
Article:
Du, J., Zhao, L., Feng, J. et al. (1 more author) (2017) Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee. IEEE Transactions on Communications. ISSN 0090-6778

https://doi.org/10.1109/TCOMM.2017.2787700

Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee

Jianbo Du, Liqiang Zhao, Jie Feng, and Xiaoli Chu

***Abstract*—Cooperation between the fog and the cloud in mobile cloud computing environments could offer improved offloading services to smart mobile user equipment (UE) with computation intensive tasks. In this paper, we tackle the computation offloading problem in a mixed fog/cloud system by jointly optimizing the offloading decisions and the allocation of computation resource, transmit power and radio bandwidth, while guaranteeing user fairness and maximum tolerable delay. This optimization problem is formulated to minimize the maximal weighted cost of delay and energy consumption (EC) among all UEs, which is a mixed-integer non-linear programming problem. Due to the NP-hardness of the problem, we propose a low-complexity suboptimal algorithm to solve it, where the offloading decisions are obtained via semidefinite relaxation and randomization and the resource allocation is obtained using fractional programming theory and Lagrangian dual decomposition. Simulation results are presented to verify the convergence performance of our proposed algorithms and their achieved fairness among UEs, and the performance gains in terms of delay, EC and the number of beneficial UEs over existing algorithms.**

***Index Terms*—Computation offloading, cloud computing, fog computing, resource allocation, min-max fairness.**

I. Introduction

With smart mobile user equipments (UEs) gains enormous popularity, people expect to run more and more computation-intensive mobile applications [1]. Those applications usually consume huge amounts of energy and demand powerful computation capacity, and have rigorous delay constraints. However, UEs are usually resource-constrained, possessing limited computation capability and battery, which makes it impractical to run sophisticated applications on them [2]. Mobile Cloud Computing (MCC) [3] has been considered as a promising way to address the above challenges by offloading those applications to powerful cloud centers. However, the delay caused by transferring data to the remote cloud server is usually unacceptable for some latency-sensitive applications.

Mobile edge computing [3] (or fog computing [4]) has been proposed as a supplement to MCC for further energy saving and delay reduction in recent years [5]. In fog computing, access points (APs) and UEs with certain processing capabilities serve as fog nodes [4], [6], [7], and each UE is associated to a cloud clone in the fog node or cloud center, where a virtual machine (VM) executes mobile applications for the UE [8], [9]. The slight difference between Fog computing and Mobile edge computing is that Fog computing can be expanded to the core network [3], [4], however, similar to most works, we do not distinguish the two concepts in this paper.

Many works have been proposed to investigate the issues involved in computation offloading. Through the optimization of offloading decisions and the involved resource allocation, such as the allocation of transmit power, bandwidth, and computation resource, to obtain system performance gains, e.g., reduction in delay or energy consumption (EC), or an improvement in energy efficiency, etc. However, most of those previous works put their emphasis either on offloading decision making [2], [10], or resource allocation [6], without a joint consideration of both. The works in [6] and [11] focused on system-level performance improvement, without considering the performance of individual UEs, where UEs with good channel conditions (e.g., high channel gains, low interference, or both) will benefit from computation offloading, but at the cost of degraded performance of UEs under bad channel conditions, resulting in unfairness among UEs.

Different from the above approaches, in this paper, we study the joint optimization of offloading decision making, computation resource allocation, transmit power assignment, and radio bandwidth allocation for a mixed cloud/fog computing system to minimize the system cost, i.e., a weighted sum of delay and energy consumption, with the maximum tolerable delay guaranteed. To ensure the fairness of all the UEs, we minimize the maximum cost among all the UEs. We formulate the joint optimization as a mixed integer non-linear programming (MINLP) problem and propose a suboptimal algorithm with low complexity to solve it. The main contributions of this work are summarized as follows.

- Different from [6], [12] and [13], a fairness-aware cost minimization problem is formulated to minimize the maximum cost among all UEs.
- We devise a low complexity algorithm called **c**omputation **o**ffloading and **r**esource **a**llocation algorithm (CORA) to solve the formulated NP-hard optimization problem. It is first transformed into a non-convex quadratically constrained quadratic programming (QCQP) problem;

*This work was supported in part by National Natural Science Foundation of China (61771358), Intergovernmental International Cooperation on Science and Technology Innovation (2016YFE0122900), and the 111 Project (B08038).

J. Du, L. Zhao, J. Feng are with State Key Laboratory of ISN, Xidian University, No.2 Taibainan-lu, Xi'an, 710071, Shaanxi, China. (Email: dujianboo@163.com; lqzhao@mail.xidian.edu.cn; 784852087@qq.com).

X. Chu is with Department of Electronic and Electrical Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK. (Email: x.chu@sheffield.ac.uk).

and through semidefinite relaxation (SDR) and randomization, offloading decisions are obtained; then by using the **b**isection method for **c**omputation **r**esource **a**llocation (BCRA), we propose Algorithm 2 to optimize the computation resource allocation.
- We propose a fractional programming based Algorithm 3 to transform the non-convex radio resource allocation into a convex programming problem, which is solved by Lagrange dual decomposition based Algorithm 4 (which is nested in Algorithm 3) and transmit power and bandwidth allocation is obtained.
- We perform abundant simulation results to evaluate the convergence of the iterative algorithms 2, 3 and 4, the fairness of CORA, and the performance gain of CORA by comparing it with the existing prevalent algorithms in computation offloading.

The remainder of this paper is organized as follows. Related works are presented in Section II. Section III introduces the system model and problem formulation. Section IV presents the CORA algorithm with focus on the SDR based offloading decision making algorithm. The iterative algorithms for computation resources allocation is detailed in Section V. In Section VI, we present the bandwidth and power allocation algorithm. Complexity analysis of CORA algorithm is presented in Section VII. Simulation results are provided in Section VIII. Finally, the paper is concluded in Section IX.

Notation: Lower case boldface letters denote vectors, while upper case boldface letters denote matrices. For a certain matrix $\mathbf{X}$, $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is a positive semidefinite matrix, while $\mathrm{Tr}(\mathbf{X})$ and $\mathrm{rank}(\mathbf{X})$ denote the trace and the rank of $\mathbf{X}$, respectively. For a vector $\mathbf{x}$ or a matrix $\mathbf{X}$, $\mathbf{x}^T$ or $\mathbf{X}^T$ represents the transpose of them. We use $\mathbf{e}_n$ to denote an $N \times 1$ unit vector with the n^{th} entry being 1, and $\mathrm{diag}(\mathbf{e}_n)$ stands for an $N \times N$ diagonal matrix with its main diagonal elements from $\mathbf{e}_n$.

II. Related Works

Application offloading has been a hot topic owning to the appearance of cloud computing and fog computing. Each application can be offloaded in coarse-grained application level [6], [12], fine-grained task level [2], [13]–[16], or a percent of [17], [18], and can be offloaded to the cloud or fog, where the decision could be made in a centralized (most current works employ this manner) or decentralized manner [10], [19], for single UE or multiple UEs.

In single-UE case, task partitioning and assignment is usually considered where each task should be determined whether to offload or not according to some criteria [2], [14]–[16], [20]. In [17], [18] the authors considered a special kind of data-partitioned-oriented-application and partial of the application is offloaded, together with transmit power optimization, to minimize the EC of the UE. The tradeoff between the EC for local computation and for remote communications was discussed in [20], [21].

In multi-UE scenario, the computation resources of the fog node, and communication resources between UEs and the fog node (e.g., bandwidth, and power) are shared, which should be allocated elaborately for a better performance. The authors in [22] studied task-level offloading in a multi-UE multi-fog scenario where offloading decision (whether to offload or not) was optimized for each task of each UE. In [10] and [19], game theory was utilized to optimized offloading decisions in a multi-UE cloud computing environment. In [12], the authors intended to minimize the system EC by offloading applications into the cloud. Note that radio resources were not optimized in [10], [12], [19], [22]. The authors in [6] studied application offloading in a multi-UE multi-cell Multiple-Input Multiple-Output (MIMO) system by transmit power optimization under given offloading decisions, in order to minimize the total EC of all UEs. The authors in [13] intended to maximize the weighted performance improvement in time saving and energy reduction of the system, by jointly optimizing offloading decisions, the allocation of local clock frequency and transmit power. A novel three-tier optimization architecture, including the UE tier, the cloudlet tier, and the cloud tier, was proposed in [19], where the authors proposed to minimize the queue-arrival-rate weighted mean task response time of each UE by offloading strategy optimization employing game theory. However, none of the aforementioned works [4], [6], [10], [12], [13], [19], [22] have jointly considered offloading decision making, computation resource allocation and uplink communication resource assignment for multi-UE mixed fog and cloud radio access networks.

Resource allocation and offloading decision making were jointly optimized in [11] so as to conserve energy while satisfying UE delay constraints. However, the EC of each UE was set as a constant for simplicity, ignoring its time varying aspect. Besides, for resource allocation, it was only mentioned as "solve the corresponding resource allocation problem", but no detail was given. Furthermore, there was no radio channel model, and the radio resource was allocated in units of bit/s, which was oversimplified, since in actual networks, radio resource is usually in terms of resource blocks, bandwidth, and/or transmit power. Moreover, the fairness between UEs was not taken into consideration [6], [11]–[13].

III. System Model and Problem Formulation

A. Description of the Concerned Scenario

As shown in Fig. 1 [19], we consider a network consisting of N UEs, one WiFi AP as the fog node, and a remote cloud server. Denote the set of UEs as $\mathcal{N}$. Each UE is connected to the fog node via a wireless link, while the fog node and the cloud server are connected via a fiber wired link. Each UE has one application to be either handled locally or offloaded for remote processing through the following procedure. Firstly, each UE sends an offloading request (including the information of the UE (e.g., its local processing capability, power), the properties of the application (e.g., the maximum tolerable delay), etc. [13]) to the decision maker (DM) in the fog node [17]. According to the collected offloading requests of all UEs and the instantaneous wireless channel gains, the DM performs optimization to decide where should the applications

be processed, i.e., in the UE locally, in the fog, or in the cloud, and finally, the offloading decision is delivered to the corresponding UE. Since the offloading requests are usually very small, we assume that no buffer is needed for queueing the computation requests, as in [9]. To enable tractable analysis, we assume that the DM decides the offloading strategy for all the UE requests that have been received at the beginning of the offloading period, i.e., the decision making delay due to request queuing and decision making is omitted [23].

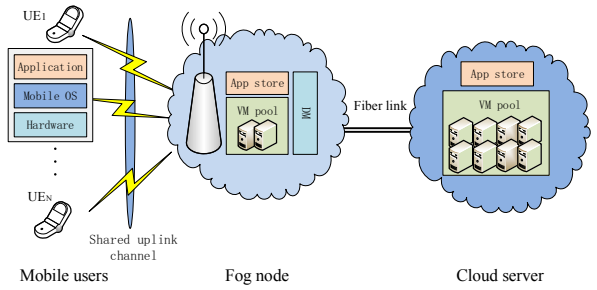


Fig. 1. System topology [19].

The offloading decisions of UE$_n$ are constrained by

$$x_n + y_n + z_n = 1, \quad \forall n \in \mathcal{N}, \tag{1}$$

where $x_n = 1,\ y_n = 1, z_n = 1$ indicate that the application is processed by UE itself, by the fog, and by the cloud, respectively; otherwise, $x_n = 0, y_n = 0, z_n = 0$. The constraint in (1) implies that among $x_n,\ y_n,$ and z_n, one and only one of them is 1 at any time.

Similar to the existing works [6], [9], [10], [13], [20], [22], [23], to enable tractable analysis, we consider a quasi-static scenario where all UEs and the wireless network remain stationary during an offloading period (usually within several seconds [10]). This assumption holds for many applications, e.g., natural language processing, and face recognition, where the input data size is not large so that the application offloading could be completed during a time shorter than the timescales of UE mobility and the dynamics of wireless networks. We perform the joint optimization of offloading decision and resource allocation at each new request from a UE within an offloading period independently.

