Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing

Chenmeng Wang, Chengchao Liang, F. Richard Yu, *Senior Member, IEEE*, Qianbin Chen, *Senior Member, IEEE*, and Lun Tang

***Abstract*— Mobile edge computing has risen as a promising technology for augmenting the computational capabilities of mobile devices. Meanwhile, in-network caching has become a natural trend of the solution of handling exponentially increasing Internet traffic. The important issues in these two networking paradigms are computation offloading and content caching strategies, respectively. In order to jointly tackle these issues in wireless cellular networks with mobile edge computing, we formulate the computation offloading decision, resource allocation and content caching strategy as an optimization problem, considering the total revenue of the network. Furthermore, we transform the original problem into a convex problem and then decompose it in order to solve it in a distributed and efficient way. Finally, with recent advances in distributed convex optimization, we develop an alternating direction method of multipliers-based algorithm to solve the optimization problem. The effectiveness of the proposed scheme is demonstrated by simulation results with different system parameters.**

***Index Terms*— Mobile edge computing, small cell networks, computation offloading, resource allocation, in-network caching.**

I. Introduction

With the radically increasing popularity of smart phones, new mobile applications such as face recognition, natural language processing, augmented reality, etc. are emerging constantly. However, traditional wireless cellular networks are becoming incapable to meet the exponentially growing demand not only in *high data rate* but also in *high computational capability* [1].

In order to address the data rate issue, the heterogeneous network structure was recently proposed, in which multiple low-power, local coverage enhancing small cells are deployed in one macro cell [2]. Since the same radio resource could be shared among small cells and with the macro cell, small cell networks have been considered as a promising solution to improving spectrum efficiency and energy efficiency, therefore consisting one of the key components of next generation wireless cellular networks [3]. Nevertheless, severe inter-cell interference may be incurred due to spectrum reuse, which will significantly deteriorate network performance. Without effective *spectrum resource allocation mechanism*, the overall spectrum efficiency and energy efficiency of the network might become even worse than that of a network without small cells [4]. To address the spectrum allocation issue, the work in [5] proposes a graph colouring method to assign physical resource blocks (PRBs) to user's equipment (UEs). The study of [4] presents a spectrum allocation algorithm based on game theory, in which the PRB allocation can reach a Nash equilibrium of the game.

On the other hand, to address the computational capability issue, mobile cloud computing (MCC) systems have been proposed to enable mobile devices to utilize the powerful computing capability in the cloud [6], [7]. In order to further reduce the latency and make the solution more economical, the *fog computing* has been proposed to deploy computing resources closer to end devices [8]–[10]. A similar technique, called *mobile edge computing* (MEC), has attracted great interest in wireless cellular networks recently [11]–[13]. MEC enables the mobile to UEs to perform *computation offloading* to send their computation tasks to the MEC server via wireless cellular networks. Then each UE is associated with a clone in MEC server, which executes the computation tasks on behalf of that UE. A number of previous works have discussed the computation offloading problem [14]–[18], from latency reduction and energy saving, or QoS (Quality of Service) promoting perspectives.

In addition, the server in MEC system can realize an in-network caching function [11], similar to the function provided by information-centric networking (ICN) [19]–[22], which is able to reduce replicate information transmissions. According to the study of [23], in-network caching has the capability of significantly improving the quality of Internet content transmissions (e.g., reducing latency and increasing throughput) by moving the content closer to users. A number of research efforts have been dedicated to content caching strategies. The caching strategies proposed in [24] and [25] are based on where routers are located in the topology, while [26] designs the strategy according to the content popularity.

Manuscript received November 3, 2016; revised March 3, 2017; accepted April 24, 2017. Date of publication May 16, 2017; date of current version August 10, 2017. This work was supported in part by Graduates Research Innovation Program of Chongqing under Grant CYB15106, in part by the National Natural Science Foundation of China under Grant 61571073, in part by the National High Technology Research and Development Program of China under Grant 2014AA01A701, and in part by the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was J. Tang. *(Corresponding author: Chenmeng Wang.)*

C. Wang, Q. Chen, and L. Tang are with the Chongqing Key Lab of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: wangchenmeng09@gmail.com; chenqb@cqupt.edu.cn; tangl@cqupt.edu.cn).

C. Liang and F. R. Yu are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada (e-mail: chengchaoliang@sce.carleton.ca; richard.yu@carleton.ca).

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

Digital Object Identifier 10.1109/TWC.2017.2703901

Although some outstanding works have been dedicated to studying computation offloading, resource allocation and content caching, these important aspects were generally considered separately in the existing works. However, as shown in the following, it is necessary to jointly address these issues together to improve the performance of next generation wireless networks. Therefore, in this paper, we propose to jointly consider computation offloading, spectrum and computation resource allocation and content caching in order to improve the performance of wireless cellular networks with mobile edge computing. The motivations behind our work are based on the following observations.

- Computation offloading, resource allocation and content caching are all parts of the entire system, and they all contribute to the end-to-end user experience, which can be hardly guaranteed by the optimization of one single segment of the whole system [27].
- If multiple UEs choose to offload their computation tasks to the MEC server via small cell networks simultaneously, severe interference can be generated, which will decrease the data rate. Moreover, the MEC server could be overloaded. In this case, it is not beneficial for all the UEs to offload their tasks to the MEC server. Instead, some UEs should be selected to offload their computations, while others should execute their computations locally.
- Different amounts of spectrum and computation resources should be allocated to different UEs to fulfill different user demands.
- Due to limited caching space of the MEC server, different caching strategies should be applied upon different contents, in order to maximize the caching revenue.

Therefore, an integrated framework for computation offloading, resource allocation and content caching has the potential to significantly improve the performance of wireless cellular networks with mobile edge computing.

To the best of our knowledge, the joint design of computation offloading, resource allocation and content caching has not been addressed in previous works. The distinct features of this paper are as follows.

- We formulate the computation offloading decision, resource allocation, and content caching in wireless cellular networks with mobile edge computing as an optimization problem.
- We transform the original non-convex problem into a convex problem and provide the proof of the convexity of the transformed problem.
- We decompose the problem and apply alternating direction method of multipliers (ADMM) to solve the problem in an efficient and practical way.
- Simulation results are presented to show the effectiveness of the proposed scheme with different system parameters.

The rest of this paper is organized as follows. The system model under consideration is described in Section II. The original optimization problem is formulated and is transformed into a convex problem in Section III. Furthermore, it is decomposed in order to employ a distributed problem solving method. Section IV presents the progress of problem solving by ADMM. Simulation results are discussed in Section V. Finally, we conclude this study in Section VI.

II. System Model

In this section, the system model adopted in this work is described. We first describe the network model, then we present the communication model, computation model and caching model in details. Finally, the utility function of the optimization problem is proposed.

A. Network Model

An environment of one macrocell and N small cells in the terminology of LTE standards is considered here. The macro cell is connected to the Internet through the core network of cellular communication system. An MEC server is placed in the macro eNodeB (MeNB), and all the N small cell eNodeBs (SeNBs) are connected to the MeNB as well as the MEC server. In this paper, it is assumed that the SeNBs are connected to the MeNB in wired manner [28]. The set of small cells is denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$, and we use n to refer to the nth small cell (SeNB). It is assumed that SeNB n is associated with K_n mobile UEs. We let $\mathcal{K}_n = \{1, 2, \ldots, K_n\}$ denote the set of UEs associating with SeNB n, and k_n refers to the kth UE which associates with the nth SeNB. In this paper, we consider single-antenna UEs and SeNBs. The network model is illustrated in Fig. 1.

We assume that each UE has a computationally intensive and delay sensitive task to be completed. Each UE can offload the computation to the MEC server through the SeNB with which it is associated, or execute the computation task locally. UEs can request content from the Internet, then the Internet content will be transmitted through macro base station (MeNB) to UEs. Upon the first transmission of any particular Internet content, the MEC server can choose whether to store the content or not. If the content were stored, it can be used by other UEs without another transmission from Internet in the future. In this paper, we consider two logical roles in the network: *mobile network operator* (MNO) and *MEC system operator* (MSO). The mobile network operators possess and operate the radio resources and physical infrastructures of the wireless networks, including spectrum, backhaul, radio access networks, transmission networks, core networks, etc., while the MEC system operators own the MEC servers, lease physical resources (e.g., spectrum and backhaul) from MNO and provide mobile edge computing services to UEs. The MSO will charge the UEs for receiving mobile edge computing services.

Similar to many previous works in mobile cloud computing [29] and mobile networking [30]–[34], to enable tractable analysis and get useful insights, we employ a quasi-static scenario where the set of mobile device users $\mathcal{K}_n, \forall n$ remains unchanged during a computation offloading period (e.g., within several seconds), while it may change across different periods. Since both the communication and computation aspects play a key role in mobile edge computing, next the communication and computation models are introduced in detail.

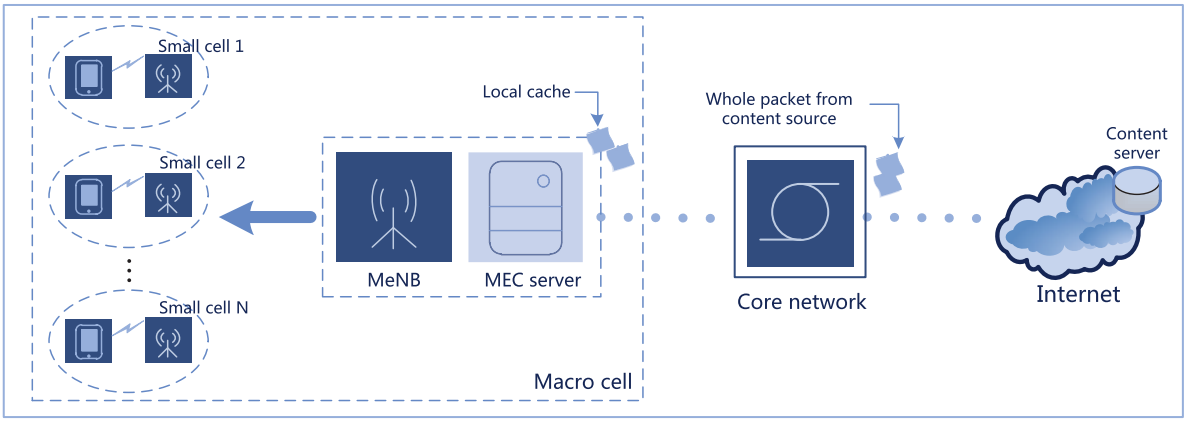


Fig. 1. Network model.

TABLE I
NOTATION

Notation	Definition	Notation	Definition
$\mathcal{N}$	The set of small cells.	$\boldsymbol{s}$	Radio spectrum allocation vector.
$\mathcal{K}_n$	The set of UEs associate with SeNB n.	$\boldsymbol{c}$	Computation resource allocation profile.
k_n	kth UE associating with the nth SeNB.	$\boldsymbol{h}$	Internet content caching decision vector.
p_{k_n}	The transmission power of UE k_n.	δ_n	Unit price for leasing spectrum from small cell n.
N	Total number of small cells.	η_n	Unit price for leasing backhaul from small cell n.
B	Total system bandwidth.	θ_n	Unit price for transmitting computation input data.
F	Total computational capability of the MEC server.	λ	Unit price for leasing computation resource.
L	Backhaul capacity between MeNB and MEC server.	ζ	Unit price for leasing backhaul connecting Internet.
L_n	Backhaul capacity of SeNB n.	ϖ	The cost in the memory for caching one content.
Y	Total storage capability of the MEC server.	ι_{k_n}	Revenue of assigning radio resource to UE k_n.
$G_{k_n,n}$	The channel gain between UE k_n and SeNB n.	Ω_{k_n}	Revenue of allocating comput. resource to UE k_n.
W_{k_n}	Computation task of UE k_n.	Λ_{k_n}	Revenue of caching the content of UE k_n.
$f_{k_n}^{(l)}$	Computation capability of UE k_n.	$f_{k_n}^{(e)}$	Computation resource assigned to UE k_n.
$\boldsymbol{a}$	Computation offloading decision vector.		

