Joint Data Compression and Computation Offloading in Hierarchical Fog-Cloud Systems

Ti Ti Nguyen, *Student Member, IEEE*, Vu Nguyen Ha, *Member, IEEE*, Long Bao Le, *Senior Member, IEEE*, and Robert Schober, *Fellow, IEEE*

***Abstract*— Data compression (DC) has the potential to significantly improve the computation offloading performance in hierarchical fog-cloud systems. However, it remains unknown how to optimally determine the compression ratio jointly with the computation offloading decisions and the resource allocation. This optimization problem is studied in this paper where we aim to minimize the maximum weighted energy and service delay cost (WEDC) of all users. First, we consider a scenario where DC is performed only at the mobile users. We prove that the optimal offloading decisions have a threshold structure. Moreover, a novel three-step approach employing convexification techniques is developed to optimize the compression ratios and the resource allocation. Then, we address the more general design where DC is performed at both the mobile users and the fog server. We propose three algorithms to overcome the strong coupling between the offloading decisions and the resource allocation. Numerical results show that the proposed optimal algorithm for DC at only the mobile users can reduce the WEDC by up to 65% compared to computation offloading strategies that do not leverage DC or use sub-optimal optimization approaches. The proposed algorithms with additional DC at the fog server lead to a further reduction of the WEDC.**

***Index Terms*— Fog computing, resource allocation, computation offloading, hierarchical fog/cloud, data compression, energy saving, latency, mixed integer non-linear programming.**

I. Introduction

CURRENTLY, mobile edge/cloud computing (MEC/MCC) technologies are considered as promising solutions for enhancing the mobile usability and prolonging the mobile battery life by offloading computation heavy applications to a remote fog/cloud server [1]–[3]. In an MCC system, enormous computing resources are available in the core network, but the limited backhaul capacity can induce significant delay for the underlying applications. In contrast, an MEC system, with computing resources deployed at the network edge in close proximity to the mobile devices, can enable computation offloading and meet demanding application requirements [4].

Hierarchical fog-cloud computing systems which leverage the advantages of both MCC and MEC can further enhance the system performance [5]–[9] where fog servers deployed at the network edge can operate collaboratively with the more powerful cloud servers to execute computation-intensive user applications. Specifically, when the users' applications require high computing power or low latency, their computation tasks can be offloaded and processed at the fog and/or remote cloud servers. However, the upsurge of mobile data and the constrained radio spectrum may result in significant delays in transferring offloaded data between the mobile users and the fog/cloud servers, which ultimately degrades the quality of service (QoS) [10]. To overcome this challenge, advanced data compression (DC) techniques can be leveraged to reduce the amount of incurred data (i.e., the input data of a user's application) [11], [12]. However, DC entails additional computations needed for the execution of the corresponding compression and decompression algorithms [13]. Therefore, an efficient joint design of DC, offloading decisions, and resource allocation is needed to take full advantage of DC while meeting all QoS requirements and other system constraints.

A. Related Works

Computation offloading design for MCC/MCE systems has been studied extensively in the literature, see recent surveys [14], [15] and the references therein. Most existing works consider two main performance metrics for their designs, namely energy-efficiency [16]–[19] and delay-efficiency [20]–[23]. Focusing on energy-efficiency, the authors of [16] develop partial offloading frameworks for multiuser MEC systems employing time division multiple access (TDMA) and frequency-division multiple access (FDMA). In [17], wireless power transfer is integrated into the computation offloading design. Moreover, different binary offloading frameworks are developed in [18], [19] where various branch-and-bound and heuristic algorithms are proposed to tackle the resulting mixed integer optimization problems.

Manuscript received March 19, 2019; revised July 29, 2019; accepted September 21, 2019. Date of publication October 4, 2019; date of current version January 8, 2020. This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA01. The associate editor coordinating the review of this article and approving it for publication was L. Galluccio. *(Corresponding author: Ti Ti Nguyen.)*

T. T. Nguyen and L. B. Le are with INRS, Université du Québec, Montréal, QC H5A1K6, Canada (e-mail: titi.nguyen@emt.inrs.ca; long.le@emt.inrs.ca).

V. N. Ha is with the École Polytechnique de Montréal, Montréal, QC H3T1J4, Canada (e-mail: vu.ha-nguyen@polymtl.ca).

R. Schober is with the Institute for Digital Communications, Friedrich-Alexander University Erlangen-Nuremberg, D-91058 Erlangen, Germany (e-mail: robert.schober@fau.de).

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

Digital Object Identifier 10.1109/TWC.2019.2944165

Considering computation offloading from the delay-efficiency point of view, an iterative heuristic algorithm to optimize the binary offloading decisions for minimization of the overall computation and transmission delay in a hierarchical fog-cloud system is proposed in [20]. The authors in [21] formulate the computation offloading and resource allocation problem as a student-project-allocation game with the objective to maximize the ratio between the average offloaded data rate and the offloading cost at the users. In [22], the authors study a binary computation offloading problem for maximization of the weighted sum computation rate. Then, they propose a coordinate descent based algorithm in which the offloading decision and time-sharing variables are iteratively updated until convergence. Considering partial computation offloading, the authors in [23] propose a framework for minimization of the weighted-sum latency of the mobile users via collaborative cloud and fog computing assuming a TDMA based resource sharing strategy.

