Joint Computation Offloading and Resource Allocation in Cloud Based Wireless HetNets

Nguyen Ti Ti and Long Bao Le

Abstract—In this paper, we study the joint computation of-floading and resource allocation problem in the two-tier wireless heterogeneous network (HetNet). Our design aims to optimize the computation offloading to the cloud jointly with the subchannel allocation to minimize the maximum (min-max) weighted energy consumption subject to practical constraints on bandwidth, computing resource and allowable latency for the multi-user multi-task computation system. To tackle this non-convex mixed integer non-linear problem (MINLP), we employ the bisection search method to solve it where we propose a novel approach to transform and verify the feasibility of the underlying problem in each iteration. In addition, we propose a low-complexity algorithm, which can decrease the number of binary optimization variables and enable more scalable computation offloading optimization in the practical wireless HetNets. Numerical studies confirm that the proposed design achieves the energy saving gains about 55% in comparison with the local computation scheme under the strict required latency of 0.1s.

Index Terms—Mobile edge computing, energy saving, computing and resource allocation, HetNet, MINLP, and ILP.

I. INTRODUCTION

The number of global devices has increased drastically in recent years. Moreover, with the multi-task processing capacity, advanced mobile devices have led to proliferation of many computation-intensive applications covering different areas including entertainment, communication, social networking, e-health, image recognition, language processing, and gaming. These computation-extensive applications have demanded more powerful central processing unit (CPU) with higher clock frequency, which will result in significant increase in the mobile energy consumption [1]. Moreover, advancement in mobile battery technology is usually not sufficiently fast to keep up with practical applications' requirements; therefore, the battery can become the bottleneck to improve the quality of experience (QoE) for mobile users. Consequently, reducing mobile energy consumption in power-hungry applications is of great importance and one very potential solution for this problem is to offload heavy computations tasks to the edge cloud servers using the so-called mobile cloud computing (MCC) technology.

Recent development of small-cell based wireless HetNets promises enormous benefits from both the network and mobile user perspectives. First, deployment of low-power small cells enables efficient reuse of the radio spectrum, which helps increase the spectral efficiency. Second, the close transmitter-receiver proximity allows small cell users to achieve high signal-to-noise ratio (SNR) even with low transmit power. This enables them to meet the low-latency requirements of many emerging applications. Finally, realization of the MCC

in small-cell based wireless HetNets can lead to significant benefits such as prolonging battery lifetime and providing high-speed and ultra-low latency communications services in future 5G wireless systems.

Several MCC platforms have been proposed and developed in the literature such as MAUI [2], CloneCloud [3], ThinkAir [4] and a good survey for them with the corresponding computation offloading designs can be found in [5]. In particular, the tradeoff between transmission and computation energy was studied in [6], [7]. The joint computation task offloading scheduling and transmit power allocation of a single-user system was investigated in [8]. Moreover, the authors in [9] studied the multi-user radio resource management problem for the HetNet-MCC system, which always offloads the entire computation task to the cloud. Dynamic computation offloading policies based on Lyapunov optimization were developed in [1], [10]. These existing works, however, only consider the single-cell setting and many practical design aspects of the multi-cell MCC system such as dynamic computation offloading, joint multi-user resource allocation and computing resource assignment, and consideration of practical constraints on bandwidth, operating frequency and tolerable delay limits are not satisfactorily accounted for. Our current work aims to fill this gap in the literature.

In this paper, we study the joint optimization problem for computation offloading and resource allocation where computation tasks are either processed locally at the mobile or offloaded and processed in the cloud. Moreover, offloaded tasks require radio resources for transmissions of the involved data (i.e., programming states). Importantly, this design is conducted for the multi-task multi-user multi-cell setting, which has not been addressed in the literature to the best of our knowledge. The underlying joint computation task, subchannel, and operating clock speed assignment problem, which aims to minimum the maximum weighted consumed energy subject to delay and resource constraints is a non-convex and difficult MINLP problem. Therefore, we employ the bisection search method to solve it where we transform the underlying non-convex INLP problem into an ILP for feasibility verification in each iteration. We also propose a low-complexity algorithm which is based on the decoupled optimization of the resource assignments for the macrocell and small cells.

The remaining of this paper is organized as follows. Section II presents the system model, computation and transmission energy models and problem formulation. Section III describes the solution method to solve the studied problem. Section IV evaluates the performance of proposed algorithms. Finally, Section V concludes the work.

T. T. Nguyen and L.B. Le are with INRS-EMT, University of Québec, Montréal, Québec, Canada. Emails: {titi.nguyen,long.le}@emt.inrs.ca.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a two-tier wireless HetNet with M small cells (SC), one macro cell (MC), and K users (UE). We assume that each cell is serviced by one base station (BS) connected to a common cloud provider via a high-speed fiber cable. Moreover, the cloud is assumed to have unlimited computing resource to serve offloaded computation demands from multiple SUs simultaneously. We denote the set of all BSs as $\mathcal{M}_0 = \{0, \mathcal{M}\}$, where 0 denotes the MC BS, and $\mathcal{M} = \{1, ..., M\}$ denotes the set of SC BSs, and the set of all users is denoted as $\mathcal{K} = \{\mathcal{K}_0, \mathcal{K}_1, ..., \mathcal{K}_M\}$, where $\mathcal{K}_0$ is the set of macro users (MUE) and $\mathcal{K}_m$ is the set of SC users (SUE) associated with BS $m \in \mathcal{M}$.