The notations that will be used in the rest of this paper are summarized in Table I.

B. Communication Model

Every SeNB in the network is linked to the MEC server, so each UE could offload its computation task to the MEC server via the SeNB to which it is connected. We denote $a_{k_n} \in \{0,1\}, \forall n,k$ as the computation offloading decision of UE k_n. Specifically, we have $a_{k_n}=0$ if UE k_n was determined to compute its task locally on the mobile device. We have $a_{k_n}=1$ if UE k_n was chosen to offload the computation to the MEC server via wireless access. So we have $\boldsymbol{a}=\{a_{k_n}\}_{k_n\in\mathcal{K}_n,n\in\mathcal{N}}$ as the offloading decision profile.

In this paper, we consider the case where spectrum used by small cells is overlaid, which means there exists interference between small cells. However, spectrum within one small cell is orthogonally assigned to every UE, so there will be no interference within one small cell. Only uplink direction transmissions are considered, which means transmission is from a UE to the SeNB to which it is associated, and interference is from a UE to a neighboring SeNB. In this paper, we assume that the interference only occurs when UEs served by various SeNBs are occupying the same frequency simultaneously. The whole available spectrum bandwidth is B Hz. The backhaul capacity between MeNB and MEC server is L bps, and the backhaul capacity of SeNB n is L_n bps. According to Shannon bound, the spectrum efficiency of UE k_n is given by,

$$e_{k_n}=\log_2\left(1+\frac{p_{k_n}G_{k_n,n}}{\sigma+\sum\limits_{m=1,m\neq n}^{N}\sum\limits_{i=1}^{K_m}p_{i_m}G_{i_m,n}}\right),\quad \forall n,k, \tag{1}$$

where p_{k_n} is the transmission power density of UE k_n, and $G_{k_n,n}$, $G_{i_m,n}$ stand for the channel gain between UE k_n and SeNB n, the channel gain between UE i_m and SeNB n, respectively. σ denotes the power spectrum density of additive white Gaussian noise.

We denote $s_{k_n}\in[0,1],\forall n,k$ as the percentage of radio spectrum allocated to UE k_n by small cell n, thus $\sum_{k_n\in\mathcal{K}_n}s_{k_n}\leq 1,\forall n$. We have $\boldsymbol{s}=\{s_{k_n}\}_{k_n\in\mathcal{K}_n,n\in\mathcal{N}}$ as the radio spectrum allocation profile. Then the expected instantaneous data rate of UE k_n, R_{k_n} is calculated as

$$R_{k_n}(\boldsymbol{a},\boldsymbol{s})=a_{k_n}s_{k_n}Be_{k_n},\quad \forall n,k. \tag{2}$$

The data rate cannot exceed the backhaul capacity of SeNB n, thus $\sum_{k_n\in\mathcal{K}_n}R_{k_n}\leq L_n,\forall n$ must hold. The total data rate of

all the UEs cannot exceed the backhaul capacity of MeNB, thus $\sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} R_{k_n} \le L$ must hold.

C. Computation Model

For the computation model, we consider that each UE k_n has a computation task $W_{k_n} \triangleq (Z_{k_n}, D_{k_n})$, which can be computed either locally on the mobile device or remotely on the MEC server via computation offloading, as in [16]. Here Z_{k_n} stands for the size of input data, including program codes and input parameters, and D_{k_n} denotes the total number of CPU cycles required to accomplish the computation task W_{k_n}. A UE k_n can use the method in [29] and [35] to obtain the information of Z_{k_n} and D_{k_n}. We next discuss the computation overhead in terms of processing time for both local and MEC computing approaches.

1) Local Computing: For the local computing approach, the computation task W_{k_n} is executed locally on each mobile device. We denote $f_{k_n}^{(l)}$ as the computational capability (i.e., CPU cycles per second) of UE k_n. It is allowed that different UEs may have different computational capabilities. The computation execution time $T_{k_n}^{(l)}$ of task W_{k_n} executed locally by UE k_n is expressed as

$$T_{k_n}^{(l)} = \frac{D_{k_n}}{f_{k_n}^{(l)}}. \tag{3}$$

2) MEC Server Computing: For the MEC server computing approach, a UE k_n will offload its computation task W_{k_n} through wireless access to SeNB n, then through the connection from SeNB n to the MEC server. Then the MEC server will execute the computation task instead of UE k_n. For offloading the computation task, a UE k_n will incur the consumption on time when transmitting computation input data to the MEC server. According to the communication model presented in Subsection II-B, the time costs for transmitting the computation input data of size Z_{k_n} are calculated as

$$T_{k_n,off}^{(e)}(\boldsymbol{a},\boldsymbol{s}) = \frac{Z_{k_n}}{R_{k_n}(\boldsymbol{a},\boldsymbol{s})}. \tag{4}$$

The MEC server will execute the computation task after offloading. Let $f_{k_n}^{(e)}$ denote the computational capability (i.e., CPU cycles per second) of the MEC server assigned to UE k_n. Then the execution time of the MEC server on task W_{k_n} is given as

$$T_{k_n,exe}^{(e)} = \frac{D_{k_n}}{f_{k_n}^{(e)}}. \tag{5}$$

Then the total execution time of the task of UE k_n is given by

$$T_{k_n}^{(e)}(\boldsymbol{a},\boldsymbol{s}) = T_{k_n,off}^{(e)}(\boldsymbol{a},\boldsymbol{s}) + T_{k_n,exe}^{(e)}. \tag{6}$$

In Section V, we will use this expression to assess the average UE time consumption for executing computation tasks in simulations.

Similar to the study in [16], the time consumption of computation outcome transmission from the MEC server to UE k_n is neglected in this work, due to the fact that the size of computation outcome data in general is much smaller than that of the computation input data including the mobile system settings, program codes and input parameters.

D. Caching Model

We denote $h_{k_n} \in \{0,1\}, \forall n,k$ as the caching strategy for UE k_n. Specifically, we have $h_{k_n} = 1$ if the MEC server decides to cache the content requested by UE k_n and $h_{k_n} = 0$ otherwise. So we have $\boldsymbol{h} = \{h_{k_n}\}_{k_n\in\mathcal{K}_n, n\in\mathcal{N}}$ as the caching decision profile.

According to [36] and [37], the reward of caching in wireless networks can be the reduction of backhaul delay or the alleviation of backhaul bandwidth. In this paper the alleviated backhaul bandwidth between macro cell and the Internet is adopted as the caching reward. Thus, the reward (alleviated backhaul bandwidth) of caching the content requested by UE k_n can be given as

$$Caching\ reward = q_{k_n}\bar{R}h_{k_n}, \tag{7}$$

where $\bar{R}$ is the average single UE data rate in the system, and q_{k_n} is the request rate (by other UEs) of the content first requested by UE k_n. According to the statistics of [38], if the requested content has a constant size, the request rate follows Zipf popularity distribution, therefore can be calculated as $q(i) = 1/i^{\beta}$, where i stands for the i-th most popular content, and β is a constant whose typical value is 0.56 [39]. Therefore, if the size of the content first requested by UE k_n is known, the request rate of other UEs upon the same content could be derived from the equation given above. In fact, the modeling of request rates of the caching content is still under research by many scholars. Since we adopt constant request rates in this paper, the modeling of request rates is above the scope of this paper.

It is worth noting that the storage capability of the MEC server is not unlimited, thus the sum size of all the cached content cannot exceed the total storage capability of the MEC server. In other words, $\sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} h_{k_n}o_{k_n} \le Y$ must hold, where Y is the total storage capability of the MEC server, and o_{k_n} is the size of the content first requested by UE k_n. In this paper, it is assumed that $o_{k_n}\forall k,n$ is constant and we adopt $o_{k_n} = 1$.

E. Utility Function

In this paper, we set the maximization of the revenue of MSO as our goal. MSO rents spectrum and backhaul from MNO, and the unit price for leasing spectrum from small cell n is defined as δ_n per Hz, while the unit price of backhaul between small cell n and macro cell is defined as η_n per bps. The MSO will charge UEs for transmitting computation input data to MEC server, and the unit price being charged is defined as θ_n per bps. So the net revenue of MSO for assigning radio resources to UE k_n is calculated as $\iota_{k_n} = s_{k_n}\Psi_{k_n} = s_{k_n}(\theta_n Be_{k_n} - \delta_n B - \eta_n Be_{k_n})$.

We next calculate the revenue of MSO for allocating computation resource to UEs. First, we define $c_{k_n} \in [0,1], \forall n,k$ as the percentage of MEC server computation resource allocated to UE k_n, thus $\sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} c_{k_n} \le 1$. We have $\boldsymbol{c} = \{c_{k_n}\}_{k_n\in\mathcal{K}_n, n\in\mathcal{N}}$ as the computation resource allocation profile.

Without losing generality, the different sizes of computation tasks and different local computation capability of different UEs should be taken into account. So we define that the MSO will charge UE k_n only for the difference between MEC computation resource allocated to every unit computation task and the local computation resource assigned to every unit computation task, and the unit price is λ_n for small cell n. Then the net revenue of allocating computation resource to UE k_n is given as $\Omega_{k_n} = \lambda_n(c_{k_n}F/D_{k_n} - f_{k_n}^{(l)}/D_{k_n})$, where F stands for the total computation resource of MEC server. Note that the reciprocal of $c_{k_n}F/D_{k_n}$ is the time consumption for MEC server executing computation task D_{k_n}, and the reciprocal of $f_{k_n}^{(l)}/D_{k_n}$ is the time consumption for UE k_n locally executing task D_{k_n}. This implies that the amount of computation resource assigned to every unit computation task can reflect the time consumption of executing this task.

We next discuss the revenue of MSO for caching Internet content requested by UEs. We define the unit price of leasing the backhaul between the macro cell and Internet is ζ per bps, and the cost in the memory for caching one content is ϖ. If the content first requested by UE k_n was stored by the MEC server, the alleviated backhaul bandwidth in the future should be $\zeta q_{k_n}\bar{R}$. And the memory cost for storing that content is ϖ. So the long term revenue of caching the Internet content first requested by UE k_n is calculated as, $\Lambda_{k_n} = \zeta q_{k_n}\bar{R} - \varpi$.

