict-tropic.eu>
- [29] ETSI. (2014). *ETSI First Meeting of New Standardization Group on Mobile-edge Computing* [Online]. Available: <http://www.etsi.org/news-events/news/838-2014-10-news-etsi-announces-first-meeting-of-new-standardization-group-on-mobile-edge-computing>
- [30] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Parallel and distributed methods for nonconvex optimization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP'14)*, Florence, Italy, May 4–9, 2014, pp. 840–844.
- [31] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Distributed methods for nonconvex constraints multi-agent problems – Part I: Theory," *IEEE Trans. Signal Process.*, submitted for publication, Oct. 2014. [Online]. Available: <http://arxiv.org/abs/1410.4754>
- [32] G. Scutari, F. Facchinei, J.-S. Pang, and D. P. Palomar, "Real and complex monotone communication games," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4197–4231, Jul. 2014.
- [33] G. Scutari, F. Facchinei, P. Song, D. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 641–656, Feb. 2014.
- [34] H. A. L. Thi and P. Tao, "The DC programming and DCA revised with DC models of real world nonconvex optimization problems," *Ann. Oper. Res.*, vol. 133, no. 1–4, pp. 23–46, Jan. 2005.
- [35] A. Alvarado, G. Scutari, and J. Pang, "A new decomposition method for multiuser DC-programming and its applications to physical layer security," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2984–2998, Jun. 2014.
- [36] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, 2004.
- [37] B. K. Sriperumbudur and G. R. G. Lanckriet, "A proof of convergence of the concave-convex procedure using Zangwill's theory," *Neural Comput.*, vol. 21, pp. 1391–1407, Jun. 2012.
- [38] S. Boyd. (2013). *Sequential Convex Programming, Lecture Note* [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf
- [39] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Opt.*, vol. 23, no. 2, pp. 1126–1153, 2013.
- [40] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874–1889, Apr. 2015.
- [41] M. Patriksson, "Partial linearization methods in nonlinear programming," *J. Optim. Theory Appl.*, vol. 78, no. 2, pp. 227–246, Aug. 1993.
- [42] J. Bolte, S. Shoham, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1–2, pp. 459–494, Aug. 2014.
- [43] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problem*. Berlin, Germany: Springer-Verlag, 2003.
- [44] 3GPP TR 36.814, "Technical specification group radio access network; Further advancements for E-UTRA, physical layer aspects," Release 9, v.9.0.0., Mar. 2010.
- [45] M. Avriel, W. E. Diewert, S. Schaible, and I. Zang, *Generalized Concavity*. Philadelphia, PA, USA: SIAM, 2010.



Stefania Sardellitti (M'12) received the M.Sc. degree in electronic engineering from the University of Rome "La Sapienza," Rome, Italy, in 1998, and the Ph.D. degree in electrical and information engineering from the University of Cassino, Italy, in 2005. Since 2005, she has been an Appointed Professor of Digital Communications with the University of Cassino, Cassino, Italy. She is currently a Research Assistant with the Department of Information, Electronics, and Telecommunications, University of Rome, Sapienza, Italy. She is currently involved in

the European project TROPIC, on distributed computing, storage and radio resource allocation over cooperative femtocells. She has participated in the European project WINSOC (on wireless sensor networks) and in the European project FREEDOM (on femtocell networks). Her research interests include statistical signal processing, mobile cloud computing, femtocell networks, and wireless sensor networks, with emphasis on distributed optimization. She was the recipient of the 2014 IEEE Best Paper Award from the IEEE Signal Processing Society.



Gesualdo Scutari (S'05–M'06–SM'11) received the Electrical Engineering and Ph.D. degrees (both with Hons.) from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2005, respectively. He is an Associate Professor with the Department of Industrial Engineering, Purdue University, West Lafayette, IN, USA. He had previously held several research appointments, namely, at the University of California at Berkeley, Berkeley, CA, USA; Hong Kong University of Science and Technology, Hong Kong; University of Rome, "La Sapienza," Rome, Italy; University of Illinois at Urbana-Champaign, Urbana, IL, USA. His research interests include theoretical and algorithmic issues related to big data optimization, equilibrium programming, and their applications to signal processing, medical imaging, machine learning, and networking. Dr. Scutari is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and he served as an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS. He serves on the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications (SPCOM). He was the recipient of the 2006 Best Student Paper Award at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2006, the 2013 NSF Faculty Early Career Development (CAREER) Award, and the 2013 UB Young Investigator Award.



Sergio Barbarossa (S'84–M'88–F'12) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Rome "La Sapienza," Rome, Italy, in 1984 and 1988, respectively. He has held positions as a Research Engineer with Selenia SpA (1984–1986) and with the Environmental Institute of Michigan (1988), as a Visiting Professor at the University of Virginia, Charlottesville, VA, USA (1995 and 1997) and at the University of Minnesota, Minneapolis, MN, USA (1999). Currently, he is a Full Professor with the University of Rome "La Sapienza."

He was an IEEE Distinguished Lecturer from the Signal Processing Society in 2012–2013. He is the author of a research monograph titled *Multiantenna Wireless Communication Systems*. He has been the Scientific Coordinator of various European projects on wireless sensor networks, femtocell networks, and mobile cloud computing. His research interests include signal processing for self-organizing networks, mobile-edge computing, signal processing over graphs, and distributed optimization algorithms. Dr. Barbarossa is an EURASIP Fellow. From 1997 to 2003, he was a member of the IEEE Technical Committee for Signal Processing in Communications. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1998–2000 and 2004–2006) and the IEEE SIGNAL PROCESSING MAGAZINE. He is currently an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He has been the General Chairman of the IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2003 and the Technical Co-Chair of SPAWC, 2013. He has been the Guest Editor for Special Issues on the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *EURASIP Journal of Applied Signal Processing*, *EURASIP Journal on Wireless Communications and Networking*, the IEEE SIGNAL PROCESSING MAGAZINE, and the IEEE SELECTED TOPICS ON SIGNAL PROCESSING. He was the recipient of the 2010 EURASIP Technical Achievements Award and the 2000 and 2014 IEEE Best Paper Awards from the IEEE Signal Processing Society. He is the coauthor of papers that received the Best Student Paper Award at ICASSP 2006, SPAWC 2010, EUSIPCO 2011, and CAMSAP 2011.