# An Energy-efficient Task Offloading Solution for MEC-based IoT in Ultra-dense Networks

Elie El Haber, Tri Minh Nguyen, Chadi Assi, and Wessam Ajib

***Abstract*—By pushing computation to the mobile network edge, Multi-access Edge Computing (MEC) has been an enabler for the stringent latency and energy requirements of the new Internet of Things (IoT) services. On the other hand, ultra-dense heterogeneous networks with wireless backhaul have been proposed as a low-cost solution, allowing Network Operators (NOs) to extend the network capability, by deploying densified close-proximity small-cells and hence supporting a large number of low-latency low-energy IoT devices. In this paper, we study the problem of IoT task offloading in a MEC-enabled heterogeneous network, which to the best of our knowledge, is the first attempt to thoroughly explore the task offloading problem in a heterogeneous network with MEC support and wireless backhaul. We jointly optimize the offloading decision, transmission power, and the allocation of radio and computational resources, with the objective of minimizing the devices energy consumption, while respecting their latency deadline. We mathematically formulate our problem as a non-convex mixed-integer program, and due to its complexity, we propose an iterative algorithm based on the Successive Convex Approximation (SCA) method for providing an approximate solution on the original problem. Through numerical analysis, we perform simulations based on multiple scenarios, and find out how NOs can respond to the requested load and help in minimizing the total devices energy consumption.**

## I. Introduction

The world is witnessing a fast expansion of smart objects such as sensor-embedded wearable devices, enabling them to sense real-time information [1], culminating into the concept of Internet of Things (IoT) [2]. The introduction of IoT however is disrupting the existing communication processes, necessitating new models to be established. Mobile Cloud Computing (MCC), which was the go-to solution for alleviating the load of the energy-sensitive devices [3], incurs a long communication delay and a huge amount of network bandwidth, rendering it inadequate for meeting the strict requirements imposed by many IoT services. Multi-access Edge Computing (MEC) [4] has emerged as a solution for enabling IoT services, by moving the computation and storage to the network edge as small computation units denoted as cloudlets [5], allowing devices to compute their latency-sensitive tasks.

Meanwhile, heterogeneous networks were introduced for allowing Network Operators (NOs) to extend the network edge capability, by deploying low-power Small-Cells (SCs) close to the end-users, and connected to a Macro-Cell (MC) [6]. Moreover, the Ultra-Dense Network (UDN) concept was introduced as an improvement, where SCs deployment is further densified to accommodate the immense amount of traffic generated in 5G networks [7]. This model constitutes a perfect solution for the large number of IoT devices, since it can respond to their low-energy low-latency requirements, and provide better coverage and higher data rates. Moreover, equipping the SCs with cloudlets will further decrease the load by alleviating the load on the MC-cloudlet, creating a multi-tier MEC-enabled ultra-dense network where cloudlets are located at multiple tiers [8]. This forms an ideal model for actualizing IoT-based services, such as self-driving which requires processing real-time images using the cellular network.

In this paper, we present a novel solution for realizing energy-efficient task offloading for MEC-based IoT in densified heterogeneous networks. We present a framework that aims at minimizing IoT devices' energy consumption while respecting their latency deadline, by optimizing the devices offloading decision, and the joint allocation of transmission power, radio and computational resources while managing the resulting interference. To the best of our knowledge, this is the first attempt at studying a MEC-enabled heterogeneous network with wireless backhaul, leveraging cloudlets located at different tiers. We formulate the problem as a non-convex mixed-integer program, and apply multiple transformations to convert it into a more tractable form. We then present a low-complexity iterative algorithm based on the successive convex approximation (SCA) technique, for obtaining an approximate solution on the original problem. Through numerical results, we validate the performance of our proposed algorithm, and perform varying simulations and analyze the produced results.

The remainder of this paper is structured as follows: in section II we present the related work. In section III we present our system model and we mathematically formulate our non-convex problem. Section IV presents our solution approach to transform the non-convex problem into a more tractable form, and presents our iterative SCA-based algorithm. Section V presents and analyzes the numerical results for validating our solutions. Finally, section VI concludes the paper.

## II. Literature Review

The following studies addressed task offloading in a two-tier MEC-enabled system. [9] used a central cloud as a second level computing node, and allocated computational and radio resource. In [10], a multi-BS model is studied with one cloudlet, and the allocation of BSs caching and Resource Blocks (RBs) are optimized. The following studies addressed task offloading in a heterogeneous network. In [11] and [12], a cloudlet was considered on the MC where computational resources were allocated to the offloaded tasks, while radio resources where allocated on the uplink channel as a continuous spectrum in [11], and as discrete RBs in [12] while considering spectrum reuse and negligible backhaul delay. In [13], considering one SC connected to a cloudlet-enabled MC with limited backhaul capacity, the offloading decision and the uplink RBs allocation

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

were optimized. In [14], an algorithm is proposed for optimizing the radio resources allocation and the UEs-to-SCs mapping, in addition to caching and computational resources on the SCs, and transmission power allocation. In [15], the RAN downlink radio resources are orthogonally allocated to UEs, while also optimizing the UE to BS association and caching decisions on all BSs while respecting the UEs computation and communication rate requirements. Also, [16] proposed a solution to minimize UEs' latency with respect to energy constraint when offloading tasks in a UDN system, by optimizing the offloading decision and the computational resources allocation.

Our work differs from [11] and [12] in that cloudlets are co-located on all BSs, and the network has a wireless backhaul communication. Also, unlike [13], we consider a multi-SC model where cloudlets are co-located on all BSs, and optimize the radio resource allocation in the network backhaul, and unlike [14] and [15], we consider the Orthogonal Division Multiple Access (OFDMA) model so that radio resources are allocated as discrete RBs, and also consider frequency reuse among SCs which increases the spectrum efficiency and hence the transmission rates. Finally, unlike [16], we optimize UEs transmission power and radio resources allocation, and we consider the backhaul communication. The innovation of our solution can be characterized by considering a heterogeneous network with low-cost wireless backhaul requiring the management of OFDMA radio resources, and also considering cloudlet-enabled SCs which would make the solution more practical for IoT-based services. We jointly allocate radio and computational resources, which would result in a improved solution as compared to the related works.

## III. System Model

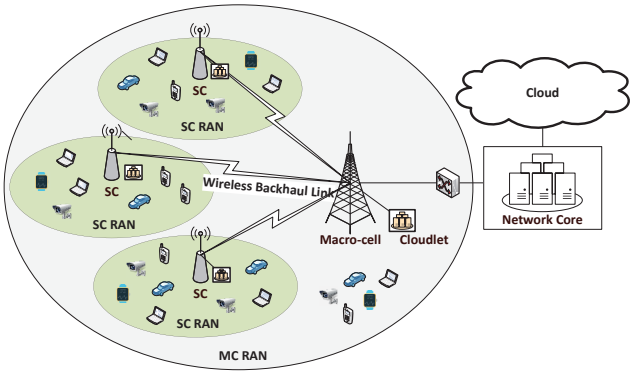



Fig. 1: System model.