Next we formulate the utility function of MSO as

$$\begin{aligned} U &= \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} u\left(a_{k_n}\iota_{k_n} + a_{k_n}\Omega_{k_n}\right) + h_{k_n}\Lambda_{k_n} \\ &= \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} u\left[a_{k_n}s_{k_n}\Psi_{k_n} + a_{k_n}\lambda_n\left(\frac{c_{k_n}F}{D_{k_n}} - \frac{f_{k_n}^{(l)}}{D_{k_n}}\right)\right] \\ &\quad + h_{k_n}\Lambda_{k_n} \\ &= \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n}u\left(s_{k_n}\Psi_{k_n} + c_{k_n}\frac{\lambda_n F}{D_{k_n}} - \frac{\lambda_n f_{k_n}^{(l)}}{D_{k_n}}\right) \\ &\quad + h_{k_n}\Lambda_{k_n}, \end{aligned} \tag{8}$$

where $u(\cdot)$ is an utility function which is nondecreasing and convex. Since $h_{k_n}\Lambda_{k_n}$ is always non-negative due to problem optimality, it can be put outside of the function $u(\cdot)$. It is equivalent to take a_{k_n} outside of the function $u(\cdot)$. If $a_{k_n} = 0$, it means UE k_n will not offload the task to the MEC server, so MSO will not earn, then $a_{k_n}u(s_{k_n}, c_{k_n}) = u(a_{k_n}, s_{k_n}, c_{k_n}) = 0$; if $a_{k_n} = 1$, it means MSO may earn, and $a_{k_n}u(s_{k_n}, c_{k_n}) = u(a_{k_n}, s_{k_n}, c_{k_n})$. Here the logarithmic function, which has been used frequently in literature [40], is adopted as the utility function, given as, $u(x) = logx$ when $x > 0$ and $u(x) = -\infty$ otherwise.

Define

$$U' = \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n}u\left(s_{k_n}\Psi_{k_n} + c_{k_n}\frac{\lambda_n F}{D_{k_n}}\right) + h_{k_n}\Lambda_{k_n}. \tag{9}$$

Because $\lambda_n f_{k_n}^{(l)}/D_{k_n}$ is constant, when U' reaches the maximum value, U reaches the maximum as well, i.e., the MSO reaches the maximum income. Let $\lambda_n F/D_{k_n} = \Phi_{k_n}$, next we will use

$$U' = \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n}u\left(s_{k_n}\Psi_{k_n} + c_{k_n}\Phi_{k_n}\right) + h_{k_n}\Lambda_{k_n} \tag{10}$$

as our objective function of the optimization problem.

III. Problem Formulation, Transformation and Decomposition

In order to maximize the utility function of MSO, we formulate it as an optimization problem and transform it into a convex optimization problem.

A. Problem Formulation

We adopt the utility function proposed in (10) as the objective function of our optimization problem, and the problem is formulated as

$$\begin{aligned} \underset{\boldsymbol{a},\boldsymbol{s},\boldsymbol{c},\boldsymbol{h}}{\text{Maximize}} \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n}u\left(s_{k_n}\Psi_{k_n} + c_{k_n}\Phi_{k_n}\right) + h_{k_n}\Lambda_{k_n} \\ s.t.\ C1: \quad & \sum_{k_n\in\mathcal{K}_n} a_{k_n}s_{k_n} \le 1, \quad \forall n \\ C2: \quad & \sum_{k_n\in\mathcal{K}_n} a_{k_n}s_{k_n}Be_{k_n} \le L_n, \quad \forall n \\ C3: \quad & \sum_{m\in\mathcal{N}/\{n\}}\sum_{k_m\in\mathcal{K}_m} a_{k_m}p_{k_m}G_{k_m,n} \le I_n, \quad \forall n \\ C4: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n}c_{k_n} \le 1 \\ C5: \quad & a_{k_n}\left(\frac{c_{k_n}F}{D_{k_n}} - \frac{f_{k_n}^{(l)}}{D_{k_n}}\right) \ge 0, \quad \forall k, n \\ C6: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} h_{k_n} \le Y. \end{aligned} \tag{11}$$

The first set of constraints (11) $C1$ guarantee that in every small cell, the sum of spectrum allocated to all the offloading UEs cannot exceed the total available spectrum of that small cell. Constraints (11) $C2$ mean the sum data rate of all offloading UEs which associate with SeNB n cannot exceed the backhaul capacity of small cell n. If too many UEs are allowed to offload computation tasks to the MEC server, the transmitting delay will be high due to high interference. In order to guarantee a relatively high data rate, constraints (11) $C3$ are proposed to ensure that the interference on SeNB n caused by all offloading UEs which are served by other SeNBs doesn't exceed a predefined threshold, I_n. Constraint (11) $C4$ is due to the request that the sum of computation resource allocated to all offloading UEs in the whole system cannot exceed the total amount of computation resource (total computational capability) of the MEC server. Because we removed $(-\lambda_n f_{k_n}^{(l)}/D_{k_n})$ in (8), we need constraints (11) $C5$ to guarantee that the computation resource allocated to each offloading UE k_n is no less than that of itself. Constraint (11) $C6$ guarantees that the sum size of all the cached content doesn't exceed the total storage capability of the MEC server.

B. Problem Transformation

Problem (11) is difficult to solve due to the following observations:

- Due to the fact that $\boldsymbol{a}$ and $\boldsymbol{h}$ are binary variables, the feasible set of problem (11) is not convex.
- There exist product relationships between $\{a_{k_n}\}$ and linear function of $\{s_{k_n}\}$, as well as $\{c_{k_n}\}$, so that the objective function of problem (11) is not a convex function.
- The problem has a quite large size. If we assume that the average number of UEs in one small cell is k, the number of variables in this problem could reach $4kN$, and the complexity for a central algorithm to find a globally optimal solution will be $O((kN)^x)$ ($x > 0$, $x = 1$ implies a linear algorithm while $x > 1$ implies a polynomial time algorithm) even if we simply consider all the variables as binary variables. In addition, the number of small cells in one macro cell is increasing as time goes on, which results in an even more radically increasing complexity in our problem.

As is shown, problem (11) is a mixed discrete and non-convex optimization problem, and such problems are usually considered as NP-hard problems [41]. Therefore, a transformation and simplification of the original problem are necessary. The transformation of the problem is composed of the following two steps:

1) Binary Variable Relaxation: In order to transform the non-convex feasible set of problem (11) into a convex set, we need to relax binary variables $\boldsymbol{a}$ and $\boldsymbol{h}$ into real value variables as $0 \le a_{k_n} \le 1$, $0 \le h_{k_n} \le 1$ [41]. The relaxed variables can be interpreted as the time fraction of access to the MEC computation resource of UE k_n and the time fraction of sharing the content cache introduced by the request of UE k_n, respectively.

2) Substitution of the Product Term: Due to the non-convex objective function, the problem is still intractable even though we relax the variables. Next we will propose a proposition of the equivalent problem of (11) to make the problem solvable.

Proposition 1: If we define $\tilde{s}_{k_n} = s_{k_n} a_{k_n}$, $\tilde{c}_{k_n} = c_{k_n} a_{k_n}$, and $a_{k_n} u[(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})/a_{k_n}] = 0$ when $a_{k_n} = 0$, the following formulation (12) is equivalent to problem (11):

$$
\begin{aligned}
\underset{\boldsymbol{a},\tilde{\boldsymbol{s}},\tilde{\boldsymbol{c}},\boldsymbol{h}}{\text{Maximize}} \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n} u\left(\frac{\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n}}{a_{k_n}}\right) + h_{k_n}\Lambda_{k_n} \\
s.t.\ \ C1: \quad & \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} \le 1, \quad \forall n \\
C2: \quad & \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} Be_{k_n} \le L_n, \quad \forall n \\
C3: \quad & \sum_{m\in\mathcal{N}/\{n\}}\sum_{k_m\in\mathcal{K}_m} a_{k_m} p_{k_m} G_{k_m,n} \le I_n, \quad \forall n \\
C4: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} \tilde{c}_{k_n} \le 1 \\
C5: \quad & \tilde{c}_{k_n}\frac{F}{D_{k_n}} - a_{k_n}\frac{f_{k_n}^{(l)}}{D_{k_n}} \ge 0, \quad \forall k, n \\
C6: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} h_{k_n} \le Y \\
C7: \quad & a_{k_n} \ge \tilde{s}_{k_n}, \quad a_{k_n} \ge \tilde{c}_{k_n}, \quad \forall k, n.
\end{aligned} \tag{12}
$$

Proof: This proof of proposition 1 is motivated by [42]. If we substitute $\tilde{s}_{k_n} = s_{k_n} a_{k_n}$ and $\tilde{c}_{k_n} = c_{k_n} a_{k_n}$ into (12), we can recover the original optimization problem (11) except the point when $a_{k_n} = 0$. Next we will discuss about this point. Suppose $a_{k_n} = 0$, then $s_{k_n} = 0$ and $c_{k_n} = 0$ will certainly hold because of the problem optimality. Apparently, if UE k_n will not offload computation task to MEC server, SeNB n will not allocate any spectrum resource to UE n, and MEC server will not assign any computation resource to it, either. Thus, the complete mapping between $\{a_{k_n}, s_{k_n}, c_{k_n}\}$ and $\{a_{k_n}, \tilde{s}_{k_n}, \tilde{c}_{k_n}\}$ is as shown in (13) and (14).

$$
s_{k_n} = \begin{cases} \tilde{s}_{k_n}/a_{k_n}, & a_{k_n} > 0, \\ 0, & otherwise, \end{cases} \tag{13}
$$

$$
c_{k_n} = \begin{cases} \tilde{c}_{k_n}/a_{k_n}, & a_{k_n} > 0, \\ 0, & otherwise. \end{cases} \tag{14}
$$

Now it's a one-to-one mapping. Note that constraints (12) C7 guarantee that $\tilde{s}_{k_n}$ and $\tilde{c}_{k_n}$ don't exceed a_{k_n}, and that is because of $s_{k_n} \in [0, 1]$ and $c_{k_n} \in [0, 1]$. □

C. Convexity

In this subsection, we will discuss the convexity of problem (12) using the well known perspective function [43].

Proposition 2: If problem (12) is feasible, it is jointly convex with respect to all the optimization variables $\boldsymbol{a}$, $\tilde{\boldsymbol{s}}$, $\tilde{\boldsymbol{c}}$ and $\boldsymbol{h}$.

Proof: This proof of proposition 2 is similar to [42]. $f(t, x) = x\log(t/x), t \ge 0, x \ge 0$ is the well-known perspective function of $f(x) = \log x$. Next we will give a proof of the continuity of the perspective function $f(t, x) = x\log(t/x), t \ge 0, x \ge 0$ on the point $x = 0$. Let $s = t/x$,

$$
f(t, 0) = \lim_{x\to 0} x\log\frac{t}{x} = \lim_{s\to\infty}\frac{t}{s}\log s = t\lim_{s\to\infty}\frac{\log s}{s} = 0. \tag{15}
$$

So we have $a_{k_n}\log[(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})/a_{k_n}] = 0$ for $a_{k_n} = 0$. Since $(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})$ are linear with respect to $\tilde{s}_{k_n}$ and $\tilde{c}_{k_n}$, $\log(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})$ is a concave function. Then $a_{k_n}\log[(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})/a_{k_n}]$ is concave due to the fact that it is the perspective function of $\log(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})$. The perspective function of a concave function is concave [43]. Furthermore, $h_{k_n}\Lambda_{k_n}$ is linear, then it is obvious that our objective function of problem (12), i.e., $a_{k_n}\log[(\tilde{s}_{k_n}\Psi_{k_n} + \tilde{c}_{k_n}\Phi_{k_n})/a_{k_n}] + h_{k_n}\Lambda_{k_n}$ is concave. On the other hand, all the constraints of problem (12) are linear (the feasible set of the problem is a convex set), so problem (12) is a convex optimization problem. □

A lot of methods could be applied to solve a convex optimization problem. But as far as our problem (12) is concerned, as mentioned above, the size of the problem becomes appreciably large as the number of small cells grows. In addition, if a centralized algorithm is adopted in the MEC server, the signaling overhead of delivering local information (e.g., channel status information (CSI)) to the MEC server could be extremely high. Therefore, it will be more efficient to employ a distributed algorithm which is running on each SeNB as well as the MEC server. In the next section, we

will decouple the optimization problem (12) in order to enable the application of a distributed optimization problem solving method, namely alternating direction method of multipliers (ADMM).