For fog processing, the fog node needs to allocate the limited computation resources (in CPU cycles/s) to the application of corresponding UEs. Denote the set of fog-processing UEs by $\mathcal{N}_1$, and $N_1 = |\mathcal{N}_1|$ is the number of UEs in $\mathcal{N}_1$. For remote processing in the cloud, the applications need to be transmitted from the UE to the fog node through the shared wireless links, and then forwarded by the fog node to the cloud through a wired link. Since the cloud has plenty of computation resources and the wired link between the fog node and the cloud server is of a sufficiently large capacity, the allocation of these resources will not be discussed. However, the limited radio bandwidth needs to be allocated among all the fog-processing and cloud-executing UEs for communicating with the fog node. All the fog-processing and cloud-executing UEs are referred to as remote-processing UEs, collected in the set $\mathcal{N}_2$, and $N_2 = |\mathcal{N}_2|$ is the number of remote-processing UEs. Assume the total radio bandwidth is B Hz. We allocate to each remote-processing UE a portion of the total bandwidth orthogonally to avoid interference between them [24], [25].[1] Denote the normalized assigned portion of bandwidth to UE$_n$ as a_n, we have $a_n \in [0, 1]$ and $\sum_{n\in\mathcal{N}_2} a_n \le 1$.

The application of UE n is described by $J_n = \{D_n, App_n, \tau_n^{max}\},\ n \in \mathcal{N}$, where D_n denotes the size of input data (in bits), τ_n^{max} is the tolerable maximum latency (in second), and App_n is the *processing density* (in CPU cycles/bit), which depends on the computational complexity of the application [1], [17]. We model the size C_n of calculation amount, i.e., the number of CPU cycles necessary to accomplish the application, as $C_n = D_n App_n$ [17], [18]. In the following, we assume that the D_n, C_n and App_n are known, which can be obtained by employing *program profilers* as in [2], [13], [22]. We assume there is a clone for each UE in the fog node, so the application code of J_n with size C_n is backed up in the fog node [1], [20], and can be downloaded by the cloud server through a high-speed wired link [2], [6]. Therefore, only the data of D_n bits need to be transmitted from the UE to cloud server when offloading. It should be noted that $D_n,\ C_n,\ App_n$ and τ_n^{max} are inherent parameters of the application of UE$_n$, and they will not change with where the application is processed.

In the following we will discuss the EC and the delay caused by local processing, fog executing and cloud computing, respectively. Since the output after processing is usually small, only the uplink communication is considered hereafter for simplicity of analysis [9], [10], [13], [20].

B. Cost Under Different Scenarios

1) Local Processing: Let f_n^{loc} and p_n^{loc} be the local computation capability (in CPU cycles/s) and the local executing power consumption (in watt) of UE$_n$, respectively. The delay and EC of processing application J_n locally are [6], [13], [16]

$$T_n^{loc} = C_n / f_n^{loc}, \tag{2}$$

$$E_n^{loc} = p_n^{loc}(C_n / f_n^{loc}). \tag{3}$$

2) Fog Computing: For analytical tractability, we assume that the fog processing for an application starts only after all the input data has been received by the fog node. More specifically, if the application J_n is to be processed in the fog, UE$_n$ needs to transmit the input data D_n to the fog through the shared wireless links. After all the input data D_n is received, application J_n is executed by the fog node. Denote the channel gain between UE$_n$ and the fog node as h_n, then the achievable transmit rate of UE$_n$ is

$$r_n = a_n B \log_2\left(1 + \frac{p_n^{com} h_n}{a_n N_0 B}\right), \tag{4}$$

where p_n^{com} is the transmit power of UE$_n$, which is restricted by the maximum value p_n^{max}.

[1] When the total allocated bandwidth is less than the system bandwidth, the frequency band of each UE does not overlap and can be accessed orthogonally.

Denote the power consumption (in watt) of UE$_n$ in idle state as p_n^{id}, then the delay and EC of fog processing are given respectively by [6]

$$T_n^{fog} = D_n/r_n + C_n/f_n^{fog}, \tag{5}$$

$$E_n^{fog} = p_n^{com}(D_n/r_n) + p_n^{id}(C_n/f_n^{fog}), \tag{6}$$

where f_n^{fog} (in CPU cycles/s) denotes the computation resources allocated to UE$_n$.

3) Cloud Computing: If application J_n is offloaded to the cloud server, then UE$_n$ first transmits the data of size D_n through a wireless link to the fog node, which then forwards J_n to the cloud server through a high-speed wired link. We denote the rate of the wired link allocated to UE$_n$ as R_n^{fc} (in bit/s), and the cloud processing capability assigned to UE$_n$ as f_n^c (in CPU cycles/s). The delays in wired transmission and cloud processing are given by $T_n^{fc} = D_n/R_n^{fc}$ and $T_n^c = C_n/f_n^c$, respectively. The total delay and total EC of cloud processing for UE$_n$ are given respectively by

$$T_n^{cloud} = D_n/r_n + T_n^{fc} + T_n^c, \tag{7}$$

$$E_n^{cloud} = p_n^{com}(D_n/r_n) + p_n^{id}(T_n^{fc} + T_n^c). \tag{8}$$

According to (2)-(8), the EC and delay of UE$_n$ can be expressed respectively as

$$E_n = E_n^{loc}x_n + E_n^{fog}y_n + E_n^{cloud}z_n, \tag{9}$$

$$T_n = T_n^{loc}x_n + E_n^{fog}y_n + E_n^{cloud}z_n. \tag{10}$$

All the notations used are listed in Table I.

TABLE I
NOTATION DEFINITIONS

Symbol	Definition
E_n^{loc}, E_n^{fog}, E_n^{cloud}	EC for UE$_n$ in local/fog/cloud processing
T_n^{loc}, T_n^{fog}, T_n^{cloud}	Delay for UE$_n$ in local/fog/ cloud processing
f_n^{loc}, f_n^{fog}, f_n^c	Processing ability of UE$_n$ of local/fog/cloud processing
p_n^{loc}, p_n^{id}, p_n^{com}	Power of UE$_n$ in local processing/idle/transmit
D_n, C_n, App_n	Data size/size of calculation amount/ processing density of J_n
τ_n^{max}	The maximum processing delay of J_n
r_n	Transmit rate of UE$_n$
a_n	Normalized allocated bandwidth to UE$_n$
F^{fog}	Total computation capability of the fog
T_n^{fc}, T_n^c	Wired transmit/cloud-processing delay of UE$_n$
R_n^{fc}	Rate of UE$_n$ in wired link
x_n, y_n, z_n	Offloading decisions of UE$_n$
$\boldsymbol{\pi}$	Set of offloading decision of all UEs
$\mathcal{N}, N$	Set/number of UEs
$\mathcal{N}_1, N_1$	Set/number of fog processing UEs
$\mathcal{N}_2, N_2$	Set/number of remote processing UEs
L	Number of runs (i.e., randomization trails)
B	Total radio bandwidth between UEs and the fog
N_0	Additive noisy power spectral density

C. Problem Formulation

In this section, we formulate the problem of jointly optimizing offloading decision making and resource allocation for a mixed fog/cloud computing system and show that it is NP-hard.

The cost of UE$_n$ is defined as the weighted sum of EC and latency as $Cost_n = \lambda_n^e E_n + \lambda_n^t T_n$ where $\lambda_n^e, \lambda_n^t \in [0,1]$, $n \in N$ denote the weights of EC and delay for UE$_n$, respectively. We propose to minimize the maximum cost among all UEs while meeting the maximum delay constraints. We formulate the joint optimization of the offloading decisions $\boldsymbol{\pi} = [\mathbf{x}, \mathbf{y}, \mathbf{z}] = [x_1, y_1, z_1, ..., x_N, y_N, z_N]$, the power and bandwidth assignment $\mathbf{p}^{com} = [p_1^{com}, ..., p_N^{com}]$ and $\mathbf{a} = [a_1, ..., a_N]$ (for local processing UE$_n$, we let $p_n^{com} = 0,\ a_n = 0$), and the computation resource allocation $\mathbf{f}^{fog} = [f_1^{fog}, ..., f_N^{fog}]$ (for non-fog processing UE$_n$, let $f_n^{fog} = 0$) as follows

$$\begin{aligned}
(\mathcal{P}_\mathbf{1}): \min_{\boldsymbol{\pi}, \mathbf{f}^{fog}, \mathbf{p}^{com}, \mathbf{a}} \ & \max_{n \in \mathcal{N}} \ Cost_n \\
\text{s.t. } & (\text{C1}): x_n, y_n, z_n \in \{0,1\},\ \forall n \in \mathcal{N}, \\
& (\text{C2}):\ x_n + y_n + z_n = 1,\ \forall n \in \mathcal{N}, \\
& (\text{C3}):\ \sum_{n \in \mathcal{N}} f_n^{fog} \le F^{fog}, \\
& (\text{C4}):\ f_n^{fog} \ge 0,\ \forall n \in \mathcal{N}, \\
& (\text{C5}):\ 0 < a_n \le 1,\ \forall n \in \mathcal{N}, \\
& (\text{C6}):\ \sum_{n \in \mathcal{N}} a_n \le 1, \\
& (\text{C7}):\ 0 \le p_n^{com} \le p_n^{max}, \forall n \in \mathcal{N}, \\
& (\text{C8}):\ T_n \le \tau_n^{max}, \forall n \in \mathcal{N},
\end{aligned} \tag{11}$$

where F^{fog} is the total computation capacity of the fog node; (C1) and (C2) are the constraints on the offloading decision of each UE; (C3) indicates that the allocated computation resources cannot exceed the total computation capability of the fog node; (C4) is the non-negative constraint on computation resource allocation; (C5) and (C6) are the constraints on bandwidth allocation; (C7) is the transmit power constraint of each UE; and (C8) indicates each application should be performed before a tolerable deadline. Note that $(\mathcal{P}_\mathbf{1})$ minimizes the maximum cost among all UEs. Therefore, it guarantees fairness among UEs from the perspective of system cost.

Remark 1. Problem $(\mathcal{P}_\mathbf{1})$ is not convex due to: 1) the min-max formulation; and 2) the binary variables $\boldsymbol{\pi}$. It is a mixed-integer non-linear programming problem, which can be generally NP-hard [26].

IV. OFFLOADING DECISION MAKING

A. Equivalent Transformation into a QCQP Problem

In the following, to reduce the computational complexity, we transform $(\mathcal{P}_\mathbf{1})$ into a QCQP problem, which is then converted into a standard convex problem via semidefinite relaxation. The converted problem can be solved using convex optimization toolbox CVX [27].

Firstly, we introduce a slack variable ζ, and let $\max_{n\in\mathcal{N}} Cost_n = \zeta$. By merging items containing x_n, y_n, z_n and $y_n + z_n$, respectively, we have

$$\begin{aligned}&(\lambda_n^e p_n^{loc} + \lambda_n^t)\frac{C_n}{f_n^{loc}}x_n + (\lambda_n^e p_n^{id} + \lambda_n^t)(T_n^{fc} + T_n^c)z_n \\ &+ (\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}}y_n + (\lambda_n^e p_n^{com} + \lambda_n^t)\frac{D_n}{r_n}(y_n + z_n) \\ &\le \zeta, \end{aligned} \tag{12}$$

where $(\lambda_n^e p_n^{loc} + \lambda_n^t)\frac{C_n}{f_n^{loc}}$ and $(\lambda_n^e p_n^{id} + \lambda_n^t)(T_n^{fc} + T_n^c)$ are constants. Letting $\max_{n\in\mathcal{N}}\{(\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}}y_n\} = D_n^{fog}$ and $\max_{n\in\mathcal{N}}\{(\lambda_n^e p_n^{com} + \lambda_n^t)\frac{D_n}{r_n}(y_n + z_n)\} = D_n^{com}$, we have

$$\begin{aligned}C_n(\lambda_n^e p_n^{id} + \lambda_n^t)y_n \le D_n^{fog} f_n^{fog}, \\ D_n(\lambda_n^e p_n^{com} + \lambda_n^t)(y_n + z_n) \le D_n^{com} r_n.\end{aligned} \tag{13}$$

Based on the above definitions, $(\mathcal{P}_\mathbf{1})$ is transformed as

$$\begin{aligned}(\mathcal{P}_\mathbf{2}): &\min_{\boldsymbol{\pi},\mathbf{f}^{fog},\mathbf{p}^{com},\mathbf{a},\mathbf{d},\zeta} \zeta \\ \text{s.t. } &(\text{C1}_1): x_n(x_n-1)=0,\ y_n(y_n-1)=0, \\ &\qquad z_n(z_n-1)=0, \quad \forall n\in\mathcal{N}, \\ &(\text{C2})-(\text{C8}), \\ &(\text{C9}): Cost_n \le \zeta,\ \forall n\in\mathcal{N}, \\ &(\text{C10}): C_n(\lambda_n^e p_n^{id} + \lambda_n^t)y_n \le D_n^{fog} f_n^{fog},\ \forall n\in\mathcal{N}, \\ &(\text{C11}): D_n(\lambda_n^e p_n^{com} + \lambda_n^t)(y_n+z_n) \le D_n^{com} r_n, \\ &\qquad\qquad \forall n\in\mathcal{N},\end{aligned} \tag{14}$$

where $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N]$, $\mathbf{d_n} = \left[D_n^{fog}, D_n^{com}\right]$, and the integer constraint (C1) is replaced with (C1_1). However, problem $(\mathcal{P}_\mathbf{2})$ is still nonconvex due to the quadratic terms in (C1_1). Next, we transform $(\mathcal{P}_\mathbf{2})$ into an equivalent standard QCQP problem. To enable tractable analysis, the constraints with respect to resource allocation including (C5)–(C8) are not considered temporarily in offloading decision making for simplicity.