### A. Spatial Model

As depicted in Figure 1, we consider a MEC-enabled heterogeneous network that consists of $S$ single-antenna SCs indexed by $\mathcal{S} = \{1, ..., S\}$, and are coexisted within the coverage of one multi-antenna MC with index 0. Each cell $j \in \{0 \cup \mathcal{S}\}$ has $U$ User Equipments (UEs) in its coverage, indexed by $\mathcal{U}_j = \{1, ..., U\}$. SC UEs are denoted by SUEs, and MC UEs are denoted by MUEs. For ease of presentation, we denote by $F = S + U$ as the number of units transmitting to the MC indexed by $\mathcal{F} = \{\mathcal{S}; \mathcal{U}_0\}$. We consider the UEs to have computational tasks, and they need to communicate wirelessly when offloading these tasks.

### B. Communication Model

We consider the uplink communication used by SUEs and MUEs for task offloading in the SCs RAN, and a wireless backhaul for the communication between the SCs and MBS for task migration. We ignore the downlink communication knowing that the output size is in general much smaller than the task input size [12], [13], e.g. face recognition. We consider an OFDMA multiplexing system with perfect channel state information (CSI), where the radio spectrum is separated into $B$ OFDMA resource blocks (RBs), indexed by $\mathcal{B} = \{1, ..., B\}$. We adopt a split-spectrum approach, i.e. $\mathcal{B} = \{\mathcal{B}_1; \mathcal{B}_2\}$, where $\mathcal{B}_1$ is the set of RBs dedicated for the communication between SUEs and their SCs, and $\mathcal{B}_2$ is the set of RBs dedicated for the communication between MUEs (and SCs) with the MC.

**1) SUEs–SCs Communication:** We denote by $\boldsymbol{x} = \{x_{i,j,b}, \forall j \in \mathcal{S}, i \in \mathcal{U}_j, b \in \mathcal{B}_1\}$ as the binary decision variables indicating if RB $b$ is assigned to SUE $i$ for the communication with SC $j$, where the value is 1 in case the RB is assigned, and is 0 otherwise. When task $i$ is offloaded, multiple RBs can be assigned to SUE $i$. By denoting $p_{i,j,b} \geq 0$ as the optimization variable indicating the power allocated for the transmission from SUE $i \in \mathcal{U}$ to SC $j \in \mathcal{S}$ on RB $b \in \mathcal{B}_1$, $\boldsymbol{p}_b = \{p_{i,j,b}, \forall j \in \mathcal{S}, i \in \mathcal{U}_j\}$, $\boldsymbol{p} = \{\boldsymbol{p}_b, \forall b \in \mathcal{B}_1\}$, and $h_{i,j,l,b}$ as the channel gain from SUE $i$ to SC $l$ on RB $b$ which includes fading and path loss components, the uplink transmission rate for SUE $i$ in the range of SC $j$ on RB $b$ is defined using the Shannon capacity as $r_{i,j,b}(\boldsymbol{p}_b) = \log\left(1 + \frac{p_{i,j,b}|h_{i,j,j,b}|^2}{\sum_{l \in \mathcal{S} \setminus} \sum_{k \in \mathcal{U}_l} p_{k,l,b}|h_{k,l,j,b}|^2 + N_0}\right)$, where $N_0$ is the white noise power level. As it can be seen, the inter-cell interference perceived is caused by the SUEs transmissions in the nearby SCs on the same RB $b$, knowing that frequency reuse is adopted among SCs to achieve high spectrum efficiency. Thus, the achievable rate for SUE $i$ is:

$$r_{i,j}(\boldsymbol{p}) = \sum_{b \in \mathcal{B}_1} r_{i,j,b}(\boldsymbol{p}_b) \tag{1}$$

In addition, by denoting $\overline{\boldsymbol{P}} = \{\overline{P}_{i,j}, \forall j \in \mathcal{S}, i \in \mathcal{U}_j\}$ as the maximum power budget for all SUEs in dBm, the following constraints are imposed to govern a proper relationship domain of the involved variables:

$$\sum_{i \in \mathcal{U}_j} x_{i,j,b} \leq 1 \quad \forall j \in \mathcal{S}, b \in \mathcal{B}_1 \tag{2a}$$

$$p_{i,j,b} \leq x_{i,j,b}\overline{P}_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j, b \in \mathcal{B}_1 \tag{2b}$$

$$\sum_{b \in \mathcal{B}_1} p_{i,j,b} \leq \overline{P}_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j \tag{2c}$$

where (2a) ensures orthogonal radio resource allocation in the SCs RAN, (2b) assures no transmission power on a non-assigned RB, and (2c) is for respecting the SUEs' power.

**2) Communication with the MC:** Communication with the MC is needed by MUEs to offload their tasks, and by SCs in case they are migrating offloaded tasks from their SUEs. Without loss of generality, we consider Zero-forcing (ZF) as the method applied for signal processing on the MC [17], assuming

that its number of antennas, denoted by $N$, is able to support simultaneous connections from the transmitting units $\mathcal{F}$ on a given RB $b$, i.e. $F \leq N$. By denoting $\rho_{i,b} \geq 0$ as the variable for deciding the power allocated for transmission from MUE (or SC) $i \in \mathcal{F}$ to the MC on RB $b \in \mathcal{B}_2$, $\boldsymbol{\rho}_i = \{\rho_{i,b}, \forall b \in \mathcal{B}_2\}$, and $\boldsymbol{\rho} = \{\boldsymbol{\rho}_i, \forall i \in \mathcal{F}\}$, the uplink transmission rate for MUE (or SC) $i$ on RB $b$, is defined as $R_{i,b}(\rho_{i,b}) = \log\left(1 + \frac{\rho_{i,b}}{||\mathbf{w}_{i,b}||^2 N_0}\right)$, where $\mathbf{w}_{i,b} \in \mathbb{C}^{N\times 1}$ is the ZF receive beamforming row vector obtained from matrix $\boldsymbol{w}_b \in \mathbb{C}^{F\times N}$, where $\boldsymbol{w}_b = (\boldsymbol{h}_b^T \boldsymbol{h}_b)^{-1}\boldsymbol{h}_b^T$ is the pseudo-inverse of the channel state matrix $\boldsymbol{h}_b \in \mathbb{C}^{N\times F}$ for RB $b$, which includes fading and path loss components. It follows that the achievable rate for MUE (or SC) $i$ is:

$$R_i(\boldsymbol{\rho}_i) = \sum_{b\in\mathcal{B}_2} R_{i,b}(\rho_{i,b}) \tag{3}$$

In addition, by denoting $\bar{\boldsymbol{\rho}} = \{\bar{\rho}_i, \forall i \in \mathcal{F}\}$ as the maximum power budget for all MUEs (and SCs) in dBm, and $\boldsymbol{r} = \{r_{i,j} \geq 0, \forall j \in \mathcal{S}, i \in \mathcal{U}_j\}$ as the fraction of the achievable rate on SC $j$ assigned for migrating task $i$ to the MC, the following constraints are imposed:

$$\sum_{b\in\mathcal{B}_2} \rho_{i,b} \leq \bar{\rho}_i \quad \forall i \in \mathcal{F} \tag{4a}$$

$$\sum_{i\in\mathcal{U}_j} r_{i,j} \leq R_j(\boldsymbol{\rho}_j) \quad \forall j \in \mathcal{S} \tag{4b}$$

where (2c) is for respecting the MUEs (and SCs)' power threshold, and (4b) ensures the sum of all partitioned rates on an SC $j$ does not exceed its achievable rate.