A. Computation Offloading Model

We assume that UE $k_m \in \mathcal{K}_m$, $m \in \mathcal{M}_0$ has the set of $\mathcal{L}_{k_m} = \{1, 2, ..., l_{k_m}\}$ independent tasks for his/her application and these tasks can be executed locally at the mobile device or offloaded and executed in the cloud independently over the time interval T where T is chosen to meet the delay requirement of the underlying application. Moreover, each task $l \in \mathcal{L}_{k_m}$ has the corresponding number of required CPU cycles $c_{k_m,l}$ (CPUs) and the number of transmission bits of $b_{k_m,l}$ (bits) (i.e., to transmit the involved programming states). We now introduce a binary offloading decision variable for each task $l \in \mathcal{L}_{k_m}$ as $x_{k_m,l}$, which can be defined as follows:

$$x_{k_m,l} = \begin{cases} 1, & \text{if task } l \text{ is executed at mobile device} \\ 0, & \text{if task } l \text{ is offloaded to the cloud} \end{cases}. \quad (1)$$

It is further assumed that the processor of UE $k_m \in \mathcal{K}_m$ has the DVFS (Dynamic Voltage and Frequency Scaling) capability so it can adjust the operating frequency clock speed (FCS) in the clock speed set $f_{k_m} \in \mathcal{F}_{k_m} = \{f_1^{k_m}, f_2^{k_m}, ..., f_{V_{k_m}}^{k_m}\}$ (CPU/s) which corresponds to the underlying application requirements [1]. By introducing V_{k_m} binary variables $\mu_v^{k_m}$, the operating clock speed can be expressed as $f_{k_m} = \sum_{v=1}^{V_{k_m}} \mu_v^{k_m} f_v^{k_m}$, where $\sum_{v=1}^{V_{k_m}} \mu_v^{k_m} = 1$. Then, the local computation energy in Joule at UE $k_m \in \mathcal{K}_m$ can be expressed as

$$E_c(k_m) = \frac{(\beta_{k_m,1}(f_{k_m})^{\beta_{k_m,2}} + \beta_{k_m,3}) \sum_{l \in \mathcal{L}_{k_m}} x_{k_m,l} c_{k_m,l}}{f_{k_m}}, \quad (2)$$

where $\beta_{k_m,1}, \beta_{k_m,2}, \beta_{k_m,3}$ denote the coefficients specified in the CPU model [1]. For mobile devices such as Samsung Galaxy Note or Nexus, the value of $\beta_{k,2}$ is in range of [2.7-3]. In order to satisfy the application QoS, UE k_m should complete its program execution within the delay $\tau_{1,k_m} \leq T$ if such program is executed locally at the mobile device. We assume that the cloud has the replicated version of the execution file of the offloading tasks, which can be, therefore, executed in the cloud in the offloading case.

B. Transmission Model

We assume that the available spectrum is partitioned into non-overlapping bands, which are then allocated to the MC and SC tiers to avoid the cross-tier interference. Moreover, the spectrum allocated to small cells is assumed to be fully reused over these cells (i.e., the SC spectrum reuse factor is

one). Moreover, the OFDMA scheme is assumed where the set of available sub-channels is denoted as $\mathcal{N} = \{1, 2, ..., N\}$.

Let $h_{k_j,m}^{(n)}$ denote the channel gain from UE $k_j \in \mathcal{K}_j$ to BS $m \in \mathcal{M}_0$ on subchannel $n \in \mathcal{N}$. The transmit power of user $k_j \in \mathcal{K}$ on each subchannel is assumed to be the constant $P_t W$, where W is the bandwidth of each subchannel, and the noise power density on subchannel n at BS m is $\sigma_m^{(n)}$. We represent the subchannel assignment by a binary variable $\rho_{k_m}^{(n)}$, where $\rho_{k_m}^{(n)} = 1$ if subchannel n is assigned to UE $k_m \in \mathcal{K}_m$, and $\rho_{k_m}^{(n)} = 0$, otherwise. Then, the transmission rate from UE k_m to the BS m can be expressed as

$$r_{k_m} = \sum_{n \in \mathcal{N}} \rho_{k_m}^{(n)} W \log_2\left(1 + \gamma_{k_m}^{(n)}\right), \quad (3)$$

where $\gamma_{k_m}^{(n)} = \begin{cases} \frac{P_t h_{k_m,m}^{(n)}}{\sigma_m^{(n)}}, & \text{if } m = 0 \\ \frac{P_t h_{k_m,m}^{(n)}}{\sum_{j \neq m} \sum_{k_j \in \mathcal{K}_j} \rho_{k_j}^{(n)} P_t h_{k_j,m}^{(n)} + \sigma_m^{(n)}}, & \text{if } m \neq 0 \end{cases}$.