We first define the following $(7N+1)\times 1$ vector as follows

$$\begin{aligned}\mathbf{s} = \Big[&x_1, y_1, z_1, ..., x_N, y_N, z_N, f_1^{fog}, ..., f_N^{fog}, \\ &D_1^{fog}, ..., D_N^{fog}, r_1, ..., r_N, D_1^{com}, ..., D_N^{com}, \zeta\Big]^T.\end{aligned}$$

Then problem $(\mathcal{P}_\mathbf{2})$ is transformed into the following standard QCQP problem

$$\begin{aligned}(\mathcal{P}_\mathbf{3}): &\min_{\mathbf{s}} (\mathbf{u}_0)^T\mathbf{s} \\ \text{s.t. } &(\text{C1}'): \mathbf{s}^T diag(\mathbf{e}_p')\mathbf{s} - (\mathbf{e}_p')^T\mathbf{s} = 0,\ p=1,...,3N, \\ &(\text{C2}'): (\mathbf{u}_n^I)^T\mathbf{s} = 1, \quad \forall n\in\mathcal{N}, \\ &(\text{C3}'): (\mathbf{u}^{fog})^T\mathbf{s} \le F^{fog}, \\ &(\text{C4}'): (\mathbf{u}_n^f)^T\mathbf{s} \ge 0, \quad \forall n\in\mathcal{N}, \\ &(\text{C9}'): (\mathbf{u}_n^c)^T\mathbf{s} \le 0,\ \forall n\in\mathcal{N}, \\ &(\text{C10}'): \mathbf{s}^T\mathbf{Q}_n^{fog}\mathbf{s} + (\mathbf{u}_n^{fog})^T\mathbf{s} \le 0, \quad \forall n\in\mathcal{N}, \\ &(\text{C11}'): \mathbf{s}^T\mathbf{Q}_n^{com}\mathbf{s} + (\mathbf{u}_n^{com})^T\mathbf{s} \le 0,\ \forall n\in\mathcal{N},\end{aligned} \tag{15}$$

where $\mathbf{e}_i$ and $\mathbf{e}_i'$ are standard unit vectors with size of $N\times 1$ and $(7N+1)\times 1$, respectively, and

$$\begin{aligned}&\mathbf{u}_0 = [\mathbf{0}_{1\times 7N} \ \ 1]^T, \qquad \mathbf{u}_n^I = \mathbf{e}_{3n-2}' + \mathbf{e}_{3n-1}' + \mathbf{e}_{3n}', \\ &\mathbf{u}^{fog} = [\mathbf{0}_{1\times 3N} \ \ \mathbf{1}_{1\times N} \ \ \mathbf{0}_{1\times(3N+1)}]^T, \\ &\mathbf{u}_n^{fog} = C_n(\lambda_n^e p_n^{id} + \lambda_n^t)\mathbf{e}_{3n-1}', \qquad \mathbf{u}_n^f = \mathbf{e}_{3N+n}', \\ &\mathbf{u}_n^{com} = D_n(\lambda_n^e p_n^{com} + \lambda_n^t)(\mathbf{e}_{3n-1}' + \mathbf{e}_{3n}'), \\ &\mathbf{u}_n^c = (\lambda_n^e p_n^{loc} + \lambda_n^t)\frac{C_n}{f_n^{loc}}\mathbf{e}_{3n-2}' + (\lambda_n^e p_n^{id} + \lambda_n^t)(T_n^{fc} + T_n^c)\mathbf{e}_{3n}' \\ &\qquad + \mathbf{e}_{4N+n}' + \mathbf{e}_{6N+n}' - \mathbf{e}_{7N+1}', \\ &\mathbf{Q}_n^{fog} = \begin{bmatrix} \mathbf{0}_{3N\times 3N} & \mathbf{0}_{3N\times 2N} & \mathbf{0}_{3N\times(2N+1)} \\ \mathbf{0}_{2N\times 3N} & \mathbf{Q}_{n1}^{fog} & \mathbf{0}_{2N\times(2N+1)} \\ \mathbf{0}_{(2N+1)\times 3N} & \mathbf{0}_{(2N+1)\times 2N} & \mathbf{0}_{(2N+1)\times(2N+1)} \end{bmatrix}, \\ &\mathbf{Q}_n^{com} = \begin{bmatrix} \mathbf{0}_{5N\times 5N} & \mathbf{0}_{5N\times 2N} & \mathbf{0}_{5N\times 1} \\ \mathbf{0}_{2N\times 5N} & \mathbf{Q}_{n1}^{com} & \mathbf{0}_{2N\times 1} \\ \mathbf{0}_{1\times 5N} & \mathbf{0}_{1\times 2N} & \mathbf{0}_{1\times 1} \end{bmatrix}, \\ &\mathbf{Q}_{n1}^{com} = \mathbf{Q}_{n1}^{fog} = -\frac{1}{2}\begin{bmatrix} \mathbf{0}_{N\times N} & \text{diag}\,(\mathbf{e}_n) \\ \text{diag}\,(\mathbf{e}_n) & \mathbf{0}_{N\times N} \end{bmatrix}.\end{aligned}$$

However, the QCQP problem $(\mathcal{P}_\mathbf{3})$ is still nonconvex and is hard to solve.

B. Semidefinite Relaxation

SDR is an efficient way to simplify QCQP problems [28]. In $(\mathcal{P}_\mathbf{3})$, all the matrices are real symmetric, and all the vectors are real, satisfying the conditions for SDR. In order to apply SDR to $(\mathcal{P}_\mathbf{3})$, we define

$$\begin{aligned}\mathbf{w} &= [\mathbf{s}_{(7N+1)\times 1} \ \ 1_{1\times 1}]^T, \\ \mathbf{W} &= [\mathbf{w}\mathbf{w}^T]_{(7N+2)\times(7N+2)}.\end{aligned} \tag{16}$$

Notice $\mathbf{W}$ is a rank one symmetric positive semidefinite matrix, then we obtain the equivalent version of $(\mathcal{P}_\mathbf{3})$ as follows

$$\begin{aligned}(\mathcal{P}_\mathbf{4}): &\min_{\mathbf{W}} \text{Tr}(\mathbf{M}_0\mathbf{W}) \\ \text{s.t. } &(\text{C1}''): \text{Tr}(\mathbf{M}_p\mathbf{W}) = 0,\ p=1,...,3N, \\ &(\text{C2}''): \text{Tr}(\mathbf{M}_n^I\mathbf{W}) = 1, \quad \forall n\in\mathcal{N}, \\ &(\text{C3}''): \text{Tr}(\mathbf{M}^{fog}\mathbf{W}) \le F^{fog}, \\ &(\text{C4}''): \text{Tr}(\mathbf{M}_n^f\mathbf{W}) \ge 0, \quad \forall n\in\mathcal{N}, \\ &(\text{C9}''): \text{Tr}(\mathbf{M}_n^c\mathbf{W}) \le 0,\ \forall n\in\mathcal{N}, \\ &(\text{C10}''): \text{Tr}(\mathbf{M}_n^{fog}\mathbf{W}) \le 0,\ \forall n\in\mathcal{N}, \\ &(\text{C11}''): \text{Tr}(\mathbf{M}_n^{com}\mathbf{W}) \le 0,\ \forall n\in\mathcal{N}, \\ &(\text{C12}): \mathbf{W}(7N+2, 7N+2) = 1, \\ &(\text{C13}): \mathbf{W}\succeq 0, \\ &(\text{C14}): \text{rank}(\mathbf{W}) = 1,\end{aligned} \tag{17}$$

where

$$\mathbf{M}_0 = \begin{bmatrix} \mathbf{0}_{(7N+1)\times(7N+1)} & \frac{1}{2}\mathbf{u}_0 \\ \frac{1}{2}(\mathbf{u}_0)^T & 0 \end{bmatrix},$$

$$\mathbf{M}_n^a = \begin{bmatrix} \mathbf{0}_{(7N+1)\times(7N+1)} & \frac{1}{2}\mathbf{u}_n^a \\ \frac{1}{2}(\mathbf{u}_n^a)^T & 0 \end{bmatrix},\ a = I, f, c;\ \forall n,$$

$$\mathbf{M}^{fog} = \begin{bmatrix} \mathbf{0}_{(7N+1)\times(7N+1)} & \frac{1}{2}\mathbf{u}^{fog} \\ \frac{1}{2}(\mathbf{u}^{fog})^T & 0 \end{bmatrix},$$

$$\mathbf{M}_n^v = \begin{bmatrix} \mathbf{Q}_n^v & \frac{1}{2}\mathbf{u}_n^v \\ \frac{1}{2}(\mathbf{u}_n^v)^T & 0 \end{bmatrix},\ v = com, fog;\ \forall n,$$

$$\mathbf{M}_p = \begin{bmatrix} \mathrm{diag}(\mathbf{e}'_{\mathrm{p}}) & -\frac{1}{2}\mathbf{e}'_p \\ -\frac{1}{2}(\mathbf{e}'_p)^T & 0 \end{bmatrix},\ \forall p.$$

In problem $(\mathcal{P}_\mathbf{4})$, the only non-convex constraint is the rank constraint (C14). By dropping the rank constraint (C14), we relax problem $(\mathcal{P}_\mathbf{4})$ into a positive semidefinite programming (PSD) problem as follows

$$\begin{aligned}(\mathcal{P}_\mathbf{5}):\ &\min_{\mathbf{W}}\ \mathrm{Tr}(\mathbf{M}_0\mathbf{W}) \\ \text{s.t. }&(C1'')-(C4''),(C9'')-(C11''), \\ &(C12),(C13).\end{aligned} \tag{18}$$

Now, we have transformed the original problem $(\mathcal{P}_\mathbf{1})$ into a standard convex optimization problem $(\mathcal{P}_\mathbf{5})$, which could be solved in polynominal time using standard CVX tools such as SeDuMi [27].

C. Extracting Offloading Decisions

In this subsection, we extract a feasible solution $\widetilde{\mathbf{s}}$ to $(\mathcal{P}_\mathbf{3})$ from the global optimal solution $\mathbf{W}^*$ to $(\mathcal{P}_\mathbf{5})$. We adopt the method proposed in [11], [16], to obtain the offloading decisions in the feasible solution $\widetilde{\mathbf{s}}$.

According to the definition of $\mathbf{W}$, we know that only the top left $3N \times 3N$ submatrix of $\mathbf{W}^*$, denoted as $\mathbf{W}'^*$, is necessary to obtain the offloading decisions $\boldsymbol{\pi}$; and all the diagonal elements in $\mathbf{W}'^*$ are positive real numbers between 0 and 1. We define $\mathbf{pr} = [pr_1^l, pr_1^f, pr_1^c, ..., pr_N^l, pr_N^f, pr_N^c]^T \triangleq \mathrm{diag}(\mathbf{W}'^*)$, where each entry of $\mathbf{pr}$ indicates the probability of the corresponding entry of $\boldsymbol{\pi}$ being 1.

To satisfy constraint (C2), we define $\Xi_n^l = pr_n^l(1-pr_n^f)(1-pr_n^c)$, $\Xi_n^f = (1-pr_n^l)pr_n^f(1-pr_n^c)$ and $\Xi_n^c = (1-pr_n^l)(1-pr_n^f)pr_n^c$. Based on them, the probabilities of local, fog, and cloud processing for UE$_n$ are given respectively as $Pr_n^l = \Xi_n^l/(\Xi_n^l+\Xi_n^f+\Xi_n^c)$, $Pr_n^f = \Xi_n^f/(\Xi_n^l+\Xi_n^f+\Xi_n^c)$, and $Pr_n^c = \Xi_n^c/(\Xi_n^l+\Xi_n^f+\Xi_n^c)$.