### C. Computation Model

Each UE $i \in \mathcal{U}_j$ in the range of cell $j \in \{0 \cup \mathcal{S}\}$ has a task represented by the tuple $\{D_{i,j}, C_{i,j}, \overline{L}_{i,j}\}$, concatenating: Task input size $D_{i,j}$ (bits), computational demand $C_{i,j}$ (CPU cycles), and required latency deadline $\overline{L}_{i,j}$ (ms). The SCs and MC are equipped with cloudlets for supporting local computation. The offloading decision can be modeled using the binary variables $\boldsymbol{y} = \{y_{i,j}, \forall j \in \{0 \cup \mathcal{S}\}, i \in \mathcal{U}_j\}$ and $\mathbf{z} = \{z_{i,j}, \forall j \in \mathcal{S}, i \in \mathcal{U}_j\}$. When $y_{i,j}$ is equal to 1, task $i$ will be offloaded to its associated cell $j$, otherwise it will be executed on the local device. In addition, for the SUEs, when $z_{i,j}$ is equal to 1, task $i$ will be further migrated to the MC, otherwise it will be executed on the local SC $j$ cloudlet. Next, we model the latency and energy consumption resulting from the computation of task $i$, which depends on whether that task was executed locally, or offloaded to the SC or MC cloudlet.

**Local Computation:** We denote by $f_{i,j}^{\text{loc}}$ as the local computation capability in cycles/second on UE $i$ in the range of cell $j$. When task $i$ is executed locally, the resulting latency is defined as $L_{i,j}^{\text{loc}} = \frac{C_{i,j}}{f_{i,j}^{\text{loc}}}$. The resulting energy consumption for UE $i$ in the range of cell $j$ is defined as $E_{i,j}^{\text{loc}} = C_{i,j}E_{i,j}^{\text{cyc}}$, where $E_{i,j}^{\text{cyc}}$ is a constant representing the consumed energy per CPU cycle, which depends on the UE circuit architecture and can be obtained by the measurement method in [18].

**Cloudlet Computation:** The latency for SUE and MUE $i \in \mathcal{U}_j$ incurred from transmission to the associated cell $j$ is given by $L_{i,j}^{\text{sue}}(\boldsymbol{p}) = \frac{D_{i,j}}{r_{i,j}(\boldsymbol{p})}$ and $L_i^{\text{mue}}(\boldsymbol{\rho}_i) = \frac{D_{i,0}}{R_i(\boldsymbol{\rho}_i)}$, respectively, which depends on the achievable rate in the SC RAN.

The energy consumption resulting from task transmission to cell $j$ for SUE and MUE $i \in \mathcal{U}_j$ is given by $E_{i,j}^{\text{sue}}(\boldsymbol{p}) = \sum_{b\in\mathcal{B}_1} \frac{p_{i,j,b}d_{i,j,b}}{r_{i,j,b}(\boldsymbol{p}_b)}$ and $E_i^{\text{mue}}(\boldsymbol{\rho}_i) = \sum_{b\in\mathcal{B}_2} \frac{\rho_{i,b}\phi_{i,b}}{R_{i,b}(\rho_{i,b})}$, respectively, where $d_{i,j,b} \geq 0$ and $\phi_{i,b} \geq 0$ are the decision variables indicating the data chunk allocated for the transmission on RB $b$. In addition, the following constraints hold:

$$\sum_{b\in\mathcal{B}_1} x_{i,j,b}d_{i,j,b} \leq D_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j \tag{5a}$$

$$\sum_{b\in\mathcal{B}_2} \phi_{i,b} \leq D_{i,j} \quad \forall i \in \mathcal{U}_0 \tag{5b}$$

which are used to make sure the whole task is transmitted using the resource blocks, for the SUEs and MUEs, respectively. Increasing the transmission power for a given UE will decrease the upload latency, but this will result in an increased energy consumption. By denoting $s_j \in \{0; 1\}$ as an indicator returning 1 if cell $j$ is a SC and 0 if it is a MC, the energy consumption for UE $i$ in the range of cell $j$ can be defined as

$$E_{i,j}(\boldsymbol{p}; \boldsymbol{\rho}_i) = s_j E_{i,j}^{\text{sue}}(\boldsymbol{p}) + (1 - s_j)E_i^{\text{mue}}(\boldsymbol{\rho}_i) \tag{6}$$

In case task $i$ is for an SUE and is being further migrated to the MC through the backhaul, the migration latency is also considered, which depends on the SC assigned rate, and is defined as $L_{i,j}^{\text{mig}}(r_{i,j}) = \frac{D_{i,j}}{r_{i,j}}$. Whether the offloaded task $i$ ends up on the SC or MC, it needs to be executed on the associated cloudlet. We denote by $\boldsymbol{f} = \{f_{i,j} \geq 0, \forall j \in \mathcal{S}, i \in \mathcal{U}_j\}$ as an optimization variable which represents the computational resources allocated for tasks cloudlet computation. The resulting execution latency is given by $L_{i,j}^{\text{cld}}(f_{i,j}) = \frac{C_{i,j}}{f_{i,j}}$. Allocating more computational resources to task $i$ will decrease its computation latency, but this will limit the resources availability for executing other offloaded tasks on that cloudlet, which will in turn increase their computation latency.