D. Problem Decomposition

In order to make it possible for each SeNB to participate in the computation for problem solving, we need to separate problem (12) so that it can be solved in a distributed manner. However, optimization variables $\boldsymbol{a}$, $\tilde{\boldsymbol{c}}$ and $\boldsymbol{h}$ in problem (12) are considered as global variables, which are not separable in the problem (Specifically speaking, it is constraints (12) C3, C4 and C6 that make the problem inseparable). Thus, in order to make the problem separable, we introduce the local copies of the global variables. Since the global variables concern all the small cells in the network, which means they cannot be handled in any single small cell, we create a copy for every global variable in each small cell. Thus each small cell can independently conduct their computation for problem solving with their local copies. For small cell n, we denote $\hat{\boldsymbol{a}}^n = \{\hat{a}^n_{k_j}\}_{k_j\in\mathcal{K}_j, j\in\mathcal{N}, n\in\mathcal{N}}$,* $\hat{\boldsymbol{c}}^n = \{\hat{c}^n_{k_j}\}_{k_j\in\mathcal{K}_j, j\in\mathcal{N}, n\in\mathcal{N}}$ and $\hat{\boldsymbol{h}}^n = \{\hat{h}^n_{k_j}\}_{k_j\in\mathcal{K}_j, j\in\mathcal{N}, n\in\mathcal{N}}$ as the local copies of $\boldsymbol{a}$, $\tilde{\boldsymbol{c}}$ and $\boldsymbol{h}$, respectively. We have

$$\begin{cases} \hat{a}^n_{k_j} = a_{k_j}, & \forall n, k, j, \\ \hat{c}^n_{k_j} = \tilde{c}_{k_j}, & \forall n, k, j, \\ \hat{h}^n_{k_j} = h_{k_j}, & \forall n, k, j. \end{cases} \tag{16}$$

Letting

$$U'' = \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} \hat{a}^n_{k_n} u\left(\frac{\tilde{s}_{k_n}\Psi_{k_n} + \hat{c}^n_{k_n}\Phi_{k_n}}{\hat{a}^n_{k_n}}\right) + \hat{h}^n_{k_n}\Lambda_{k_n}, \tag{17}$$

next we give the equivalent global consensus version of problem (12) as

$$\begin{aligned} &\underset{\substack{\{\hat{\boldsymbol{a}}^n,\tilde{\boldsymbol{s}},\hat{\boldsymbol{c}}^n,\hat{\boldsymbol{h}}^n\},\\ \{\boldsymbol{a},\tilde{\boldsymbol{c}},\boldsymbol{h}\}}}{\text{Maximize}} \ U'' \\ &s.t.\ C1: \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} \le 1, \quad \forall n \\ &\quad C2: \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} Be_{k_n} \le L_n, \quad \forall n \\ &\quad C3: \sum_{j\in\mathcal{N}/\{n\}}\sum_{k_j\in\mathcal{K}_j} \hat{a}^n_{k_j} p_{k_j} G_{k_j,n} \le I_n, \quad \forall n \\ &\quad C4: \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \hat{c}^n_{k_j} \le 1, \quad \forall n \\ &\quad C5: \hat{c}^n_{k_j}\frac{F}{D_{k_j}} - \hat{a}^n_{k_j}\frac{f^{(l)}_{k_j}}{D_{k_j}} \ge 0, \quad \forall n, k, j \\ &\quad C6: \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \hat{h}^n_{k_j} \le Y, \quad \forall n \\ &\quad C7: \hat{a}^n_{k_n} \ge \tilde{s}_{k_n}, \quad \hat{a}^n_{k_n} \ge \hat{c}^n_{k_n}, \quad \forall k, n \\ &\quad C8: \hat{a}^n_{k_j} = a_{k_j}, \quad \hat{c}^n_{k_j} = \tilde{c}_{k_j}, \ \hat{h}^n_{k_j} = h_{k_j}, \quad \forall n, k, j. \end{aligned} \tag{18}$$

*Here we need to introduce another small cell index $j \in \mathcal{N}$ to indicate each small cell in the local copy of small cell n.

The consensus constraint (18) C8 imposes that all the local copy variables in all small cells (i.e., $\{\hat{a}^n_{k_j}, \hat{c}^n_{k_j}, \hat{h}^n_{k_j}\}_{n\in\mathcal{N}}$) *must be consistent with* the corresponding global variables (i.e., $\{a_{k_j}, \tilde{c}_{k_j}, h_{k_j}\}$).

For ease of description, we define the following set as the local variable *feasible set* of each small cell $n \in \mathcal{N}$:

$$\xi_n = \left\{ \begin{matrix} \hat{\boldsymbol{a}}^n \\ \tilde{\boldsymbol{s}}_n \\ \hat{\boldsymbol{c}}^n \\ \hat{\boldsymbol{h}}^n \end{matrix} \middle| \begin{array}{l} \sum\limits_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} \le 1 \\ \sum\limits_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} Be_{k_n} \le L_n \\ \sum\limits_{j\in\mathcal{N}/\{n\}}\sum\limits_{k_j\in\mathcal{K}_j} \hat{a}^n_{k_j} p_{k_j} G_{k_j,n} \le I_n \\ \sum\limits_{j\in\mathcal{N}}\sum\limits_{k_j\in\mathcal{K}_j} \hat{c}^n_{k_j} \le 1 \\ \hat{c}^n_{k_j} F/D_{k_j} - \hat{a}^n_{k_j} f^{(l)}_{k_j}/D_{k_j} \ge 0, \forall k, j \\ \sum\limits_{j\in\mathcal{N}}\sum\limits_{k_j\in\mathcal{K}_j} \hat{h}^n_{k_j} \le Y \\ \hat{a}^n_{k_n} \ge \tilde{s}_{k_n}, \hat{a}^n_{k_n} \ge \hat{c}^n_{k_n}, \forall k \end{array} \right\}, \quad \forall n. \tag{19}$$

Note that ξ_n is proprietary for small cell n and is completely decoupled from other small cells.

Next we give the local utility function of each small cell $n \in \mathcal{N}$ as follows

$$\upsilon_n = \begin{cases} -\left[\sum\limits_{k_n\in\mathcal{K}_n} \hat{a}^n_{k_n} u\left(\frac{\tilde{s}_{k_n}\Psi_{k_n} + \hat{c}^n_{k_n}\Phi_{k_n}}{\hat{a}^n_{k_n}}\right) + \hat{h}^n_{k_n}\Lambda_{k_n}\right], \\ \qquad when \quad \{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\} \in \xi_n, \\ +\infty, \quad otherwise. \end{cases} \tag{20}$$

With (19) and (20), an equivalent formulation of problem (18) is given as

$$\begin{aligned} &\underset{\{\hat{\boldsymbol{a}}^n,\tilde{\boldsymbol{s}},\hat{\boldsymbol{c}}^n,\hat{\boldsymbol{h}}^n\},\{\boldsymbol{a},\tilde{\boldsymbol{c}},\boldsymbol{h}\}}{\text{Minimize}} \ \sum_{n\in\mathcal{N}} \upsilon_n(\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n) \\ &s.t.\ C1: \ \hat{a}^n_{k_j} = a_{k_j}, \quad \forall n, k, j \\ &\quad C2: \ \hat{c}^n_{k_j} = \tilde{c}_{k_j}, \quad \forall n, k, j \\ &\quad C3: \ \hat{h}^n_{k_j} = h_{k_j}, \quad \forall n, k, j. \end{aligned} \tag{21}$$

Now it is obvious that in problem (21) the objective functions υ_n with feasible sets ξ_n are separable with respect to all the small cells in the system. But the *consensus constraints* (21) C1–C3 remain coupled upon all the small cells. That is exactly what we want. The separation of the objective functions enables each small cell to independently handle the subproblem related to itself, while the persistence of coupling of the consensus constraints (21) C1–C3 guarantees the consistency of all the local copies with each other, as well as with the real global variables. In the next section we will apply *Alternating Direction Method of Multipliers (ADMM)* to solve the problem in a distributed fashion.

IV. Problem Solving via Alternating Direction Method of Multipliers

In this section, first, we will derive the augmented Lagrangian with corresponding global consensus constraints

and formulate the ADMM iteration steps [44]–[46]; secondly, the update methods for ADMM iterations are presented; thirdly, the relaxed variables are recovered to binary variables; finally, the overall algorithm is summarized.

A. Augmented Lagrangian and ADMM Sequential Iterations

According to [44], problem (21) is called a *global consensus problem*, due to the fact that all the local variables are consistent (with the global variables). According to [44], the augmented Lagrangian of problem (21) is given as

$$\begin{aligned} &L_\rho(\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\}_{n\in\mathcal{N}}, \{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}, \{\boldsymbol{\sigma}^n, \boldsymbol{\omega}^n, \boldsymbol{\tau}^n\}_{n\in\mathcal{N}}) \\ &= \sum_{n\in\mathcal{N}} v_n(\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n) + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \sigma_{k_j}^n(\hat{a}_{k_j}^n - a_{k_j}) \\ &\quad + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \omega_{k_j}^n(\hat{c}_{k_j}^n - \tilde{c}_{k_j}) \\ &\quad + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \tau_{k_j}^n(\hat{h}_{k_j}^n - h_{k_j}) \\ &\quad + \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} (\hat{a}_{k_j}^n - a_{k_j})^2 \\ &\quad + \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} (\hat{c}_{k_j}^n - \tilde{c}_{k_j})^2 \\ &\quad + \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} (\hat{h}_{k_j}^n - h_{k_j})^2, \end{aligned} \tag{22}$$

where $\boldsymbol{\sigma}^n = \{\sigma_{k_j}^n\}_{n\in\mathcal{N}}$, $\boldsymbol{\omega}^n = \{\omega_{k_j}^n\}_{n\in\mathcal{N}}$ and $\boldsymbol{\tau}^n = \{\tau_{k_j}^n\}_{n\in\mathcal{N}}$ are the Lagrange multipliers with respect to (18) C8, and $\rho \in \mathbb{R}_{++}$ is the so called *penalty parameter*, which is a constant parameter intended for adjusting the convergence speed of ADMM [44]. Compared to standard Lagrangian, the additional ρ-terms in augmented Lagrangian (22) can improve the property of the iterative method [47]. Please note that for any feasible solution, the ρ-terms added in augmented Lagrangian (22) are actually *equal to zero* [44].

With ADMM being applied to solving problem (21), the following sequential iterative optimization steps are presented as findings [44].

