Computation Offloading and Resource Allocation for Backhaul Limited Cooperative MEC Systems

Phuong-Duy Nguyen[†], Vu Nguyen Ha[‡], and Long Bao Le[†]

[†]INRS-EMT, University of Quebec, Montreal, Quebec, Canada; emails: {phuongnguyen,le}@emt.inrs.ca
[‡]École Polytechnique de Montréal, Montreal, Quebec, Canada; email: vu.ha-nguyen@polymtl.ca.

Abstract—In this paper, we jointly optimize computation offloading and resource allocation to minimize the weighted sum of energy consumption of all mobile users in a backhaul limited cooperative MEC system with multiple fog servers. Considering the partial offloading strategy and TDMA transmission at each base station, the underlying optimization problem with constraints on maximum task latency and limited computation resource at mobile users and fog servers is non-convex. We propose to convexify the problem exploiting the relationship among some optimization variables from which an optimal algorithm is proposed to solve the resulting problem. We then present numerical results to demonstrate the significant gains of our proposed design compared to conventional designs without exploiting cooperation among fog servers and a greedy algorithm.

I. INTRODUCTION

Mobile Edge Computing (MEC) has been considered very potential in overcoming the limited computing resource constraint of mobile users (MUs) [1] where MUs can offload partially or fully their computation load (CL) to the fog servers (FSs) at the network edge, which will be called partial and binary offloading, respectively. Such computation offloading (CO) can result in smaller energy consumption and/or better execution latency [2], [3]. However, CO typically requires transmission of some incurred data from MUs to the fog servers. From the MU's perspective, CO can reduce the local CL but it also consumes extra energy for the data transmission. Therefore, joint CO decision and resource allocation (RA) becomes a major challenge in MEC systems which has been studied in several recent works [3]–[10]. In particular, the papers [3]–[6] consider MEC systems in which one FS serves multiple MUs. Moreover, [4] designs a binary CO scheme to improve energy-efficiency considering radio interference while [3] proposes fair binary CO algorithms aiming to minimize the maximum weighted energy consumption. The partial CO strategy is considered in [5], [6]. Both papers focus on minimizing the total energy consumption of all MUs with the delay constraint where [5] investigates the offloading for the frequency division multiple access (FDMA) while [6] studies the offloading for both OFDMA and TDMA scenarios.

A joint CO and RA design is recently investigated in multiple-FS MEC systems where a group of FSs is available to help the MUs reduce their CLs [8]–[11]. In particular, [8] studies the fair partial CO for the multiple-FS MEC system to minimize the task execution delay. In this work, a task from an MU is first partitioned into multiple subtasks which are offloaded independently to different FSs within their coverage using the corresponding wireless links. In [9], a binary CO strategy in the multi-fog system is proposed where CL from an MU can be executed at its FS or at one of its neighboring access points. Similarly, the authors in [10] propose a joint CO and RA framework which maximizes the weighted sum of the reductions in task completion time and energy consumption. These papers do not consider cooperation among the FSs. Considering FSs' cooperation, [11] studies the multiple-cloud load balancing problem for the system with limited backhaul capacity where minimization of the maximum CL of all cloud servers is the design objective. However, optimization of the CO decisions is not addressed in this work.

Our current paper studies the joint CO and RA in a cooperative MEC system with multiple FSs where cooperative computation load balancing among different FSs is considered to efficiently serve the offloaded CLs. Specifically, an overloaded FS can offload its CL to neighboring FSs through the capacity limited backhaul links. Our design aims at minimizing the weighted sum of energy consumption considering constraints on the maximum task execution latency and computation resource (CR) at the MUs and FSs as well as the backhaul capacity. The design can be applied to any multi-FS MEC systems with arbitrary backhaul network topologies. The optimization problem is formulated as a non-convex problem which is hard to solve due to the complex coupling among the variables. To overcome this challenge, we first study the relationship among the variables based on which the problem is convexified by using the penalty method. Then, a duality-based algorithm is proposed to solve the problem optimally. For comparison purpose, a greedy algorithm is also discussed. Numerical results are then presented to illustrate the various insights and significant performance gains compared to conventional design without exploiting cooperation among FSs and the greedy algorithm.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a multi-FS system consisting S FSs deployed at S one-antenna base stations (BS). Each BS is connected with a corresponding co-located FS which provide wireless communication and edge computation services to MUs inside its coverage. Let $\mathcal{S}$ denote the set of all FSs (or BSs), and $\mathcal{K}_s$ denote the set of K_s MUs served by FS/BS s. Furthermore, it is assumed that each FS can offload its CL to other FSs in a FS set through the corresponding backhaul links. The helping
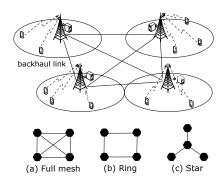
Fig. 1. Backhaul-limited cooperative MEC systems with different backhaul topologies.

FS set for each FS can be determined from the underlying backhaul network topology, which is denoted as $\mathcal{L}_s$ for FS s. For convenience, we also define $\mathcal{L}_s^{\mathsf{O}}$ as the set of FSs which can offload their CLs to FS s.