### D. Problem Formulation

Our objective is to minimize the total energy consumption for all UEs while respecting their required latency. This is done by optimizing the tasks offloading decision, the transmission power and radio resources allocation for the UEs and SCs, as well as the computational resources allocation on each cloudlet. By denoting $\boldsymbol{\chi}_1 = \{\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\rho}, \boldsymbol{r}, \boldsymbol{f}, \boldsymbol{d}, \boldsymbol{\phi}\}$, the joint transmission power and computational/radio resources allocation for IoT task offloading problem in heterogeneous networks, denoted as $\mathcal{P}_1$, is formulated as

$$\min_{\boldsymbol{\chi}_1} \sum_{j\in\{0\cup\mathcal{S}\}} \sum_{i\in\mathcal{U}_j} (1 - y_{i,j})E_{i,j}^{\text{loc}} + y_{i,j}E_{i,j}(\boldsymbol{p}; \boldsymbol{\rho}_i) \tag{7a}$$

$$\text{s.t. } (1 - y_{i,j})L_{i,j}^{\text{loc}} + y_{i,j}(L_{i,j}^{\text{sue}}(\boldsymbol{p}) + L_{i,j}^{\text{cld}}(f_{i,j})) + z_{i,j}L_{i,j}^{\text{mig}}(r_{i,j}) \leq \overline{L}_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j \tag{7b}$$

$$(1 - y_{i,0})L_{i,0}^{\text{loc}} + y_{i,0}(L_i^{\text{mue}}(\boldsymbol{\rho}_i) + L_{i,0}^{\text{cld}}(f_{i,0})) \leq \overline{L}_{i,0} \ \forall i \in \mathcal{U}_0 \tag{7c}$$

$$x_{i,j,b} \leq y_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j, b \in \mathcal{B}_1 \tag{7d}$$

$$z_{i,j} \leq y_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j \tag{7e}$$

$$z_{i,j}r_{i,j}(\boldsymbol{p}) \leq r_{i,j} \quad \forall j \in \mathcal{S}, i \in \mathcal{U}_j \tag{7f}$$

$$\sum_{i\in\mathcal{U}_j}(1-z_{i,j})f_{i,j}\leq \overline{F}_j \quad \forall j\in\mathcal{S} \tag{7g}$$

$$\sum_{j\in\mathcal{S}}\sum_{i\in\mathcal{U}_j} z_{i,j}f_{i,j}+\sum_{i\in\mathcal{U}_0} f_{i,0}\leq \overline{F}_0 \tag{7h}$$

$$\text{(2), (4), (5)} \tag{7i}$$

$$y_{i,j}, z_{i,j}, x_{i,j,b}\in\{0,1\} \tag{7j}$$

$$p_{i,j,b}, \rho_{i,b}, r_{i,j}, f_{i,j}, d_{i,j,b}, \phi_{i,b}\geq 0 \tag{7k}$$

where (7a) is minimizing the total energy consumption (in millijoule) for all UEs. Constraints (7b) and (7c) respect the tasks' latency deadline. (7d) makes sure an RB can be only assigned to offloading UEs. (7e) is to make sure a given SUE task can be migrated to the MC only if it was offloaded. (7f) prevents the backhaul from being a bottleneck when migrating a task to the MC, and hence adds a major significance by ensuring transmission stability in the wireless backhaul. Constraints (7g) and (7h) are for respecting the computational resources capacity, denoted by $\overline{F}_j$, on the SCs and MC cloudlet, respectively. This problem always has a feasible solution by computing UEs tasks locally but with a much higher assumed energy consumption. Therefore, all UEs will try first to either offload to their associated SC, or further migrate to the MC for cloudlet computation. It can be seen that the following terms in problem (7) are non-convex: the energy function (6) used in (7a), the latency functions used in the latency constraints (7b) and (7c), and the SC and MC rate functions (1) and (3) used in constraints (4b) and (7f). In addition, constraint (7j) implies that (7) is an integer optimization problem. In fact, the formulated problem (7) is a non-convex Mixed-Integer Nonlinear Program (MINLP), which is generally difficult to solve.

## IV. Proposed Low-Complexity Algorithm

Due to the high complexity of the proposed non-convex MINLP problem, in this section, we propose to approach a solution of (7) with a more pragmatic, efficient, and low computational complexity algorithm.

### A. Big-M based Equivalent Linear Transformation

Let us start first by equivalently transforming problem (7) into a more tractable form using the well-known big-M technique, where $M\gg 1$, to facilitate the difficulty of handling the binary-related objective function and constraints. Specifically, we introduce the non-negative slack variables $o_{i,j}$, $q_{i,j}$, $t_i$, $u_{i,j}$, $v_{i,j}$, $w_{i,j}$, to substitute the terms $y_{i,j}E_{i,j}(\boldsymbol{p};\boldsymbol{\rho}_i)$ in (7a), $y_{i,j}(L^{\text{sue}}_{i,j}(\boldsymbol{p})+L^{\text{cld}}_{i,j}(f_{i,j}))$ in (7b), $y_{i,0}(L^{\text{mue}}_{i}(\boldsymbol{\rho}_i)+L^{\text{cld}}_{i,0}(f_{i,0}))$ in (7c), $z_{i,j}L^{\text{mig}}_{i,j}(r_{i,j})$ in (7b), $z_{i,j}r_{i,j}(\boldsymbol{p})$ in (7f), and $z_{i,j}f_{i,j}$ in (7g)–(7h), respectively. The new slack constraints for variables $\boldsymbol{o}=\{o_{i,j}\geq 0, \forall j\in\mathcal{S}, i\in\mathcal{U}_j\}$ can be presented as

$$o_{i,j}\leq y_{i,j}M \tag{8a}$$

$$(y_{i,j}-1)M+E_{i,j}(\boldsymbol{p};\boldsymbol{\rho}_i)\leq o_{i,j} \tag{8b}$$

$$o_{i,j}\leq E_{i,j}(\boldsymbol{p};\boldsymbol{\rho}_i) \tag{8c}$$

The same technique can be applied for linearizing the other terms, where each constraint will be linear with respect to the involved variables. We note the non-linear constraints resulting from the big-M transformations as follows:

$$(y_{i,j}-1)M+L^{\text{sue}}_{i,j}(\boldsymbol{p})+L^{\text{cld}}_{i,j}(f_{i,j})\leq q_{i,j} \tag{9a}$$

$$q_{i,j}\leq L^{\text{sue}}_{i,j}(\boldsymbol{p})+L^{\text{cld}}_{i,j}(f_{i,j}) \tag{9b}$$

$$(y_{i,0}-1)M+L^{\text{mue}}_{i}(\boldsymbol{\rho}_i)+L^{\text{cld}}_{i,0}(f_{i,0})\leq t_i \tag{9c}$$

$$t_i\leq L^{\text{mue}}_{i}(\boldsymbol{\rho}_i)+L^{\text{cld}}_{i,0}(f_{i,0}) \tag{9d}$$

$$(z_{i,j}-1)M+L^{\text{mig}}_{i,j}(r_{i,j})\leq u_{i,j} \tag{9e}$$

$$u_{i,j}\leq L^{\text{mig}}_{i,j}(r_{i,j}) \tag{9f}$$

$$(z_{i,j}-1)M+r_{i,j}(\boldsymbol{p})\leq v_{i,j} \tag{9g}$$

$$v_{i,j}\leq r_{i,j}(\boldsymbol{p}) \tag{9h}$$

respectively. By denoting $\boldsymbol{\chi}_2=\{\boldsymbol{\chi}_1;\boldsymbol{o},\boldsymbol{q},\boldsymbol{t},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\}$, problem $\mathcal{P}_1$ after applying the big-$M$ based equivalent transformation, denoted as $\mathcal{P}_2$, can be casted as