Local variables:

$$\begin{aligned} &\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\}_{n\in\mathcal{N}}^{[t+1]} \\ &= \underset{\{\hat{a}_{k_j}^n, \tilde{s}_{k_n}, \hat{c}_{k_j}^n, \hat{h}_{k_j}^n\}}{\arg\min} \left\{ \begin{aligned} &v_n(\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n) \\ &+ \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \sigma_{k_j}^{n[t]}\left(\hat{a}_{k_j}^n - a_{k_j}^{[t]}\right) \\ &+ \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \omega_{k_j}^{n[t]}\left(\hat{c}_{k_j}^n - \tilde{c}_{k_j}^{[t]}\right) \\ &+ \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \tau_{k_j}^{n[t]}\left(\hat{h}_{k_j}^n - h_{k_j}^{[t]}\right) \\ &+ \frac{\rho}{2}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{a}_{k_j}^n - a_{k_j}^{[t]}\right)^2 \\ &+ \frac{\rho}{2}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{c}_{k_j}^n - \tilde{c}_{k_j}^{[t]}\right)^2 \\ &+ \frac{\rho}{2}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{h}_{k_j}^n - h_{k_j}^{[t]}\right)^2 \end{aligned} \right\} \end{aligned} \tag{23}$$

Global variables:

$$\begin{aligned} &\{\boldsymbol{a}\}^{[t+1]} \\ &= \underset{\{a_{k_j}\}}{\arg\min} \left\{ \begin{aligned} &\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \sigma_{k_j}^{n[t]}\left(\hat{a}_{k_j}^{n[t+1]} - a_{k_j}\right) \\ &+ \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{a}_{k_j}^{n[t+1]} - a_{k_j}\right)^2 \end{aligned} \right\} \end{aligned} \tag{24}$$

$$\begin{aligned} &\{\tilde{\boldsymbol{c}}\}^{[t+1]} \\ &= \underset{\{\tilde{c}_{k_j}\}}{\arg\min} \left\{ \begin{aligned} &\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \omega_{k_j}^{n[t]}\left(\hat{c}_{k_j}^{n[t+1]} - \tilde{c}_{k_j}\right) \\ &+ \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{c}_{k_j}^{n[t+1]} - \tilde{c}_{k_j}\right)^2 \end{aligned} \right\} \end{aligned} \tag{25}$$

$$\begin{aligned} &\{\boldsymbol{h}\}^{[t+1]} \\ &= \underset{\{h_{k_j}\}}{\arg\min} \left\{ \begin{aligned} &\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j} \tau_{k_j}^{n[t]}\left(\hat{h}_{k_j}^{n[t+1]} - h_{k_j}\right) \\ &+ \frac{\rho}{2}\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left(\hat{h}_{k_j}^{n[t+1]} - h_{k_j}\right)^2 \end{aligned} \right\} \end{aligned} \tag{26}$$

Lagrange multipliers:

$$\{\boldsymbol{\sigma}^n\}_{n\in\mathcal{N}}^{[t+1]} = \boldsymbol{\sigma}^{n[t]} + \rho(\hat{\boldsymbol{a}}^{n[t+1]} - \boldsymbol{a}^{[t+1]}) \tag{27}$$

$$\{\boldsymbol{\omega}^n\}_{n\in\mathcal{N}}^{[t+1]} = \boldsymbol{\omega}^{n[t]} + \rho(\hat{\boldsymbol{c}}^{n[t+1]} - \tilde{\boldsymbol{c}}^{[t+1]}) \tag{28}$$

$$\{\boldsymbol{\tau}^n\}_{n\in\mathcal{N}}^{[t+1]} = \boldsymbol{\tau}^{n[t]} + \rho(\hat{\boldsymbol{h}}^{n[t+1]} - \boldsymbol{h}^{[t+1]}), \tag{29}$$

where the superscript $[t]$ stands for the iteration index.

It is obvious that the iteration steps (23) concerning local variables are completely separable with respect to the small cell index n, thus can be executed by each SeNB. The iteration steps (24)–(29) concerning global variables and Lagrange multipliers would be executed by the MEC server. In the following subsections, we will discuss the methods for solving these iterations.

B. Local Variables $\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\}_{n\in\mathcal{N}}$ Update

As described above, iteration (23) is decomposed into N subproblems, with each of them being solved by an SeNB. Thus, after eliminating the constant terms, it is equivalent for SeNB $n \in \mathcal{N}$ to solve the following optimization problem at iteration $[t+1]$,

$$\begin{aligned} &\underset{\substack{\{\hat{a}_{k_j}^n, \tilde{s}_{k_n}, \\ \hat{c}_{k_j}^n, \hat{h}_{k_j}^n\}}}{\text{Minimize}}\ v_n(\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n) \\ &\qquad + \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left[\sigma_{k_j}^{n[t]}\hat{a}_{k_j}^n + \frac{\rho}{2}\left(\hat{a}_{k_j}^n - a_{k_j}^{[t]}\right)^2\right] \\ &\qquad + \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left[\omega_{k_j}^{n[t]}\hat{c}_{k_j}^n + \frac{\rho}{2}\left(\hat{c}_{k_j}^n - \tilde{c}_{k_j}^{[t]}\right)^2\right] \\ &\qquad + \sum_{j\in\mathcal{N}}\sum_{k_j\in\mathcal{K}_j}\left[\tau_{k_j}^{n[t]}\hat{h}_{k_j}^n + \frac{\rho}{2}\left(\hat{h}_{k_j}^n - h_{k_j}^{[t]}\right)^2\right] \\ &s.t.\ \{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\} \in \xi_n. \end{aligned} \tag{30}$$

Obviously, problem (30) is a convex problem due to its quadric objective function and convex feasible set. So here

Algorithm 1 Primal-Dual Interior-Point Method for Local Variables Updating

1: Initialization
Given $\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\} \in \xi_n$, $\varrho > 0$, $\varsigma > 1$, $\epsilon_{feas} > 0$, $\epsilon > 0$.
2: **Repeat**
a) Determine t. Set $t := \varsigma m/\hat{\eta}$;
b) Compute primal-dual search direction $\Delta_{y_{pd}}$;
c) Line search and update
Determine step length $s > 0$ and set $y := y + s\Delta_{y_{pd}}$.
Until $\|r_{pri}\|_2 \leq \epsilon_{feas}$, $\|r_{dual}\|_2 \leq \epsilon_{feas}$, and $\hat{\eta} \leq \epsilon$.

we employ *primal-dual interior-point method* [43] to solve this problem, which is briefly described in Algorithm 1.

In Algorithm 1, $\hat{\eta}$ stands for the surrogate duality gap, and m denotes the number of constraints. Due to the limited space, the detailed description about this method is omitted here, and readers could turn to [43] for more information, where the method is described in detail.

We only need to consider how to provide an initial feasible solution $\{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}^{[0]}$ for algorithm 1. In order to do so, let us consider an extreme case, where only one UE among all the UEs in system is allowed to offload computation task to MEC server. And we designate this UE as UE $\bar{k}_{\bar{j}}$, so we set $a^{[0]}_{\bar{k}_{\bar{j}}} = 1$ and $a^{[0]}_{k_j} = 0, \forall k_j \neq \bar{k}_{\bar{j}}$. Naturally, when only one UE is offloading computation task to MEC server, all the computation resource would be allocated to this UE, thus $\tilde{c}^{[0]}_{\bar{k}_{\bar{j}}} = 1$ and $\tilde{c}^{[0]}_{k_j} = 0, \forall k_j \neq \bar{k}_{\bar{j}}$. And all the radio resource of small cell $\bar{j}$ will be assigned to UE $\bar{k}_{\bar{j}}$, so $s_{\bar{k}_{\bar{j}}} = 1$, while $s_{k_{\bar{j}}} = 0, \forall k \neq \bar{k}$. As all the other small cells are concerned, they will not allocate any spectrum resource to any of their associating UEs, since all their UEs will execute the computation tasks locally. Thus $s_{k_j} = 0, \forall k_j \neq \bar{k}_{\bar{j}}$. As the caching strategy is concerned, we assume that the MEC server chose to store the Internet content requested by UE $\bar{k}_{\bar{j}}$ and not to store content requested by any other UEs, thus we have $h^{[0]}_{\bar{k}_{\bar{j}}} = 1$ and $h^{[0]}_{k_j} = 0, \forall k_j \neq \bar{k}_{\bar{j}}$. By doing so, the constraints of problem (30) are automatically satisfied.

C. Global Variables $\{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}$ and Lagrange Multipliers $\{\boldsymbol{\sigma}^n, \boldsymbol{\omega}^n, \boldsymbol{\tau}^n\}_{n\in\mathcal{N}}$ Update

Now we move on to the global variables. Since problem (24), (25) and (26) are unconstrained quadratic problems and are strictly convex due to the added quadratic regularization terms in augmented Lagrangian (22), we can solve them by simply setting the *gradients* of $\boldsymbol{a}$, $\tilde{\boldsymbol{c}}$ and $\boldsymbol{h}$ to zeros, i.e.,

$$\sum_{n\in\mathcal{N}} \sigma^{n[t]}_{k_j} + \rho \sum_{n\in\mathcal{N}} \left(\hat{a}^{n[t+1]}_{k_j} - a_{k_j}\right) = 0, \quad \forall k, j \tag{31}$$

$$\sum_{n\in\mathcal{N}} \omega^{n[t]}_{k_j} + \rho \sum_{n\in\mathcal{N}} \left(\hat{c}^{n[t+1]}_{k_j} - \tilde{c}_{k_j}\right) = 0, \quad \forall k, j \tag{32}$$

$$\sum_{n\in\mathcal{N}} \tau^{n[t]}_{k_j} + \rho \sum_{n\in\mathcal{N}} \left(\hat{h}^{n[t+1]}_{k_j} - h_{k_j}\right) = 0, \quad \forall k, j \tag{33}$$

and this result in

$$a^{[t+1]}_{k_j} = \frac{1}{N\rho} \sum_{n\in\mathcal{N}} \sigma^{n[t]}_{k_j} + \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{a}^{n[t+1]}_{k_j}, \quad \forall k, j \tag{34}$$

$$\tilde{c}^{[t+1]}_{k_j} = \frac{1}{N\rho} \sum_{n\in\mathcal{N}} \omega^{n[t]}_{k_j} + \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{c}^{n[t+1]}_{k_j}, \quad \forall k, j \tag{35}$$

$$h^{[t+1]}_{k_j} = \frac{1}{N\rho} \sum_{n\in\mathcal{N}} \tau^{n[t]}_{k_j} + \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{h}^{n[t+1]}_{k_j}, \quad \forall k, j. \tag{36}$$

By initializing the Lagrange multipliers as zeros at iteration $[t]$ [44], i.e., $\sum_{n\in\mathcal{N}} \sigma^{n[t]}_{k_j} = 0$, $\sum_{n\in\mathcal{N}} \omega^{n[t]}_{k_j} = 0$, $\sum_{n\in\mathcal{N}} \tau^{n[t]}_{k_j} = 0$, $\forall k, j$, equations (34)– (36) reduce to

$$a^{[t+1]}_{k_j} = \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{a}^{n[t+1]}_{k_j}, \quad \forall k, j \tag{37}$$

$$\tilde{c}^{[t+1]}_{k_j} = \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{c}^{n[t+1]}_{k_j}, \quad \forall k, j \tag{38}$$

$$h^{[t+1]}_{k_j} = \frac{1}{N} \sum_{n\in\mathcal{N}} \hat{h}^{n[t+1]}_{k_j}, \quad \forall k, j. \tag{39}$$

Equations (37)–(39) imply that at each iteration the global variables are calculated by averaging out all the corresponding local copies in all the small cells, which can be philosophically interpreted as the summary of the small cells' opinions on the optimal global variables.

The process of Lagrange multipliers $\{\boldsymbol{\sigma}^n, \boldsymbol{\omega}^n, \boldsymbol{\tau}^n\}_{n\in\mathcal{N}}$ updating is simple compared to $\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\}_{n\in\mathcal{N}}$ and $\{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}$ updating. With the current local variables received from each SeNB, the MEC server can easily obtain the Lagrange multipliers using equations (27)–(29) in each iteration.

