Joint Optimization of Radio and Computational Resources for Multicell Mobile Cloud Computing

S. Sardellitti,[1] G. Scutari,[2] and S. Barbarossa[1]
[1]Dept. of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy
[2]Dept. of Electrical Engineering, State University of New York at Buffalo, Buffalo, USA
E-mail: stefania.sardellitti@uniroma1.it, gesualdo@buffalo.edu, sergio@infocom.uniroma1.it.

***Abstract*—We consider a MIMO multicell system wherein several Mobile Users (MUs) ask for computation offloading to a common cloud server through their femto-access points. We formulate the computation offloading problem as a *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 the latency constraints imposed by the applications running on the MUs. The resulting optimization problem is nonconvex (in the objective function and the constraints), and there are constraints coupling all the optimization variables. To cope with the nonconvexity, we hinge on successive convex approximation techniques and propose an iterative algorithm converging to a local optimal solution of the original nonconvex problem. The algorithm is also suitable for a parallel implementation across the access point, with limited coordination/signaling with the cloud. Numerical results show that the proposed joint optimization yields significant energy savings with respect to more traditional schemes performing a separate optimization of the radio and computational resources.**

Keywords*—Computation offloading, cloud computing, resource allocation, successive convex approximation.

I. Introduction

Mobile terminals, such as smartphones, tablets and netbooks, are increasingly penetrating into our everyday lives as the most convenient tools for communication, entertainment, business, social networking, news, etc. [1]. Many current applications running on a mobile handset are severely energy demanding, such as video gaming, internet video streaming, location-based social networking, just to name a few. Despite the fast developments of key components such as CPU, memory, and wireless access technologies, one of the major impediment to run sophisticated applications on mobile handsets remains their limited battery lifetime [2], [3]. A way to overcome this obstacle is to enable smart mobile devices to offload their most energy-consuming tasks to nearby more resourceful servers. This strategy is known in literature under different names, such as *cyber foraging* [4] and *computation offloading* [5]. Several offloading methods have been proposed in the computer science literature, mainly focusing on computational aspects (e.g., program/code partitioning, static versus dynamic offloading, virtualization, etc.); examples are MAUI [6], and *ThinkAir* [7]. The major limitations of current mobile cloud computing [8] strategies over cellular networks are the mobile energy consumption associated with the radio access as well as the latency experienced in reaching the cloud server via radio and wide area network accesses. A possible way to alleviate this issue is to bring computational resources closer to MUs through the so-called cloudlets [9], [10]. The idea is to endow small-cell base stations with enhanced cloud functionalities: MUs will thus be able to find within a short distance a radio access point while having access also to computational and storage resources.

In this envisioned scenario it is clear that the effectiveness of the offloading depends on *both* radio access and computational aspects; this calls for a *joint* optimization of the radio and computational resources. In [11] the authors considered a SISO (flat-fading) interference-free small-cell network, and proposed a method to jointly optimize the transmit power, the number of bits per symbol, and the CPU cycles assigned to each application in order to minimize the power consumption at the mobile side, under an average latency constraint. In this paper we consider a more general formulation and realistic scenario, a MIMO multi-cell network wherein MUs of different cells may interfere against each other. We formulate the offloading problem as a *joint* optimization of the mobile radio resources−the transmit convariance matrices of the MUs−and the computational resources−the CPU cycles/second assigned by the cloud to each MU. The objective is to minimize the overall energy consumption at the mobile sides, under power budget and application-dependent latency constraints (the latter constraint is what couples the computation and communication optimization variables). Differently from [11], the resulting optimization problem is nonconvex in both the objective function and constraints. Building on recent Successive Convex Approximation (SCA) techniques [12]–[14] and exploting the structure of the problem, we propose an iterative algorithm with provable convergence to local optimal solutions of the nonconvex problem. The algorithm is also suitable for a distributed implementation across the cells, with limited signaling between the cells and the cloud server. Numerical results show that the algorithm converges in a few iteration to very "good" solutions, yielding a significant saving on the mobile energy consumption with respect to more classical schemes performing a separate optimization of the radio and computational resources.

II. Offloading over femtocloud networks

Let us consider a femto-cell cloud network composed of N_c femto-cells, each of them served by one Small Cell enhanced Node B (SCeNB in LTE terminology), and K_n MUs in each cell $n = 1, \ldots, N_c$. We denote by i_n the i-th user in the cell n, and by $\mathcal{I} \triangleq \{i_n : i = 1, \ldots, K_n, n = 1, \ldots, N_c\}$ 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 femto-cells are connected to a common cloud provider that is able to serve concurrently multiple users accessing through the associated femto-cell. We assume that MUs in the same cell

The work of Barbarossa and Sardellitti was funded by the European Community 7th Framework Programme Project ICT-TROPIC, under grant no. 318784. The work of Scutari was supported by the USA NSF under Grants CMS 1218717 and CAREER Award no. 1254739.