Some recently proposed schemes for computation offloading consider both energy and delay efficiency aspects [7], [9], [24]. In particular, the work in [7] proposes a radio and computing resource allocation framework where the computational loads of the fog and cloud servers are determined and the trade-off between power consumption and service delay is investigated. Additionally, the authors of [24] jointly optimize the transmit power and offloading probability for minimization of the average weighted energy, delay, and payment cost. In [9], the authors study fair computation offloading design minimizing the maximum weighted cost of delay and energy consumption among all users in a hierarchical fog-cloud system. In this work, a two-stage algorithm is proposed where the offloading decisions are determined in the first stage using a semidefinite relaxation and probability rounding based method while the radio and computing resource allocation is determined in the second stage. However, references [7], [9], [16]–[22], [24] have not exploited DC for computation offloading.

There are few existing works that explore DC for computation offloading. Specifically, the authors of [10] propose an analytical framework to evaluate the outage performance of a hierarchical fog-cloud system. Moreover, the work in [13] considers DC for computation offloading for systems with a single server but assumes a fixed compression ratio (i.e., this parameter is not optimized). In general, the compression ratio should be optimized jointly with the computation offloading decisions and the resource allocation to achieve optimal system performance. However, the computational load incurred by compression/decompression is a non-linear function of the compression ratio, which makes this joint optimization problem very challenging.

B. Contributions and Organization of the Paper

To the best of our knowledge, the joint design of DC, computation offloading, and resource allocation for hierarchical fog-cloud systems has not been considered in the existing literature. The main contributions of this paper can be summarized as follows:

- We propose a non-linear computation model which can be fitted to accurately capture the computational load incurred by DC and decompression. In particular, the compression and decompression computational load as well as the quality of data recovery are modeled as functions of the compression ratio.
- For DC at only the mobile users, we formulate the fair joint design of the compression ratio, computation offloading, and resource allocation as a mixed-integer non-linear programming (MINLP) optimization problem. This problem formulation takes into account practical constraints on the maximum transmit power, wireless access bandwidth, backhaul capacity, and computing resources. We propose an optimal algorithm, referred to as Joint DC, Computation offloading, and Resource Allocation (JCORA) algorithm, which solves this challenging problem optimally. To develop this algorithm, we first prove that users incurring higher weighted energy and service delay cost (WEDC) when executing their application locally should have higher priority for offloading. Based on this result, the bisection search method is employed to optimally classify users into two user sets, namely the set of offloading users, and the set of remaining users, and JCORA globally optimizes the decision variables for both user sets.
- We then study a more general design where DC is performed at both the mobile users and the fog server (with different compression ratios) before the compressed data are transmitted over the wireless link and the backhaul link to the fog server and the cloud server, respectively. This enhanced design can lead to a significant performance gain when both the wireless access and the backhaul networks are congested. Three different solution approaches are proposed to solve this more general problem. In the first approach, we extend the design principle of the JCORA algorithm by employing the piecewise linear approximation (PLA) method to tackle the coupling of the optimization variables. In the remaining approaches, we utilize the Lagrangian method and solve the dual optimization problem. Specifically, in the second approach, referred to as One-dimensional λ-Search based Two-Stage (OSTS) algorithm, a one-dimensional search is employed to determine the optimal value of the Lagrangian multiplier, while in the third approach, referred to as Iterative λ-Update based Two-Stage (IUTS) algorithm, a low-complexity iterative sub-gradient projection technique is adopted to tackle the problem.
- Extensive numerical results are presented to evaluate the performance gains of the proposed designs in comparison with conventional strategies that do not employ DC. Moreover, our results confirm the excellent performance achievable by joint optimization of DC, computation offloading decisions, and resource allocation in a hierarchical fog-cloud system.

The remainder of this paper is organized as follows. Section II presents the system model, the computation and transmission energy models, and the problem formulation. Section III develops the proposed optimal algorithm for

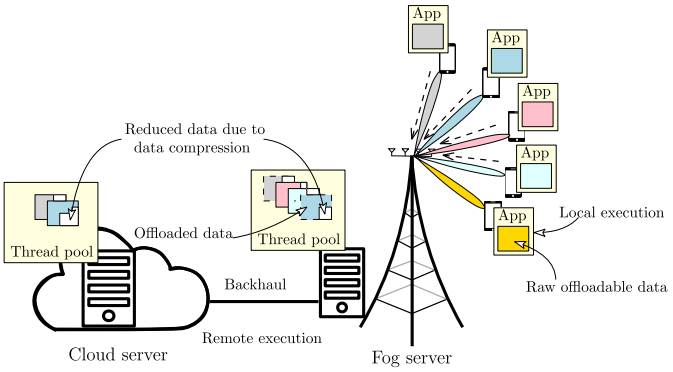

Fig. 1. DC and computation offloading in hierarchical fog-cloud systems.

the case when DC is performed only at the mobile users. Section IV provides the enhanced problem with DC also at the fog server and three methods for solving it. Section V evaluates the performance of the proposed algorithms. Finally, Section VI concludes this work.