The power P required to transmit data related to the program states from each mobile user comprises two parts, namely circuit power P_c (W/Hz) and transmit power P_t (W/Hz), which can be expressed as $P = P_c + P_t$. Then, the total required energy related to the transmission of UE $k_m \in \mathcal{K}_m$ can be computed as follows:

$$E_t(k_m) = t_{k_m} P W \sum_{n \in \mathcal{N}} \rho_{k_m}^{(n)}, \quad (4)$$

where t_{k_m} is the transmission time of the program states from the mobile to its BS, which can be calculated as

$$t_{k_m} = \frac{1}{r_{k_m}} \sum_{l \in \mathcal{L}_{k_m}} (1 - x_{k_m,l}) b_{k_m,l}. \quad (5)$$

The total latency experienced by an offloaded task comprises the time required for sending program states/bits to the cloud, the computation time in the cloud, and the time required for downloading the results to the mobile. However, cloud computation time is relatively small due to the high cloud computation power and the data related to computation results' download has much smaller size compared to the offloading data in general. Therefore, we neglect the cloud energy consumption and data download transferred time. Moreover, to ensure the constrained latency, the transmit time t_{k_m} is required to be smaller than the maximum delay, i.e., $\tau_{2,k_m} < T$.

C. Problem Formulation

We now present the formulation for the considered problem where our design objective is to minimize the maximum weighted users' consumed energy. The energy weight, denoted as w_{k_m}, represents the priorities or the battery/computation levels of different users [11]. Then, the joint computation offloading and resource allocation problem with latency, radio and computational resource constraints can be stated as

$$(\mathcal{P}_1) \quad \min_{\boldsymbol{\rho}, \boldsymbol{x}, \boldsymbol{\mu}} \max_{k_m} w_{k_m}(E_c(k_m) + E_t(k_m))$$

subject to

$$(C1): t_{k_m} \leq \tau_{2,k_m}, \ \forall m \in \mathcal{M}_0, \forall k_m \in \mathcal{K}_m$$

$$(C2): \sum_{k_m \in \mathcal{K}_m} \rho_{k_m}^{(n)} + \sum_{k_0 \in \mathcal{K}_0} \rho_{k_0}^{(n)} \leq 1, \forall m \neq 0, \forall n \in \mathcal{N}$$

$(C3): \rho_{k_m}^{(n)} \in \{0,1\}, \forall m \in \mathcal{M}_0, \forall k_m \in \mathcal{K}_m, \forall n \in \mathcal{N}$

$(C4): \dfrac{\sum\limits_{l \in \mathcal{L}_{k_m}} x_{k_m,l} c_{k_m,l}}{f_{k_m}} \leq \tau_{1,k_m}, \ \forall m \in \mathcal{M}_0, \forall k_m \in \mathcal{K}_m$

$(C5): x_{k_m,l} \in \{0,1\}, \ \forall m \in \mathcal{M}_0, \forall k_m \in \mathcal{K}_m, \forall l \in \mathcal{L}_{k_m}$

$(C6): f_{k_m} = \sum\limits_{v=1}^{V_{k_m}} \mu_v^{k_m} f_v^{k_m}, \ \mu_v^{k_m} \in \{0,1\}, \forall k_m \in \mathcal{K}_m$

$(C7): \sum\limits_{v=1}^{V_{k_m}} \mu_v^{k_m} = 1, \forall k_m \in \mathcal{K}_m.$

In this problem formulation, constraint (C1) captures the transmission latency requirements for offloading process. Constraint (C2) and (C3) represent the MC and SCs bandwidth sharing where each subchannel can be allocated to at most one MUE or one SUE in each SC. Constraint (C4) represents the delay requirements for local computation. Furthermore, constraint (C5) captures the binary offloading decisions while the remaining constraints express the computational capacity of mobile devices.

III. ALGORITHM DEVELOPMENT

The considered problem $(\mathcal{P}_1)$ is indeed a non-convex INLP due to the integer optimization variables for allocating tasks, frequency clock speeds' selection and subchannel assignments and due to the non-convexity of the objective function and constraint functions in (C1). Therefore, this problem is very difficult to solve. To have an insightful description, we first reformulate the min-max objective of $(\mathcal{P}_1)$ as follows:

$$(\mathcal{P}_2) \ \min \zeta$$
subject to
$$(C8): w_{k_m}(E_c(k_m) + E_t(k_m)) \leq \zeta, \ \forall k_m$$
$$(C1)-(C7).$$

A. Proposed Algorithm (Optimal alg.)

We can now apply the bisection search method to find the optimal min-max users' energy consumption for the reformulated problem $(\mathcal{P}_2)$. Specifically, the bisection search method iteratively updates an upper-bound $\zeta_{\max}$ and a lower-bound $\zeta_{\min}$ of the objective value ζ of problem $(\mathcal{P}_2)$. In particular, in each iteration, we have to verify the feasibility of problem $(\mathcal{P}_2)$ for a given value of ζ based on which we can update $\zeta_{\max}$ and $\zeta_{\min}$. If the set of constraints is feasible, then upper-bound of objective function will decrease, and inversely its lower-bound will increase. This algorithm will terminate when the difference between upper-bound and lower-bound values becomes sufficiently small. The proposed algorithm which can find the optimal solution of $(\mathcal{P}_2)$ is given in Algorithm 1.