978-1-4799-4903-8/14/$31.00 ©2014 IEEE

transmit over orthogonal channels, whereas users of different cells may interfere against each other.

In this scenario, each MU i_n is willing to run an application within a given time T_{i_n} (imposed by the application) while minimizing the resulting energy consumption. The application is structured as a graph composed by computational components whose parameters are: i) the number b_{i_n} of input bits; ii) the number of CPU cycles $\omega_{i_n} = \omega(b_{i_n})$ associated with the b_{i_n} bits; and iii) the number $b^\circ_{i_n}$ of output bits (the result of the computation). We assume that the instructions to be executed are available at the server, otherwise they can be downloaded by the server through a high speed wired link. The MU can perform its computations locally or offload them to the cloud, based on which strategy requires less energy, while satisfying the latency constraint. In case of offloading, the overall latency experienced by each MU i_n is given by

$$\Delta_{i_n} = \Delta^{\mathtt{t}}_{i_n} + \Delta^{\mathtt{exe}}_{i_n} + \Delta^{\mathtt{tx/rx}}_{i_n} \tag{1}$$

where $\Delta^{\mathtt{t}}_{i_n}$ is the time necessary for the MU i_n to transfer the input bits b_{i_n} (associated with the set of instructions to be run on the cloud) to its SCeNB; $\Delta^{\mathtt{exe}}_{i_n}$ is the time for the server to execute ω_{i_n} CPU cycles; and $\Delta^{\mathtt{tx/rx}}_{i_n}$ is the round-trip time of the information (b_{i_n} and $b^\circ_{i_n}$) between SCeNB n and the cloud through the backhaul link. We assume that this link is a dedicated high speed connection (e.g., fiber optics) with constant latency, resulting thus in a constant $\Delta^{\mathtt{tx/rx}}_{i_n}$. We derive next an explicit expression of $\Delta^{\mathtt{t}}_{i_n}$ and $\Delta^{\mathtt{exe}}_{i_n}$ as a function of the radio and computational resources (to be optimized).

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 power budget constraints

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

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 the MU i_n is:

$$r_{i_n}(\mathbf{Q}) = \log_2 \det \left(\mathbf{I} + \mathbf{H}^H_{i_n n} \mathbf{R}_{i_n}(\mathbf{Q}_{-n})^{-1} \mathbf{H}_{i_n n} \mathbf{Q}_{i_n}, \right) \tag{2}$$

where

$$\mathbf{R}_{i_n}(\mathbf{Q}_{-n}) \triangleq \mathbf{R}_n + \sum_{n \neq m = 1}^{N_c} \sum_{j=1}^{K_m} \mathbf{H}_{j_m n} \mathbf{Q}_{j_m} \mathbf{H}^H_{j_m n}, \tag{3}$$

is the covariance matrix of the noise (with $\mathbf{R}_n \succ \mathbf{0}$) plus the inter-cell interference at the SCeNB n (treated as additive noise); $\mathbf{H}_{i_n n}$ is the channel matrix of the (up)link i in the cell n, and $\mathbf{H}_{j_m n}$ is the (cross-)channel matrix between the interferer MU j in the cell m and the SCeNB of cell n; and we denoted by $\mathbf{Q}_{-n} \triangleq ((\mathbf{Q}_{j_m})_{j=1}^{K_m})_{n \neq m = 1}^{N_c}$ the tuple of the covariance matrices of all users interfering with the SCeNB n.

Given each $r_{i_n}(\mathbf{Q})$, the time $\Delta^{\mathtt{t}}_{i_n}$ necessary for user i to transmit the input bits b_{i_n} of duration T_b to its SCeNB can be written as

$$\Delta^{\mathtt{t}}_{i_n} = \Delta^{\mathtt{t}}_{i_n}(\mathbf{Q}) = \frac{c_{i_n}}{r_{i_n}(\mathbf{Q})} \tag{4}$$

where $c_{i_n} = b_{i_n} T_b$. The overall energy consumption of the MU i_n due to offloading is then

$$E_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-n}) = \mathrm{tr}(\mathbf{Q}_{i_n}) \cdot \Delta^{\mathtt{t}}_{i_n}(\mathbf{Q}), \tag{5}$$

which depends also on the covariance matrices $\mathbf{Q}_{-n}$ of the users in the other cells (due to the intercell interference).

Computational resources. The cloud provider is able to serve concurrently multiple users (accessing through the associated SCeNB). The computational resources made available by the cloud and shared among all the users are quantified in terms of the number of CPU cycles/second, set to f_T; let $f_{i_n} \geq 0$ be the fraction of f_T assigned to each user i_n. All the f_{i_n} are thus (nonnegative) optimization variables (to be determined) subject to the budget constraint $\sum_{i_n \in \mathcal{I}} f_{i_n} \leq f_T$. Given the resource assignment f_{i_n}, the time $\Delta^{\mathtt{exe}}_{i_n}$ needed to remotely run ω_{i_n} CPU cycles of user i_n is then

$$\Delta^{\mathtt{exe}}_{i_n} = \Delta^{\mathtt{exe}}_{i_n}(f_{i_n}) = \omega_{i_n} / f_{i_n}. \tag{6}$$

The expression of the overall latency Δ_{i_n} [cf. (1), (4), and (6)] 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 cloud frequency assignment $\mathbf{f} \triangleq (\mathbf{f}_{i_n})_{i_n \in \mathcal{I}}$.