A. Local-Computing and Offloading Model

Assume that MU k in $\mathcal{K}_s$ has to execute a computation task with $D_{k,s}$-bit input data within the maximum latency of $\bar{T}_{k,s}$ seconds. The partial offloading strategy is considered in this paper. In particular, $\ell_{k,s}$ bits of the incurred data are executed locally at the MU while the remaining $u_{k,s}$ bits will be offloaded to its FS for processing where $u_{k,s} = D_{k,s} - \ell_{k,s}$. Let $f_{k,s}$ be the CPU clock speed of MU k associated with BS s, which can be chosen in the range $(0, \bar{F}_{k,s}]$. The local execution energy per CPU cycle can be expressed as $\rho_{k,s} = \alpha_{k,s} f_{k,s}^2$ [12] where $\alpha_{k,s}$ represents the energy coefficient specified in the CPU model. Denote $c_{k,s}$ as the number of CPU cycles required for executing 1 data bit of MU k. Then, the energy and time consumed for local execution at MU k can be expressed, respectively as

$$E_{k,s}^{\mathsf{Lo}} = c_{k,s}\alpha_{k,s}\ell_{k,s}f_{k,s}^2 = c_{k,s}\alpha_{k,s}(D_{k,s} - u_{k,s})f_{k,s}^2, \quad (1)$$
$$t_{k,s}^{\mathsf{Lo}} = c_{k,s}\ell_{k,s}/f_{k,s} = c_{k,s}(D_{k,s} - u_{k,s})/f_{k,s}. \quad (2)$$

Once MU k decides to offload $u_{k,s}$ bits to FS s, it transmits these data bits to BS s over the wireless link. The TDMA transmission strategy is employed in this paper. In cell s (corresponding to FS s), an interval of T_s seconds is divided into K_s time slots each of which is then assigned to one corresponding MU in the set $\mathcal{K}_s$ to support the computation offloading. Let $h_{k,s}$ be the channel gain between MU k and BS s, which is assumed to be unchanged during the offloading duration; and $p_{k,s}$ denote the transmission power of MU k. To avoid the strong inter-cell interference, we assume that different frequency bands are allocated to different BSs [13]. Then, the data rate corresponding to the transmission from MU k to its BS can be expressed as

$$r_{k,s} = W \log_2\left(1 + p_{k,s}h_{k,s}/\sigma^2\right), \quad (3)$$

where W is the bandwidth and σ^2 is the noise power. Let $t_{k,s}$ denote the duration of the time slot assigned to MU k, which must satisfy $\sum_{k \in \mathcal{K}_s} t_{k,s} = T_s$ due to the employed TDMA strategy. Hence, the transmission power of MU k in BS s can

be set so that $r_{k,s} = u_{k,s}/t_{k,s}$. Thus, $p_{k,s}$ can be expressed as $p_{k,s} = \chi\left(u_{k,s}/t_{k,s}\right)$ where $\chi(x) = \tilde{h}_{k,s}^{-1}\left(2^{x/W} - 1\right)$, and $\tilde{h}_{k,s} = h_{k,s}/\sigma^2$ [6]. Then, the energy consumed by offloading MU k can be expressed as

$$E_{k,s}^{\mathsf{Off}} = t_{k,s}p_{k,s} = t_{k,s}\chi\left(u_{k,s}/t_{k,s}\right). \quad (4)$$

Combination of the results in (1) and (4) yields the total energy consumed for MU k at BS s as

$$E_{k,s} = E_{k,s}^{\mathsf{Lo}} + E_{k,s}^{\mathsf{Off}} = c_{k,s}\alpha_{k,s}\ell_{k,s}f_{k,s}^2 + t_{k,s}\chi\left(u_{k,s}/t_{k,s}\right). \quad (5)$$

B. Multi-Fog Cooperation Computing Model

The offloaded CL from MU k (corresponding to $u_{k,s}$ bits) can be executed at FS s or be further offloaded to FSs in the set $\mathcal{L}_s$. Let $u_{k,s,m}$ denote the number of bits, which is a part of $u_{k,s}$ bits, offloaded to FS m ($m \in \mathcal{L}_s$) for processing, we have $\sum_{m \in \mathcal{L}_s} u_{k,s,m} = u_{k,s}$. Assume that the allocated backhaul rate between FSs s and m is fixed at $d_{s,m}$, and the CPU clock speed that FS m allocates for executing $u_{k,s,m}$ is $f_{k,s,m}$. Then, the time consumed for further offloading and executing $u_{k,s,m}$ bits at BS m can be given, respectively as

$$t_{k,s,m}^{\mathsf{TF}} = u_{k,s,m}/d_{s,m} \text{ and } t_{k,s,m}^{\mathsf{Fog}} = u_{k,s,m}c_{k,s}/f_{k,s,m}. \quad (6)$$

Note that $u_{k,s,s}$ stands for the number of bits processed directly at BS s; hence, $t_{k,s,s}^{\mathsf{TF}}$ should be set to be zero. To yield this required outcome, we set $d_{s,s}$ as $+\infty$, $s \in \mathcal{S}$. For many practical applications, the amount of data describing the computation outcome to be sent back to every MU, is usually much smaller than that due to the task offloading. Therefore, we omit the downlink transmission in this paper. Hence, the total time required for task execution of MU k can be written as

$$T_{k,s} = \max\left(t_{k,s}^{\mathsf{Lo}}, T_s + \max_{m \in \mathcal{L}_s}\left(t_{k,s,m}^{\mathsf{TF}} + t_{k,s,m}^{\mathsf{Fog}}\right)\right). \quad (7)$$