II. System Model and Problem Formulation

A. System Model

We consider a hierarchical fog-cloud system consisting of K single-antenna mobile users, one cloud server, and one fog server co-located with a base station (BS) equipped with a large number of antennas. In this system, the BS communicates with the users through wireless links while a (wired) backhaul link is deployed between the BS co-located with the fog server and the cloud server as in Fig. 1. For convenience, we denote the set of users as $\mathcal{K}$. We assume that each user k needs to execute an application requiring c_k CPU cycles within an interval of $T_k^{\max}$ seconds, where $c_{k,0}$ CPU cycles must be executed locally at the mobile device and the remaining offloadable $c_{k,1}$ CPU cycles can be processed locally or offloaded and processed at the fog/cloud server for energy saving and delay improvement. Sequential processing of the unoffloadable and offloadable computing tasks is assumed in this paper. Let b_k^{in} be the number of bits representing the corresponding incurred data (i.e., programming states, input text/image/video) of the possibly-offloaded $c_{k,1}$ CPU cycles. To overcome the wireless transmission bottleneck caused by the capacity-limited wireless links between the users and the BS, DC is employed at the users for reducing the amount of data transferred to the fog server.

In particular, once $c_{k,1}$ CPU cycles are offloaded, user k first compresses the corresponding b_k^{in} bits down to $b_k^{\mathsf{out,u}}$ bits before sending them to the remote fog server. The ratio between b_k^{in} and $b_k^{\mathsf{out,u}}$ is called the compression ratio and is denoted as $\omega_k^{\mathsf{u}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,u}}$. Depending on the available fog computing resources, the offloaded computation task can be directly processed at the fog server or be further offloaded to the cloud server. The amount of data required to represent the computation outcome sent back to the users is usually much smaller than that incurred by offloading the task. Therefore, similar to [9], [16], [24], we do not consider the downlink transmission of the computation results in this paper.[1]

[1]The design in this paper can be extended to also include the downlink transmission of feedback data as in [25].

Remark 1: Running an application requires executing several unoffloadable sub-tasks that handle user interaction or access local I/O devices and cannot be executed remotely and other offloable sub-tasks that can be executed locally or remotely based on the employed offloading strategy [24], [26]. Practically, the workload corresponding to each sub-task of a specific application has to be pre-determined and remains unchanged according to the pre-programmed source code. Hence, the total workload of the offloadable components is typically fixed and cannot be optimized. In this work, we assume a binary offloading decision for all offloadable sub-tasks of each user. This corresponds to the practical scenario where all offloadable sub-tasks are strongly related such that they cannot be executed at different locations.

1) Data Compression Model: DC can be achieved by eliminating only statistical redundancy (i.e., lossless compression) or by also removing unnecessary information (i.e., lossy compression). To realize it, compression and decompression algorithms must be executed at the data source and destination, respectively, which induces additional computational load. To the best of our knowledge, in the literature, there is no theoretical model for the computational workload incurred by DC. Hence, we adopt a practical data-fitting approach to model the compression computational load, decompression computational load, and compression quality as non-linear functions of the compression ratio as follows:

$$c_k^{\mathsf{x,u}} = \gamma_{k,0}^{\mathsf{u}}\left[\gamma_{k,1}^{\mathsf{x,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{x,u}}}+\gamma_{k,3}^{\mathsf{x,u}}\right], \text{ for } \omega_k^{\mathsf{u}}\in[\omega_{k,1}^{\mathsf{u,min}},\omega_{k,1}^{\mathsf{u,max}}], \quad (1)$$

$$q_k^{\mathsf{qu,u}} = \gamma_{k,3}^{\mathsf{qu,u}}-\left[\gamma_{k,1}^{\mathsf{qu,u}}(\omega_k^{\mathsf{u}})^{\gamma_{k,2}^{\mathsf{qu,u}}}\right], \quad \text{for } \omega_k^{\mathsf{u}}\in[\omega_{k,1}^{\mathsf{u,min}},\omega_{k,1}^{\mathsf{u,max}}], \quad (2)$$

where 'x' = 'co' and 'de' stands for compression and decompression, respectively, $[\omega_{k,1}^{\mathsf{u,min}},\omega_{k,1}^{\mathsf{u,max}}]$ represents the possible range of ω_k^{u} and depends on the compression algorithm employed at user k, $c_k^{\mathsf{co,u}}$ and $c_k^{\mathsf{de,u}}$ denote the additional CPU cycles at source and destination needed for compression and decompression, respectively[2]; $q_k^{\mathsf{qu,u}}$ represents the perceived QoS (i.e., this parameter, which is only considered for lossy compression, measures the deviation between the true data and the decompressed data); $\gamma_{k,0}^{\mathsf{u}}$ is the maximum number of CPU cycles; $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i=1,2,3$, are constant parameters where $\gamma_{k,1}^{\mathsf{co/de/qu,u}}, \gamma_{k,3}^{\mathsf{co/de/qu,u}}\geq 0$. The values of the $\gamma_{k,i}^{\mathsf{co/de/qu,u}}, i=1,2,3$, employed in this paper are determined based on experimental data collected by running the compression algorithms GZIP, BZ2, and JPEG in Python 3.0.[3]

[2]Note that when the compression and decompression algorithms are executed at a fixed CPU clock speed, the computational load in CPU cycles is linearly proportional to the execution time.