System design. We formulate the offloading problem as minimization of the overall energy spent by the MUs to remotely run their applications, under latency and power constraints. More formally, we have the following:

$$\begin{aligned}
\min_{\mathbf{Q},\mathbf{f}}\ & E(\mathbf{Q}) \triangleq \sum_{n,i} E_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-n}) \\
\text{s.t. }\ & \mathtt{a}) \ g_{i_n}(\mathbf{Q}, f_{i_n}) \triangleq \frac{c_{i_n}}{r_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}_{-n})} + \frac{\omega_{i_n}}{f_{i_n}} - \tilde{T}_{i_n} \leq 0,\ \forall i_n \in \mathcal{I}, \\
& \mathtt{b}) \ \sum_{i_n \in \mathcal{I}} f_{i_n} \leq f_T \quad \text{and} \quad f_{i_n} \geq 0, \quad \forall i_n \in \mathcal{I}, \\
& \mathtt{c}) \ \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I},
\end{aligned} \tag{P}$$

where a) reflects the users' latency constraints $\Delta_{i_n} \leq T_{i_n}$ [cf. (1)], with $\tilde{T}_{i_n}$ in a) capturing all the constant terms, i.e., $\tilde{T}_{i_n} \triangleq T_{i_n} - \Delta^{\mathtt{tx/rx}}_{i_n}$; and b) imposes a limit on the cloud computational resources made available to the users. We denote by $\mathcal{X}$ the feasible set of the optimization problem (P).

Feasibility: Depending on the system parameters, Problem (P) may be feasible or not. In the latter case, offloading is not possible and thus the users will perform their computations locally. The following conditions are sufficient for $\mathcal{X}$ to be nonempty and thus the offloading be feasible: $\tilde{T}_{i_n} > 0$, and there exists a $\bar{\mathbf{Q}} \triangleq (\bar{\mathbf{Q}}_{i_n})_{i_n \in \mathcal{I}} \in \mathcal{Q}$ such that

$$\tilde{T}_{i_n} > \frac{c_{i_n}}{r_{i_n}(\bar{\mathbf{Q}})},\ \forall i_n \in \mathcal{I}, \quad \text{and} \quad \sum_{i_n \in \mathcal{I}} \frac{\omega_{i_n}}{\tilde{T}_{i_n} - \frac{c_{i_n}}{r_{i_n}(\bar{\mathbf{Q}})}} \leq f_T.$$

Hereafter we will assume that Problem (P) is feasible.

Problem (P) is nonconvex, due to the nonconvexity of the objective function and the constraints a). In what follows we will exploit the structure of (P) and building on some recent Successive Convex Approximation (SCA) techniques proposed in [13], [14] we develop an efficient iterative inner approximation algorithm converging to a local optimal solution of (P). Numerical results show that the proposed algorithm converges in a few iterations to "good" local solutions of (P); furthermore, it is suitable for a distributed implementation

across the SCeNBs, with limited coordination and signalling with the cloud.

III. Algorithmic design

To solve the non-convex problem (P) efficiently, we develop a SCA-based method where the original problem (P) is replaced by a sequence of *strongly convex* problems. At the basis of the proposed technique there is a suitably chosen *convex* approximations of the nonconvex objective function $E(\mathbf{Q})$ and the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$, which are preliminary discussed next.

A. Approximant of $E(\mathbf{Q})$

The main idea is to approximate around the current (feasible) iterate $\mathbf{Q}^\nu \triangleq (\mathbf{Q}^\nu_{i_n})_{i_n\in\mathcal{I}} \in \mathcal{X}$ the original nonconvex nonseparable objective function $E(\mathbf{Q})$ with a *strongly convex* function, say $\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu)$, that is *separable* in the MUs' variables and has the same first order behaviour of $E(\mathbf{Q})$ at $\mathbf{Q}^\nu$ [13], [14]. To motivate the proposed choice of $\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu)$, observe preliminary that, for any given $\mathbf{Q}_{-n} = \mathbf{Q}^\nu_{-n}$, each term $E_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-n})$ in $E(\mathbf{Q})$ is bi-convex in $\mathbf{Q}_{i_n}$ [cf. (5)], and the other terms of the sum, $\sum_{n\neq m=1}^{N_c}\sum_{j=1}^{K_m} E_{j_m}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-i_n,j_m})$ with $\mathbf{Q}^\nu_{-i_n,j_m} \triangleq (\mathbf{Q}^\nu_{j_m}, (\mathbf{Q}^\nu_{l_q})_{\forall l,q\neq m,l_q\neq i_n})$, are not convex in $\mathbf{Q}_{i_n}$. Exploiting such a structure, a convex approximation of the sum-energy function $E(\mathbf{Q})$ for each MU i_n can be obtained by convexifying the bi-convex term in $E_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-n})$ [cf. (5)], and linearizing the nonconvex part $\sum_{n\neq m=1}^{N_c}\sum_{j=1}^{K_m} E_{j_m}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-i_n,j_m})$. More formally, for each i_n we introduce the "approximation" function $\tilde{E}_{i_n}(\mathbf{Q}_{i_n};\mathbf{Q}^\nu)$:

$$\begin{aligned}\tilde{E}_{i_n}(\mathbf{Q}_{i_n};\mathbf{Q}^\nu) \triangleq\ & \frac{c_{i_n}\cdot \mathrm{tr}(\mathbf{Q}_{i_n})}{r_{i_n}(\mathbf{Q}^\nu_{i_n}, \mathbf{Q}^\nu_{-n})} + \frac{c_{i_n}\cdot \mathrm{tr}(\mathbf{Q}^\nu_{i_n})}{r_{i_n}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-n})} \\ & + \sum_{n\neq m=1}^{N_c}\sum_{j=1}^{K_m}\left\langle \nabla_{\mathbf{Q}^*_{i_n}} E_{j_m}(\mathbf{Q}^\nu), \mathbf{Q}_{i_n} - \mathbf{Q}^\nu_{i_n}\right\rangle \\ & + \tau_{i_n}\left\|\mathbf{Q}_{i_n} - \mathbf{Q}^\nu_{i_n}\right\|^2\end{aligned} \tag{7}$$

where the first term on the right-hand side is the aforementioned convexification of the bi-convex term in (5); and the second term comes from the linearization of $\sum_{n\neq m=1}^{N_c}\sum_{j=1}^{K_m} E_{j_m}(\mathbf{Q}_{i_n}, \mathbf{Q}^\nu_{-i_n,j_m})$, with $\langle \mathbf{A}, \mathbf{B}\rangle \triangleq \mathrm{tr}(\mathbf{A}^H\mathbf{B})$ and $\nabla_{\mathbf{Q}^*_{i_n}} E_{j_m}(\mathbf{Q})$ denoting the conjugate gradient of $E_{j_m}(\mathbf{Q})$ with respect to $\mathbf{Q}_{i_n}$ evaluated at $\mathbf{Q}^\nu$, and given by

$$\begin{aligned}\nabla_{\mathbf{Q}^*_{i_n}} E_{j_m}(\mathbf{Q}) = & \frac{\mathrm{tr}(\mathbf{Q}^\nu_{j_m})\Delta_{j_m}(\mathbf{Q}^\nu)}{r_{j_m}(\mathbf{Q}^\nu)}\cdot\left[\mathbf{H}^H_{i_n m}\left(\mathbf{R}_{j_m}(\mathbf{Q}^\nu_{-m})^{-1}\right.\right. \\ & \left.\left. -(\mathbf{R}_{j_m}(\mathbf{Q}^\nu_{-m}) + \mathbf{H}_{j_m m}\mathbf{Q}^\nu_{j_m}\mathbf{H}^H_{j_m m})^{-1}\right)\mathbf{H}_{i_n m}\right].\end{aligned} \tag{8}$$

Note that in (7) we also added a quadratic regularization, making $\tilde{E}_{i_n}(\mathbf{Q}_{i_n};\mathbf{Q}^\nu)$ uniformly strongly convex on $\mathcal{Q}_{i_n}$. Based on each $\tilde{E}_{i_n}(\mathbf{Q}_{i_n};\mathbf{Q}^\nu)$, we can now define the candidate sum-energy approximation $\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu)$ as: given $\mathbf{Q}^\nu \succeq \mathbf{0}$,

$$\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu) \triangleq \sum_{n=1}^{N_c}\sum_{i=1}^{K_n}\tilde{E}_{i_n}(\mathbf{Q}_{i_n};\mathbf{Q}^\nu). \tag{9}$$