$$\min_{\boldsymbol{\chi}_2}\ \sum_{j\in\{0\cup\mathcal{S}\}}\sum_{i\in\mathcal{U}_j}(1-y_{i,j})E^{\text{loc}}_{i,j}+o_{i,j} \tag{10a}$$

$$\text{s.t. } (1-y_{i,j})L^{\text{loc}}_{i,j}+q_{i,j}+u_{i,j}\leq \overline{L}_{i,j} \quad \forall j\in\mathcal{S}, i\in\mathcal{U}_j \tag{10b}$$

$$(1-y_{i,0})L^{\text{loc}}_{i,0}+t_i\leq \overline{L}_{i,0}\ \forall i\in\mathcal{U}_0 \tag{10c}$$

$$x_{i,j,b}\leq y_{i,j} \quad \forall j\in\mathcal{S}, i\in\mathcal{U}_j, b\in\mathcal{B}_1 \tag{10d}$$

$$z_{i,j}\leq y_{i,j} \quad \forall j\in\mathcal{S}, i\in\mathcal{U}_j \tag{10e}$$

$$v_{i,j}\leq r_{i,j} \quad \forall j\in\mathcal{S}, i\in\mathcal{U}_j \tag{10f}$$

$$\sum_{i\in\mathcal{U}_j} f_{i,j}-w_{i,j}\leq \overline{F}_j \quad \forall j\in\mathcal{S} \tag{10g}$$

$$\sum_{j\in\mathcal{S}}\sum_{i\in\mathcal{U}_j} w_{i,j}+\sum_{i\in\mathcal{U}_0} f_{i,0}\leq \overline{F}_0 \tag{10h}$$

$$\text{(2), (4), (5), (8), (9)} \tag{10i}$$

$$y_{i,j}, z_{i,j}, x_{i,j,b}\in\{0,1\} \tag{10j}$$

$$p_{i,j,b}, \rho_{i,b}, r_{i,j}, f_{i,j}, d_{i,j,b}, \phi_{i,b}\geq 0 \tag{10k}$$

where the equivalence between (7) and (8)–(10) is omitted due to lack of space. In the next subsection, we reveal the factors that make the formulated problem $\mathcal{P}_1$ non-convex by separating the convex and non-convex constraints.

### B. Successive Convex Approximation (SCA) Method

At this point, we observe that the underlying issues which make (10) a non-convex MINLP problem, and hence difficult to solve, are due to the existence of functions $E_{i,j}(\boldsymbol{p};\boldsymbol{\rho}_i)$, $L^{\text{sue}}_{i,j}(\boldsymbol{p})$, and $r_{i,j}(\boldsymbol{p})$, which are non-convex with respect to $\boldsymbol{p}$ and $\boldsymbol{\rho}_i$, in constraints (8b)–(8c), (9a)–(9b), and (9g)–(9h), respectively. Also, the existence of functions $L^{\text{mue}}_{i}(\boldsymbol{\rho}_i)$, $L^{\text{cld}}_{i,j}(f_{i,j})$, and $L^{\text{mig}}_{i,j}(r_{i,j})$, forming a difference of convex (DC) form, which renders constraints (9b), (9d), and (9f) as non-convex. It is worth noting that constraint (4b) is recognized as a convex exponential cone due to the existence of the log function $R_{i,b}(\rho_{i,b})$, and can be easily approximated by a system of second order cone constraints, similar to the transformation done in [19] from (12) to (13) in Section III. Also, functions $L^{\text{mue}}_{i}(\boldsymbol{\rho}_i)$, $L^{\text{cld}}_{i,0}(f_{i,0})$, and $L^{\text{mig}}_{i,j}(r_{i,j})$ in (9c) and (9e) are convex functions of the form $f(x)=\frac{A}{x}$, and constraints (9c) and (9e) can be easily linearized by introducing a set of rotated quadratic cones. Those conversions ensure the resulting problem is a standard

Mixed-Integer Second-Order Cone Program (MISOCP), where a modern dedicated solver such as MOSEK [20] can solve the problem efficiently in each iteration. We address next the non-convex constraints, where we approximate (10) into a series of approximated MISOCP problems. Then, we develop a SCA-based MISOCP algorithm to iteratively solve until convergence.

**1) Constraints** (9g) **and** (9h)**:** After some Algebraic manipulation, function $r_{i,j}(\boldsymbol{p})$ can be rewritten as follows:

$$r_{i,j}(\boldsymbol{p}) = \underbrace{\sum_{b\in\mathcal{B}_1}\log\left(\overbrace{\sum_{l\in\mathcal{S}^{\setminus}}\sum_{k\in\mathcal{U}_l}p_{k,l,b}|h_{k,l,j,b}|^2+N_0+p_{i,j,b}|h_{i,j,j,b}|^2}^{\breve{\gamma}_{i,j,b}(\boldsymbol{p}_b)}\right)}_{\breve{r}_{i,j,b}(\boldsymbol{p}_b)} - \underbrace{\log\left(\overbrace{\sum_{l\in\mathcal{S}^{\setminus}}\sum_{k\in\mathcal{U}_l}p_{k,l,b}|h_{k,l,j,b}|^2+N_0}^{\hat{\gamma}_{j,b}(\boldsymbol{p}_b)}\right)}_{\hat{r}_{j,b}(\boldsymbol{p}_b)} \tag{11}$$

In order to resolve the DC form in (9g), $r_{i,j}(\boldsymbol{p})$ should be convexified. Thus, we are motivated by the inner-approximation method in [21] to approximate $\breve{r}_{i,j,b}(\boldsymbol{p}_b)$ by its upper-bounded convex function $\breve{R}_{i,j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)})$ around the point $\boldsymbol{p}_b^{(n)}$ as

$$\breve{R}_{i,j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)}) = \breve{r}_{i,j,b}(\boldsymbol{p}_b^{(n)}) + \frac{\breve{\gamma}_{i,j,b}(\boldsymbol{p}_b)-\breve{\gamma}_{i,j,b}(\boldsymbol{p}_b^{(n)})}{\breve{\gamma}_{i,j,b}(\boldsymbol{p}_b^{(n)})} \tag{12}$$

Similarly, to resolve the DC form in (9h), $r_{i,j}(\boldsymbol{p})$ should be made concave. We also approximate function $\hat{r}_{j,b}(\boldsymbol{p}_b)$ by its upper-bounded convex function $\hat{R}_{j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)})$ around the point $\boldsymbol{p}_b^{(n)}$. By replacing function $r_{i,j}(\boldsymbol{p})$ by its approximates, constraints (9g) and (9h) can now be written as

$$(z_{i,j}-1)M + \sum_{b\in\mathcal{B}_1}\left(\breve{R}_{i,j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)}) - \hat{r}_{j,b}(\boldsymbol{p}_b)\right) \le v_{i,j} \tag{13a}$$

$$v_{i,j} \le \sum_{b\in\mathcal{B}_1}\left(\breve{r}_{i,j,b}(\boldsymbol{p}_b) - \hat{R}_{j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)})\right) \tag{13b}$$

Constraints (13) are now convex due to the log functions, and constraints (13) can be easily linearized similar to (4b).