C. Problem Formulation

In this paper, we focus on minimizing the weighted sum energy consumption of all MUs and our design can be formulated in the following problem:

$$(\mathcal{P}_1) \quad \min_{\Omega_1} \quad \sum_{\forall(k,s)} \beta_{k,s}E_{k,s} \quad (8a)$$

$$\text{s.t} \quad \ell_{k,s} + \sum_{m \in \mathcal{L}_s} u_{k,s,m} = D_{k,s}, \ \forall(k,s), \quad (8b)$$

$$\sum_{s \in \mathcal{L}_m^{\mathsf{O}}} \sum_{k \in \mathcal{K}_s} f_{k,s,m} \leq \bar{F}_m, \ \forall m, \quad (8c)$$

$$f_{k,s} \leq \bar{F}_{k,s}, \ \forall(k,s), \quad (8d)$$

$$c_{k,s}\ell_{k,s}/f_{k,s} \leq \bar{T}_{k,s}, \forall(k,s), \quad (8e)$$

$$t_{k,s,m}^{\mathsf{TF}} + t_{k,s,m}^{\mathsf{Fog}} + \sum_{k \in \mathcal{K}_s} t_{k,s} \leq \bar{T}_{k,s}, \forall(k,s,m), \quad (8f)$$

where $\beta_{k,s}$'s are the weighted coefficients, $\Omega_1 = \{T_s, \ell_{k,s}, u_{k,s,m}, t_{k,s}, f_{k,s}, f_{k,s,m}\}$'s denotes the set of all variables. Here, the constraints represent the backhaul capacity, computing resource, and different delay constraints where $\bar{F}_m$ is the maximum clock speed of FS m.

III. Optimal Joint Computation Offloading and Resource Allocation

A. Penalty based Convex Transformation

This section describes how to convexify problem $(\mathcal{P}_1)$ which is non-convex due to the non-linear term $\ell_{k,s}f_{k,s}^2$ in $E_{k,s}^{\mathsf{Lo}}$ and the non-convex constraint (8f). Toward this end, we first study the relationship between the optimal value of $f_{k,s}$ with other variables in the following proposition.

Proposition 1. *The optimal value of $f_{k,s}$ for any values of $\ell_{k,s}$ and $\bar{T}_{k,s}$ can be expressed as*

$$f_{k,s}^\star = (c_{k,s}\ell_{k,s})/\bar{T}_{k,s}. \tag{9}$$

Proof: As can be seen, constraint (8e) can provide the lower bound on $f_{k,s}$ as $\frac{c_{k,s}\ell_{k,s}}{T_{k,s}} \leq f_{k,s}$. Besides, it can be verified from the local computing energy consumption given in (1) that the objective function of problem $(\mathcal{P}_1)$ is a monotonic increasing function of $f_{k,s}$. Therefore, the optimal value of $f_{k,s}$ for any value of $\ell_{k,s}$ and $\bar{T}_{k,s}$ can be expressed in (9). ∎

The proof of Proposition 1 also shows that the constraint (8d) will be violated if $\frac{c_{k,s}\ell_{k,s}}{T_{k,s}} > \bar{F}_{k,s}$. Hence, one yields $\ell_{k,s} \leq \frac{\bar{T}_{k,s}\bar{F}_{k,s}}{c_{k,s}}$ which is equivalent to

$$D_{k,s} - \bar{T}_{k,s}\bar{F}_{k,s}/c_{k,s} \leq u_{k,s} \leq D_{k,s}. \tag{10}$$

By replacing $\ell_{k,s} = D_{k,s} - u_{k,s}$ and $f_{k,s} = (c_{k,s}\ell_{k,s})/\bar{T}_{k,s}$ using Proposition 1, the variable set can be reduced to $\Omega_2 = \Omega_1/\{f_{k,s}, \ell_{k,s}\}$'s and $E_{k,s}^{\mathsf{Lo}}$ can be rewritten as $\alpha_{k,s}c_{k,s}^3(D_{k,s}-u_{k,s})^3/\bar{T}_{k,s}^2$ which is a convex function with respect to $u_{k,s} > 0$. However, problem $(\mathcal{P}_1)$ is still non-convex due to non-convex term $\frac{u_{k,s,m}c_{k,s}}{f_{k,s,m}}$ in constraint (8f). To deal with this challenge, we introduce a new variable $a_{k,s,m}$ which satisfies $a_{k,s,m}^2 = u_{k,s,m} \ \forall(k,s,m)$. Using these new variables, problem $(\mathcal{P}_1)$ can be rewritten as

$$(\mathcal{P}_2) \min_{\Omega_3} \sum_{\forall(k,s)} \beta_{k,s}\left(\frac{\alpha_{k,s}c_{k,s}^3(D_{k,s}-u_{k,s})^3}{\bar{T}_{k,s}^2} + t_{k,s}\chi\left(\frac{u_{k,s}}{t_{k,s}}\right)\right) \tag{11a}$$