Note that $\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu)$ has many desirable properties, such as: i) it has the same first-order behavior of the original nonconvex function $E(\mathbf{Q})$ at $\mathbf{Q}^\nu$, i.e., $\nabla_{\mathbf{Q}^*}E(\mathbf{Q}^\nu) = \nabla_{\mathbf{Q}^*}\tilde{E}(\mathbf{Q}^\nu;\mathbf{Q}^\nu)$; ii) it is separable in the users variables $\mathbf{Q}_{i_n}$ (which is instrumental to obtain distributed algorithms, see [15] for details); and iii) it is uniformly strongly convex on $\mathcal{Q}$ [15]; we will denote by $c_\tau > 0$ the constant of strong convexity, that is c_τ is the largest positive scalar such that [16]

$$\begin{aligned}&\left\langle \mathbf{Q}^1 - \mathbf{Q}^2, \nabla_{\mathbf{Q}^*}\tilde{E}\left(\mathbf{Q}^1;\mathbf{Q}^\nu\right) - \nabla_{\mathbf{Q}^*}\tilde{E}\left(\mathbf{Q}^2;\mathbf{Q}^\nu\right)\right\rangle \\ &\qquad \geq c_\tau\left\|\mathbf{Q}^1 - \mathbf{Q}^2\right\|^2, \quad \forall\, \mathbf{Q}^1,\, \mathbf{Q}^2 \in \mathcal{Q}.\end{aligned} \tag{10}$$

An explicit expression of c_τ is given in [15] and is omitted here because of space limitation; we only observe that $c_\tau \geq \min_{i_n\in\mathcal{I}}\{\tau_{i_n}\}$.

B. Inner convexification of the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$

We aim at introducing an inner convex approximation of the constraints $g_{i_n}(\mathbf{Q}, f_{i_n})$ around $\mathbf{Q}^\nu \in \mathcal{X}$, denoted by $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu)$. To do so, let us exploit first the concave-convex structure of the rate functions $r_{i_n}(\mathbf{Q})$ [cf. (2)]:

$$r_{i_n}(\mathbf{Q}) = r^+_{i_n}(\mathbf{Q}) + r^-_{i_n}(\mathbf{Q}_{-n}), \tag{11}$$

with

$$\begin{aligned}r^+_{i_n}(\mathbf{Q}) &\triangleq \log_2\det\left(\mathbf{R}_{i_n}(\mathbf{Q}_{-n}) + \mathbf{H}_{j_n n}\mathbf{Q}_{j_n}\mathbf{H}^H_{j_n n}\right) \\ r^-_{i_n}(\mathbf{Q}_{-n}) &\triangleq -\log_2\det\left(\mathbf{R}_{i_n}(\mathbf{Q}_{-n})\right)\end{aligned} \tag{12}$$

and $\mathbf{R}_{i_n}(\mathbf{Q}_{-n})$ defined in (3). 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 (11), and observing that at any feasible $\mathbf{Q}, \mathbf{f}$, it must be $r_{i_n}(\mathbf{Q}) > 0$ and $f_{i_n} > 0$ for all i and n, the constraints $g_{i_n}(\mathbf{Q}, f_{i_n}) \leq 0$ in (P) can be rewritten as

$$g_{i_n}(\mathbf{Q}, f_{i_n}) = -r^+_{i_n}(\mathbf{Q}) - r^-_{i_n}(\mathbf{Q}_{-n}) + \frac{c_{i_n}\cdot f_{i_n}}{f_{i_n}\cdot\bar{T}_{i_n} - \omega_{i_n}} \leq 0. \tag{13}$$

The desired inner convex approximation $\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu)$ is obtained from $g_{i_n}(\mathbf{Q}, f_{i_n})$ by retaining the convex part in (13) and linearizing the concave term $-r^-_{i_n}(\mathbf{Q}_{-n})$, resulting in:

$$\begin{aligned}\tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu) \triangleq\ & -r^+_{i_n}(\mathbf{Q}) + \frac{c_{i_n}\cdot f_{i_n}}{f_{i_n}\cdot\bar{T}_{i_n} - \omega_{i_n}} \\ & - r^-_{i_n}(\mathbf{Q}^\nu_{-n}) - \sum_{m\neq n, j\neq i}\left\langle \nabla_{\mathbf{Q}^*_{j_m}} r^-_{i_n}(\mathbf{Q}^\nu_{-n}), \mathbf{Q}_{j_m} - \mathbf{Q}^\nu_{j_m}\right\rangle,\end{aligned} \tag{14}$$

with $\nabla_{\mathbf{Q}^*_{j_m}} r^-_{i_n}(\mathbf{Q}^\nu_{-n}) = -\mathbf{H}^H_{j_m n}\mathbf{R}_{i_n}(\mathbf{Q}^\nu_{-n})^{-1}\mathbf{H}_{j_m n}$.