Then the location where the application of UE$_n$ will be executed is given by

$$\mathbf{O}_n = \begin{cases} (1,0,0), & \text{local processing with prob. } Pr_n^l, \\ (0,1,0), & \text{fog processing with prob. } Pr_n^f, \\ (0,0,1), & \text{cloud processing with prob. } Pr_n^c. \end{cases} \tag{19}$$

By randomly setting the value of the vector according to the probabilities in (19), problem $(\mathcal{P}_\mathbf{6})$ is resolved and the offloading decisions x_n, y_n and z_n of UE$_n$ can be obtained.

D. Joint Offloading Decision and Resource Allocation algorithm

In the above procedure for offloading decision extraction in (19), since the offloading decisions are obtained randomly according to the obtained probabilities Pr_n^l, Pr_n^f, Pr_n^c, we can run the above procedure several times to obtain more accurate offloading decisions. Each run is referred to as a randomization trial and the number of randomization trails is denoted by L. After that L i.i.d feasible offloading decisions $\boldsymbol{\pi}^l, l = 1, ..., L$ are obtained. Then we perform radio and computation resource allocation under each $\boldsymbol{\pi}^l$, and L solutions including offloading decision and resource allocation are obtained, among which the one with the minimum objective value is considered as the final solution $\boldsymbol{\pi}^*$. A small value of L will be sufficient to obtain a satisfying result [11]. The global framework of CORA is shown in Algorithm 1.

Algorithm 1 **C**omputation **O**ffloading Decision Making and **R**esource **A**llocation Algorithm (CORA)

Initialization:

1: Initialize L, N, B, N_0, F^{fog}.
2: Initialize $D_n, App_n, f_n^{loc}, p_n^{max}, p_n^{id}, p_n^{loc}, R_n^{fc}, f_n^c, \tau_n^{max}$ of each UE.
3: Initialize all the matrixes involved in $(\mathcal{P}_\mathbf{5})$;

Iteration:

4: Solve the SDR problem $(\mathcal{P}_\mathbf{5})$ adopting standard CVX tool SeDuMi to get its optimal $\mathbf{W}^*$.
5: Extract the top left corner $3N \times 3N$ sub-matrix $\mathbf{W}'^*$ from matrix $\mathbf{W}^*$, and denote the values of diagonal elements in $\mathbf{W}'^*$ as $\mathbf{pr} = [pr_1^l, pr_1^f, pr_1^c, ..., pr_N^l, pr_N^f, pr_N^c]^T$.
6: **for** $l = 1, ..., L$ **do**
7: Extract $\boldsymbol{\pi}^l$ from $\mathbf{pr}^l = [pr_1^l, pr_1^f, pr_1^c, ..., pr_N^l, pr_N^f, pr_N^c]^{T^l}$ according to (19).
8: Perform radio and computation resource allocation under $\boldsymbol{\pi}^l$.
9: **end for**
10: Compare the objective value of all the L solutions, and choose the solution with the minimum objective value.
11: **Output**: The corresponding offloading decision $\boldsymbol{\pi}^*$ and resource allocation is considered as the final solution.

E. Dimensional Reduction of Original Problem $(\mathcal{P}_\mathbf{1})$

In lines 6-9 in the iteration of Algorithm 1, after offloading decision $\boldsymbol{\pi}^l$ is obtained, we need to perform resource allocation in line 8 under given $\boldsymbol{\pi}^l$. For notation simplicity, we denote $\boldsymbol{\pi}^l$ as $\boldsymbol{\pi}$. Then problem $(\mathcal{P}_\mathbf{1})$ reduces to the optimization of radio and computation resource allocation, which is embedded in Step 7 of Algorithm 1 as follows

$$\begin{aligned}(\mathcal{P}_\mathbf{6}):\ &\min_{\mathbf{f}^{fog},\mathbf{p}^{com},\mathbf{a}}\ \max_{n\in\mathcal{N}}\ (\lambda_n^e p_n^{com}+\lambda_n^t)\frac{D_n}{r_n}(y_n+z_n) \\ &\qquad +(\lambda_n^e p_n^{id}+\lambda_n^t)\frac{C_n}{f_n^{fog}}y_n+u_n \\ \text{s.t. }&(C3)-(C8),\end{aligned} \tag{20}$$

where $u_n = (\lambda_n^e p_n^{loc} + \lambda_n^t)\frac{C_n}{f_n^{loc}} x_n + (\lambda_n^e p_n^{id} + \lambda_n^t)(T_n^{fc} + T_n^c) z_n$ is a constant for a given offloading decision $\boldsymbol{\pi}$. According to $(\mathcal{P}_{\mathbf{6}})$, the computation resource allocation $\mathbf{f}^{fog}$ and the radio resource allocation $\mathbf{p}^{com}, \mathbf{a}$ are decoupled both in objective function and the constraints, so $(\mathcal{P}_{\mathbf{6}})$ can be decomposed into the joint optimization of computation resource allocation and radio resource allocation, which will be detailed in the next two sections.

V. Iterative Computation Resource Allocation

Under given radio resource allocation $\mathbf{p}^{com}, \mathbf{a}$, the optimal computation resource allocation can be obtained by solving the following problem:

$$\begin{aligned}(\mathcal{P}_{\mathbf{7}}): &\min_{\mathbf{f}^{fog}} \max_{n\in\mathcal{N}_1} (\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}} + B_n \\ \text{s.t. } &(\mathrm{C3}^+): \sum_{n\in\mathcal{N}_1} f_n^{fog} \le F^{fog}, \\ &(\mathrm{C4}^+): \ f_n^{fog} \ge 0, \ \forall n \in \mathcal{N}_1,\end{aligned} \tag{21}$$

where $B_n = (\lambda_n^e p_n^{com} + \lambda_n^t)\frac{D_n}{r_n}$ is a constant now. Let $(\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}} + B_n \le \zeta_1$, the nonsmooth problem $(\mathcal{P}_{\mathbf{7}})$ is transformed into

$$\begin{aligned}(\mathcal{P}_{\mathbf{8}}): &\min_{\mathbf{f}^{fog}, \zeta_1} \ \zeta_1 \\ \text{s.t. } &(\mathrm{C3}^+), (\mathrm{C4}^+), \\ &(\mathrm{C15}): \ (\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}} + B_n \le \zeta_1, \ \forall n \in \mathcal{N}_1.\end{aligned} \tag{22}$$

As $(\lambda_n^e p_n^{id} + \lambda_n^t)\frac{C_n}{f_n^{fog}} \ge 0$, thus $\zeta_1 - B_n \ge 0$, and we have $0 \le \frac{C_n(\lambda_n^e p_n^{id} + \lambda_n^t)}{\zeta_1 - B_n} \le f_n^{fog}, \forall n \in \mathcal{N}_1$.

Based on the analysis above, we obtain $\sum_{n\in\mathcal{N}_1} \frac{C_n(\lambda_n^e p_n^{id} + \lambda_n^t)}{\zeta_1 - B_n} \le \sum_{n\in\mathcal{N}_1} f_n^{fog} \le F^{fog}$.

Next, we perform computation resource allocation among all the fog-executing UEs to minimize the maximum cost among them to guarantee min-max fairness. To this end, we need to allocate more computation resource to the UE with the maximum cost. Thus the cost of this UE is reduced while that of other UEs will increase. By performing this procedure iteratively, in the end, all the computation resource will be allocated and all the fog-executing UEs will be assigned the same quality of computation resource. Thus we have

$$\sum_{n\in\mathcal{N}_1} \frac{C_n(\lambda_n^e p_n^{id} + \lambda_n^t)}{\zeta_1 - B_n} = \sum_{n\in\mathcal{N}_1} f_n^{fog} = F^{fog}. \tag{23}$$

Then problem $(\mathcal{P}_{\mathbf{8}})$ could be transformed into

$$\begin{aligned}(\mathcal{P}_{\mathbf{9}}): &\min_{\zeta_1} \ \zeta_1 \\ \text{s.t. } &(\mathrm{C16}): \ \sum_{n\in\mathcal{N}_1} \frac{C_n(\lambda_n^e p_n^{id} + \lambda_n^t)}{\zeta_1 - B_n} = F^{fog}.\end{aligned} \tag{24}$$

As the left side of constraint (C16) is monotonic decreasing with ζ_1, the bisection method could be employed to resolve problem $(\mathcal{P}_{\mathbf{9}})$. The procedure of the proposed **b**isection method for **c**omputation **r**esource **a**llocation algorithm (BCRA) is described in Algorithm 2.

Algorithm 2 Bisection Method for **C**omputation **R**esource **A**llocation Algorithm (BCRA)

Initialization:

1: Set $\zeta_1^{min} = max\{B_n\}$, $\zeta_1^{max} = \sum_{n\in\mathcal{N}_1}\left(\frac{C_n(\lambda_n^e p_n^{id} + \lambda_n^t)N_1}{F^{fog}} + B_n\right)$, such that $\zeta_1^{min} \le \zeta^{opt} \le \zeta_1^{max}$.
2: Set $l = 1$ and the maximum tolerance $\varepsilon > 0$.

Iteration:

3: **while** 1 **do**
4: $\zeta_1^l = (\zeta_1^{min} + \zeta_1^{max})/2$.
5: **if** $\mid \zeta_1^{max} - \zeta_1^{min} \mid \le \varepsilon$ **then**
6: $\zeta_1^{opt} = \zeta_1^l$
7: **else**
8: **if** $\sum_{n\in\mathcal{N}_1} \frac{C_n y_n}{\zeta_1^l - B_n} > F^{fog}$ **then**
9: $\zeta_1^{min} = \zeta_1^l$.
10: **else**
11: $\zeta_1^{max} = \zeta_1^l$.
12: **end if**
13: **end if**
14: $l = l + 1$
15: **end while**
16: Substituting ζ_1^l into (23), computation resource allocation scheme $\mathbf{f}^{fog}$ is obtained.
17: **Output**: $\mathbf{f}^{fog}$.

VI. Iterative Radio Resource Allocation Algorithm Design

When the computational resource allocation $\mathbf{f}^{fog}$ has been obtained, problem $(\mathcal{P}_{\mathbf{6}})$ reduces to the optimization of transmit power and bandwidth allocation among all remote-executing UEs in $\mathcal{N}_2$ as follows

$$\begin{aligned}(\mathcal{P}_{\mathbf{10}}): &\min_{\mathbf{p}^{com}, \mathbf{a}} \max_{n\in\mathcal{N}_2} \frac{D_n(\lambda_n^e p_n^{com} + \lambda_n^t)}{a_n B \log_2(1 + \frac{p_n^{com} h_n}{a_n N_0 B})} \\ s.t. \ &(\mathrm{C5}^+): \ 0 \le a_n \le 1, \ \forall n \in \mathcal{N}_2, \\ &(\mathrm{C6}^+): \ \sum_{n\in\mathcal{N}_2} a_n \le 1, \\ &(\mathrm{C7}^+): \ 0 \le p_n^{com} \le p_n^{max}, \forall n \in \mathcal{N}_2, \\ &(\mathrm{C8}^+): \ T_n^{fog} y_n + T_n^{cloud} z_n \le \tau_n^{max}, \forall n \in \mathcal{N}_2,\end{aligned} \tag{25}$$

where the constant $u_n = (\lambda_n^e p_n^{loc} + \lambda_n^t)\frac{C_n}{f_n^{loc}} x_n + (\lambda_n^e p_n^{id} + \lambda_n^t)(T_n^{fc} + T_n^c) z_n$ in the objective function is omitted, because it does not affect the problem solving. As the objective function is non-convex, $(\mathcal{P}_{\mathbf{10}})$ is a non-convex optimization problem. Nevertheless, $(\mathcal{P}_{\mathbf{10}})$ can be grouped into nonlinear fractional programming problem [29], so fractional optimization could be employed to solve it.