**2) Constraints** (9a)**,** (9b)**,** (9d)**, and** (9f)**:** First, By introducing the non-negative slack variables $\theta_{i,j}$, $\xi_{i,j}$, and $\tau_i$, constraints (9a), (9b), and (9d) can be equivalently rewritten as

$$(y_{i,j}-1)M + \underbrace{\frac{D_{i,j}}{\theta_{i,j}}}_{\breve{L}^{\text{sue}}_{i,j}(\theta_{i,j})} + L^{\text{cld}}_{i,j}(f_{i,j}) \le q_{i,j} \tag{14a}$$

$$q_{i,j} \le \underbrace{\frac{D_{i,j}}{\xi_{i,j}}}_{\hat{L}^{\text{sue}}_{i,j}(\xi_{i,j})} + L^{\text{cld}}_{i,j}(f_{i,j}) \tag{14b}$$

$$t_i \le \underbrace{\frac{D_{i,0}}{\tau_i}}_{\hat{L}^{\text{mue}}_{i}(\tau_i)} + L^{\text{cld}}_{i,0}(f_{i,0}) \tag{14c}$$

$$\theta_{i,j} \le \sum_{b\in\mathcal{B}_1}\left(\breve{r}_{i,j,b}(\boldsymbol{p}_b) - \hat{R}_{j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)})\right) \tag{14d}$$

$$\sum_{b\in\mathcal{B}_1}\left(\breve{R}_{i,j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)}) - \hat{r}_{j,b}(\boldsymbol{p}_b)\right) \le \xi_{i,j} \tag{14e}$$

$$\sum_{b\in\mathcal{B}_2} R_{i,b}(\rho_{i,b}) \le \tau_i \tag{14f}$$

where (14d) and (14e) are directly converted into their linear form following the approximations done in (12), and constraint (14a) can be easily linearized as it falls in the same category as constraints (9c) and (9e). Therefore, constraints (14b), (14c) and (14f) are non-convex and need to be addressed.

The non-convexity of (14f) is caused by the existence of concave function $R_{i,b}(\rho_{i,b})$ on the left side of the inequality, causing a DC form. To address (14f), we approximate function $R_{i,b}(\rho_{i,b})$ by its upper-bounded convex function $\tilde{R}_{i,b}(\rho_{i,b};\rho_{i,b}^{(n)})$ around the point $\rho_{i,b}^{(n)}$ as

$$\tilde{R}_{i,b}(\rho_{i,b};\rho_{i,b}^{(n)}) = R_{i,b}(\rho_{i,b}^{(n)}) + \frac{\rho_{i,b}-\rho_{i,b}^{(n)}}{\rho_{i,b}^{(n)}+||\mathbf{w}_{i,b}||^2N_0} \tag{15}$$

The non-convexity of constraints (9f), (14b), and (14c) is also due to a DC form. In fact, with a slight abuse of notation $x$, all of these functions have the same form, and can be represented by $f(x) = \frac{A}{x}$. To linearize the aforementioned constraints, we also approximate function $f(x)$ by its upper-bounded convex function $\tilde{f}(x;x^{(n)})$ around the point $x^{(n)}$ as $\tilde{f}(x;x^{(n)}) = \frac{A}{x^{(n)}} - \frac{A}{(x^{(n)})^2}(x-x^{(n)})$. At this point, constraints (9f), (14b), (14c), and (14f) can now be written in a linear form, by replacing functions $L^{\text{mig}}_{i,j}(r_{i,j})$, $\hat{L}^{\text{sue}}_{i,j}(\xi_{i,j})$, $L^{\text{cld}}_{i,j}(f_{i,j})$, $\hat{L}^{\text{mue}}_{i}(\tau_i)$, $L^{\text{cld}}_{i,0}(f_{i,0})$, and $R_{i,b}(\rho_{i,b})$ by their approximates.

**3) Constraints** (8b) **and** (8c)**:** For addressing function $E_{i,j}(\boldsymbol{p};\boldsymbol{\rho}_i)$, we introduce the non-negative slack variables $\zeta_{i,j,b}$, $\psi_{i,j,b}$, $\nu_{i,j,b}$, $\beta_{i,j,b}$, $\eta_{i,b}$, $\mu_{i,b}$, $\kappa_{i,b}$, and $\alpha_{i,b}$, and equivalently rewrite both constraints as

$$(y_{i,j}-1)M + s_j\sum_{b\in\mathcal{B}_1}\frac{\beta^2_{i,j,b}}{\zeta_{i,j,b}} + (1-s_j)\sum_{b\in\mathcal{B}_2}\frac{\alpha^2_{i,b}}{\eta_{i,b}} \le o_{i,j} \tag{16a}$$

$$o_{i,j} \le s_j\sum_{b\in\mathcal{B}_1}\underbrace{\left(\frac{\nu^2_{i,j,b}}{\psi_{i,j,b}}\right)}_{\hat{E}^{\text{sue}}_{ijb}(\nu_{ijb};\psi_{ijb})} + (1-s_j)\sum_{b\in\mathcal{B}_2}\underbrace{\left(\frac{\kappa^2_{i,b}}{\mu_{i,b}}\right)}_{\hat{E}^{\text{mue}}_{ib}(\kappa_{ib};\mu_{ib})} \tag{16b}$$

$$d_{i,j,b} \le \frac{\beta^2_{i,j,b}}{p_{i,j,b}} \tag{16c}$$

$$\phi_{i,b} \le \frac{\alpha^2_{i,b}}{\rho_{i,b}} \tag{16d}$$

$$\nu^2_{i,j,b} \le d_{i,j,b}p_{i,j,b} \tag{16e}$$

$$\kappa^2_{i,b} \le \phi_{i,b}\rho_{i,b} \tag{16f}$$

$$\zeta_{i,j,b} \le \breve{r}_{i,j,b}(\boldsymbol{p}_b) - \hat{R}_{j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)}) \tag{16g}$$

$$\breve{R}_{i,j,b}(\boldsymbol{p}_b;\boldsymbol{p}_b^{(n)}) - \hat{r}_{j,b}(\boldsymbol{p}_b) \le \psi_{i,j,b} \tag{16h}$$

$$\eta_{i,b} \le R_{i,b}(\rho_{i,b}) \tag{16i}$$

$$\tilde{R}_{i,b}(\rho_{i,b};\rho_{i,b}^{(n)}) \le \mu_{i,b} \tag{16j}$$

where (16g), (16h) and (16j) are converted into a linear form, following the approximations in (12), and (15). Constraints (16a) and (16e)–(16j) are convex, where (16a), (16e), and (16f) can be easily converted to quadratic cones. Constraints (16b), (16c), and (16d) are non-convex having a DC form.