In order to verify the feasibility of problem $(\mathcal{P}_2)$, we take three major steps to transform all constraints of problem $(\mathcal{P}_2)$ into the linear form. In the first step, we linearize the involved logarithmic functions in (C1) and (C8). In the second step, we attempt to determine whether UEs can locally process their tasks or not for a given value of ζ. In the final step, we introduce some further auxiliary variables to transform the product-form of the obtained constraint functions into the

---

Algorithm 1 Multi-task and Multi-user Computation Offloading and Resource Allocation

1: **Initialize**: choose ϵ, $\zeta_{\min} = 0$ and $\zeta_{\max} = \max\limits_{m \in \mathcal{M}, k_m \in \mathcal{K}_m} E_c(k_m) | \{x_{k_m,l} = 1, \forall l \in \mathcal{L}_{k_m}\}$.
2: **while** $\zeta_{\max} - \zeta_{\min} < \epsilon$ **do**
3:     Compute $\zeta = (\zeta_{\max} + \zeta_{\min})/2$.
4:     **for** each user k_m **do**
5:         **if** optimal value of $\mathcal{P}_{s1}^{k_m} = 0$ **then**
6:             neglect user k_m.
7:         **else**
8:             Assign $x_{k_m,l} = 0$ as in *Proposition 2*.
9:         **end if**
10:     **end for**
11:     Check the feasibility of $(\mathcal{P}_2)$ for users (optimal value of $\mathcal{P}_{s1}^{k_m} \neq 0, \forall k_m$) with the set of available subchannels $\mathcal{N}$ as in Section III.A.3.
12:     **if** feasibility **then**
13:         Assign $\zeta_{\max} = \zeta$.
14:     **else**
15:         Assign $\zeta_{\min} = \zeta$.
16:     **end if**
17: **end while**

---

desirable linear form. The obtained linear program after step three can then be solved effectively.

These steps are described in more details for a given value of ζ in the following.

1) Step one: We introduce some auxiliary binary variables as follows:

$$\alpha_{k_1,k_2,\ldots,k_M}^{(n)} = \begin{cases} 1, & \text{if } \prod_{m \in \mathcal{M}} \rho_{k_m}^{(n)} = 1, \ k_m \in \mathcal{K}_m \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

The above expression means that the variable $\alpha_{k_1,k_2,\ldots,k_M}^{(n)}$ will be active if users $k_1 \in \mathcal{K}_1, k_2 \in \mathcal{K}_2, \ldots, k_M \in \mathcal{K}_M$ transmit on the same subchannel n. We have to now re-write $E_t(k_m)$ in (4), which depends on t_{k_m} given in (5). Toward this end, the transmission rate from SUE $k_m \in \mathcal{K}_m$ to BS $m \in \mathcal{M}$ is re-expressed in (7) which is needed in the expression of t_{k_m}. We also need (9) to re-write $E_t(k_m)$ in (4). Moreover, constraint (C2) can be now rewritten as in (8).

2) Step two: For a given ζ, UEs will not offload their tasks if they can process all tasks locally. Therefore, to determine whether UEs offload or not, we find the minimum number of transmission bits of UE k_m as $\sum_{l \in \mathcal{L}_K} (1 - x_{k_m,l}) b_{k_m,l}$ when its computing energy is less than ζ. If this value is equal to zero, UE k_m can locally execute its application; therefore the transmission rate r_{k_m} will be qual to zero. This problem is formulated as follows:

$$(\mathcal{P}_{s1}^{k_m}) \ \min_{\boldsymbol{x}_{k_m}, \boldsymbol{\mu}_{k_m}} \sum_{l \in \mathcal{L}_{k_m}} (1 - x_{k_m,l}) b_{k_m,l}$$

subject to
$$(C9): w_{k_m} E_c(k_m) \leq \zeta,$$
$$(C4)-(C7).$$

To solve problem $\mathcal{P}_{s1}^{k_m}$, we re-express one term in computation energy expression as follows:

$$r_{k_m} = \sum_{n \in \mathcal{N}} \sum_{k_1 \in \mathcal{K}_1} \cdots \sum_{k_{m-1} \in \mathcal{K}_{m-1}} \sum_{k_{m+1} \in \mathcal{K}_{m+1}} \cdots \sum_{k_M \in \mathcal{K}_M} \alpha_{k_1,k_2,\ldots,k_M}^{(n)} W \log_2(1 + \frac{P_t h_{k_m,m}^{(n)}}{\sum_{j \in \mathcal{M} \backslash m} P_t h_{k_j,m}^{(n)} + \sigma_m^{(n)}}). \tag{7}$$