$$\text{s.t} \quad \text{constraints (8c), (10),}$$

$$\sum_{m\in\mathcal{L}_s} a_{k,s,m}^2 = u_{k,s}, \forall(k,s), \tag{11b}$$

$$\frac{a_{k,s,m}^2}{d_{s,m}} + \frac{c_{k,s}a_{k,s,m}^2}{f_{k,s,m}} + \sum_{k\in\mathcal{K}_s} t_{k,s} \leq \bar{T}_{k,s}, \forall(k,s,m), \tag{11c}$$

where $\Omega_3 = \Omega_2 \cup \{a_{k,s,m}\}$'s. A can be seen, the left hand side of constraint (11c) is a convex function of $(a_{k,s,m}, f_{k,s,m})$. Thus, problem $(\mathcal{P}_2)$ is convex which can be solved optimally if one omits constraint (11b). Therefore, we employ the Penalty-Function method to solve problem $(\mathcal{P}_2)$ by dealing with the following problem

$$(\mathcal{P}_3) \min_{\Omega_3} \sum_{\forall(k,s)} \left[\beta_{k,s}E_{k,s} + \rho\left(\sum_{m\in\mathcal{L}_s} a_{k,s,m}^2 - u_{k,s}\right)^2\right] \tag{12a}$$

$$\text{s.t} \quad \text{constraints (8c), (10), (11c).} \tag{12b}$$

In particular, the optimal solution of $(\mathcal{P}_2)$ can be obtained by repeatedly solving problem $(\mathcal{P}_3)$ and increasing ρ until

constraint (11b) holds. In the next section, we study how to tackle problem $(\mathcal{P}_3)$ and propose an optimal algorithm to solve problem $(\mathcal{P}_2)$. Problem $(\mathcal{P}_3)$ is convex thanks to its convex-form objective function and constraints. Therefore, it can be solved by employing the standard duality method as presented in the following section.

B. Duality-based Optimal Algorithm

The Lagrangian of problem $(\mathcal{P}_3)$ is expressed in (17) where $\boldsymbol{\mu} = \{\mu_m\}$'s, $\boldsymbol{\gamma} = \{\gamma_{k,s,m}\}$'s are the Lagrangian multipliers associated with the constraints of problem $(\mathcal{P}_3)$. Then, the dual function $g(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ can be defined as

$$g(\boldsymbol{\mu}, \boldsymbol{\gamma}) = \inf_{\Omega_3} \mathcal{L}(\Omega_3, \boldsymbol{\mu}, \boldsymbol{\gamma}) \text{ s.t. (10)}, \tag{14}$$

and the dual problem can be stated as

$$\max_{\boldsymbol{\mu}, \boldsymbol{\gamma}} g(\boldsymbol{\mu}, \boldsymbol{\gamma}) \text{ s.t. } \mu_m, \gamma_{k,s,m} \geq 0, \forall(k,s,m). \tag{15}$$

From the KKT conditions [14], the optimal solution $\Omega_3^\star = \{u_{k,s}^\star, t_{k,s}^\star, a_{k,s,m}^\star, f_{k,s,m}^\star\}$ of problem (14) for given $\boldsymbol{\mu}, \boldsymbol{\gamma}$ can be determined as stated in the following propositions.

Proposition 2. *Let $z_{k,s}^\star = \frac{u_{k,s}^\star}{Wt_{k,s}^\star}$. For given $\boldsymbol{\mu}, \boldsymbol{\gamma}$, the value of $z_{k,s}^\star$ can be expressed as*

$$z_{k,s}^\star = W_n\left(\left(\gamma_{k,s,m}\tilde{h}_{k,s}/\beta_{k,s} - 1\right)/e + 1\right)/ln2, \tag{16}$$

where $W_n(.)$ is the Lambert function [15].

Proof: Due to the space constraint, the proof is given briefly as follows. Applying the KKT conditions, we first take the derivative of $\mathcal{L}(\Omega_3, \boldsymbol{\mu}, \boldsymbol{\gamma})$ with respect to $t_{k,s}$, set it to zero, which results in an equation of $u_{k,s}/t_{k,s}$. Substituting $z_{k,s} = \frac{u_{k,s}}{Wt_{k,s}}$ into this equation yields a Lambert function of $z_{k,s}$. Solving this equation as in [15], we can obtain $z_{k,s}^\star$ as given in (16). ∎

Proposition 3. *Let $A(z_{k,s}) = \frac{\sigma^2}{h_{k,s}}\left(2^{z_{k,s}} - 1\right)$ where $z_{k,s} = \frac{u_{k,s}}{Wt_{k,s}}$. For given $\boldsymbol{\mu}, \boldsymbol{\gamma}$, the following value of $\Omega_3^\star$ satisfy the KKT conditions:*

$$a_{k,s,m}^\star = \sqrt{\frac{\mu_m}{\gamma_{k,s,m}c_{k,s}}}f_{k,s,m}^\star, \quad t_{k,s}^\star = u_{k,s}^\star/(Wz_{k,s}^\star), \tag{17}$$

$$f_{k,s,m}^\star = \left[\left(\frac{2\rho\Delta_{k,s}}{\gamma_{k,s,m}} - \frac{1}{d_{s,m}}\right)^{-1}c_{k,s}\right]^+ \tag{18}$$

$$u_{k,s}^\star = \min\left(\max\left(D_{k,s} - \bar{T}_{k,s}\bar{F}_{k,s}/c_{k,s}, \Gamma_{k,s}\right), D_{k,s}\right) \tag{19}$$