[3]For validation, we collected three experimental data sets for three algorithms (GZIP, BZ2, or JPEG) by running each algorithm in Python 3.0 via a Linux terminal using Ubuntu 18.04.1 LTS on a computer equipped with CPU chipset Intel(R) core(TM) i7-4790 and 12 GB RAM. To keep the CPU clock speed almost constant, we turned off all other applications when executing the compression and decompression algorithms by using the *'cpupower tool'* in Linux. In each realization for each algorithm, we measured the execution time of running that algorithm with different compression ratios. Then, the experimental data sets are compiled from the average execution time for each compression ratio value over 1000 realizations of running each algorithm. This allowed us to estimate the normalized execution time, which is proportional to the normalized computational load.

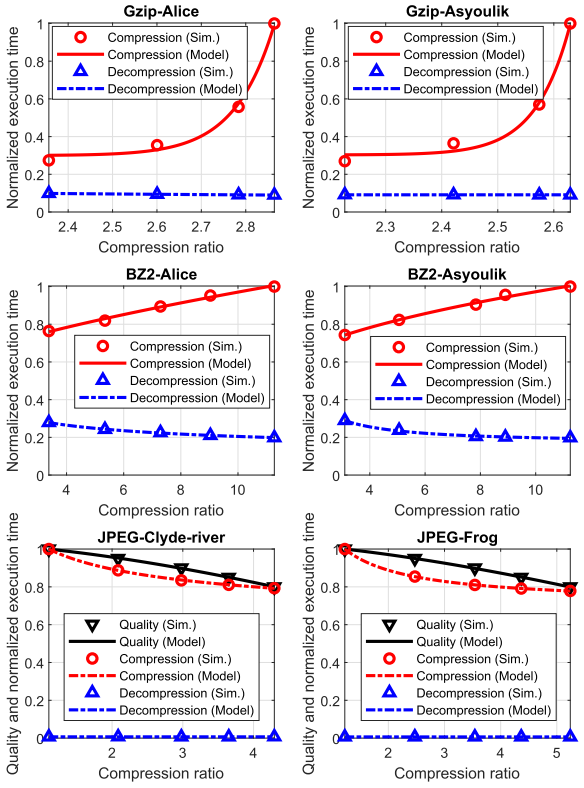

Fig. 2. Compression quality and normalized execution time.

The accuracy of the proposed model is validated in Fig. 2 which illustrates the relation between the normalized compression/decompression execution time and the compression ratio using the lossless algorithms GZIP and BZ2 for the benchmark text files *"alice.txt"* and *"asyoulik.txt"* from Canterbury Corpus [27], and the lossy algorithm 'JPEG' for images *"clyde-river.jpg"* and *"frog.jpg"* from the Canadian Museum of Nature [28], obtained by simulating and fitting the proposed model. Here, the normalized execution time is the ratio of the actual execution time and the maximum execution time over all values of the compression ratio. The figure shows that the curves obtained through fitting using the proposed model match the simulation results well.

Remark 2: A detailed comparison of the accuracy of the proposed compression computational load model and that of existing models is provided in Appendix G of our technical report [29].

2) Computing and Offloading Model: We now introduce the binary offloading decision variables s_k^{u}, s_k^{f}, and s_k^{c} for the computation task of user k, where $s_k^{\mathsf{u}}=1$, $s_k^{\mathsf{f}}=1$, and $s_k^{\mathsf{c}}=1$ denote the scenarios where the application is executed at the mobile device, the fog server, and the cloud server, respectively; and these variables are zero otherwise. Moreover, we assume that the $c_{k,1}$ CPU cycles can be executed at exactly one location, which implies $s_k^{\mathsf{u}}+s_k^{\mathsf{f}}+s_k^{\mathsf{c}}=1$. Then, the total computational load of user k at the mobile device, denoted as c_k^{u}, and at the fog server, denoted as c_k^{f}, are given as, respectively,

$$c_k^{\mathsf{u}}=c_{k,0}+s_k^{\mathsf{u}}c_{k,1}+(1-s_k^{\mathsf{u}})c_k^{\mathsf{co,u}} \text{ and } c_k^{\mathsf{f}}=s_k^{\mathsf{f}}\left(c_{k,1}+c_k^{\mathsf{de,u}}\right). \quad (3)$$

As the fog and cloud servers are generally connected to the power grid while the capacity of a mobile battery is limited, we will focus on the energy consumption of the users [9]. The local computation energy consumed by user k and the local computation time can be expressed, respectively, as $\xi_{1,k}^{\mathsf{u}}=\alpha_k {f_k^{\mathsf{u}}}^2 c_k^{\mathsf{u}}$ and $t_{1,k}^{\mathsf{u}}=c_k^{\mathsf{u}}/f_k^{\mathsf{u}}$, where f_k^{u} is the CPU clock speed of user k and α_k denotes the energy coefficient specified by the CPU model [30]. Let f_k^{f} denote the CPU clock speed used at the fog server to process $c_{k,1}$. Then, the computing time at the fog server is given by $t_{1,k}^{\mathsf{f}}=c_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. We assume that the computation task of each user is executed at the cloud server with a fixed delay of T^{c} seconds.[4]