C. Inner SCA algorithm

We are now ready to introduce the proposed convex approximation of the nonconvex problem (P), which consists in replacing the nonconvex objective function $E(\mathbf{Q})$ and constraints $g_{i_n}(\mathbf{Q}, f_{i_n}) \leq 0$ in (P) with the approximations $\tilde{E}(\mathbf{Q};\mathbf{Q}^\nu)$ and $\tilde{g}_{i_n}(\mathbf{Q}_{i_n}, f_{i_n}; \mathbf{Q}^\nu) \leq 0$, respectively. More formally, given the feasible point $\mathbf{Q}^\nu$, we have

$$\begin{aligned}\widehat{\mathbf{Z}}(\mathbf{Z}^\nu) \triangleq\ & \underset{\mathbf{Q},\mathbf{f}}{\mathrm{argmin}}\ \tilde{E}(\mathbf{Q};\mathbf{Q}^\nu) + \frac{c_f}{2}\|\mathbf{f} - \mathbf{f}^\nu\|^2 \\ & \text{s.t.}\quad \mathtt{a)}\ \tilde{g}_{i_n}(\mathbf{Q}, f_{i_n}; \mathbf{Q}^\nu) \leq 0,\ \forall i_n \in \mathcal{I}, \\ & \qquad \mathtt{b)}\ \sum_{i_n\in\mathcal{I}} f_{i_n} \leq f_T \text{ and } f_{i_n} \geq 0,\ \forall i_n \in \mathcal{I}, \\ & \qquad \mathtt{c)}\ \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I},\end{aligned} \tag{P$^\nu$}$$

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

that we added in the objective function a quadratic term in the f-variables, in order to make the objective strongly convex in $\mathbf{f}$, with c_f being an arbitrary positive constant).

The proposed solution method consists in solving the sequence of problems (P$^\nu$), starting from a feasible $\mathbf{Z}^0 \triangleq (\mathbf{Q}^0, \mathbf{f}^0)$. The formal description of the algorithm is given in Algorithm 1 below, and its convergence to local optimal solution of the original nonconvex problem (P) are stated in Theorem 1 (the proof of the theorem is omitted because of the space limitation; it consists in showing that all the conditions in [13, Th.1] are satisfied; see [15]). Note that in Step 2 of the algorithm we allow a memory in the update of the iterate $\mathbf{Z}^\nu \triangleq (\mathbf{Q}^\nu, \mathbf{f}^\nu)$. A practical termination criterion in Step 1 is to stop the iterates when $|E(\mathbf{Q}^{\nu+1}) - E(\mathbf{Q}^\nu)| \leq \delta$, where $\delta > 0$ is the prescribed accuracy.

Algorithm 1 : Inner SCA Algorithm for (P)

Data: $\mathbf{Z}^0 \triangleq (\mathbf{Q}^0, \mathbf{f}^0) \in \mathcal{X}$; $\{\gamma^\nu\}_\nu \in (0,1]$; $c_\tau > 0$; $c_f > 0$.
Set $\nu = 0$.
(S.1): If $\mathbf{Z}^\nu$ 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. (P^ν)];
(S.3): Set $\mathbf{Z}^{\nu+1} = \mathbf{Z}^\nu + \gamma^\nu \left(\hat{\mathbf{Z}}(\mathbf{Z}^\nu) - \mathbf{Z}^\nu\right)$;
(S.4): $\nu \leftarrow \nu + 1$ and go to (S.1).

Theorem 1 *Given the nonconvex problem (P), choose $c_\tau > 0$, $c_f > 0$, and $\{\gamma^\nu\}_\nu$ such that*

$$(0,1] \ni \gamma^\nu \to 0,\ \forall \nu \geq 0, \quad \textit{and} \quad \sum_\nu \gamma^\nu = +\infty. \tag{15}$$

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

Theorem 1 offers some flexibility in the choice of the free parameters c_τ, c_f, and $\{\gamma^\nu\}_\nu$ while guaranteeing convergence of Algorithm 1. For instance, c_τ is positive if all τ_{i_n} 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 $c_\tau > 0$). Any $c_f > 0$ is instead admissible. Many choices are possible for the step-size γ^ν; a practical rule satisfying (15) that we found effective in our experiment is [12]:

$$\gamma^{\nu+1} = \gamma^\nu (1 - \alpha \gamma^\nu), \quad \gamma^0 = 1, \tag{16}$$

with $\alpha \in [0,1)$.

Distributed implementation. We remark that problem (P^ν) can be decomposed across the SCeNBs (with limited coordination with the cloud) by i) introducing some slack variables whose goal is to decouple $\mathbf{Q}_{i_n}$ from $\mathbf{Q}_{-n}$ in $r^+_{i_n}(\mathbf{Q})$; ii) dualizing the coupling constraints of the resulting equivalent problem; and iii) solving the dual problem via standard (accelerated) gradient algorithms. We omit the details because of the space limitation, we refer the interested reader to [15].

IV. Numerical results

In this section we present some numerical results to assess the performance of the proposed offloading strategy. Our experiments are run under the following setup. We consider a $N_c = 2$ cell MIMO network, where all the transceivers are equipped with $n_T = n_R = 2$ antennas (unless stated otherwise); in each cell there are $K_n = 3$ active users, randomly deployed. The other system parameters are set as follows: $f_T = 2 \cdot 10^7$, $\mathbf{R}_n = \sigma^2 \mathbf{I}$, with $\mathtt{snr} \triangleq P_T/\sigma^2 = 3$dB. We simulated Algorithm 1 using the diminishing step-size rule (15), with $\alpha = 1\mathrm{e}{-5}$; the termination accuracy δ is set to 10^{-2}.