where $\Gamma_{k,s} = D_{k,s} - \sqrt{\frac{\bar{T}_{k,s}^2\left(A(z_{k,s}^\star) + 2\rho\Delta_{k,s}\right)}{3\alpha_{k,s}\beta_{k,s}c_{k,s}^3}}$, and $\Delta_{k,s}$ is determined by solving the following equation

$$D_{k,s} - \sqrt{\frac{\bar{T}_{k,s}^2\left(A(z_{k,s}) + 2\rho\Delta_{k,s}\right)}{3\alpha_{k,s}\beta_{k,s}c_{k,s}^3}} - \sum_{m\in\mathcal{L}_s}\frac{c_{k,s}\mu_m}{\gamma_{k,s,m}}\left(\frac{2\rho\Delta_{k,s}}{\gamma_{k,s,m}} - \frac{1}{d_{s,m}}\right)^{-2} = \Delta_{k,s}. \tag{20}$$

$$\mathcal{L}(\Omega_3,\boldsymbol{\mu},\boldsymbol{\gamma}) = \sum_{\forall(k,s)} \beta_{k,s}\left(\frac{\alpha_{k,s}c_{k,s}^3(D_{k,s}-u_{k,s})^3}{\bar{T}_{k,s}^2} + t_{k,s}\tilde{h}_{k,s}^{-1}\left(2^{\frac{u_{k,s}}{t_{k,s}}\cdot\frac{1}{W}}-1\right)\right) + \rho\sum_{\forall(k,s)}\left(\sum_{m\in\mathcal{L}_s}a_{k,s,m}^2-u_{k,s}\right)^2$$
$$+ \sum_{m\in S}\mu_m\left(\sum_{s\in\mathcal{L}_m^O}\sum_{k\in\mathcal{K}_s}f_{k,s,m}-\bar{F}_m\right) + \sum_{\forall(k,s)}\sum_{m\in\mathcal{L}_s}\gamma_{k,s,m}\left(\frac{a_{k,s,m}^2}{d_{s,m}}+\frac{c_{k,s}a_{k,s,m}^2}{f_{k,s,m}}+\sum_{r\in\mathcal{K}_s}t_{r,s}-\bar{T}_{k,s}\right) \quad (17)$$

Proof: Due to the space constraint, the proof is given briefly as follows. First, we set

$$\Delta_{k,s} = u_{k,s} - \sum_{m\in\mathcal{L}_s}a_{k,s,m}^2. \quad (21)$$

Applying the KKT conditions, we first take the derivative of $\mathcal{L}(\Omega_3,\boldsymbol{\mu},\boldsymbol{\gamma})$ with respect to all variables in Ω_3 and set the results equal to zeros to form a system of equations based on which the variables in Ω_3 can be stated as a function of $\Delta_{k,s}$ as in (17)-(19). Substituting (19) and (17) into (21) yields (20). Then, $\Omega_3^\star$ hence can be expressed as in (17)-(19) after obtaining $\Delta_{k,s}$ by solving (20). Since $\Omega_3^\star$ are calculated from the KKT conditions, $\Omega_3^\star$ must satisfy these conditions. ∎

Thanks to Proposition 2-3, $g(\boldsymbol{\mu},\boldsymbol{\gamma})$ can be determined for given $\boldsymbol{\mu},\boldsymbol{\gamma}$. The remaining work is to solve the duality problem (15) which is considered in the following proposition.

Proposition 4. *The dual function $g(\boldsymbol{\mu},\boldsymbol{\gamma})$ is a concave function and its sub-gradient at μ_m is $\sum_{s\in\mathcal{L}_m^O}\sum_{k\in\mathcal{K}_s}f_{k,s,m}-\bar{F}_m$, and at $\gamma_{k,s,m}$ is $\frac{a_{k,s,m}^2}{d_{s,m}}+\frac{c_{k,s}a_{k,s,m}^2}{f_{k,s,m}}+\sum_{k\in\mathcal{K}_s}t_{k,s}-\bar{T}_{k,s}$ where $\Omega_4^\star$ is obtained from the previous section.*

Proof: The dual function is a concave function by nature [14]. The choice of the sub-gradient for μ_m and $\gamma_{k,s,m}$ is justified by the fact that $\boldsymbol{\mu}$ and $\gamma_{k,s,m}$ are the Lagrangian multipliers associated with constraints (8c) and (11c), respectively. Details of this proof are omitted for brevity. ∎

Due to proposition 4, the dual variables can be solved using the following iterative updates:

$$\mu_m^{(n+1)} = \mu_m^{(n)} + \delta_m^{(n)}\left(\sum_{s\in\mathcal{L}_m^O}\sum_{k\in\mathcal{K}_s}f_{k,s,m}-\bar{F}_m\right), \quad (22)$$

$$\gamma_{k,s,m}^{(n+1)} = \gamma_{k,s,m}^{(n)} + \alpha_{k,s,m}^{(n)}\times$$
$$\left(\frac{a_{k,s,m}^2}{d_{s,m}}+\frac{c_{k,s}a_{k,s,m}^2}{f_{k,s,m}}+\sum_{k\in\mathcal{K}_s}t_{k,s}-\bar{T}_{k,s}\right), \quad (23)$$

where $\delta_m^{(n)}$ and $\alpha_{k,s,m}^{(n)}$ are suitable small step-sizes. If $\delta_m^{(n)},\alpha_{k,s,m}^{(n)}\xrightarrow{n\to\infty}0$, the above sub-gradient based updates are guaranteed to converge to the optimal solution of problem (15). By iteratively updating $\{\Omega_3^\star,\boldsymbol{\mu},\boldsymbol{\gamma}\}$, we can obtain the optimal solution. The proposed solution approach is summarized in the Algorithm 1 where $h(\Omega_3^{(n)}) = \sum_{\forall(k,s)}|\Delta_{k,s}^{(n)}|$.