D. Algorithm Stopping Criterion and Convergence

Apparently, all the variables of problem (21) are bounded and the objective function of the problem is bounded, too, so inequality $\sum_{n\in\mathcal{N}} \upsilon_n(\hat{\boldsymbol{a}}^{n*}, \tilde{\boldsymbol{s}}^*_n, \hat{\boldsymbol{c}}^{n*}, \hat{\boldsymbol{h}}^{n*}) < \infty$ holds, where $\{\hat{\boldsymbol{a}}^{n*}, \tilde{\boldsymbol{s}}^*_n, \hat{\boldsymbol{c}}^{n*}, \hat{\boldsymbol{h}}^{n*}\}$ is the optimal solution of problem (21). Since problem (21) is a convex optimization problem (the proof of the convexity of the problem has been given in Section III-C), the strong duality holds [43]. According to [44], the objective function of problem (21) is convex, closed and proper, and the Lagrangian (22) has saddle point, so the ADMM iterations described above satisfy residual convergence, objective convergence and dual variable convergence when $t \to \infty$.

For implementation purposes, we can employ the rational stopping criterion proposed in [44], which is given as

$$\| r_p^{[t+1]} \|_2 \leq \vartheta_{pri} \quad and \quad \| r_d^{[t+1]} \|_2 \leq \vartheta_{dual}, \tag{40}$$

where $\vartheta_{pri} > 0$ and $\vartheta_{dual} > 0$ are small positive constant scalars, which are called the feasibility tolerances for the primal and dual feasibility conditions, respectively. This stopping criterion implies that the primal residual $r_p^{[t+1]}$ and the dual residual $r_d^{[t+1]}$ must be small.

As this stopping criterion in [44] being applied to our algorithm, the residual for the primal feasibility condition of

Algorithm 2 Binary Variables Recovery

1: Computing first partial derivations
Compute the first partial derivations of augmented Lagrangian $Q_{k_n} = \partial L_\rho / \partial a_{k_n}$ with respect to each a_{k_n}.
2: Sort all the partial derivations $Q_{k_n}, \forall k, n$ from largest to smallest. Mark them with $Q_1, Q_2 \ldots Q_i \ldots$, and mark the corresponding a_{k_n} as $a_1, a_2 \ldots a_i \ldots$
3: **For** i=1,2,..., **Do**
Set $a_i = 1$ and $a_{i+1}, a_{i+2}, a_{i+3} \ldots = 0$;
If Any of the constrains (11) C1-C6 does not hold, **Then Break**.
End for
4: Output the recovered binary variables $\{a_{k_n}\}, \forall k, n$.

small cell n in iteration $[t+1]$ must be small enough so that

$$\| \hat{\boldsymbol{a}}^{n[t+1]} - \boldsymbol{a}^{[t+1]} \|_2 \leq \vartheta_{pri}, \quad \forall n, \tag{41}$$

$$\| \hat{\boldsymbol{c}}^{n[t+1]} - \tilde{\boldsymbol{c}}^{[t+1]} \|_2 \leq \vartheta_{pri}, \quad \forall n, \tag{42}$$

$$\| \hat{\boldsymbol{h}}^{n[t+1]} - \boldsymbol{h}^{[t+1]} \|_2 \leq \vartheta_{pri}, \quad \forall n, \tag{43}$$

and the residual for the dual feasibility condition in iteration $[t+1]$ must be small enough so that

$$\| \boldsymbol{a}^{[t+1]} - \boldsymbol{a}^{[t]} \|_2 \leq \vartheta_{dual}, \tag{44}$$

$$\| \tilde{\boldsymbol{c}}^{[t+1]} - \tilde{\boldsymbol{c}}^{[t]} \|_2 \leq \vartheta_{dual}, \tag{45}$$

$$\| \boldsymbol{h}^{[t+1]} - \boldsymbol{h}^{[t]} \|_2 \leq \vartheta_{dual}. \tag{46}$$

E. Binary Variables Recovery

In order to transform the original problem (11) into a convex problem, we have relaxed the binary variables $\boldsymbol{a}$ and $\boldsymbol{h}$ into continuous variables in Section III-B.1. So we need to recover the binary variables after the convergence of ADMM process. In order to maximize the revenue of MSO, we try to maximize the number of offloading UEs and to store as much Internet content as possible. The recovery deals with the marginal benefit of each UE. We adopt the following algorithm 2 to recover the binary variables $\boldsymbol{a}$ and $\boldsymbol{h}$. In algorithm 2 we use $\boldsymbol{a}$ as an example, and the same algorithm is applied to $\boldsymbol{h}$. It should be mentioned that the recovery of binary variables creates a gap between our results and the upper bound results. (Since the problem is NP-hard, it is very difficult to examine the existence of the optimal results.) However, as will be shown in the simulation, the gap is not significant.

F. Feasibility, Complexity and Summary of the Algorithm

If the computational capability and storage capability of the MEC server is too low, or the cost of the spectrum is too high, the utility function of our problem may become non-positive under all possible solutions. In that case, the optimal solution would be $\{\boldsymbol{a}^*, \boldsymbol{s}^*, \boldsymbol{c}^*, \boldsymbol{h}^*\} = \{\vec{0}, \vec{0}, \vec{0}, \vec{0}\}$, which means that all the UEs in system will execute their computation tasks locally, and no spectrum and computation resources are allocated to any UE. Besides, no requested Internet content will be stored by the MEC server. This case may be treated as the problem becomes infeasible. However, except for the extreme cases, the MEC server could allow at least one UE offload its computation task or store at least one content, and gain a positive revenue, due to the fact that a typical MEC server could always have a much higher computational capability and storage capability than a single UE.

Now let us discuss about the complexity of our algorithm by comparing it with the complexity of the centralized algorithm. First we assume that the average number of UEs in each small cell is k, and the total number of small cells is N. Thus the size of input for the centralized algorithm would be kN. If the centralized algorithm adopted the primal-dual interior-point method for convex optimization problem solving at each input, the complexity would be $O((kN)^x)$ with $x > 0$, where $x = 1$ implies a linear algorithm while $x > 1$ implies a polynomial time algorithm. Now we move on to our proposed distributed algorithm. In local variables updating (23), the size of input is k, due to the fact that each small cell only needs to mind its own associating UEs, and we employ primal-dual interior-point method for problem solving in each iteration, thus the complexity would be $O(k^x)$ with $x > 0$. In global variables updating (24)–(26), we denote y as the number of elementary steps needed for calculation in (24)–(26). Then the complexity is given as kNy. In Lagrange multipliers updating (27)–(29), we use z as the number of elementary steps needed for calculation in (27)–(29), so the time complexity is calculated as kz. Thus the sum of time complexity in each iteration would be $O(k^x) + kNy + kz = O(k^x)$. Assuming P stands for the number of iterations needed for the algorithm convergence, the overall time complexity of the distributed algorithm would be $O(k^x)P$. As will be shown in simulation, the number of iterations before algorithm convergence is not large. So it can be seen that our proposed distributed algorithm can significantly reduce the time complexity compared to the centralized algorithm.

Our overall resources allocation algorithm is summarized in Algorithm 3.

V. Simulation Results and Discussions

In this section, simulation results of the proposed decentralized scheme are presented in comparison with the centralized scheme and several baseline schemes. The simulation is run on a Matlab-based simulator. Unless otherwise mentioned, most of the simulations employ the following scenario. We consider 10-50 small cells that are randomly deployed in a 120×120 m^2 area. It is worth noting that most of the results of simulation studies in this section are based on an average over a number of Monte Carlo simulations for various system parameters. There are 4-10 UEs connected to one SeNB, as mentioned in Section II. The transmission power of single UE, P_n is set to 100 mW. The channel gain models presented in 3GPP standardization are adopted here. The total size of the Internet content is 1000 files, and the storage capability of the MEC server is 1000 files. The main simulation parameters employed in the simulations, unless mentioned otherwise, are summarized in table II.

We first present the convergence of the proposed ADMM-based algorithm with different values of parameter ρ.

Algorithm 3 Decentralized Resources Allocation Algorithm in MEC System via ADMM

1: Initialization
a) MEC server determines the stopping criterion threshold ϑ_{pri} and ϑ_{dual};
b) MEC server initializes the initial feasible solution $\{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}^{[0]}$ as described in Section IV-B and sends them to each SeNB;
c) Each SeNB n collects CSI of all its associating UEs;
d) Each SeNB n determines its initial Lagrange multipliers vectors $\{\boldsymbol{\sigma}^{n[0]} > \mathbf{0}, \boldsymbol{\omega}^{n[0]} > \mathbf{0}, \boldsymbol{\tau}^{n[0]} > \mathbf{0}\}$ and sends them to MEC server;
t=0.
2: Iterations
Repeat
a) Each SeNB n updates its local variables $\{\hat{\boldsymbol{a}}^n, \tilde{\boldsymbol{s}}_n, \hat{\boldsymbol{c}}^n, \hat{\boldsymbol{h}}^n\}_{n\in\mathcal{N}}^{[t+1]}$ by solving problem (30) and transmits them to MEC server;
b) The MEC server updates the global variables $\{\boldsymbol{a}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}^{[t+1]}$ and transmits them to each SeNB;
c) The MEC server updates the Lagrange multipliers $\{\boldsymbol{\sigma}^n, \boldsymbol{\omega}^n, \boldsymbol{\tau}^n\}_{n\in\mathcal{N}}^{[t+1]}$ and transmits them to each SeNB;
t=t+1.
Until $\| \boldsymbol{a}^{[t+1]} - \boldsymbol{a}^{[t]} \|_2 \leq \vartheta_{dual}$, $\| \tilde{\boldsymbol{c}}^{[t+1]} - \tilde{\boldsymbol{c}}^{[t]} \|_2 \leq \vartheta_{dual}$ and $\| \boldsymbol{h}^{[t+1]} - \boldsymbol{h}^{[t]} \|_2 \leq \vartheta_{dual}$.
3: Output
Output the optimal solution $\{\boldsymbol{a}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{c}}, \boldsymbol{h}\}^*$.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Bandwidth	20MHz
Transmission power of UE n, P_n	100 mWatts
Background noise σ^2	-100 dBm
Data size for computation offloading Z_{k_n}	420 KB
Number of CPU cycles of computation task D_{k_n}	1,000 Megacycles
Computation capability of UE n, $f_{k_n}^{(l)}$	0.7 GHz [48]
Computation capability of the MEC server F	100 GHz [48]

As is shown in Fig. 2, the utilities of ADMM-based algorithm increase dramatically in the first 15 iterations and then enter a stable status within the first 40 iterations. So it is proper to say that the decentralized algorithm can converge quickly. All the three iterative progresses converge to the same utility value eventually. The progress with a ρ value $\rho = 1.2$ converges fastest, while $\rho = 0.4$ slowest, but the difference is not significant. As can be seen in Fig. 2, the gap between ADMM-based algorithm and centralized algorithm is narrow.

Next the percentages of offloading UEs in all UEs with an increasing total number of small cells are shown in Fig. 3. Here the number of UEs connected to each SeNB is set as 6. The offloading UE percentages of ADMM-based algorithm and centralized algorithm are compared in Fig. 3. The percentage of offloading UEs remains 100 with small total number of small cells, but as the total number of small cells keeps increasing, the percentage of offloading UEs begins to decline.

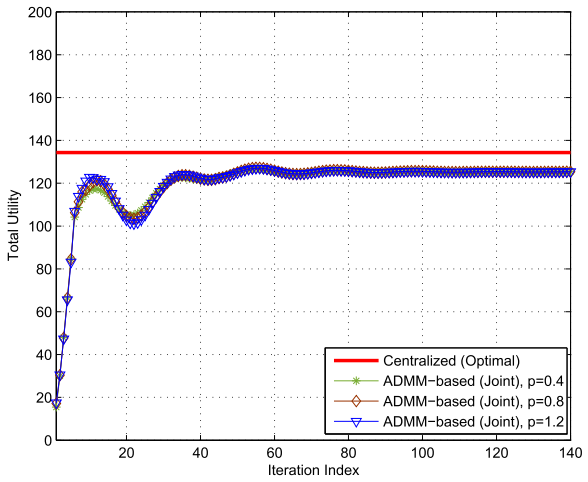


Fig. 2. Convergence progresses of ADMM-based algorithm with different values of ρ.