3) Communication Model: In order to send the incurred data during the offloading process, we assume that zero-forcing beamforming is applied at the BS and the average uplink rate from user k to the BS (fog server) is expressed as $r_k=\rho_k \log_2(1+p_k\beta_{k,0})$, where p_k is the uplink transmit power per Hz of user k, ρ_k denotes the transmission bandwidth, and $\beta_{k,0}=M_0\beta_k/\sigma_{\mathsf{bs}}$. Here, β_k represents the large-scale fading coefficient, σ_{bs} is the noise power density (watts per Hz), and M_0 is the multiple-input multiple-output (MIMO) beamforming gain [32]. It is assumed that the number of antennas is sufficiently large so that M_0 is identical for all users. Then, the uplink transmission time and energy of user k can be computed, respectively, as $t_{2,k}^{\mathsf{u}}=(1-s_k^{\mathsf{u}})b_k^{\mathsf{out,u}}/r_k$ and $\xi_{2,k}^{\mathsf{u}}=\rho_k(p_k+p_{k,0})t_{2,k}^{\mathsf{u}}$, where $p_{k,0}$ denotes the circuit power consumption per Hz. For the data transmission between the fog server and the cloud server, a backhaul link with capacity D^{max} bps (bits per second) is assumed. Let d_k denote the backhaul rate allocated to user k. Then, the transmission time from the fog server to the cloud server is $t_{2,k}^{\mathsf{f}}=s_k^{\mathsf{c}}b_k^{\mathsf{out,u}}/d_k$.

B. Problem Formulation

Assume the users have to pay for their usage of the radio and computing resources at the fog/cloud servers. Then, the service cost of user k can be modeled as $\Theta_k=(1-s_k^{\mathsf{u}})(w^{\mathsf{BW}}\rho_k+w^{\mathsf{C}}c_{k,1})$, where w^{BW} is the price per 1 Hz of bandwidth for wireless data transmission, and w^{C} is the price paid to execute one CPU cycle at the fog/cloud servers. Assuming that a predetermined contract agreement specifies a maximum service cost Θ_k^{max} then $\Theta_k\le\Theta_k^{\mathsf{max}}$. This constraint can be rewritten equivalently as $(1-s_k^{\mathsf{u}})\rho_k\le\rho_k^{\mathsf{max}}=\frac{\Theta_k^{\mathsf{max}}-w^{\mathsf{C}}c_{k,1}}{w^{\mathsf{BW}}}$. Besides the constrained service cost, two important metrics for each user are the service latency and the consumed energy. Specifically, the total delay for completing the computation task of user k includes the computation delay of the mobile device, the average transmission delay of the mobile device, the computation delay of the fog server, the average transmission delay of the fog server over the backhaul link, and the computation delay of the cloud server. Therefore, the total delay is

[4]The delay time for the cloud server consists of two components: the execution time and the CPU set-up time. Due to the huge computing resources at the cloud server, the execution time is generally much smaller than the CPU set-up time [31], which is identical for all users.

given by

$$T_k = t^{\mathsf{u}}_{1,k} + t^{\mathsf{u}}_{2,k} + t^{\mathsf{f}}_{1,k} + t^{\mathsf{f}}_{2,k} + s^{\mathsf{c}}_k T^{\mathsf{c}}$$
$$= \frac{c_{k,0}+s^{\mathsf{u}}_k c_{k,1}+(1-s^{\mathsf{u}}_k)c^{\mathsf{co,u}}_k}{f^{\mathsf{u}}_k} + \frac{(1-s^{\mathsf{u}}_k)b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k \rho_k \log_2(1+p_k\beta_{k,0})}$$
$$+\frac{s^{\mathsf{f}}_k\left(c_{k,1}+c^{\mathsf{de,u}}_k\right)}{f^{\mathsf{f}}_k} + \frac{s^{\mathsf{c}}_k b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k d_k} + s^{\mathsf{c}}_k T^{\mathsf{c}}. \quad (4)$$

Since we assume massive MIMO transmission with zero-forcing beamforming, multiple mobile users can transmit their data to the fog server at the same time over the same frequency band. Unlike [23], we do not adopt the TDMA transmission protocol where the users are scheduled and have to wait for their turns to transmit their data in the uplink. For the considered massive MIMO system, time-based scheduling is not required since all users can transmit concurrently.

Furthermore, the overall energy consumed at user k for processing its task comprises the energy for local computation and for data transmission in the offloading case. Hence, the energy consumption of user k is given by

$$\xi_k = \xi^{\mathsf{u}}_{1,k} + \xi^{\mathsf{u}}_{2,k} = \alpha_k f^{\mathsf{u}2}_k (c_{k,0}+s^{\mathsf{u}}_k c_{k,1}+(1-s^{\mathsf{u}}_k)c^{\mathsf{co,u}}_k)$$
$$+\frac{(p_k+p_{k,0})(1-s^{\mathsf{u}}_k)b^{\mathsf{in}}_k}{\omega^{\mathsf{u}}_k \log_2(1+p_k\beta_{k,0})}. \quad (5)$$