IV. GREEDY ALGORITHM

For a comparison purpose, a greedy algorithm is described in Algorithm 2. This algorithm aims to perform load balancing

---

Algorithm 1 DUALITY-BASED OPTIMAL ALGORITHM

1: Initialization: Set $\boldsymbol{\mu}=\mathbf{0}$ for all (k,s,m), choosing $\rho>0$, a tolerance ε, and set $n=0$.
2: **repeat**
3:    **repeat**
4:       Update $\Omega_3^{(n)}$ by following all steps in Propositions 2-3.
5:       Update $\boldsymbol{\mu}^{(n+1)}$ and $\boldsymbol{\gamma}^{(n+1)}$ as in (22)-(23).
6:       Update $n=n+1$.
7:    **until** Convergence.
8:    Increasing ρ by setting $\rho:=M\rho$ $(M>1)$.
9: **until** $h(\Omega_3^{(n)})\le\varepsilon$
10: Output: Obtain $\Omega_3^{opt}=\{f_{k,s,m}^{opt},u_{k,s}^{opt},a_{k,s,m}^{opt}\,t_{k,s}^{opt}\}$ and compute $E_{k,s}$ for each MU k of BS s

---

among FSs. Specifically, we attempt to determine if one FS is overload and there exists a *Ready-To-Help (RTH)* FS with available computing resource (CR) as follows. First, we optimize the CO and RA at each BS/FS separately, from which the available CR of each FS is calculated as $\Delta f_s = \bar{F}_s - \sum_{\forall(k,m)}f_{k,m,s}, \forall s$. Then, the FS having the largest available CR is selected as *RTH* FS $\hat{m}$. This *RTH* FS will utilize all available CR to help the FS that has used all CR and its MUs spend the highest energy consumption, named as *Asking-For-Help (AFH)* FS $s^\star$. Specifically, FS $s^\star$ is allowed to offload its CL to FS $\hat{m}$. The CPU clock speed that FS $\hat{m}$ assigns MUs in cell $s^\star$ can be determined as in (26). Then, the offloaded data size $u_{k,s^\star,\hat{m}}$ can be determined to achieve time balancing as follows:

$$\frac{c_{k,s^\star}(u_{k,s^\star,s^\star}-u_{k,s^\star,\hat{m}})}{f_{k,s^\star,s^\star}} = \frac{u_{k,s^\star,\hat{m}}}{d_{s^\star,\hat{m}}}+\frac{c_{k,s^\star}u_{k,s^\star,\hat{m}}}{f_{k,s^\star,\hat{m}}}, \quad (24)$$

which yields (27). Thanks to the load balancing between FSs $\hat{m}$ and $s^\star$ as in **Step 11** of Algorithm 2, the processing time at the FS for MUs in cell $s^\star$ is reduced by $\Delta T_{s^\star} = \min_{k\in\mathcal{K}_{s^\star}}(\frac{c_{k,s^\star}u_{k,s^\star,\hat{m}}}{f_{k,s^\star,s^\star}})$. This time reduction can be exploited for increasing the offloading time of all MUs in cell $s^\star$. Hence, the new offloading duration can be defined as $T'_{s^\star} = T_{s^\star}+\Delta T_{s^\star}$. Then, the join CO and RA design for cell $s^\star$ can be re-optimized by solving the following problem:

$$(\mathcal{P}_{s^\star})\ \min_{\Omega_s}\ \sum_{k\in\mathcal{K}_{s^\star}}\beta_{k,s^\star}\left[\frac{\alpha_{k,s^\star}c_{k,s^\star}^3(D_{k,s^\star}-u_{k,s,s}-u_{k,s^\star,\hat{m}})^3}{\bar{T}_{k,s^\star}^2}\right.$$
$$\left.+t_{k,s}\frac{\sigma^2}{h_{k,s^\star}}\left(2^{\frac{u_{k,s,s}+u_{k,s^\star,\hat{m}}}{Wt_{k,s}}}-1\right)\right]$$

s.t (8c), (10), (8f) for cell $s^\star$ and $\sum_{k\in\mathcal{K}_{s^\star}}t_{k,s}\le T'_{s^\star}$,