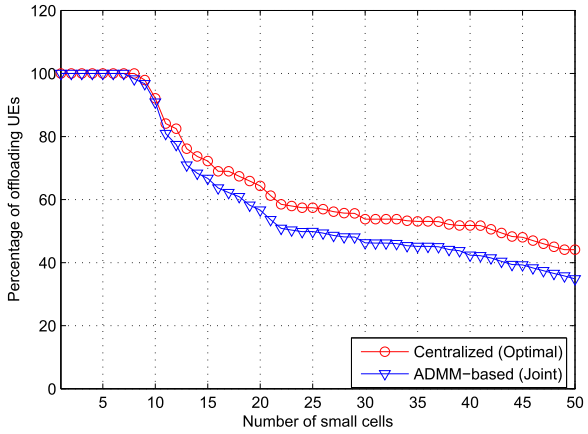


Fig. 3. Percentage of offloading UEs versus number of small cells.

This is because when the total number of small cells is small enough, all the UEs are allowed to offload their computation tasks in order to maximize the network revenue, on the other hand, when the total number of small cells becomes large enough, there will be more UEs that tend to offload their computation tasks to the MEC server, and this would cause severe interference to each other, so the algorithm automatically rejects some of the offloading requests generated by UEs.

Fig. 4 and Fig. 5 show the spectrum and computation resources distribution among all the UEs in system, respectively. Here we only set 4 small cells in system, and there are 4 UEs associating to each SeNB. In this case all the 16 UEs are allowed to offload their computation tasks to the MEC server. As we can see, due to different channel conditions and sizes of computation tasks of different UEs, the resource allocation among UEs is not uniform, in order to reach the optimal utility value. In Fig. 4 and Fig. 5, the resource allocation decisions of our ADMM-based algorithm are compared with that of the centralized algorithm. As is shown, except for slight discrepancies on a few UEs, they approximately coincide with each other.

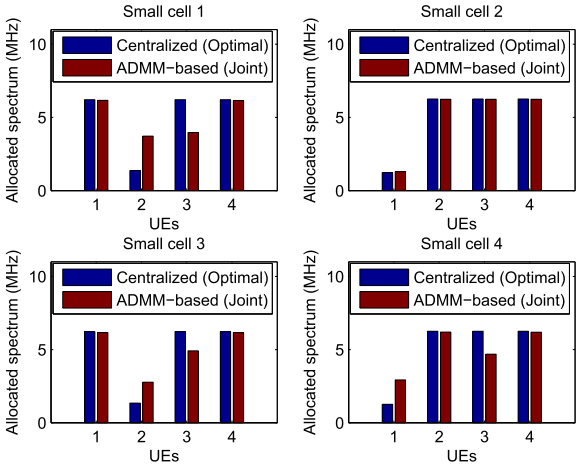


Fig. 4. Spectrum allocation among UEs.

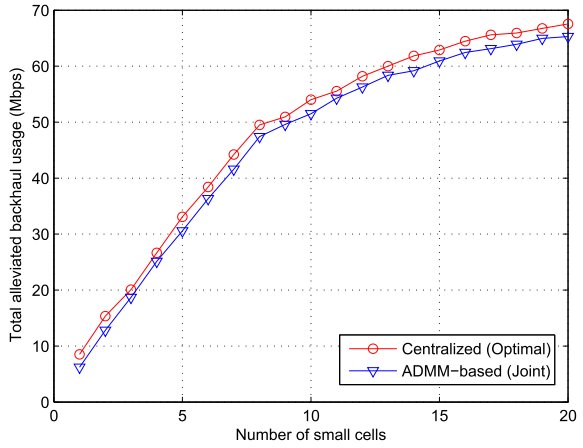


Fig. 6. Total alleviated backhaul usage versus number of small cells.

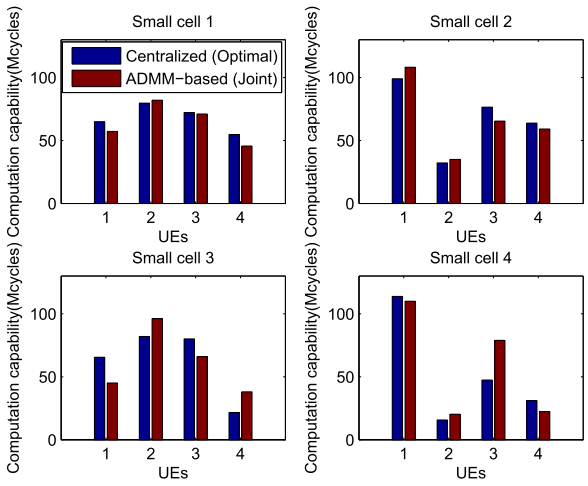


Fig. 5. Computation resource allocation among UEs.

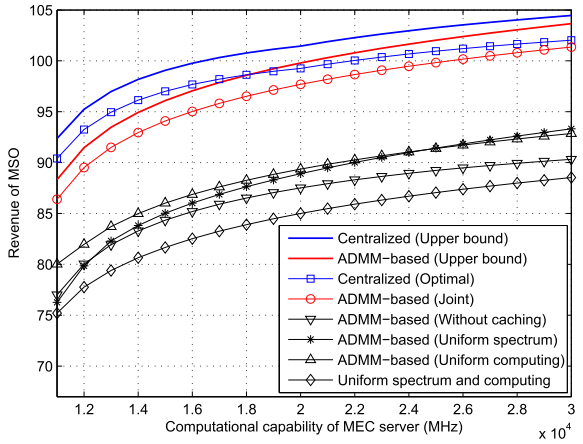


Fig. 7. MSO revenue versus MEC server computational capability.

Fig. 6 shows the total alleviated backhaul usage with respect to an increasing number of small cells. Note that total alleviated backhaul means the accumulated alleviation of backhaul usage upon all the UEs in system. As is shown in Fig. 6, the alleviated backhaul of our proposed ADMM-based algorithm is very close to that of the centralized algorithm. The alleviated backhaul of both algorithms keeps increasing with an increasing number of small cells (UEs).

Fig. 7 shows the revenue of MSO (utility value) with respect to the increasing computational capability, where the total number of SeNBs is 20. The revenue of ADMM-based algorithm is shown by the red line, in comparison with revenue of centralized algorithm (in blue line) and other benchmark solutions (Which are, ADMM solution without caching, ADMM solution with spectrum uniformly allocated, ADMM solution with computation resource uniformly allocated, spectrum and computation resource uniformly allocated, respectively). In Fig. 7, 8 and 9, we also calculate the revenue before the binary variables are recovered and present them, which serve as the upper bound results and are given the legend "Upper bound", to demonstrate that the binary variables recovery operation does not significantly reduce the algorithm performance. The centralized algorithm achieves the highest revenue among all the solutions, but it is obvious that the gap between ADMM-based algorithm and centralized algorithm is not wide. In contrast, ADMM with uniform spectrum allocation solution and ADMM with uniform computation resource allocation solution can just achieve much lower revenue. This is because uniform resource allocation usually cannot reach the optimal revenue. Then it is no wonder the solution with uniform spectrum and computation resource allocation achieves the lowest MSO revenue. Finally, without the revenue of alleviated backhaul bandwidth, ADMM solution without caching can only achieve a much lower total revenue compared to joint ADMM-based solution.

Fig. 8 shows the revenue of MSO (utility value) with respect to the increasing number of small cells. It can be seen that with an increasing number of small cells, the revenues of all the solutions increase dramatically at first, because with more and more small cells joining into the system, the spectrum can be reused among more and more small cells, then the MSO could gain more from allocating spectrum to more small cells. Nevertheless, when the number of small cells reaches about 10, the acceleration of the increasing revenue significantly goes down. The main reason is that when there are too many small cells in the system, all those UEs will cause severe interference

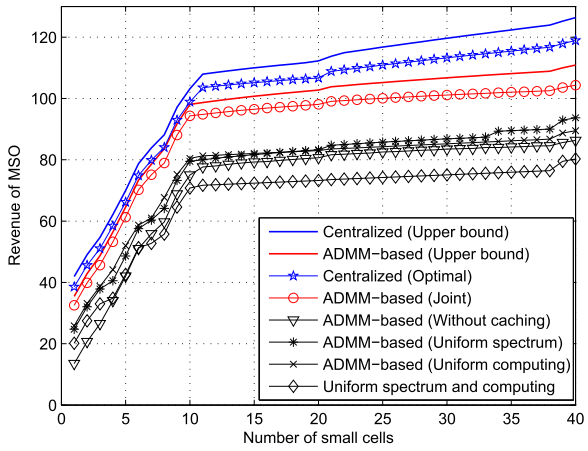


Fig. 8. MSO revenue versus the number of small cells.

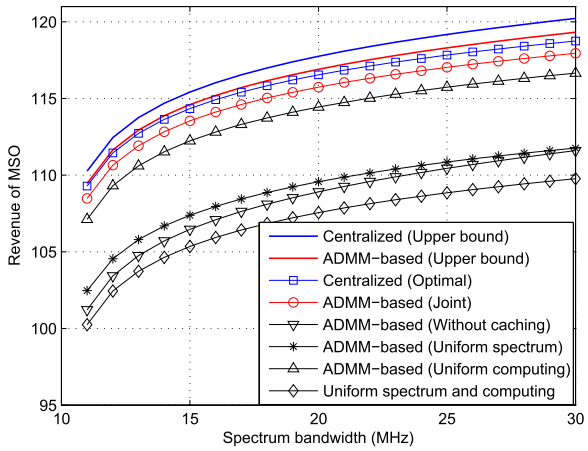


Fig. 9. MSO revenue versus the available bandwidth.

to each other during the computation tasks offloading process, so the algorithm automatically declined some of the offloading requests generated by the UEs. Besides, the computation resource of the MEC server cannot be reused at the same time, so if there are too many offloading UEs, the amount of computation resource assigned to each UE will decrease, which implies that the MSO will gain less from a single UE. The centralized algorithm and our proposed ADMM-based algorithm achieve relatively high revenue among all the six solutions. Again, because of the uniform allocation of spectrum and computation resources, the other three solutions in which uniform resource allocation strategies are adopted can just achieve much lower revenue. Similarly, due to lack of revenue from alleviated backhaul bandwidth, the revenue of ADMM solution without caching is also low.

In Fig. 9, the revenue of ADMM is compared with those of the centralized algorithm and the other four baseline solutions. In this figure, the number of small cells is set as 20. It can be seen from Fig. 9 that the revenue of our proposed ADMM algorithm is close to the revenue achieved by the centralized algorithm under various spectrum bandwidth conditions. The solution of ADMM with uniform computation resource allocation but optimal spectrum allocation can achieve relatively higher revenue compared with the other two uniform

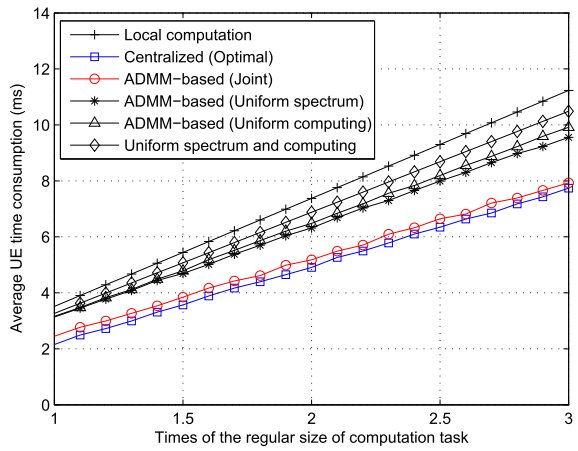


Fig. 10. Average UE time consumption versus the size of computation task.

solutions (ADMM with uniform spectrum allocation and uniform spectrum and computation resource allocation). This is mainly because of the fact that spectrum allocation usually plays a more important role in earning interest of MSO, and this in turn is due to the fact that unlike computation resource of MEC server, the spectrum resource could be reused simultaneously among UEs in different small cells.