For notational simplicity, we define the feasible solutions set of $(\mathcal{P}_{\mathbf{10}})$ as $\mathcal{F}(\mathcal{F} \ne 0)$. Denote the optimal solution and

optimal value of $(\mathcal{P}_{\mathbf{10}})$ as $\{\mathbf{p}^{com*}, \mathbf{a}^*\}$ and V^*, respectively, we have

$$V^* = \min_{\{\mathbf{p}^{com},\mathbf{a}\}\in\mathcal{F}} \max_{n\in\mathcal{N}_2} \frac{D_n(\lambda_n^e p_n^{com} + \lambda_n^t)}{a_n B \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right)} = \max_{n\in\mathcal{N}_2} \frac{D_n(\lambda_n^e p_n^{com*} + \lambda_n^t)}{a_n^* B \log_2\left(1+\frac{p_n^{com*} h_n}{a_n^* N_0 B}\right)}. \quad (26)$$

Proposition 1: The optimal value V^* is reached if and only if

$$\min_{\{\mathbf{p}^{com},\mathbf{a}\}\in\mathcal{F}} \max_{n\in\mathcal{N}_2} \left[D_n(\lambda_n^e p_n^{com} + \lambda_n^t) - V^* a_n B \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right)\right] = \max_{n\in\mathcal{N}_2} \left[D_n(\lambda_n^e p_n^{com*} + \lambda_n^t) - V^* a_n^* B \log_2\left(1+\frac{p_n^{com*} h_n}{a_n^* N_0 B}\right)\right] = 0. \quad (27)$$

Proof: See Appendix A. □

Proposition 1 indicates that $(\mathcal{P}_{\mathbf{10}})$ can be solved via solving its equivalent problem (27). Nevertheless, V^* is usually unknown in advance. To tackle the difficulty, we replace V^* with an update parameter V [29]. The procedure is elaborated in Algorithm 3, where the optimization problem in line 4 under a given V (e.g., V^i at iteration i) is

$$(\mathcal{P}_{\mathbf{11}}): \min_{\mathbf{p}^{com},\mathbf{a}} \max_{n\in\mathcal{N}_2} \left[D_n(\lambda_n^e p_n^{com} + \lambda_n^t) - V a_n B \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right)\right] \quad s.t.\ (\mathrm{C5}^+) - (\mathrm{C8}^+). \quad (28)$$

To solve $(\mathcal{P}_{\mathbf{11}})$, substituting equations (5) and (7) into constraint (C8$^+$)and noting that $y_n + z_n = 1, n \in \mathcal{N}_2$, we obtain

$$a_n B \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right) \geq \frac{D_n}{\tau_n^{max} - v_n}, \quad (29)$$

where $v_n = \frac{C_n}{f_n^{fog}} y_n + (T_n^{fc} + T_n^c) z_n$ is a constant.

Similar to (12), let $D_n(\lambda_n^e p_n^{com} + \lambda_n^t) - V a_n \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right) \leq \zeta_2$, $(\mathcal{P}_{\mathbf{11}})$ is recasted as

$$(\mathcal{P}_{\mathbf{12}}): \min_{\mathbf{p}^{com},\mathbf{a},\zeta_2} \ \zeta_2 \quad (30)$$
$$\text{s.t. } (\mathrm{C5}^+), (\mathrm{C6}^+),$$
$$(\mathrm{C7}_1^+): p_n^{com} \geq 0, n \in \mathcal{N}_2,$$
$$(\mathrm{C7}_2^+): p_n^{com} \leq p_n^{max}, n \in \mathcal{N}_2,$$
$$(\mathrm{C17}): a_n B \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right) \geq \frac{D_n}{\tau_n^{max} - v_n}, n \in \mathcal{N}_2,$$
$$(\mathrm{C18}): D_n(\lambda_n^e p_n^{com} + \lambda_n^t) - V a_n \log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right) \leq \zeta_2, \ n \in \mathcal{N}_2.$$

Proposition 2: Problem $(\mathcal{P}_{\mathbf{12}})$ is jointly convex in $\mathbf{p}^{com}$, $\mathbf{a}$ and ζ_2.

Proof: See Appendix B. □

As problem $(\mathcal{P}_{\mathbf{12}})$ is convex, the Slaters condition [30] is satisfied and the zero duality gap is guaranteed, thus the problem could be resolved using Lagrange dual decomposition and subgradient projection [31].

The Lagrange function of $(\mathcal{P}_{\mathbf{12}})$ is given by (31), where $\beta \geq 0$, $\boldsymbol{\omega} = [\omega_1, ..., \omega_{N_2}] \succeq 0$, $\boldsymbol{\mu} = [\mu_1, ..., \mu_{N_2}] \succeq 0$, and $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_{N_2}] \succeq 0$ are Lagrange multipliers corresponding to (C6$^+$), (C7$_2^+$), (C17), and (C18), respectively.

The Lagrange dual function is given by

$$D(\beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma}) = \min_{\mathbf{p}^{com},\mathbf{a},\zeta_2} L(\mathbf{p}^{com}, \mathbf{a}, \zeta_2, \beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma}) \quad (32)$$
$$\text{s.t. } (\mathrm{C5}^+), (\mathrm{C7}_1^+).$$

From (32), we minimize $L(\mathbf{p}^{com}, \mathbf{a}, \zeta_2, \beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma})$ for a given set of dual variables $\beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma}$ to obtain the transmit power $\mathbf{p}^{com}$ and bandwidth allocation $\mathbf{a}$, by resolving the following two problems.

Algorithm 3 Iterative Power and Bandwidth Allocation Algorithm to solve $(\mathcal{P}_{\mathbf{10}})$

Initialization:
1: Set the maximum iteration number i_{max} and precision ϵ.
2: Set the initial iteration index $i = 0$ and the initial optimal value $V^i = 1$.
Iteration:
3: **while** $i < i_{max}$ **do**
4: For given V^i, solve $(\mathcal{P}_{\mathbf{11}})$ to obtain $\{\mathbf{p}^{comi}, \mathbf{a}^i\}$.
5: **if**
$$\left| \max_{n\in\mathcal{N}_2} \left[D_n(\lambda_n^e p_n^{comi} + \lambda_n^t) - V^i a_n^i B \log_2\left(1+\frac{p_n^{comi} h_n^i}{a_n^i N_0 B}\right)\right] \right| < \epsilon$$
then
6: $\{\mathbf{p}^*, \mathbf{a}^*\} = \{\mathbf{p}^{comi}, \mathbf{a}^i\}$.
7: $V^* = \max_{n\in\mathcal{N}_2} \frac{D_n(\lambda_n^e p_n^{comi} + \lambda_n^t)}{a_n^i B \log_2\left(1+\frac{p_n^{comi} h_n^i}{a_n^i N_0 B}\right)}$.
8: **else**
9: Set $V^{i+1} = \max_{n\in\mathcal{N}_2} \frac{D_n(\lambda_n^e p_n^{comi} + \lambda_n^t)}{a_n^i B \log_2\left(1+\frac{p_n^{comi} h_n^i}{a_n^i N_0 B}\right)}$.
10: break.
11: **end if**
12: **end while**

$$
\begin{aligned}
&L(\mathbf{p}^{com},\mathbf{a},\zeta_2,\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})\\
&=\zeta_2+\beta\left(\sum_{n\in\mathcal{N}_2}a_n-1\right)+\sum_{n\in\mathcal{N}_2}\omega_n\left(p_n^{com}-p_n^{max}\right)+\sum_{n\in\mathcal{N}_2}\mu_n\left[\frac{D_n}{\tau_n^{max}-v_n}-a_nB\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)\right]\\
&\quad+\sum_{n\in\mathcal{N}_2}\gamma_n\left[D_n(\lambda_n^e p_n^{com}+\lambda_n^t)-Va_nB\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)-\zeta_2\right].
\end{aligned}
\tag{31}
$$

A. Resource Allocation Update

$$
\begin{aligned}
(\mathcal{P}_{\mathbf{13}}):\min_{\mathbf{p}^{com},\mathbf{a}}\ &\left\{\sum_{n\in\mathcal{N}_2}\omega_n p_n^{com}-\sum_{n\in\mathcal{N}_2}\mu_n a_nB\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)\right.\\
&+\beta\sum_{n\in\mathcal{N}_2}a_n+\sum_{n\in\mathcal{N}_2}\gamma_n\Big[D_n(\lambda_n^e p_n^{com}+\lambda_n^t)\\
&\left.-Va_nB\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)\Big]\right\}\\
\text{s.t.}\ \ &(\mathrm{C5}^+),(\mathrm{C7}_1^+).
\end{aligned}
\tag{33}
$$

1) Power and Bandwidth Allocation:

- Optimal Transmit Power Allocation

For a given bandwidth allocation $\mathbf{a}$, the Karush-Kuhn-Tucker (KKT) conditions are satisfied. By differentiating $L(\mathbf{p}^{com},\mathbf{a},\zeta_2,\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})$ with respect to p_n^{com}, $n\in\mathcal{N}_2$, and let it equal 0, we obtain the optimal transmit power allocation for each UE as follows,

$$
p_n^{com*}=\left\{a_n\left[\frac{B(\mu_n+V\gamma_n)}{\ln 2(\omega_n+\gamma_nD_n\lambda_n^e)}-\frac{N_0B}{h_n}\right]\right\}^+,n\in\mathcal{N}_2,
\tag{34}
$$

where $x^+\triangleq max\{0,x\}$.

- Optimal Bandwidth Assignment

After $\mathbf{p}^{com*}$ has been obtained, by differentiating $L(\mathbf{p}^{com},\mathbf{a},\zeta_2,\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})$ w.r.t. a_n and letting it equal 0, we obtain

$$
\begin{aligned}
&(\mu_n+V\gamma_n)B\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)\frac{\mu_n+V\gamma_n}{\ln 2}\frac{p_n^{com}h_nB}{a_nN_0B+p_n^{com}h_n}\\
&=\beta.
\end{aligned}
\tag{35}
$$

Rearranging (35) and denoting $X_n=X_n(\mu_n,\gamma_n,\beta)=\frac{1}{1+\frac{p_n^{com}h_n}{a_nN_0B}}$ as the solution to

$$
X_n-\ln 2\log_2(X_n)=\frac{\beta\ln 2}{(\mu_n+V\gamma_n)B}+1,
\tag{36}
$$

it is easy to show $0<X_n<1$. For $X_n\in(0,1)$, we have $X_n-\ln 2\log_2(X_n)\in(1,+\infty)$, which suggests that $X_n\in(0,1)$ always exists as all the dual variables μ_n,γ_n,β are non-negative, and $X_n-\ln 2\log_2(X_n)-\frac{\beta\ln 2}{(\mu_n+V\gamma_n)B}-1=0$ decreases with X_n, and has a root in $(0,1)$. Consequently, a bisection search in $(0,1)$ could be used for X_n^*, and the bandwidth allocation a_n^* is given as

$$
a_n^*=\frac{p_n^{com}h_n}{N_0B}\frac{X_n^*(\mu_n,\gamma_n,\beta)}{1-X_n^*(\mu_n,\gamma_n,\beta)},\ (0<X_n^*(\mu_n,\gamma_n,\beta)<1).
\tag{37}
$$

2) Adaptive ζ_2 Selection:

$$
\begin{aligned}
&(\mathcal{P}_{\mathbf{14}}):\min_{\zeta_2}\ \zeta_2\\
&\text{s.t.}(\mathrm{C18}^+):D_n(\lambda_n^e p_n^{com}+\lambda_n^t)-Va_nB\log_2\left(1+\frac{p_n^{com}h_n}{a_nN_0B}\right)\\
&\qquad\le\zeta_2\le 0,\ n\in\mathcal{N}_2.
\end{aligned}
\tag{38}
$$

From $(\mathcal{P}_{\mathbf{14}})$, the optimal solution ζ_2^* is

$$
\zeta_2^*=\begin{cases}0, & 1<\sum\limits_{n\in\mathcal{N}_2}\gamma_n\\ G_n^*, & 1>\sum\limits_{n\in\mathcal{N}_2}\gamma_n\end{cases},\ \forall n\in\mathcal{N}_2,
\tag{39}
$$

where $G_n^*=\max\limits_{n\in\mathcal{N}_2}\left[D_n(\lambda_n^e p_n^{com}+\lambda_n^t)-Va_n^*B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^*N_0B}\right)\right]$.

B. Lagrange Multipliers Update

The optimal resource allocations in (34) and (37) depend on the dual variables $\beta,\boldsymbol{\omega},\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$, which can be updated by solving the dual problem of $(\mathcal{P}_{\mathbf{12}})$ as follows

$$
\begin{aligned}
(\mathcal{P}_{\mathbf{15}}):&\max_{\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma}}\ D(\beta,\boldsymbol{\omega},\mu,\boldsymbol{\gamma})\\
&\text{s.t.}\ \ \beta\ge 0,\boldsymbol{\omega}\succeq 0,\boldsymbol{\mu}\succeq 0,\boldsymbol{\gamma}\succeq 0.
\end{aligned}
\tag{40}
$$

According to (31) and (32), $(\mathcal{P}_{\mathbf{15}})$ is convex as $D(\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})$ is a linear function w.r.t. the dual variables $\beta,\boldsymbol{\omega},\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$. Thus, subgradient projection could be applied to solve $(\mathcal{P}_{\mathbf{15}})$.