With a slight abuse of notations $x$ and $y$, these functions

have the same form, and can be represented by $g(x) = \frac{x^2}{y}$. To linearize constraints (16b), (16c), and (16d), we approximate function $g(x)$ by its upper-bounded convex function $\tilde{g}(x, y; x^{(n)}, y^{(n)})$ around the points $x^{(n)}$ and $y^{(n)}$ as $\tilde{g}(x, y; x^{(n)}, y^{(n)}) = \frac{2x^{(n)}x}{y^{(n)}} - \frac{(x^{(n)})^2}{(y^{(n)})^2}y$. Constraints (16b), (16c), and (16d) can now be written in a linear form, after replacing the non-convex functions by their approximates.

### C. SCA-based Algorithm

The MISOCP approximation of problem (10) still pauses scalability limitations, due to the mixed-integer nature of the problem, which is caused by (7j). To solve that problem, we adapt a similar approach to [22], and relax the integrality constraint of the binary variables $\boldsymbol{y}$, $\boldsymbol{z}$, and $\boldsymbol{x}$. For instance, for relaxing variable $\boldsymbol{y}$, by defining slack variable $\delta^{\mathrm{y}}_{i,j} \geq 0$, we introduce the following constraints which will force $\boldsymbol{y}$ to take binary value with a penalty term added to the objective:

$$0 \leq y_{i,j} \leq 1 \tag{17a}$$

$$0 \leq y_{i,j} - \tilde{h}(y_{i,j}; y^{(n)}_{i,j}) \leq \delta^{\mathrm{y}}_{i,j} \tag{17b}$$

By employing all conic transformations, an approximated MI-SOCP of the mixed-integer non-convex problem (10), denoted by $\tilde{\mathcal{P}}^{(n)}$, can be formulated at the $n$th iteration as

$$\min_{\boldsymbol{\chi}} \sum_{j \in \{0 \cup \mathcal{S}\}} \sum_{i \in \mathcal{U}_j} \left( (1 - y_{i,j}) E^{\mathrm{loc}}_{i,j} + o_{i,j} + A\delta^{\mathrm{y}}_{i,j} \right) \tag{18a}$$

$$\text{s.t. (2), (4), (5), (8a), (9c), (9e), (10b), (10c),}$$

$$\text{(10d), (10e), (10f), (10g), (10h), (13), (14a),} \tag{18b}$$

$$\text{(14d)–(14e), (16a), (16e)–(16j), (17).} \tag{18c}$$

$$p_{i,j,b}, \rho_{i,b}, r_{i,j}, f_{i,j}, d_{i,j,b}, \phi_{i,b} \geq 0 \tag{18d}$$

where $A > 0$ is the penalty parameter.

By denoting $\boldsymbol{\Phi}^{(n)} = \{\boldsymbol{y}^{(n)}, \boldsymbol{z}^{(n)}, \boldsymbol{x}^{(n)}, \boldsymbol{p}^{(n)}, \boldsymbol{\rho}^{(n)}, \boldsymbol{\xi}^{(n)}, \boldsymbol{f}^{(n)}, \boldsymbol{\tau}^{(n)}, \boldsymbol{r}^{(n)}, \boldsymbol{\nu}^{(n)}, \boldsymbol{\psi}^{(n)}, \boldsymbol{\kappa}^{(n)}, \boldsymbol{\mu}^{(n)}, \boldsymbol{\beta}^{(n)}, \boldsymbol{\alpha}^{(n)}\}$ and $\boldsymbol{\chi}^{(n)} = \{\boldsymbol{y}^{\star}, \boldsymbol{z}^{\star}, \boldsymbol{x}^{\star}, \boldsymbol{p}^{\star}, \boldsymbol{\rho}^{\star}, \boldsymbol{\xi}^{\star}, \boldsymbol{f}^{\star}, \boldsymbol{\tau}^{\star}, \boldsymbol{r}^{\star}, \boldsymbol{\nu}^{\star}, \boldsymbol{\psi}^{\star}, \boldsymbol{\kappa}^{\star}, \boldsymbol{\mu}^{\star}, \boldsymbol{\beta}^{\star}, \boldsymbol{\alpha}^{\star}\}$, the pseudo-code for the corresponding SCA algorithm is given in Algorithm 1. We note that the convergence analysis has been omitted due to lack of space.

**Algorithm 1** SCA-based MISOCP Algorithm.

1: **Initialize:**
2: $n = 0$;
3: Choose an initial point $\Phi^{(n)}$;
4: **repeat**
5: Solve $\tilde{\mathcal{P}}^{(n)}$ to obtain the optimal solution at the $n$th iteration $\boldsymbol{\chi}^{\star}$;
6: Update $\boldsymbol{\Phi}^{(n)} = \boldsymbol{\chi}^{(n)}$;
7: $n = n + 1$;
8: **until** Convergence of the objective of $\tilde{\mathcal{P}}^{(n)}$.

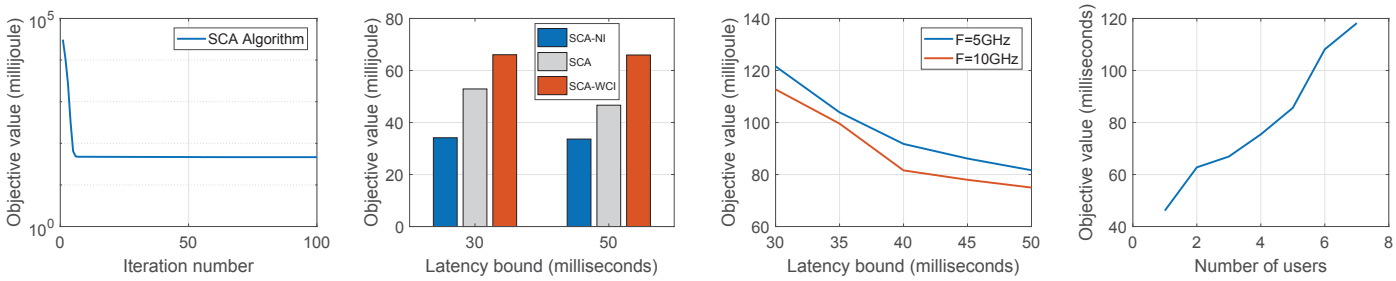
## V. Numerical Results

In this section, we present the numerical results for our joint transmission power and computational/radio resources allocation problem. We consider a heterogeneous network with a set of cloudlet-enabled SCs connected to a MC, having one user in the range of each cell. The channel gain $h$ follows an exponential distribution with mean 1. Both the beamforming vector norm $||\mathbf{w}||^2$ and the noise power level are normalized to 1. RB bandwidth is set to be 5 megahertz, data size $D$ is set to be 100 kilobits, computational demand $C$ is set to be 100 megacycles, latency bound $\overline{L}$ is set to be 50 milliseconds, cloudlets capacity $\overline{F}$ is set to be 10 gigahertz, and UEs power threshold $\overline{P}_{i,j}$ is set to be 30 dBm. For the other parameters, instance **A** uses 3 SCs and 10 RBs, instance **b** uses 15 SCs and 10 RBs, and instance **C** uses 6 SCs and 50 RBs. The convergence criteria of Algorithm 1 is established when $\epsilon$, i.e. the difference of objective value between $\Gamma^{(n)}$ and $\Gamma^{(n+1)}$ of the approximated problem, is $\epsilon \leq 10^{-3}$.