Example # 1: Energy consumption. We start comparing the energy consumption of the proposed offloading strategy with that of more traditional methods where the optimization of the communication and computational resources is performed separately. A good heuristic for the latter approach is the following procedure: i) optimizing the users' covariance matrices first (based on some optimality criterion) subject to the power constraints $\mathcal{Q}$; and ii) performing offloading using as users' rates those resulting from the optimization in step i) (provided that the offloading is feasible). As optimality criterion in i) we adopted the maximization of the weighted geometric mean, i.e., $\Pi_{i,n} r_{i_n}(\mathbf{Q})^{\bar{b}_{i_n}}$, with $\bar{b}_{i_n} = b_{i_n} / \sum_{i,n} b_{i_n}$ (the idea is to assign higher priorities to users having more bits to transfer to the cloud), resulting in the following optimization problem:

$$\begin{aligned} \max_{\mathbf{Q}} \quad & \sum_{i_n \in \mathcal{I}} \bar{b}_{i_n} \cdot \log\left(r_{i_n}(\mathbf{Q})\right) \\ \text{s.t.} \quad & \mathbf{Q}_{i_n} \in \mathcal{Q}_{i_n}, \quad \forall i_n \in \mathcal{I}. \end{aligned} \tag{17}$$

The above problem is nonconvex; we used the SCA-based algorithm proposed in [12] to compute a local optimal solution of (17). The resulting rates are then used to support MUs' offloading (when it is feasible). We will term this procedure *heuristic algorithm*. As further heuristic strategy, one can consider the optimization problem (P) wherein the optimization variables are only the MUs' covariance matrices, while the f_{i_n} are fixed and equal to $f_{i_n} = f_T / \sum_n K_n$ CPU cycles/second (the cloud computational rate f_T is equally divided across the users); we termed such a method *Equal Computational resources Assignment (ECA)*. In Fig. 1 we plot the energy consumption resulting from Algorithm 1, the heuristic algorithm, and ECA versus the ratio $\eta = w_{i_n}/b_{i_n}$ (assumed to be equal for all the users). The curves are averages over 100 independent realizations of Rayleigh fading channels. The system parameters have been chosen so that offloading is feasible for all the algorithms over the simulated channel realizations. The analysis of the figure clearly shows that our

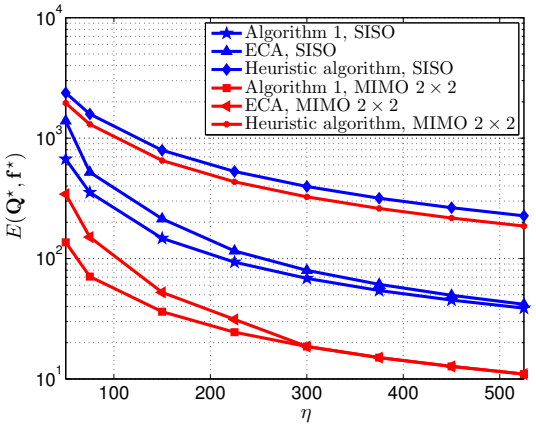

Fig. 1. Energy consumption vs. $\eta = w_{i_n}/b_{i_n}$: Proposed scheme (Algorithm 1) and heuristic methods.

scheme yields an energy saving with respect to the other simulated schemes, which validates the proposed approach of jointly optimizing radio and computational resources. Note that

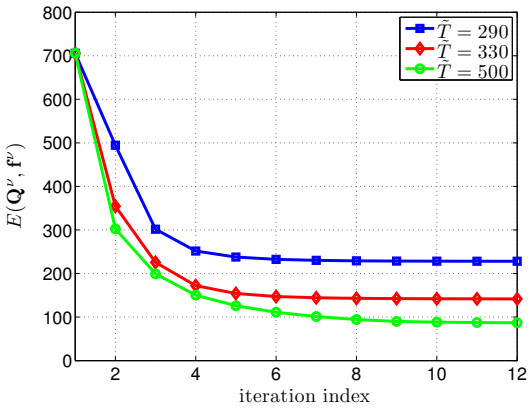

Fig. 2. Convergence speed: Optimal energy vs. the iterations for different values of $\tilde{T}$.

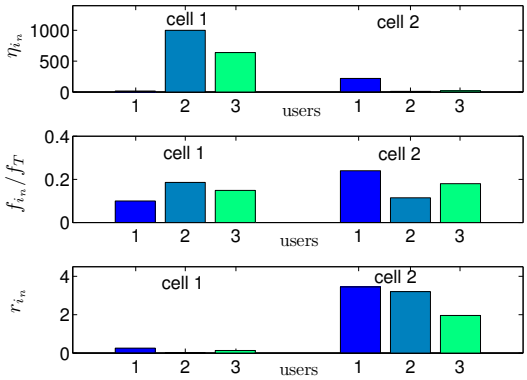

Fig. 3. Radio and computational resource allocation (SISO channels): η_{i_n}, f_{i_n}/f_T, and r_{i_n} of the users in the cells.

the energy consumption decreases as η increases, showing that the applications more suitable for offloading are those characterized by a limited input (or state) size (i.e., small b_{i_n}) and a high number of CPU cycles to be executed (i.e., large ω_{i_n}). The figure also shows that offloading takes advantage from the channel spatial diversity: increasing the number of antennas will contribute to reduce the energy consumption.