Proposition 3: The subgradients of $D(\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})$ are given as

$$
\nabla\beta=\sum_{n\in\mathcal{N}_2}a_n^*-1,
\tag{41}
$$

$$
\nabla\omega_n=p_n^{com*}-p_n^{max},\ \forall n\in\mathcal{N}_2,
\tag{42}
$$

$$
\nabla\mu_n=\frac{D_n}{\tau_n^{max}-v_n}-a_n^*B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^*N_0B}\right),\forall n\in\mathcal{N}_2,
\tag{43}
$$

$$
\begin{aligned}
\nabla\gamma_n=&D_n(\lambda_n^e p_n^{com*}+\lambda_n^t)-Va_n^*B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^*N_0B}\right)\\
&-\zeta_2,\ \forall n\in\mathcal{N}_2,
\end{aligned}
\tag{44}
$$

where $p_n^{com*},\ a_n^*,\ n\in\mathcal{N}_2$ are the optimal solutions to (37) for a given set of dual variables $\beta,\boldsymbol{\omega},\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$.

Proof: See Appendix C. □

Based on (41)–(44), the Lagrange multipliers are updated

using the subgradient projection method as follows:

$$\beta(t+1) = [\beta(t) - h(t) \bigtriangledown \beta(t)]^+, \quad (45)$$

$$\omega_n(t+1) = [\omega_n(t) - i(t) \bigtriangledown \omega_n(t)]^+, \ \forall n \in \mathcal{N}_2, \quad (46)$$

$$\mu_n(t+1) = [\mu_n(t) - j(t) \bigtriangledown \mu_n(t)]^+, \ \forall n \in \mathcal{N}_2, \quad (47)$$

$$\gamma_n(t+1) = [\gamma_n(t) - k(t) \bigtriangledown \gamma_n(t)]^+, \ \forall n \in \mathcal{N}_2, \quad (48)$$

where t is iteration index; $h(t), i(t), j(t)$ and $k(t)$ are positive step sizes. We adopt square-summable but not summable step sizes [30], where $h(t) = 1/(10^{-2}t)$, $i(t) = 1/(10t)$, $j(t) = -1/(10^{13.8}t)$, and $k(t) = 1/(10^{18}t)$. The Lagrange multipliers are updated iteratively until the terminal condition is med. The whole procedure to solve $(\mathcal{P}_{\mathbf{11}})$ is summarized in Algorithm 4.

Till now, the complete solution to the primal optimization problem $(\mathcal{P}_{\mathbf{1}})$ is obtained. For the sake of a clear understanding, the detailed flow diagram of CORA is shown in Fig.2.

Algorithm 4 Suboptimal Power and Bandwidth Allocation Algorithm to solve problem $(\mathcal{P}_{\mathbf{11}})$

Initialization:

Set $\beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \tau_n^{max}$ and the precision δ. Set $t = 0$.

Iteration:

while $t \leq t_{max}$ **do**
  Allocate transmit power $p_n^{com}(t)$ according to (34).
  Perform bisection search between $(0, 1)$ for $X_n(t)$.
  Assign bandwidth $a_n(t)$ from (37) based on $X_n(t)$.
  Update Lagrange multipliers $\beta, \boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\gamma}$ from (45)-(48), respectively.
  if $\|\beta(t+1) - \beta(t)\|_2 < \delta$, $\|\boldsymbol{\omega}(t+1) - \boldsymbol{\omega}(t)\|_2 < \delta$,$\|\boldsymbol{\mu}(t+1) - \boldsymbol{\mu}(t)\|_2 < \delta$, $\|\boldsymbol{\gamma}(t+1) - \boldsymbol{\gamma}(t)\|_2 < \delta$ **then**
    $a_n^* = a_n(t)$, $p_n^{com*} = p_n^{com}(t)$.
    break.
  else
    $t = t + 1$.
  end if
end while

Output: $\mathbf{a}^* = [a_n^*, ..., a_{N_2}^*]$, $\mathbf{p}^{com*} = [p_1^{com*}, ..., p_{N_2}^{com*}]$.

VII. Complexity Analysis of CORA

The computational complexity of CORA in Algorithm 1 mainly comes from Step 4 and Step 8 in the for-loop for L times. In Step 4 of Algorithm 1, the SDR problem could be resolved easily within a precision ε_1 using the interior-point method within $O(\sqrt{N}\log(\frac{1}{\varepsilon_1}))$ iterations, where the computational complexity per iteration is $O(N^6)$, thus the complexity of Step 4 is $O(N^{6.5}\log(\frac{1}{\varepsilon_1}))$ [30].

In Step 8 of Algorithm 1, it contains Algorithm 2 and Algorithm 3 in fact. In Algorithm 2, the bisection method costs $O\left(\log_2\left(\frac{\zeta_1^{max}-\zeta_1^{min}}{\varepsilon}\right)\right)$ iterations. In Algorithm 3, computational complexity mainly comes from Step 4 (i.e., Algorithm 4) in the while loop. In Algorithm 4, the complexity mainly focuses on the bisection search in $(0, 1)$ (i.e., Step 4) in the while-loop, and the complexity of Step 4 in Algorithm 4 is $O(\log_2(\frac{1-0}{\delta_1}))$. The subgradient projection method needs $O\left(\frac{1}{\delta^2}\right)$ iterations to converge [30]. Therefore, the total complexity of the while-loop in Algorithm 4 is $O\left(\frac{1}{\delta^2}\log_2\left(\frac{1}{\delta_1}\right)\right)$. The while-loop in Algorithm 3 needs M iterations to converge (as shown in simulations latter, M is usually no more than 2), so the complexity of Algorithm 3 is $O\left(\frac{1}{\delta^2}\log_2\left(\frac{1}{\delta_1}\right)M\right) = O\left(\frac{1}{\delta^2}\log_2\left(\frac{1}{\delta_1}\right)\right)$.

Thus the complexity of CORA is $O\left(N^{6.5}\log(\frac{1}{\varepsilon_1}) + L\left(\log_2(\frac{\zeta_1^{max}-\zeta_1^{min}}{\varepsilon}) + \frac{1}{\delta^2}\log_2(\frac{1}{\delta_1})\right)\right)$.

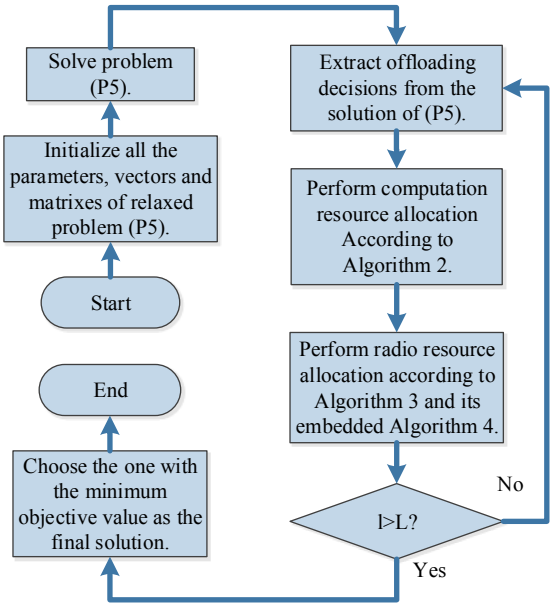


Fig. 2. The flow chart of CORA.

VIII. Simulation Results

In this section, we present simulation results to first verify the convergence of the three iterative algorithms (Algorithms 2, 3 and 4), and then evaluate the performance of the proposed algorithm CORA. Simulation is performed on a Monte Carlo simulation on a Matlab-based simulator. We simulate a mixed fog/cloud computing system with one WiFi AP based fog node, a cloud server, and multiple UEs. TGn path loss model and Rician fading with 6 dB Rician factor is considered [31], [32]. Other parameters are listed in Table II. Note that each point in the following figures (except for Fig. 5) are based on the average values of 5000 runs.

Remark 2. In Table II, the "Unchanged" parameters are kept unchanged in our simulation; while the "Default" parameters are set as default unless otherwise specified, because their values may change in our simulation. Note that in most of the simulations, we take $\lambda_n^e = 1$ and $\lambda_n^t = 0$ as the default values, i.e., we take EC as our default optimization objective. In addition, in the second sub-figure of Fig. 9, we take delay as the objective function to show our algorithm works well under different optimization objectives including EC and delay.

TABLE II
SIMULATION PARAMETERS

	Parameter	Value
Unchanged	N_0	-174 dBm/Hz [24]
	p_n^{id}	$0.001 - 0.01$ W uniformly [20]
	B	15 MHz [25]
	p_n^{max}	0.1 W [17]
	R_n^{fc}	1 M b/s [33]
	p_n^{loc}	$0.1 - 0.5$ W uniformly [20]
Default	L	6
	N	6
	F^{fog}	2 G cycles/s [16]
	f_n^c	4 G cycles/s [16]
	f_n^{loc}	$0.5 - 1.5$ G cycles/s uniformly [13]
	τ_n^{max}	4 s [18]
	D_n	0.42 MB [10]
	App_n	297.62 cycles/bit [10]
	λ_n^e	1
	λ_n^t	0

A. Convergence of Algorithms 2, 3 and 4

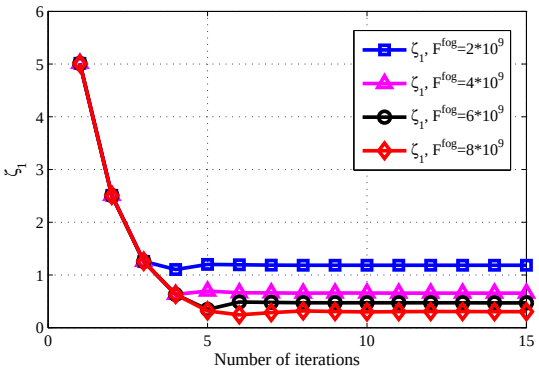


Fig. 3. Convergence of the main loop of Algorithm 2.

Fig. 3 plots ζ_1 (As the analysis in Section V, ζ_1 in fact is the maximum EC of all fog-processing UEs) versus the number of iterations to show the convergence of the main loop of Algorithm 2 for different processing capabilities of the fog server F^{fog}. As can be seen, ζ_1 keeps decreasing after each iteration until convergence. This is because the BCRA aims to minimize the maximum EC among all UEs in each iteration by performing computation resource allocation, thus the maximum EC can be reduced by an appropriate computation resource allocation. As shown in Fig. 3, the number of iterations is always no more than 10.

In Fig. 4, we plot V^i versus the number of iterations to show the convergence evolution of the outer loop of Algorithm 3, for different number of UEs N. It is observed that it converges typically in two iterations.

In Fig. 5, we further plot the dual variables $\boldsymbol{\mu} = [\mu_n]$, $\boldsymbol{\gamma} = [\gamma_n], n \in \mathcal{N}_2$ versus the number of iterations to depict the convergence of the inner loop of Algorithm 3, i.e., Algorithm 4, under $N = 6$. Observing form Fig. 5, it can be known that UEs $1, 2, 4, 6$ are remote-processing UEs, and radio resource allocation in Algorithm 4 will be performed within them; it

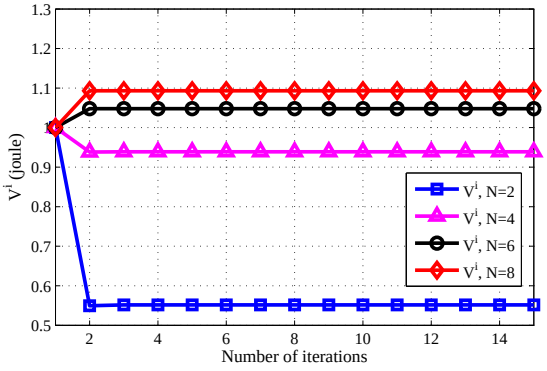


Fig. 4. Convergence of the outer loop of Algorithm 3.