$$\sum_{k_1 \in \mathcal{K}_1} \cdots \sum_{k_M \in \mathcal{K}_M} \alpha_{k_1,k_2,\ldots,k_M}^{(n)} + \sum_{k_0 \in \mathcal{K}_0} \rho_{k_0}^{(n)} \quad \leq \quad 1, \quad \forall n \in \mathcal{N}. \tag{8}$$

$$\sum_{n \in \mathcal{N}} \rho_{k_m}^{(n)} = \sum_{n \in \mathcal{N}} \sum_{k_1 \in \mathcal{K}_1} \cdots \sum_{k_{m-1} \in \mathcal{K}_{m-1}} \sum_{k_{m+1} \in \mathcal{K}_{m+1}} \cdots \sum_{k_M \in \mathcal{K}_M} \alpha_{k_1,k_2,\ldots,k_M}^{(n)}. \tag{9}$$

$$r_{k_m} \sum_{v=1}^{V_{k_m}} \sum_{l \in \mathcal{L}_{k_m}} \mu_v^{k_m} F_v^{k_m} x_{k_m,l} c_{k_m,l} + \sum_{l \in \mathcal{L}_{k_m}} (1 - x_{k_m,l}) \, b_{k_m,l} PW \sum_{n \in \mathcal{N}} \rho_{k_m}^{(n)} \leq \frac{\zeta r_{k_m}}{w_{k_m}}, \text{if } r_{k_m} > 0, \ \forall k_m. \tag{10}$$

$$\frac{(\beta_{k_m,1}(f_{k_m})^{\beta_{k_m,2}} + \beta_{k_m,3})}{f_{k_m}} = \sum_{v=1}^{V_{k_m}} \mu_v^{k_m} F_v^{k_m}, \tag{11}$$

where $F_v^{k_m} = \frac{(\beta_{k_m,1}(f_v^{k_m})^{\beta_{k_m,2}} + \beta_{k_m,3})}{f_v^{k_m}}$ for $f_v^{k_m} > 0$ and $F_v^{k_m} = 0$ for $f_v^{k_m} = 0$. The constraint (C9) now is the sum of the product of two binary variables, which can be given as

$$(\sum_{v=1}^{V_{k_m}} \mu_v^{k_m} F_v^{k_m})(\sum_{l \in \mathcal{L}_{k_m}} x_{k_m,l} c_{k_m,,l}) \leq \frac{\zeta}{w_{k_m}}. \tag{12}$$

We now deal with the non-convex term $z_{k_m,v,l} = \mu_v^{k_m} x_{k_m,l}$ in (12). In general, the product of binaries variables can be converted to the linear inequalities as suggested in [12]. Particularly, the 0/1-variable $y = \prod_{i=1}^{n} s_i$ can be expressed equivalently as

$$\begin{cases} y & \in \{0,1\}, \ s_i \in \{0,1\}, \forall i \\ y & \geq \sum_{i=1}^{n} s_i - n + 1 \\ y & \leq \min\{s_i\} \end{cases}. \tag{13}$$

Then applying (13), we can transform the non-convex term $z_{k_m,v,l} = \mu_v^{k_m} x_{k_m,l}$ to linear form of $z_{k_m,v,l}, \mu_v^{k_m}$ and $x_{k_m,l}$. In addition, constraints (C4) can be easily converted to a linear form as: $\sum_{l \in \mathcal{L}_{k_m}} x_{k_m,l} c_{k_m,,l} - \tau_{1,k_m} \sum_{v=1}^{V_{k_m}} \mu_v^{k_m} f_v^{k_m} \leq 0$.

Using these expressions, the considered problem can be transformed into an ILP with optimization variables $\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\mu}$ which can be solved effectively by using the interior-point method or the solver CVX-Gurobi [12].

3) Step three - feasibility verification for problem $\mathcal{P}_2$: We now state some important results in the following two propositions, which correspond to two cases where the offloading decision variables $x_{k_m,l}$ are zero and one, respectively.

Proposition 1: If there exists a feasible solution for $\mathcal{P}_2$ given ζ and the optimal value of $\mathcal{P}_{s_1}^{k_m}$ is 0, then the offloading decision variables $x_{k_m,l}, \forall l \in \mathcal{L}_{k_m}$ are set equal to 1 and this will form a feasible solution.

Proof. It is clear that the efficient optimization for x_{k_m}, μ_{k_m} in $\mathcal{P}_2$ must allocate the smallest amount of radio resources to meet the fixed energy level ζ. In fact, if any user, who has computing energy less than ζ and has execution time satisfying the computing delay time, offloads its tasks to

the cloud, it will occupy the radio resources of other users demanding for computation offloading, which results in the increase of transmit energy of those users. This proves the proposition. ∎

Proposition 2: If there exists a feasible solution for $\mathcal{P}_2$ given ζ and the optimal value of $\mathcal{P}_{s_1}^{k_m}$ is positive, then the offloading decision variable $x_{k_m,l}$ is set equal to 0 if the optimal solution $x_{k_m,l}$ of $\mathcal{P}_{s_1}^{k_m}$ equals to 0 and this will form a feasible solution.