In Figure 2(a), we use instance A to study the convergence behavior of the SCA-based iterative algorithm. The algorithm starts with a very high objective (the total UEs energy consumption) due to the binary approximation in (17), causing the violation term to be very high at first. Within few iterations, the objective decreases rapidly, reaching small $\epsilon$ values, until convergence with objective equal to $46.466$ millijoule.

In Figure 2(b), we compare our SCA-based algorithm with the No-Interference (SCA-NI), and the Worst-Case Interference (SCA-WCI) approaches. The SCA-NI considers a given RB to not being used by any other SUE in the neighboring cells, so the interference on this block will amount to 0. On the other hand, SCA-WCI considers the highest possible interference on each RB, which results from the maximum transmission power over the strongest channel in each neighboring cell. Adopting one of these two approaches allows the algorithm to generally have a better performance and be more scalable. As it can be seen however, this comes with the cost of a degraded solution: the BCI and WCI approaches give lower and upper bounds on our approximate solution with a significant gap. This indicates that adopting the approximate solution without relaxations has a notable advantage over the other relaxation methods, since the lower and upper bounds are relatively large. The objective generally increases when the latency bound decreases, especially for the SCA-based algorithm. This is because a lower latency bound forces UEs to spend more power for decreasing the upload time to keep up with the required latency, incurring more transmission energy in the process. For the other relaxed approaches, they are less sensitive to latency bound changes. This is mainly because the interference amount is identical in both cases, causing the transmission rate to depend only on the signal strength, and hence causing it to change slightly, while in the SCA case the amount of interference will also be diminished, adding more effect on the transmission rate and hence energy. The change in the objective value will also be more significant for bigger instances including more users.

In figure 2(c), we study the effect of cloudlet computational capacity and UEs latency bound on the objective, while using instance A with RB bandwidth capacity $=$ 3 MHz. Here, we make two observations. First, having cloudlets with lower computation capacity leads to a higher objective. Second, increasing the latency bound leads to a lower objective. The first observation can be explained by the fact that lower computation




(a) Convergence of The Algorithm. (b) Comparing SCA, BCI, and WCI. (c) Objective vs capacity and latency. (d) Objective vs number of users.

Fig. 2: Performance Evaluation

capacity leads to a higher computation latency, forcing the upload latency to be smaller for keeping up with the latency bound. This means that UEs have to increase their transmission power in order to meet that latency, leading to a higher energy and hence a higher objective. Also, when there is enough capacity on the MBS cloudlet, UEs will migrate their task there incurring additional transmission latency in the process, and hence limiting the available task upload time. On the other hand, increasing the latency bound does the opposite effect, which means that UEs can occupy less computational resources, and also decrease their transmission power in a way that will minimize the overall consumed energy. It can be seen in the figure that the first part of the lines are steeper. We remark here that when the latency bound is too low, devices will be forced to choose local computation which will incur a higher energy, and hence the high objective in the figure.

Figure 2(d) shows how the number of users in the network affects the objective. We used instance B having 15 SCs serving one SUE each. The objective clearly increases whenever the number of users in the network goes up. This is because more users have to compute their task, and hence will consume energy that results from either local device execution, or from the allocated transmission power on the assigned resource blocks. In this case, NOs have to consider increasing cloudlets capacity in order to accommodate the number of users requests, and help in decreasing the UEs energy.

## VI. Conclusion

In this paper, we studied the problem of MEC computational and radio resources allocation for enabling IoT services in ultra-dense networks. Our objective was to decrease the total UEs energy consumption resulting from local computation and task transmission. We presented a low-complexity SCA-based algorithm for providing an approximate solution on the original problem. Through numerical results, we showed the efficiency of our solution, and performed multiple simulations following different scenarios. Our work can be further extended to consider other cases, like studying the effect of users mobility, as well as exploring the problem of load balancing among the cloudlets in a heterogeneous network.

## References


[1] M. Swan, "Sensor mania! the Internet of Things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.

[2] L. Atzori and al., "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] N. Fernando and al., "Mobile cloud computing: A survey," *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.

[4] M. Patel and al., "Mobile-edge computing introductory technical white paper," *White Paper, MEC industry initiative*, 2014.

[5] M. Satyanarayanan and al., "The case for VM-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, 2009.

[6] A. Damnjanovic and al., "A survey on 3GPP heterogeneous networks," *IEEE Wireless communications*, vol. 18, no. 3, 2011.

[7] M. Kamel and al., "Ultra-dense networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2522–2545, 2016.

[8] L. Tong and al., "A hierarchical edge cloud architecture for mobile computing," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[9] M.-H. Chen and al., "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3516–3520.

[10] X. Xu and al., "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16 406–16 415, 2017.

[11] C. Wang and al., "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[12] C. Wang, F. R. Yu, and al., "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[13] K. Zhang and al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[14] Z. Tan and al., "Heterogeneous services provisioning in small cell networks with cache and mobile edge computing," *arXiv preprint arXiv:1706.09542*, 2017.

[15] Y. Zhou and al., "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 339–11 351, 2017.

[16] M. Chen and al., "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[17] Q. H. Spencer and al., "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE transactions on signal processing*, vol. 52, no. 2, pp. 461–471, 2004.

[18] Y. Wen and al., "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2716–2720.

[19] K.-G. Nguyen and al., "Achieving energy efficiency fairness in multicell miso downlink," *IEEE Communications Letters*, vol. 19, no. 8, pp. 1426–1429, 2015.

[20] E. D. Andersen and al., "The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm," in *High performance optimization*. Springer, 2000, pp. 197–232.

[21] H. Tuy, "Outer and inner approximation," in *Convex Analysis and Global Optimization*. Springer, 1998, pp. 177–222.

[22] T. M. Nguyen and al., "A novel cooperative non-orthogonal multiple access (NOMA) in wireless backhaul two-tier hetnets," *IEEE Transactions on Wireless Communications*, 2018.