Practically, all users want to save energy and enjoy low application execution latency. Hence, we adopt the WEDC as the objective function of each user k as follows:

$$\Xi_k = w^{\mathsf{T}}_k T_k + w^{\mathsf{E}}_k \xi_k,$$

where w^{T}_k and w^{E}_k represent the weights corresponding to the service latency and consumed energy, respectively. These weights can be pre-determined by the users to reflect their priorities or interests. The proposed design aims to minimize the WEDC function for each user while maintaining fairness among all users. Towards this end, we consider the following min-max optimization problem:

$$(\mathcal{P}_1) \quad \min_{\Omega_1} \max_k \ \Xi_k$$
$$\text{s.t} \quad (\text{C1}): f^{\mathsf{u}}_k \le F^{\mathsf{max}}_k, \quad \forall k, \ (\text{C6}): 0 \le \rho_k p_k \le P^{\mathsf{max}}_k, \ \forall k,$$
$$(\text{C2}): \sum\nolimits_k f^{\mathsf{f}}_k \le F^{\mathsf{f,max}}, \ (\text{C7}): 0 \le \rho_k \le \rho^{\mathsf{max}}_k, \quad \forall k,$$
$$(\text{C3}): s^{\mathsf{u}}_k, s^{\mathsf{f}}_k, s^{\mathsf{c}}_k \in \{0,1\}, \ \forall k, \ (\text{C8}): \sum\nolimits_k d_k \le D^{\mathsf{max}},$$
$$(\text{C4}): s^{\mathsf{u}}_k + s^{\mathsf{f}}_k + s^{\mathsf{c}}_k = 1, \ \forall k, \ (\text{C9}): T_k \le T^{\mathsf{max}}_k, \quad \forall k,$$
$$(\text{C5}): \omega^{\mathsf{u,min}}_k \le \omega^{\mathsf{u}}_k \le \omega^{\mathsf{u,max}}_k, \ \forall k,$$

where $\Omega_1 = \cup_{k\in\mathcal{K}}\Omega_{1,k}$, $\Omega_{1,k} = \{s^{\mathsf{u}}_k, s^{\mathsf{f}}_k, s^{\mathsf{c}}_k, \omega^{\mathsf{u}}_k, f^{\mathsf{u}}_k, f^{\mathsf{f}}_k, p_k, \rho_k, d_k\}$; F^{max}_k is the maximum CPU clock speed of user k, $F^{\mathsf{f,max}}$ is the maximum CPU clock speed of the fog server, P^{max}_k is the maximum transmit power of user k, $[\omega^{\mathsf{u,min}}_k, \omega^{\mathsf{u,max}}_k]$ denotes the feasible range of the compression ratio ω^{u}_k which can guarantee the required QoS of the recovered data. In particular, for lossless DC where the perceived QoS $q^{\mathsf{qu,u}}_k = 1$ for all ω^{u}_k, this feasible range is determined as $\omega^{\mathsf{u,min}}_k = \omega^{\mathsf{u,min}}_{k,1}$ and $\omega^{\mathsf{u,max}}_k = \omega^{\mathsf{u,max}}_{k,1}$. For lossy DC where the perceived QoS is required to be greater than $q^{\mathsf{qu,u,min}}_k$, this range is determined as $\omega^{\mathsf{u,min}}_k = \omega^{\mathsf{u,min}}_{k,1}$ and $\omega^{\mathsf{u,max}}_k = \min\left\{\omega^{\mathsf{u,max}}_{k,1}, \left((\gamma^{\mathsf{qu,u}}_{k,3} - q^{\mathsf{qu,u,min}}_k)/\gamma^{\mathsf{qu,u}}_{k,1}\right)^{1/\gamma^{\mathsf{qu,u}}_{k,2}}\right\}$. In this problem, (C1) and (C2) represent the constraints on the computing resources at the users and at the fog server, respectively, while the offloading decision constraints are characterized by (C3) and (C4). The constraints on the compression ratio are captured by (C5), while (C6) and (C7) impose constraints on the maximum user transmit power and the bandwidth, respectively. Finally, (C8) and (C9) are the constraints due to the limited backhaul capacity[5] and delay, respectively.

III. Optimal Algorithm Design for DC at only Mobile Users

A. Problem Transformation

To gain insight into its non-smooth min-max objective function, we recast $(\mathcal{P}_1)$ into the following equivalent problem:

$$(\mathcal{P}_2) \ \min_{\Omega_1 \cup \eta} \eta \ \text{ s.t } \ (\text{C0}): \ \Xi_k \le \eta, \quad \forall k, \ (\text{C1}) - (\text{C9}),$$

where η is an auxiliary variable. $(\mathcal{P}_2)$ is a MINLP problem which is difficult to solve due to the complex fractional and bilinear form of the transmission time and energy consumption, the logarithmic transmission rate function, and the mix of binary offloading decision variables and continuous variables. Conventional approaches usually decompose the problem into multiple sub-problems which optimize the offloading decision and the computing and radio resource allocation separately as in [9], [22] or relax the binary variables as in [18], [19]. These approaches can obtain only sub-optimal solutions.

To solve the problem optimally, we first study how to classify the users into two sets, namely, a *"locally executing user set"* which is the set of users executing their applications locally, and an *"offloading user set"* which is the set of users offloading their applications for processing at the fog/cloud server. This classification is important because, in all constraints of $(\mathcal{P}_2)$, the optimization variables corresponding to the locally executing users are independent from the optimization variables of the other users. Hence, the decisions for the locally executing users can be optimized by decomposing $(\mathcal{P}_2)$ into user independent sub-problems which can be solved separately. The optimal algorithm is developed based on the bisection search approach where in each search iteration, we perform: 1) user classification based on the current value of η using the results in Theorem 1 below; 2) feasibility verification for sub-problem $(\mathcal{P}_{\mathcal{B}})$ of $(\mathcal{P}_2)$ corresponding to the offloading user set $\mathcal{B}$; and 3) updates of lower and upper bounds on η according to the feasibility verification outcome. The detailed design is presented in the following.