Proof. If the optimal value of $\mathcal{P}_{s_1}^{k_m}$ is greater than 0, it means that the local computational energy must be less than $\frac{\zeta}{w_{k_m}}$ since the total computation and transmission energy must be less than or equal to $\frac{\zeta}{w_{k_m}}$. Because the objective of $\mathcal{P}_{s_1}^{k_m}$ is to minimize the number of transmission bits, UE k_m will consume the least transmission energy with a given radio resource. Therefore, the task offloading decision variables must be set equal to 0 (offload to cloud) if the solution of task allocation of $\mathcal{P}_{s_1}^{k_m}$ is equal to 0. Note that when the optimal value of $\mathcal{P}_{s_1}^{k_m}$ is positive, the transmission rate r_{k_m} must be greater than 0 to offload data to the cloud. Therefore, we can rewrite the fractional constraint functions of (C8) into non-fractional form as (10). ∎

Using the results in *Proposition 1*, we can set the zero rate for users with $x_{k_m,l}$ equal to one for all tasks l. For remaining users whose optimal $x_{k_m,l}$ are equal zero for at least one task according to *Proposition 2*, we solve problem $\mathcal{P}_{s2}$ to determine the computation offloading and resource allocation solution.

The remaining thing is to verify the feasibility of the resulting equivalent problem of the original problem $\mathcal{P}_2$. Applying the same technique as in (13) for UEs having transmission rate $r_{k_m} > 0$, we transform the product-form of (10) into linear-form of $x_{k_m,l}$, $\mu_v^{k_m}$, $\alpha_{k0,0}^{(n)}$, $q_{k_0,v,l}^{(n)}$, $u_{k_0,l}^{(n)}$, $\alpha_{k_1,\ldots k_M}^{(n)}$, $q_{k_1,k_2,\ldots,k_M,v,l}^{(n)}$ and $u_{k_1,k_2,\ldots,k_M,l}^{(n)}$, where $q_{k_1,k_2,\ldots,k_M,v,l}^{(n)} = \mu_v^{k_m} x_{k_m,l} \alpha_{k_1,k_2,\ldots,k_M}^{(n)}$, $u_{k_1,k_2,\ldots,k_M,l}^{(n)} = x_{k_m,l} \alpha_{k_1,k_2,\ldots,k_M}^{(n)}$ for SUEs, $q_{k_0,v,l}^{(n)} = \mu_v^{k_0} x_{k_0,l} \rho_{k_0}^{(n)}$, and $u_{k_0,l}^{(n)} = x_{k_0,l} \rho_{k_0}^{(n)}$ for MUEs.

Using these transformations, the set of constraints of problem $(\mathcal{P}_2)$ can be converted to the linear form of $x_{k_m,l}$, $\mu_v^{k_m}$, $\alpha_{k0,0}^{(n)}$, $q_{k_0,v,l}^{(n)}$, $u_{k_0,l}^{(n)}$, $\alpha_{k_1,\ldots k_M}^{(n)}$, $q_{k_1,k_2,\ldots,k_M,v,l}^{(n)}$ and $u_{k_1,k_2,\ldots,k_M,l}^{(n)}$. Therefore, the feasibility verification of the transformed problem can be done effectively by a standard solver.

B. Low-complexity Algorithm with Decoupled MC-SC Optimization (LC alg.)

We now present a low-complexity algorithm, which can perform well in large-scale wireless HetNets. In this algorithm, we first determine the minimum number of subchannels so that all MC UEs (MUEs) can meet the energy consumption level ζ, which can be stated as follows:

$$(\mathcal{P}_{s2}) \quad \min \sum_{n \in \mathcal{N}} \sum_{k_0 \in \mathcal{K}_0} \rho_{k_0}^{(n)}$$

subject to

$$(\text{C1}), (\text{C3}) - (\text{C8}), \text{ for } m = 0.$$

After solving problem $\mathcal{P}_{s2}$ by using the above transformations for MUEs, the set of remaining subchannels that SC UEs (SUEs) can use can be written as $\mathcal{N}_{SC} = \mathcal{N} \setminus \{n | \rho_{k_0}^{(n)}{}_{(\mathcal{P}_{s2})} = 1, \forall k_0 \in \mathcal{K}_0\}$.

We can then allocate these remaining subchannels ($n \in \mathcal{N}_{SC}$) by solving problem $\mathcal{P}_2$ with only SUEs $k_m \in \mathcal{K}_m, \forall m \in \mathcal{M}$ (i.e., we remove any terms related to MUEs in this problem). This problem can be solved by using the proposed Algorithm 1 for only SUEs with the available subchannels $\mathcal{N}_{SC}$. The number of optimization variables in this algorithm now decreases $\frac{|\mathcal{N}|}{|\mathcal{N}_{SC}|}$ times in compassion with the case where the joint subchannel allocation optimization for MC and SCs is conducted.
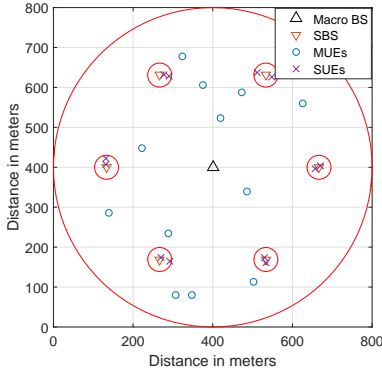
IV. NUMERICAL RESULTS

Fig. 1: Network topology and user placement in the numerical examples.