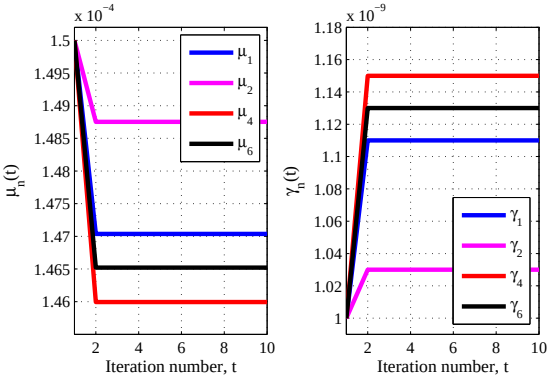


Fig. 5. Convergence of Algorithm 4.

can also be known that Algorithm 4 has a fast convergence rate. It should be note that Fig. 5 has to be plotted based on one random realization, because Fig. 5 plots the two dual variables, μ_n and $\gamma_n, n \in \mathcal{N}_2$, versus the number of iterations for each UE in set $\mathcal{N}_2$. As the offloading decision is independent in each run, and the UEs in set $\mathcal{N}_2$ will be different in different runs. Hence, the dual variables μ_n and $\gamma_n, n \in \mathcal{N}_2$, cannot be averaged over multiple runs.

B. Performance of CORA

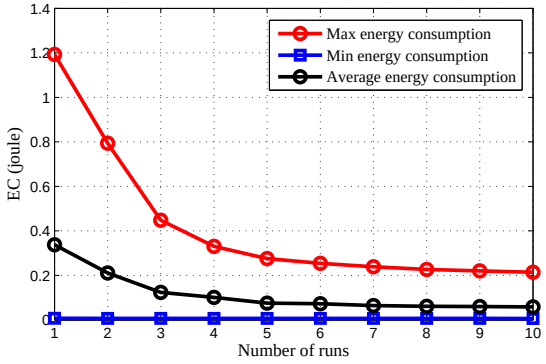


Fig. 6. The maximum, minimum, and average EC among all UEs obtained by CORA vs. the number of runs L.

In Fig. 6, we show the maximum, the minimum and the average system EC obtained by CORA vs. the number of

runs L. As is shown, the three kinds of EC keeps dropping with the increase of L. This is because CORA is proposed aiming at minimizing the maximum EC among all UEs, and the offloading decisions are extracted from the probabilities in (19) randomly and then resource allocation is performed, so the system performance increases (i.e., the objective value decrease) with the number of the runs L. Moreover, the three kinds of system EC decrease sharply at the begin and slowly with the number of L increases. So a moderate L will be the best choice to obtain better performance and without too high computational complexity, and we take $L = 6$ as the default runs in this paper.

Next we evaluate the performance of CORA in comparison with the following three algorithms: (i) Offloading-only algorithm [10], where only offloading decisions are optimized to minimize the weighted sum of EC and delay for each UE, while no resource allocation optimization. (ii) Resource-only algorithm [24], where only the allocation of resources (including transmit power, bandwidth and computation resource) is optimized to minimize the power consumption of each UE, without optimizing offloading decisions. (iii) Local-only: all UEs process their applications themselves without any optimization.

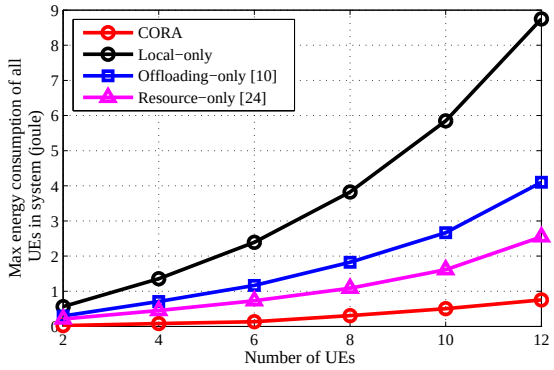


Fig. 7. The max EC among all UEs vs. the number of UEs N.

In Figures 7 and 8, we present comparisons of CORA with Local-only, Offloading-only, and Resource-only, respectively, under different number of UEs N. Fig. 7 shows the comparison on the objective value, i.e., the maximum EC among all UEs in system, which keeps increasing with the number of UEs N increases for all algorithms. However, CORA increases the slowest, while other three algorithms grow sharper and sharper with the number of UEs N more than 8, demonstrating CORA performs good in EC reduction.

In order to show the percentage of UEs benefited from computation offloading, in Fig. 8 we show the number of beneficial UEs vs. the total number of UEs, where a beneficial UE is defined as the UE that consumes less energy than when adopting Local-only method. As there's no any optimization in Local-only, no UE benefits in the method, so we plot Fig. 8 without plotting the bars of Local-only. As a result of the joint optimization of the offloading decision making, the allocation of computation resource, transmit power and radio

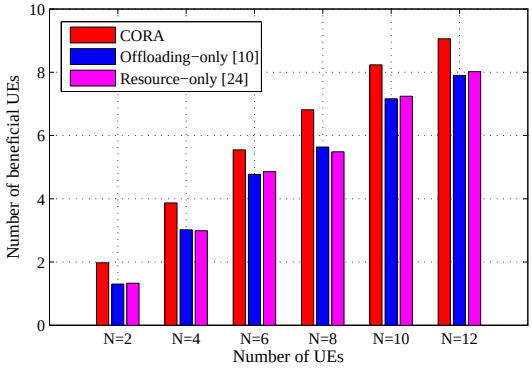


Fig. 8. The number of beneficial UEs vs. the number of UEs N.

bandwidth, CORA can always benefit the most number of UEs compared with other algorithms, which only optimize part of the optimization items.

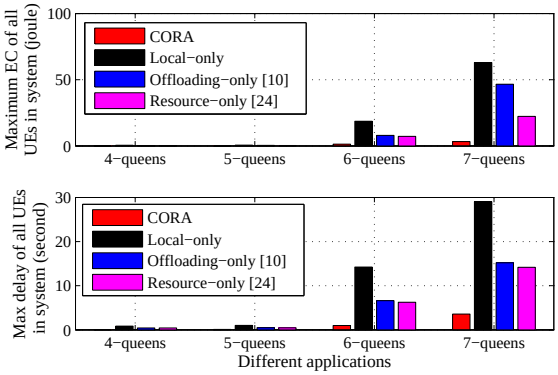


Fig. 9. The max EC and delay among all UEs vs. different applications

To verify the feasibility of CORA for different applications, in Fig. 9, we evaluate the performance achieved by CORA under different applications, and take $\max_{n\in\mathcal{N}} E_n$ and $\max_{n\in\mathcal{N}} T_n$ as the objective function, respectively. The applications are the m-queens puzzle, where $m = 4, 5, 6, 7$, respectively [1], [12]. The four applications possess the same size of data, i.e., $D_n = 200$ KB, $n \in \mathcal{N}$, but with the different size of processing density, where $App_n = 87.8,\ 263,\ 1760,\ 8250$, $n \in \mathcal{N}$, for 4-queens puzzle, 5-queens puzzle, 6-queens puzzle, and 7-queens puzzle, respectively. From the two sub-figures in Fig. 9, when m increases, the maximum EC and delay of all UEs in the system increase, which is the same for all the algorithms. However, CORA consumes the minimum energy or delay compared with other algorithms. What's more, the second sub-figure demonstrates that our algorithm can works well when only delay is considered as the optimization objective.

The impact of local processing capability f_n^{loc} on EC of the four algorithms is shown in Fig. 10. As the local processing capability grows stronger and stronger, the maximum consumed energy decreases gradually for all the methods as in Fig. 10, and CORA always consumes the least amount of energy.

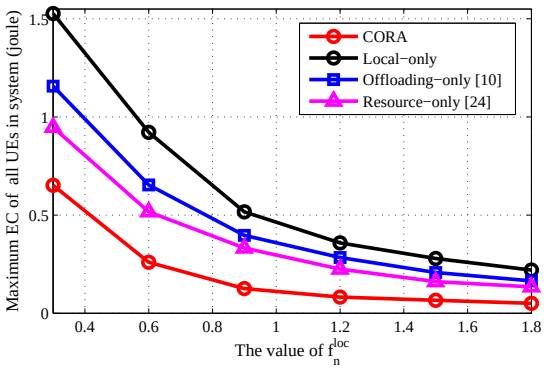


Fig. 10. The max EC among all UEs vs. f_n^{loc}.

Fig. 10 accords with our intuition that the weaker processing capability of a UE, the more benefit could be obtained by computation offloading, and vice-versa.

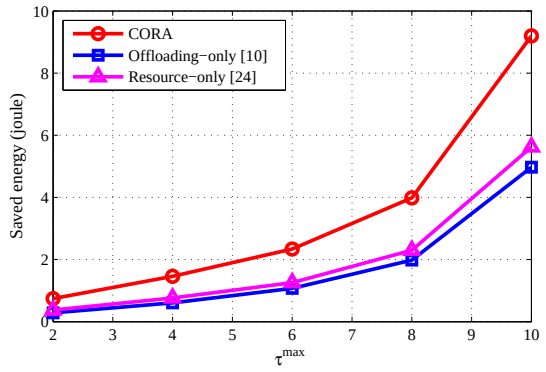


Fig. 11. The max saved energy among all UEs vs. τ_n^{max}.

In Fig. 11 we consider the impact of the delay constraint τ^{max} on the saved energy compared with Local-only method. It can be observed that the longer the delay constraint, the more saved energy. This is because a looser delay constraint will lead to more offloaded UEs, and consequently more conserved energy, which is the same for all the three algorithms. However, CORA conserves the most energy among all the three algorithms under any τ^{max}.

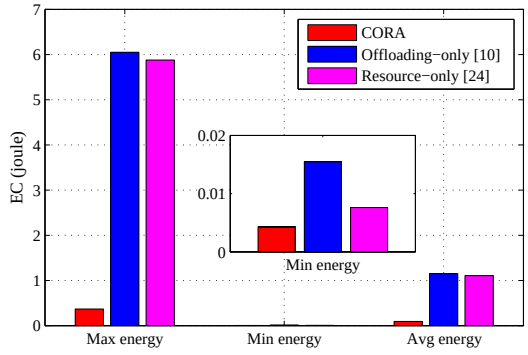


Fig. 12. Fairness comparison.

In Fig. 12, we compare the maximum, the minimum EC, and the average EC of all UEs. All parameters are set as the default values. We observe that there is a considerable difference in the maximum and minimum EC in offloading-only and resource-only schemes. However, the proposed algorithm can balance the EC among all the UEs, demonstrating CORA performs better in min-max fairness.

IX. Conclusions and Future Work

In this paper, we have investigated a min-max fairness based cost conservation problem in a mixed fog/cloud computing system by a joint optimization of offloading decision making and resource allocation. To address the NP-hard problem, we have proposed CORA algorithm, where SDR and random extracting are first adopted for offloading decision making. To solve the nested resource allocation problem in CORA, BCRA algorithm was proposed to solve computation resource allocation among all the fog-processing UEs. Employing fractional programming and Lagrangian dual decomposition, radio bandwidth and transmit power allocation was optimized among all the remote-processing UEs. Our simulation results verified the convergence of the proposed iterative algorithms, and indicated the performance gains of CORA in cost conversation and the increase in the number of beneficial UEs compared with other existing works.

Our future work are listed as follows:

- The case UEs may move dynamically in an offloading period is regarded as one of our future work.
- The long-term optimization where the offloading periods are time-coupled with each other and the wireless networks may changes dynamically during a long period of time will be regarded as one of our future work.
- We will extend the scenario from one fog node to multiple fog nodes when interference management and load balancing will be considered in our future work.
- The queue length and delay of UEs' requests will be considered in our future work.

Appendix

Appendix A. Proof of Proposition 1

Proof: We prove it from sufficiency and necessity. First, the sufficiency proof is as follows.

Assuming the optimal solution for (27) is $\{\mathbf{p}^{com\prime}, \mathbf{a}'\}$, and for any feasible solution $\{\mathbf{p}^{com}, \mathbf{a}\} \in \mathcal{F}$, we have

$$\max_{n\in\mathcal{N}_2}\left[D_n(\lambda_n^e p_n^{com}+\lambda_n^t)-V^* a_n B\log_2\left(1+\frac{p_n^{com} h_n}{a_n N_0 B}\right)\right]\geq 0,$$
$$\max_{n\in\mathcal{N}_2}\left[D_n(\lambda_n^e p_n^{com\prime}+\lambda_n^t)-V^* a_n' B\log_2\left(1+\frac{p_n^{com\prime} h_n}{a_n' N_0 B}\right)\right]=0. \quad (49)$$

From (49), we obtain

$$\max_{n\in\mathcal{N}_2}\left[\frac{D_n(\lambda_n^e p_n^{com}+\lambda_n^t)}{a_n B\log_2\left(1+\frac{p_n^{com}h_n}{a_n N_0 B}\right)}\right]\geq V^*,$$
$$\max_{n\in\mathcal{N}_2}\left[\frac{D_n(\lambda_n^e p_n^{com\prime}+\lambda_n^t)}{a_n{}' B\log_2\left(1+\frac{p_n^{com\prime}h_n}{a_n{}' N_0 B}\right)}\right]=V^*. \quad (50)$$

Hence, $\{\mathbf{p}^{com\prime},\mathbf{a}'\}$ is also the optimal solution of $(\mathcal{P}_{\mathbf{8}})$. This completes the sufficiency proof.