Example # 2: Convergence speed. In Fig. 2 we test the convergence speed of the proposed scheme; we plot the optimal energy $E(\mathbf{Q}^\nu, \mathbf{f}^\nu)$ [cf. (P^ν)] versus the iteration index ν of Algorithm 1, for different values of the maximum tolerable delay $\tilde{T}_{i_n}$ (assumed to be equal for all the users). The curves correspond to a single channel realization. Quite interestingly, the proposed algorithm converges in a few iterations (we experienced a similar behaviour in all our experiments on different channel realizations). Note also that the energy consumption increases as the delay constraints become more stringent.

Example # 3: Optimal resources allocation. To grasp how the computational and radio resources are allocated across the users, in Fig. 3, we show the allocation of the (normalized) CPU cycles f_{i_n}/f_T and the rates r_{i_n} resulting from Algorithm 1, for a given set of $\eta_{i_n} = w_{i_n}/b_{i_n}$ (also shown in the figure). It is interesting to observe how the proposed strategy tends to assign the highest computational rate f_{i_n} to the users with the most demanding applications (the highest ratio η_{i_n}).

V. Conclusion

In this paper we formulated the offloading problem in a multi-cell MIMO femto-cloud network as a joint optimization of the radio and computational resources: the transmit covariance matrices of the MUs as well as the computational resources assigned by the cloud to the MUs are jointly optimized, in order to minimize the overall energy consumption of the MUs, subject to latency constraints. Building on advanced SCA techniques, we developed an iterative algorithm with provable convergence to local optimal solutions of the proposed nonconvex optimization problem. Numerical results show that the proposed offloading strategy yields energy savings with respect to a separate optimization of the radio and computational resources. In this paper we considered a single cloud in the network and a further development will be to extend the proposed approach to a multi-server scenario.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017," February 2013, [pdf] USA: Cisco. Available at: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf.

[2] S. Robinson, "Cellphone energy gap: Desperately seeking solutions," *Tech. Rep. Strategy Analytics*, 2009.

[3] M.R. Palacin, "Recent advances in rechargeable battery materials: a chemists perspective," *Chem. Soc. Rev.*, no. 38, pp. 2565–2575, 2009.

[4] M. Sharifi, S. Kafaie, and O. Kashefi, "A Survey and Taxonomy of Cyber Foraging of Mobile Devices," *IEEE Comm. Surveys and Tutorials*, Vol. 14, no. 4, pp. 1232–1243, Fourth Quarter 2012.

[5] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, "A Survey of computation offloading for mobile systems," *Mobile Networks and Applications*, pp. 129–140, Feb. 2013.

[6] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," *Proc. of the ACM International Conference on Mobile Systems, Applications, and Services*, pp. 49-62, San Francisco, CA, USA, 15-18 June 2010.

[7] S. Kosta, A. Aucinas, P. Hui, R. Mortier, X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," *Proc. of INFOCOM 2012*, pp. 945–953, March 2012.

[8] N. Fernando, S.W. Loke, W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, Vol. 29, pp. 84–106, Jan. 2013.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, pp. 14–23, Oct.-Dec. 2009.

[10] *TROPIC: Distributed computing, storage and radio resource allocation over cooperative femtocells*, http://www.ict-tropic.eu.

[11] S. Barbarossa, S. Sardellitti and P. Di Lorenzo,"Joint allocation of computation and communication resources in multiuser mobile cloud computing," in Proc. of the *IEEE 2013 Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2013)*, Darmstadt, Germany, June 16-19, 2013.

[12] G. Scutari, F. Facchinei, P. Song, D.P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. on Signal Process.*, Vol. 63, no. 3, pp. 641–656, Feb. 2014.

[13] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Parallel and distributed methods for nonconvex optimization," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 14)*, May 4-9, 2014, Florence, Italy.

[14] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Parallel and distributed methods for nonconvex optimization-Part I&II: Theory & Applications," *IEEE Trans. on Signal Processing*, (under review).

[15] S. Sardellitti, G. Scutari, S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile cloud computing," *IEEE Trans. on Signal Process.*, (submitted) 2014.

[16] G. Scutari, F. Facchinei, J.-S. Pang, and D.P. Palomar, "Real and complex monotone communication games," *IEEE Trans. on Inf. Theory*, to appear. [Online]. Available: http://arxiv.org/abs/1212.6235.