The network setting used in our simulations is shown in Fig.1, where there are 12 MUEs in macrocell and 2 SUEs for each small cell. The MC and SC coverage radius are 400m and 30m, respectively. All UEs have 11 levels of operating clock frequency uniformly chosen in $0 - 2$ GHz. For convenience, we assume that all users has 3 tasks and total CPU requirement for each user is 0.2 Gcycles. The maximum tolerable computing delay is set equal to T for all users while the transmission delay is set randomly in $0.7T - 0.9T$. The number of transmission bits/task and CPU cycles/task are illustrated in Fig.2, which are used in scenario 1 (presented in Fig. 3) in which the ratio of $b_{k_m,l}/c_{k_m,l}$ is chosen randomly in $10^{-5} - 10^{-3}$ (as in Fig. 2). The energy weight is set randomly in $0.8 - 1$. The energy coefficients are set for all users as $\beta_{k_m,1} = 0.34(10^{-9})^{\beta_{k_m,2}}$, $\beta_{k_m,2} = 3$ and $\beta_{k_m,3} = 0.35$ [1].
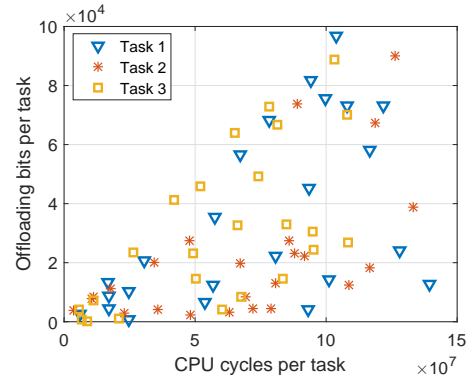

Fig. 2: Computation - transmission relation of computation tasks.

We set the number of subchannels as 20, and the bandwidth per subchannel is 180 kHz, and the noise power density equal to -140 dBm/Hz. The transmission power P_t is set equal to -33 dBm/Hz and -43 dBm/Hz for MUEs and SUEs, respectively and the circuit power $P_c = P_t/2$. The subchannel gains are generated according to $h_{k_m,m}^{(n)} = \xi^{(n)} g_{k_m,m}$ where $\xi^{(n)}$ is a random value generated according to the exponential distribution and $g_{k_m,m}$ denotes the pathloss defined according 3GPP technical report as $g_{k_0,0} = -128.1 - 37.6 \log 10(d_{k_0,0})$ (dB) for MUEs and $g_{k_m,m} = -127 - 30 \log 10(d_{k_m,m}), \forall m \neq 0$ (dB) for SUEs [13] where $d_{k_m,m}$ is the geographical distance between UE k_m and BS m (km). The stop condition of bisection search is set as $\epsilon = 10^{-3}$.
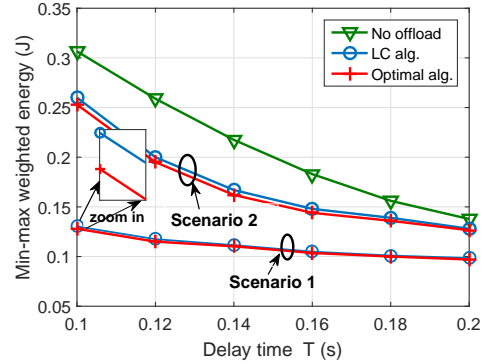

Fig. 3: Min-max weighted energy consumption.

Fig. 3 shows the min-max weighted energy required to execute the applications of all UEs, which is obtained by averaging the result over 15 system realizations under the proposed optimal (Optimal alg.) and low-complexity (LC alg.) and no computation offload (No offload). The ratio of bits per CPU cycle (BPC) in scenario 2 is 50 percent higher than that in scenario 1. These results show that the smaller the BPC, the smaller the consumed energy for all schemes. For the computation load of 0.2 Gcycles per user, the min-max weighted energy without offloading is much higher than that due to the proposed schemes under both scenarios. In particular, the proposed optimal algorithm can reduce the energy about 55% compared with the "No offload" scheme in scenario 1. Moreover, the energy consumption in the LC scheme is nearly equal to the global optimal solution due to the "Optimal alg.".
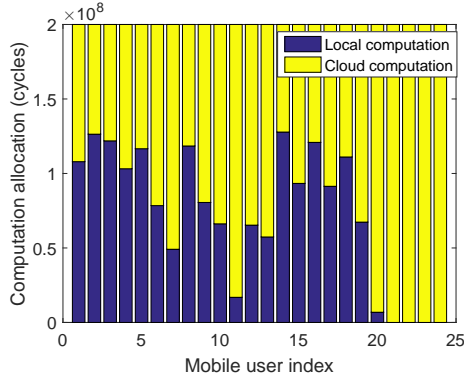
Fig. 4: Computation allocation with latency of 0.1s.