Algorithm 2 GREEDY ALGORITHM

1: Initialize $u_{k,s,m} = 0 \; \forall(k,s,m), \; m \neq s$ and $T'_s = \max_{k \in \mathcal{K}_s} \bar{T}_{k,s} \; \forall s$, $\mathcal{S}^\star = \mathcal{S}$.
2: Solve $(\mathcal{P}_s)$ for all cells to obtain optimum $\Omega_3^\star$.
3: **repeat**
4:     Calculate $\Delta f_s = \bar{F}_s - \sum\limits_{\forall(k,m)} f_{k,m,s}$.
5:     **if** $\Delta f_s > 0$ for all $s \in \mathcal{S}^\star$ **then**
6:         **Stop** since there no bottleneck at FSs.
7:     **else**
8:         Retrieve $\hat{m} = \arg\max_{m \in \mathcal{S}} \Delta f_m$ as *RTH* FS.
9:         Update $\mathcal{S}^\star = \mathcal{S}/\{\hat{m}\}$.
10:         Retrieve $s^\star = \arg\max_{s \in \mathcal{S}^\star} \sum_{k \in \mathcal{K}_s} \beta_{k,s} E_{k,s}$ s. t. $\Delta f_s = 0$.
11:         Determine $f_{k,s^\star,\hat{m}}$ and $u_{k,s^\star,\hat{m}}$ as

$$f_{k,s^\star,\hat{m}} = \Delta f_{\hat{m}} \frac{f_{k,s^\star,s^\star}}{\sum_{k \in \mathcal{K}_{s^\star}} f_{k,s^\star,s^\star}}, \qquad (26)$$

$$u_{k,s^\star,\hat{m}} = u_{k,s^\star,s^\star} \left(\frac{f_{k,s^\star,s^\star}}{c_{k,s^\star} d_{s^\star,\hat{m}}} + \frac{f_{k,s^\star,s^\star}}{f_{k,s^\star,\hat{m}}} + 1 \right)^{-1}. \; (27)$$

12:         Determine $T'_{s^\star} = \sum\limits_{k \in \mathcal{K}_{s^\star}} t_{k,s^\star} + \min_{k \in \mathcal{K}_{s^\star}} (\frac{c_{k,s^\star} u_{k,s^\star,\hat{m}}}{f_{k,s^\star,s^\star}})$.
13:         Solve $(\mathcal{P}_{s^\star})$ to update $\Omega_{s^\star}$.
14:     **end if**
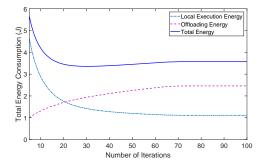15: **until** $\Delta f_s > 0$ for all $s \in \mathcal{S}^\star$.

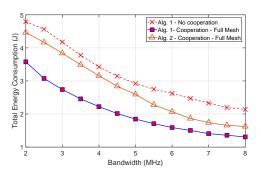
Fig. 2. Variations of energy consumption over iterations.

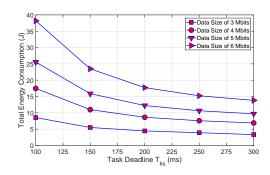Fig. 3. Total energy consumption versus the frequency bandwidth W.

Fig. 4. Total energy consumption versus allowable task latency T.

where $\Omega_s = \{u_{k,s,s}, f_{k,s,s}, t_{k,s}\}_{k \in \mathcal{K}_{s^\star}}$. Problem $(\mathcal{P}_{s^\star})$ is applied for one FS which can be solved by employing the algorithms in [6] or our proposed algorithm with low complexity. Since the previous values of $\{u_{k,s^\star,s^\star}, f_{k,s^\star,s^\star}, t_{k,s^\star}\}_{k \in \mathcal{K}_{s^\star}}$ is a feasible solution of $(\mathcal{P}_{s^\star})$, solving $(\mathcal{P}_{s^\star})$ results in a new solution with smaller weighted sum of power consumption at cell $s^\star$. **Step 4**-**Step 13** of Algorithm 2 are repeated until there is no *RTH* FS or *AFH* FS.

V. NUMERICAL RESULTS

The performance of the proposed algorithms for backhaul limited cooperative MEC system is investigated via numerical studies. We consider a simple 4-cell network where the distance between two nearest BSs is $400\,m$ as illustrated in Fig. 1. In each cell, we randomly place 7 MUs (except for Fig. 5) so that the distance from the cell center to every MU is in the range $[50m, 200m]$. The channel gains are generated by considering both Rayleigh fading and path loss which is modelled as $L_{k,s}^k = 36.8\log_{10}(y_{k,s}^k) + 43.8 + 20\log_{10}(\frac{f_{\text{freq}}}{5})$ where $y_{k,s}^k$ is the distance from MU k to FS s and $f_{\text{freq}} = 2.5\,GHz$. The noise power is set equal to $\sigma^2 = 10^{-13}\,W$. In the simulation, we choose $\beta_{k,s}$ equal to 1 for all MUs, $W_s = 4$ MHz (except for Fig. 3), and $d_{s,m} = 2$ Mbits (except for Fig. 6). Each MU needs to execute a task with data size of 20 Mbits ($D_{k,s} = 20Mbit$) within a duration of 100 ms ($\bar{T}_{k,s} = 100ms, \forall(k,s)$) (except Fig. 4). The CPU computation capacity for each MU ($\bar{F}_{k,s}$) is randomly selected from the set $\{0.3, 0.4, ..., 0.7\}$ GHz and the local computing energy per cycle is $\alpha_{k,s} = 10^{-26}$ J/cycle. The number of CPU cycles per data bit is set $c_{k,s} \in [500, 1500]$ cycles/bit. The computation capacities of four FSs are chosen as $\{1.7, 3.6, 3.8, 4.5\}$ GHz.