Proof of necessity: For any feasible solution $\{\mathbf{p}^{com},\mathbf{a}\}\in\mathcal{F}$, from $(\mathcal{P}_{\mathbf{8}})$, we have

$$\max_{n\in\mathcal{N}_2}\left[\frac{D_n(\lambda_n^e p_n^{com}+\lambda_n^t)}{a_n B\log_2\left(1+\frac{p_n^{com}h_n}{a_n N_0 B}\right)}\right]\geq V^*,$$
$$\max_{n\in\mathcal{N}_2}\left[\frac{D_n(\lambda_n^e p_n^{com*}+\lambda_n^t)}{a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)}\right]=V^*. \quad (51)$$

Rearranging (51) yields

$$\max_{n\in\mathcal{N}_2}\left[D_n(\lambda_n^e p_n^{com}+\lambda_n^t)-V^* a_n B\log_2\left(1+\frac{p_n^{com}h_n}{a_n N_0 B}\right)\right]\geq 0,$$
$$\max_{n\in\mathcal{N}_2}\left[D_n(\lambda_n^e p_n^{com*}+\lambda_n^t)-V^* a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)\right]=0. \quad (52)$$

Thus, $\{\mathbf{p}^{com*},\mathbf{a}^*\}$ is also the optimal solution of (27). The necessity proof is completed. ■

Appendix B. Proof of Proposition 2

Proof: When $f(x)$ is concave, then the perspective function $g(x,t)=tf(x/t)$ is concave, too [30]. Since $a_n B\log_2(1+\frac{p_n^{com}h_n}{a_n N_0 B})$ is the perspective function of concave function $\log_2(1+p_n^{com}h_n)$, it preserves concavity. On the other hand, the upper level set of concave function is convex [30], so (C17)−(C18) are convex. Moreover, $(\mathrm{C5}^+)-(\mathrm{C7}_2^+)$ are all linear constraints. So $(\mathcal{P}_{\mathbf{12}})$ is a convex optimization programming that minimize a convex function over a convex set. ■

Appendix C. Proof of Proposition 3

Proof: Observing the definition of $D(\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})$ of (32), we have

$$\begin{aligned}&D(\beta',\boldsymbol{\omega}',\boldsymbol{\mu}',\boldsymbol{\gamma}')\\&\geq\zeta_2+\beta'\left(\sum_{n\in\mathcal{N}_2}a_n^*-1\right)+\sum_{n\in\mathcal{N}_2}\omega_n'\left(p_n^{com*}-p_n^{max}\right)\\&+\sum_{n\in\mathcal{N}_2}\mu_n'\left[\frac{D_n}{\tau_n^{max}-v_n}-a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)\right]\\&+\sum_{n\in\mathcal{N}_2}\gamma_n'\left[D_n(\lambda_n^e p_n^{com*}+\lambda_n^t)-V a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)-\zeta_2\right].\end{aligned} \quad (53)$$

Rearranging (53), we have

$$\begin{aligned}&D(\beta',\boldsymbol{\omega}',\boldsymbol{\mu}',\boldsymbol{\gamma}')\geq D(\beta,\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\gamma})\\&+(\beta'-\beta)\left(\sum_{n\in\mathcal{N}_2}a_n^*-1\right)+\sum_{n\in\mathcal{N}_2}(\omega_n'-\omega_n)\left(p_n^{com*}-p_n^{max}\right)\\&+\sum_{n\in\mathcal{N}_2}(\mu_n'-\mu_n)\left[\frac{D_n}{\tau_n^{max}-v_n}-a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)\right]\\&+\sum_{n\in\mathcal{N}_2}(\gamma_n'-\gamma_n)\Big[D_n\left(\lambda_n^e p_n^{com*}+\lambda_n^t\right)\\&\qquad-V a_n^* B\log_2\left(1+\frac{p_n^{com*}h_n}{a_n^* N_0 B}\right)-\zeta_2\Big].\end{aligned} \quad (54)$$

Note that a subgradient ζ of a convex function $f(\cdot)$ is defined as: if $f(x)\geq f(y)+\zeta^T(x-y),\forall x,y$ holds. Thus, Proposition 3 holds. ■

REFERENCES

[1] J. Kwak, Y. Kim, J. Lee, et al., "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510-2523, 2015.

[2] E. Cuervo, A. Balasubramanian, D. Cho, et al., "MAUI: Making Smartphones Last Longer With Code Offload," in *Proc. ACM 8th Int. Conf. Mobile Syst., Appl., Services*, San Francisco, America, Jun 2010, pp. 49-62.

[3] W. Shi, J. Cao, Q. Zhang, et al., "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.

[4] R. Mahmud, and R, Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," *arXiv preprint arXiv*, vol. 1611, no. 05539, 2016.

[5] T. X. Tran, A. Hajisami, P. Pandey, et al., "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, vol. 55, no, 4, pp. 54-61, 2017.

[6] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Trans. Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89-103, 2015.

[7] H. Viswanathan, P. Pandey, and D. Pompili, "Maestro: Orchestrating Concurrent Application Workflows in Mobile Device Clouds," *IEEE International Conference on Autonomic Computing (ICAC)*, pp. 257-262, 2016.

[8] M. Satyanarayanan, P. Bahl, R. Caceres, et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE TPervasive Comput.*, vol. 8, no. 4, pp. 14-23, 2009.

[9] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590-3605, 2016.

[10] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974-983, Apr. 2015.

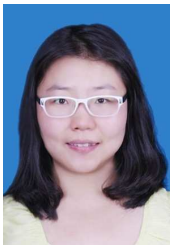
[11] M. H. Chen, M. Dong, B. Liang, et al., "Joint Offloading Decision and Resource Allocation for Mobile Cloud With Computing Access Point," in *Proc. IEEE ICASSP*, 2016, pp. 3516-3520.

[12] K. Liu, X. Zhang, and Z. Huang. "A Combinatorial Optimization for Energy-Efficient Mobile Cloud Offloading Over Cellular Networks," *IEEE Conference on Global Communications (GLOBECOM)*, pp. 1-6, 2016.

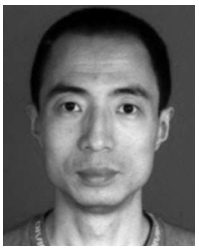
[13] X. Lyu, H. Tian, C. Sengul, et al., "Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435-3447, 2017.

[14] Y. H. Kao, B. Krishnamachari, M. R. Ra, et al., "Hermes: Latency Optimal Task Assignment for Resource-Constrained Mobile Computing," *IEEE Transactions on Mobile Computing*, 2017.

[15] H. Wu, W. Knottenbelt, K. Wolter, et al., "An Optimal Offloading Partitioning Algorithm in Mobile Cloud Computing," *Springer International Conference on Quantitative Evaluation of Systems*, pp. 311-328, 2016.
[16] T. Q. Dinh, J. Tang, Q. D. La, et al., "Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling," *IEEE Transactions on Communications*, 2017.
[17] Y. Wang, M. Sheng, X. Wang, et al., "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268-4282, 2016.
[18] O. Munoz, A. Pascual-Iserte, J. Vidal, et al., "Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738-4755, 2015.
[19] V. Cardellini, V. D. N. Person, Di Valerio V, et al., "A Game-Theoretic Approach to Computation Offloading in Mobile Cloud Computing," *Mathematical Programming*, vol. 157, no. 2, pp. 421-449, 2016.
[20] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81-93, 2015.
[21] M. V. Barbera, S. Kosta, A. Mei, et al., "To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 1285-1293.
[22] L. Yang, J. Cao, Y. Yuan, et al., "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 4, pp. 23-32, 2013.
[23] C. Wang, C. Liang, Yu F R, et al., "Computation Offloading and Resource Allocation in Wireless Cellular Networks with Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, 2017.
[24] Y. Mao, J. Zhang, S. H. Song, et al, "Power-Delay Tradeoff in Multi-User Mobile-Edge Computing Systems," *IEEE Conference on Global Communications Conference (GLOBECOM)*, pp. 1-6, 2016.
[25] X. Zhang, and F. Yang, "Joint Bandwidth and Power Allocation for Energy Efficiency Optimization over Heterogeneous LTE/WiFi Multi-Homing Networks," *IEEE Conference on Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 2017.
[26] M. Sheng, D. S. Zhai, X. Wang, et al., "Intelligent Energy and Traffic Coordination for Green Cellular Networks with Hybrid Energy Supplies," *IEEE Trans. Vehicular Technology*, 2016.
[27] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab Software for Disciplined Convex Programming," 2009. [Online]. Available: http://cvxr.com/cvx/
[28] A. M. C. So, J.W. Zhang, and Y. Ye, "On Approximating Complex Quadratic Optimization Problems via Semidefinite Programming Relaxations," in *Math. Program.*, vol.110, no.1, pp. 93-110, 2007.
[29] W. Dinkelbach, "On Nonlinear Fractional Programming," *Management science*, vol. 13, no. 7, pp. 492-498, 1967.
[30] S. Boyd and L. Vandenberghe, "Convex Optimization," Cambridge, U.K.: Cambridge Univ. Press, 2004.
[31] D.S. Zhai, M. Sheng, et al., "Rate and Energy Maximization in SC-MA Networks With Wireless Information and Power Transfer," *IIEEE Communications Letters*, vol. 20, no. 2, pp. 360-363, Feb. 2016.
[32] IEEE P802.11 Wireless LANs, "TGn Channel Models," IEEE 802.11-03/940r4, Tech. Rep., May 2004.
[33] J. Wang, J. Peng, Y. Wei, et al., "Adaptive Application Offloading Decision and Transmission Scheduling for Mobile Cloud Computing," *IEEE International Conference on Communications (ICC)*, 2016.

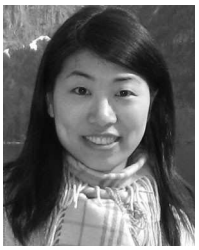


Jianbo Du received the B.S. degree and M.S. degree from Xi'an University of Posts and Telecommunications in 2007 and 2013, respectively. She is currently working towards the PH.D. degree in communication and information systems at Xidian University, Xian, Shaanxi, China. Her research interests include mobile edge computing, resource management, LTE, NOMA, and convex optimization, stochastic network optimization and heuristic algorithms and their applications in wireless communications.



Liqiang Zhao (M12) received the B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992, and the M.Sc. degree in communications and information systems and Ph.D. degree in information and communications engineering from Xidian University, Xian, China, in 2000 and 2003, respectively. From 1992 to 2005, he was a Senior Research Engineer with the 20th Research Institute, Chinese Electronics Technology Group Corporation, China, where his research focused on mobile communication systems and spread spectrum communications. From 2005 to 2007, he was an Associate Professor with the State Key Laboratory of Integrated Service Networks, Xidian University, where his research focused on WiMAX, WLAN, and wireless sensor networks. In June 2007, he was appointed as a Marie Curie Research Fellow with the Centre for Wireless Network Design, University of Bedfordshire, to conduct research in the GAWIND project funded under the EU FP6 HRM program. His activities focused on the area of automatic wireless broadband access network planning and optimization. Since June 2008, he has been with Xidian University, where he has been a Professor of information and communication engineering. His current research focuses on 5G, aerospace communications, and nanonetworks.

Jie Feng is currently pursuing the Ph.D. degree in Communication and Information System at Xidian University, Xian, China. Her current research interests include mobile edge computing, Device to Device communication, resource allocation and convex optimization and stochastic network optimization.



Xiaoli Chu (M06CSM15) received the B.Eng. degree in electronic and information engineering from Xian Jiao Tong University, Xian, China, in 2001, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2005. She is a Senior Lecturer with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K. From September 2005 to April 2012, she was with the Centre for Telecommunications Research, Kings College London. She has published more than 100 peer-reviewed journal and conference papers. She is the Lead Editor/author of the book Heterogeneous Cellular Networks: Theory, Simulation and Deployment (Cambridge University Press, 2013) and the book 4G Femtocells: Resource Allocation and Interference Management (Springer 2013).