Next we discuss the average UE time consumption in the system for executing computation tasks, and the execution time expression proposed in Section II-C.2 is employed here to present the results. Fig. 10 shows the average UE time consumption of ADMM based algorithm compared with that of the centralized algorithm, local computation solution and other three baseline solutions. The number of small cells here is 20. The y axis is the average time consumptions of all the UEs in system and the x axis is the size of computation tasks W_{k_n} (including Z_{k_n} and D_{k_n}), expressed in times of the regular size of computation task. Without losing generality, different UEs are considered to have different regular sizes of computation tasks, but the regular sizes are all around the values shown in Table II. The largest time consumption is achieved by local computation solution, in which all the UEs in system execute their computation tasks on local devices. Because of the shortage of computation resources, the local computation solution consumes much time, especially when the size of computation task is large. The benchmark solution in which both MEC computation and spectrum resources are uniformly allocated achieves less time consumption compared with local computation solution. This is due to the fact that the MEC server is more powerful than the UE devices. Because the resource allocation is not optimized, the advantage over local computation solution is not significant. ADMM with uniform computation resource allocation and ADMM with uniform spectrum resource allocation consume less time than the two solutions discussed above. The centralized algorithm consumes the least time among all the solutions. But as we can see, the gap between centralized algorithm and our proposed ADMM based algorithm is narrow.

VI. Conclusions and Future Work

In this paper, we presented an ADMM-based decentralized algorithm for computation offloading, resource allocation and

Internet content caching optimization in heterogeneous wireless cellular networks with mobile edge computing. We formulated the computation offloading decision, spectrum resource allocation, MEC computation resource allocation, and content caching issues as an optimization problem. Then, in order to tackle this problem in an efficient way, we presented an ADMM-based distributed solution, followed by a discussion about the feasibility and complexity of the algorithm. Finally, the performance evaluation of the proposed scheme was presented in comparison with the centralized solution and several baseline solutions. Simulation results demonstrated that the proposed scheme can achieve better performance than other baseline solutions under various system parameters. Future work is in progress to consider wireless network virtualization in the proposed framework.

REFERENCES

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[2] R. Xie, F. R. Yu, H. Ji, and Y. Li, "Energy-efficient resource allocation for heterogeneous cognitive radio networks with femtocells," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 3910–3920, Nov. 2012.

[3] S. Bu and F. R. Yu, "Green cognitive mobile networks with small cells for multimedia communications in the smart grid environment," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2115–2126, Jun. 2014.

[4] G. Huang and J. Li, "Interference mitigation for femtocell networks via adaptive frequency reuse," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2413–2423, Apr. 2016.

[5] A. R. Elsherif, W.-P. Chen, A. Ito, and Z. Ding, "Adaptive resource allocation for interference management in small cell networks," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2107–2125, Jun. 2015.

[6] Y. Cai, F. R. Yu, and S. Bu, "Cloud computing meets mobile wireless communications in next generation cellular networks," *IEEE Netw.*, vol. 28, no. 6, pp. 54–59, Nov. 2014.

[7] Z. Yin, F. R. Yu, S. Bu, and Z. Han, "Joint cloud and wireless networks operations in mobile cloud computing environments with telecom operator cloud," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 4020–4033, Jul. 2015.

[8] M. Zhanikeev, "A cloud visitation platform to facilitate cloud federation and fog computing," *Computer*, vol. 48, no. 5, pp. 80–83, May 2015.

[9] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, to be published.

[10] R. Gargees *et al.*, "Incident-supporting visual cloud computing utilizing software-defined networking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 182–197, Jan. 2017.

[11] M. Patel *et al.*, "Mobile-edge computing: Introductory technical white paper," ETSI, Sophia Antipolis, France, White Paper V1 18-09-14, Sep. 2014.

[12] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial Internet: A survey on the enabling technologies, applications, and challenges," *IEEE Commun. Surveys Tuts.*, to be published.

[13] O. Mäkinen, "Streaming at the edge: Local service concepts utilizing mobile edge computing," in *Proc. 9th IEEE Int. Conf. Next Generat. Mobile Appl. Services Technol.*, Cambridge, U.K., Sep. 2015, pp. 1–6.

[14] Y.-D. Lin, E. T.-H. Chu, Y.-C. Lai, and T.-J. Huang, "Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds," *IEEE Syst. J.*, vol. 9, no. 2, pp. 393–405, Jun. 2015.

[15] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[16] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[17] Y. He, F. R. Yu, N. Zhao, H. Yin, H. Yao, and R. C. Qiu, "Big data analytics in mobile cellular networks," *IEEE Access*, vol. 4, pp. 1985–1996, Mar. 2016.

[18] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation Offloading for Service Workflow in Mobile Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3317–3329, Dec. 2015.

[19] FP7. (2015). *Pursuit Project*. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[20] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart. 2015.

[21] K. Wang, H. Li, F. R. Yu, and W. Wei, "Virtual resource allocation in software-defined information-centric cellular networks with device-to-device communications and imperfect CSI," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10011–10021, Dec. 2016.

[22] C. Liang, F. Yu, and X. Zhang, "Information-centric network function virtualization over 5g mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, Jun. 2015.

[23] Z. Li, Q. Wu, K. Salamatian, and G. Xie, "Video delivery performance of a large-scale vod system and the implications on content delivery," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 880–892, Jun. 2015.

[24] W. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758–770, Apr. 2013.

[25] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal cache allocation for content-centric networking," in *Proc. IEEE ICNP*, Oct. 2013, pp. 1–10.

[26] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *Proc. IEEE INFOCOMM WKSHPS*, Mar. 2012, pp. 316–321.

[27] Z. Chen and D. Wu, "Rate-distortion optimized cross-layer rate control in wireless video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 352–365, Mar. 2012.

[28] A. H. Jafari, D. López-Pérez, H. Song, H. Claussen, L. Ho, and J. Zhang, "Small cell backhaul: Challenges and prospective solutions," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 206, Dec. 2015.

[29] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, Mar. 2013.

[30] G. Iosifidis, L. Gao, and L. Tassiulas, "An iterative double auction mechanism for mobile data offloading," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, Oct. 2015.

[31] L. Ma, F. Yu, V. C. M. Leung, and T. Randhawa, "A new method to support UMTS/WLAN vertical handover using SCTP," *IEEE Wireless Commun.*, vol. 11, no. 4, pp. 44–51, Aug. 2004.

[32] L. Ma, F. R. Yu, and V. C. M. Leung, "Performance improvements of mobile SCTP in integrated heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3567–3577, Oct. 2007.

[33] F. Yu and V. Krishnamurthy, "Optimal joint session admission control in integrated WLAN and CDMA cellular networks with vertical handoff," *IEEE Trans. Mobile Comput.*, vol. 6, no. 1, pp. 126–139, Jan. 2007.

[34] S. Bu, F. R. Yu, Y. Cai, and X. P. Liu, "When the smart grid meets energy-efficient communications: Green wireless cellular networks powered by the smart grid," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 3014–3024, Aug. 2012.

[35] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. EuroSys*, Salzburg, Austria, Apr. 2011, pp. 301–314.

[36] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[37] M. Fiore, F. Mininni, C. Casetti, and C.-F. Chiasserini, "To cache or not to cache?" in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 235–243.

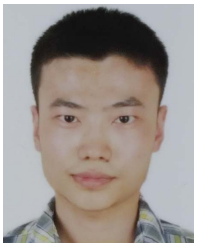
[38] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Mar. 2012, pp. 310–315.

[39] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1107–1115.

[40] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "User association and load balancing for cellular massive MIMO," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2014, pp. 1–10.

[41] D. Fooladivanda and C. Rosenberg, "Joint resource allocation and user association for heterogeneous wireless cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 1, pp. 248–257, Jan. 2013.

[42] S. Gortzen and A. Schmeink, "Optimality of dual methods for discrete multiuser multicarrier resource allocation problems," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3810–3817, Oct. 2012.
[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge univ. Press, 2009.
[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[45] L. Chen, F. R. Yu, H. Ji, G. Liu, and V. C. M. Leung, "Distributed virtual resource allocation in small-cell networks with full-duplex self-backhauls and virtualization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5410–5423, Aug. 2016.
[46] W.-C. Liao, M. Hong, H. Farmanbar, X. Li, Z.-Q. Luo, and H. Zhang, "Min flow rate maximization for software defined radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1282–1294, Jun. 2014.
[47] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed joint resource and routing optimization in wireless sensor networks via alternating direction method of multipliers," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5454–5467, Nov. 2013.
[48] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2015.



Chenmeng Wang received the M.S. degree in information and telecommunication engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Communication and Information Engineering. From 2015 to 2017, he is a visiting student with Carleton University, Ottawa, Canada. His current research interests include 5G cellular network, interference management, mobile edge computing system, and small cell networks.

Chengchao Liang received the B.Eng. and M.Eng. degrees in communication and information systems from the Chongqing University of Posts and Telecommunications, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. From 2011 to 2012, he was a visiting student with the Mobile Telecommunications Research Laboratory, Inha University, Incheon, South Korea. His current research interests include software-defined networking, network virtualization, resource allocation, and applications of convex optimization in mobile networks.

F. Richard Yu (S'00–M'04–SM'08) received the Ph.D. degree in electrical engineering from The University of British Columbia in 2003. From 2002 to 2006, he was with Ericsson, Lund, Sweden, and a start-up in CA, USA. He joined Carleton University in 2007, where he is currently a Professor. His current research interests include cross-layer/cross-system design, connected vehicles, security, and green ICT. He is a Distinguished Lecturer and a member of the Board of Governors of the IEEE Vehicular Technology Society. He has served as the technical program committee co-chair of numerous conferences. He is a registered professional engineer in the province of Ontario, Canada, and a fellow of the Institution of Engineering and Technology.

He received the IEEE Outstanding Service Award in 2016, the IEEE Outstanding Leadership Award in 2013, the Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from the Canada Foundation of Innovation in 2009, and the Best Paper Awards at the IEEE ICC 2014, the Globecom 2012, the IEEE/IFIP TrustCom 2009, and the International Conference on Networking 2005. He serves on the editorial boards of several journals, including as Co-Editor-in-Chief of the *Ad Hoc & Sensor Wireless Networks*, and the Lead Series Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and the IEEE COMMUNICATIONS SURVEYS & TUTORIALS.

Qianbin Chen (M'03–SM'14) received the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2002. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, and the Director of the Chongqing Key Lab of Mobile Communication Technology. He has authored or co-authored over 100 papers in journals and peer-reviewed conference proceedings, and has co-authored seven books. He holds 47 granted national patents.

Lun Tang received the Ph.D. degree in communication and information system from Chongqing University, Chongqing, China. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications. His current research interests include 5G cellular network, interference management, and small cell networks.