First, we examine the convergence of our proposed optimal algorithm by showing the variations of total energy consumption (TEC), the energy of local execution (ELE), and offloading energy (OE) of all MUs over iterations by using Algorithm 1 with $\rho = 100$ in Fig. 2. In this simulation, we consider the cooperation among the FSs under the full mesh backhaul topology as shown in Fig. 1-(a). It can be observed that the TEC converges after $70-80$ iterations which confirms the convergence of Algorithm 1. In addition, the ELE decreases while the TE increases over iterations before becoming saturated.

In Fig. 3, we show the benefit of cooperation among the FSs where the TEC versus the frequency bandwidth (W) is plotted for three schemes, Algorithm 1 with full cooperation (Full Mesh) and no-cooperation, and Algorithm 2 with full cooperation. As can be observed, the TEC achieved by our proposed optimal algorithm with full cooperation is much lower than those due to the schemes without FSs' cooperation and the greedy algorithm. In addition, the TEC achieved by all algorithms decrease as W increases, which demonstrates
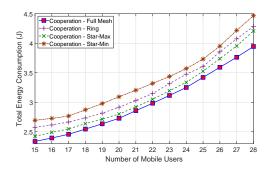
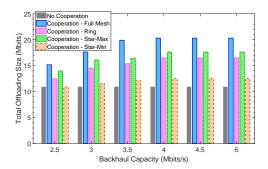Fig. 5. Total energy consumption versus the number of MUs.

Fig. 6. Size of offloaded data versus the capacity of backhaul links.

the impact of wireless links capacity on the CO design.

Then, we illustrate the variations of TEC achieved by Algorithm 1 with different values of allowable task latency ($\bar{T}_{k,s} = T, \forall (k,s)$) and application data size ($\bar{D}_{k,s} = D, \forall (k,s)$) in Fig. 4. As can be seen from this figure, the TEC decreases when $T_{k,s}$ increases while the TEC increases when $\bar{D}_{k,s}$ becomes larger. This is because the larger value of $T_{k,s}$ gives more freedom for the computation offloading and the larger data size requires higher energy for data transmission and task execution.

The performance of our proposed algorithm under different backhaul topologies is examined in Fig. 5 where the TEC achieved by employing Algorithm 1 versus the number of MUs is plotted in four backhaul network topologies: Full Mesh, Ring, Star-Max (the most powerful FS is the center), and Star-Min (the weakest FS is the center). This figure confirms that the Full-Mesh backhaul network results in the lowest TEC while Star-Max scenario achieves better performance than the Ring scenario, and the Star-Max is the worst. The figure also shows that the TEC achieved by all schemes increases with the number of MUs as expected.

Finally, we demonstrate the impact of backhaul capacity on the number of bits offloaded to FSs with and without cooperation among the FSs in Fig. 6. In this simulation, we set $d_{s,m} = d, \forall (s,m)$. As can be seen, the cooperation topologies encourage MUs to offload more data to FSs than the no-cooperation one does. In particular, the number of offloaded bits in Full-Mesh scheme is the highest, that number in Ring scheme is slightly smaller than that due to Star-Max scheme while the Star-Min scheme gains the lowest number of offloaded bits among all cooperation schemes. Considering the results in both Figs. 5 and 6, one interestingly confirms that the schemes having the higher number of offloaded bits can save more TEC. In addition, the amount of offloaded data in each scheme tends to increase as the backhaul capacity increases but it becomes saturated when d is sufficiently large.

VI. CONCLUSION

In this paper, we have considered the joint CO and RA design for backhaul limited cooperative MEC systems. This design aims to minimize the weighted sum of energy consumption of all MUs. We have developed the optimal algorithm using the convexification and duality methods and have described the greedy algorithm. Numerical results have confirmed the great performance gains of the proposed design compared to offloading design without cooperation among FSs and the greedy design.

REFERENCES

[1] E. Portal. (2014) Mobile-edge computing introductory technical white paper. [Online]. Available: https://portal.etsi.org
[2] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, "Selective offloading in mobile edge computing for the green internet of things," *IEEE Network*, vol. 32, no. 1, pp. 54–60, Jan 2018.
[3] T. T. Nguyen, L. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. Services Comput.*, vol. PP, no. 99, pp. 1–1, 2019.
[4] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug 2017.
[5] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct 2016.
[6] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, March 2017.
[7] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 14–27, Jan 2015.
[8] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.
[9] X. He, Y. Chen, and K. K. Chai, "Delay-aware energy efficient computation offloading for energy harvesting enabled fog radio access networks," in *Proc. IEEE Veh. Technol. Conf.*, June 2018, pp. 1–6.
[10] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan 2019.
[11] J. Oueis, E. Calvanese-Strinati, A. De Domenico, and S. Barbarossa, "On the impact of backhaul network on distributed cloud computing," in *Proc. IEEE Wireless Commun. and Networking Conf.* IEEE, 2014, pp. 12–17.
[12] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
[13] C. Kosta, B. Hunt, A. U. Quddus, and R. Tafazolli, "On interference avoidance through inter-cell interference coordination (icic) based on ofdma mobile systems," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 973–995, Third 2013.
[14] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.
[15] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, Dec 1996. [Online]. Available: https://doi.org/10.1007/BF02124750