The computation allocation using LC alg. for different users with $T = 0.1$s in one system realization is illustrated in Fig. 4. This figure shows that the computation load is distributed fairly equal between users thanks to min-max weighted energy design objective. Moreover, some UEs, having small BPC tasks or high SINR ratio, offload all tasks to the cloud. This figure also shows that the worst UE can offload 0.06 (Gcycles), then its FCS decreases about 30 percent leading to the decrease of computation energy by nearly 2.4 times.
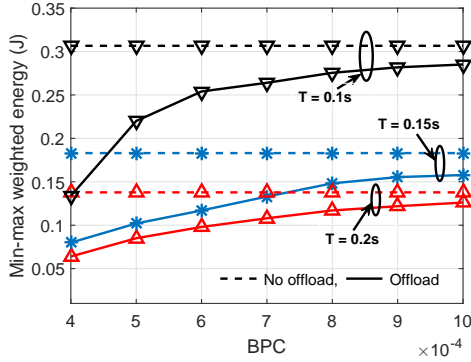

Fig. 5: Min-max weighted energy versus bits per 1 CPU cycle.

Fig. 5 shows the min-max weighted energy consumption when the number of cycles per task is fixed by $0.2/3$ (Gcycles) while the BPC is set the same for all UEs. When this parameter is small, the performance gap in terms of min-max energy consumptions between the proposed offloading ("offload") design and "No offload" scheme is quite large. This means that the proposed scheme can result in great energy reduction.
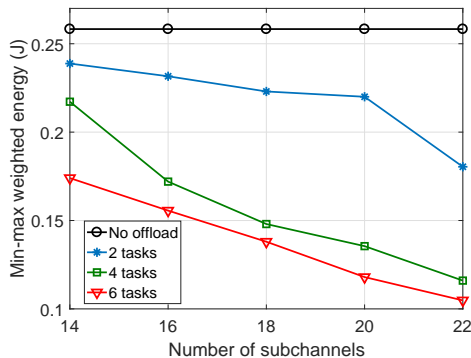

Fig. 6: Min-max weighted energy consumption for 0.2 Gcycles computation load.

Fig. 6 shows the achievable min-max weighted energy for scenarios where each user has 2, 4 and 6 tasks and the total computation load is 0.2 Gcycles per user, maximum tolerable latency $T = 0.12s$ while the BPC is fixed at 5×10^{-4}. It can be observed that the min-max weighted energy decreases quite drastically as the number of tasks increases. Moreover, if there are two tasks then users send at least one task to the cloud until $N = 22$ subchannels while if there are 4 tasks then users can send at least one task to the cloud when $N \geq 16$. However, in all cases, if sufficient radio resources are available, the UE with largest weighted energy prefers to offload first to achieve the lowest min-max weighted energy. Therefore, the "No offload" have the worst performance in term of energy consumption comparing with the proposed schemes.

V. CONCLUSION

In this paper, we have proposed a general framework for multi-task multi-user multi-cell computation offloading. Considering the practical discontinuity of operating frequency clock speed of real-world chipsets, and the partitioning of computation load into individual tasks, we have formulated the problem which minimizes the maximum weighted energy consumption while maintaining the application latency requirement. We have then developed the optimal and low-complexity algorithms to tackle this problem. Numerical results have confirmed the desirable performance of the proposed algorithms for wireless HetNets which can lead to great saving of the energy consumption.

REFERENCES

[1] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 2510–2523, Dec. 2015.

[2] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM MobiSys*, 2010, pp. 49–62.

[3] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. ACM EuroSys*, 2011, pp. 301–314.

[4] S. Kosta, A. Aucinas, H. Pan, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, 2012, pp. 945–953.

[5] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Sig. Process. Mag.*, vol. 31, pp. 45–55, Nov. 2014.

[6] N. T. Ti and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," *IEEE ICC*, pp. 1–6, 2017.

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

[8] K. B. L. Yuyi Mao, Jun Zhang, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," *IEEE WCNC*, pp. 1–6, 2017.

[9] L. C. S. Z. Jie Xu, Yuxuan Sun, "E2M2: energy efficient mobility management in dense small cells with mobile edge computing," 2017. [Online]. Available: http://arxiv.org/abs/1701.07363

[10] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, pp. 1991–1995, 2012.

[11] Y. Guo, L. Duan, and R. Zhang, "Optimal pricing and load sharing for energy saving in communications cooperation," in *IEEE Trans. Wireless Commun.*, vol. 15, Feb. 2016, pp. 951 – 964.

[12] J. Lee and S. Leyffer, *Mixed Integer Nonlinear Programming*, 1st ed., ser. The IMA Volumes in Mathematics and its Applications 154. Springer-Verlag New York, 2012.

[13] 3GPP-TR-36.814, "Evolved universal terrestrial radio access (E-UTRA); further advancements for E-UTRA physical layer aspects (release 9)," Tech. Rep., 2010.