[5]For practical scenarios, the development of sophisticated models for the communication delay over a shared backhaul link is a non-trivial task due to the complicated interactions between the routing algorithm and the other network functions (e.g. scheduling, buffering) [23]. This issue is outside the scope of this paper and left for future work. Similar to the existing work in [23], our current paper studies joint data compression and computation offloading in a hybrid fog-cloud computing system where we assume that a fixed backhaul communication capacity is allocated to each user. A fixed backhaul capacity allocation was also assumed in several recent works including [23], [33], [34].

Algorithm 1 Optimal Joint DC, Offloading, and Resource Allocation (JCORA)

1: **Initialize**: Compute $\eta_k^{\mathsf{lo}}, \forall k \in \mathcal{K}$ as in (8), choose ϵ, assign $\eta^{\mathsf{min}} = 0$, $\eta^{\mathsf{max}} = \max_k(\eta_k^{\mathsf{lo}})$, and set BOOL = *False*.
2: **while** $(\eta^{\mathsf{max}} - \eta^{\mathsf{min}} > \epsilon)$ & (BOOL = *False*) **do**
3: Assign $\eta = (\eta_{\mathsf{max}} + \eta_{\mathsf{min}})/2$, and then define sets $\mathcal{A} = \{k|\eta_k^{\mathsf{lo}} \leq \eta\}$ and $\mathcal{B} = \mathcal{K}/\mathcal{A}$.
4: Check feasibility of $(\mathcal{P}_\mathcal{B})$ as in Section III-C.
5: **if** $(\mathcal{P}_\mathcal{B})$ *is feasible* **then** $\eta^{\mathsf{max}} = \eta$, BOOL = *True*, **else** $\eta^{\mathsf{min}} = \eta$, BOOL = *False*, **end if**
6: **end while**

B. User Classification

Let $\mathcal{A}$ and $\mathcal{B}$ be the locally executing and the offloading user sets, respectively. We further define any pair of sets $(\mathcal{A}, \mathcal{B})$ satisfying $\mathcal{B} = \mathcal{K}\backslash\mathcal{A}$ as a user classification. By defining

$$\mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) = w_k^{\mathrm{E}}\alpha_k(f_k^{\mathsf{u}})^2 c_k + w_k^{\mathrm{T}} c_k / f_k^{\mathsf{u}}, \tag{6}$$

and $\Omega_\mathcal{B} = \cup_{k\in\mathcal{B}}\Omega_{1,k}$, then for a given classification $(\mathcal{A}, \mathcal{B})$, problem $(\mathcal{P}_2)$ can be tackled by solving two sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ for the users in sets $\mathcal{A}$ and $\mathcal{B}$, respectively, as follows:

$$(\mathcal{P}_\mathcal{A}) \min_{\{f_k^{\mathsf{u}}\}_{k\in\mathcal{A}},\eta} \eta \text{ s.t } (\mathrm{CA0}) : \mathcal{Q}_{k,0}(f_k^{\mathsf{u}}) \leq \eta, \forall k \in \mathcal{A},$$
$$(\mathrm{CA2}) : c_k/T_k^{\mathsf{max}} \leq f_k^{\mathsf{u}} \leq F_k^{\mathsf{max}}, \forall k \in \mathcal{A},$$
$$(\mathcal{P}_\mathcal{B}) \min_{\Omega_\mathcal{B},\eta} \eta \text{ s.t } (\mathrm{C0}) : \Xi_k \leq \eta, \forall k \in \mathcal{B},$$
$$(\mathrm{C1}) - (\mathrm{C9}), \quad \forall k \in \mathcal{B}.$$

Note that the variable set $\Omega_{1,k}$ corresponding to user k in $\mathcal{A}$ becomes $\{f_k^{\mathsf{u}}\}$ since we have $s_k^{\mathsf{u}} = 1$ and the other variables can be set equal to zero when user k executes its application locally. In such a scenario, Ξ_k can be simplified to $\mathcal{Q}_{k,0}(f_k^{\mathsf{u}})$. To attain more insight into the user classification, we now study the relationship between optimization sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ in the following lemma.

Lemma 1: We denote the optimal values of $(\mathcal{P}_2)$, $(\mathcal{P}_\mathcal{A})$, *and* $(\mathcal{P}_\mathcal{B})$ *as* $\eta^\star$, $\eta_\mathcal{A}^\star$, *and* $\eta_\mathcal{B}^\star$, *respectively. Then, we have*[6]

1) $\eta^\star \leq \max(\eta_\mathcal{A}^\star, \eta_\mathcal{B}^\star)$ *for any classification* $(\mathcal{A}, \mathcal{B})$.
2) *The merged optimal solutions of* $(\mathcal{P}_\mathcal{A})$ *and* $(\mathcal{P}_\mathcal{B})$ *are the optimal solution of* $(\mathcal{P}_2)$ *if*

$$\eta^\star = \max(\eta_\mathcal{A}^\star, \eta_\mathcal{B}^\star). \tag{7}$$

3) *If* $\mathcal{B}' \subset \mathcal{B}$, *then we have* $\eta_{\mathcal{B}'}^\star \leq \eta_\mathcal{B}^\star$.

Considering Lemma 1, instead of solving $(\mathcal{P}_2)$, we can equivalently solve the two sub-problems $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$. Moreover, a classification $(\mathcal{A}, \mathcal{B})$ is optimal if the condition in (7) holds. The optimal solution of $(\mathcal{P}_\mathcal{A})$ can be obtained as described in Proposition 1 while solving $(\mathcal{P}_\mathcal{B})$ requires a more complex approach which will be discussed in Section III-D.

[6]Due to the space constraint, the proof of Lemma 1 is given in the online technical report [29].

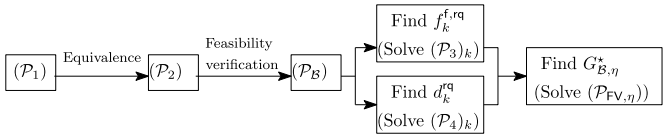

Fig. 3. Relationship between the (sub)problems when solving $(\mathcal{P}_1)$ by the JCORA algorithm.

Proposition 1: The optimal objective value of $(\mathcal{P}_\mathcal{A})$ *can be expressed as* $\eta_\mathcal{A}^\star = \max_{k\in\mathcal{A}}\eta_k^{\mathsf{lo}}$, *where* η_k^{lo} *is defined as*

$$\eta_k^{\mathsf{lo}} = \begin{cases} \mathcal{Q}_{k,0}(f_k^{\mathsf{u,sta}}), & \text{if } f_k^{\mathsf{u,sta}} \in [f_k^{\mathsf{u,min}}, F_k^{\mathsf{max}}] \\ \min\left(\mathcal{Q}_{k,0}(f_k^{\mathsf{u,min}}), \mathcal{Q}_{k,0}(F_k^{\mathsf{max}})\right), & \text{otherwise}, \end{cases} \tag{8}$$

where $f_k^{\mathsf{u,min}} = c_k/T_k^{\mathsf{max}}$ *and* $f_k^{\mathsf{u,sta}} = \sqrt[3]{w_k^{\mathrm{T}}/(2w_k^{\mathrm{E}}\alpha_k)}$.[7]

Based on the results in Lemma 1 and Proposition 1, the optimal user classification can be performed as described in the following theorem.

Theorem 1: If $\eta^\star$ *is the optimum objective value of problem* $(\mathcal{P}_2)$, *then an optimal classification,* $(\mathcal{A}^\star, \mathcal{B}^\star)$, *can be determined as* $\mathcal{A}^\star = \{k|\eta_k^{\mathsf{lo}} \leq \eta^\star\}$, *and* $\mathcal{B}^\star = \mathcal{K}\backslash\mathcal{A}^\star$.

Proof: The proof is given in Appendix A. ■

C. General Optimal Algorithm Design

The results in Theorem 1 are now employed to develop an optimal algorithm for solving $(\mathcal{P}_2)$ by iteratively solving $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$ and updating $(\mathcal{A}, \mathcal{B})$ until the optimal $(\mathcal{A}^\star, \mathcal{B}^\star)$ is obtained. The general optimal algorithm is presented in Algorithm 1. In this algorithm, we initially calculate η_k^{lo} for all users in $\mathcal{K}$ as in (8). Then, we employ the bisection search to find the optimum $\eta^\star$ where upper bound η^{max} and lower bound η^{min} are iteratively updated until the difference between them becomes sufficiently small, $(\mathcal{P}_\mathcal{B})$ is feasible, and the sets $\mathcal{A}$ and $\mathcal{B}$ do not change. At convergence, the optimal classification solution can be obtained by merging the solutions of $(\mathcal{P}_\mathcal{A})$ and $(\mathcal{P}_\mathcal{B})$. The optimal solution of $(\mathcal{P}_\mathcal{A})$ can be determined using Proposition 1 and the verification of the feasibility of $(\mathcal{P}_\mathcal{B})$ is addressed in the following. The relationship between the (sub)problems when solving $(\mathcal{P}_1)$ is illustrated in Fig. 3.

D. Feasibility Verification of $(\mathcal{P}_\mathcal{B})$

In order to verify the feasibility of $(\mathcal{P}_\mathcal{B})$, we consider the following problem

$$(\mathcal{P}_{\mathsf{FV},\eta}) \quad \min_{\Omega_\mathcal{B}} \sum_{k\in\mathcal{B}} f_k^{\mathsf{f}} \quad \text{s.t. } (\mathrm{C0}), (\mathrm{C1}), (\mathrm{C3}) - (\mathrm{C9}).$$

This problem minimizes the total required computing resource of the fog server subject to all constraints of $(\mathcal{P}_\mathcal{B})$ except (C2). Let $G_{\mathcal{B},\eta}^\star$ be the objective value of problem $(\mathcal{P}_{\mathsf{FV},\eta})$. Then, the feasibility of $(\mathcal{P}_\mathcal{B})$ can be verified by comparing $G_{\mathcal{B},\eta}^\star$ to the available fog computing resource $F^{\mathsf{f,max}}$. In particular,

[7]Due to the space constraint, the proof of Proposition 1 is given in the online technical report [29].

problem $(\mathcal{P}_{\mathcal{B}})$ is feasible if $G^{\star}_{\mathcal{B},\eta} \leq F^{\mathsf{f},\mathsf{max}}$. Otherwise, $(\mathcal{P}_{\mathcal{B}})$ is infeasible.

We propose to solve $(\mathcal{P}_{\mathsf{FV},\eta})$ as follows. First, recall that there are two possible scenarios for executing the tasks of the users in set $\mathcal{B}$ (referred to as modes): *Mode 1* - task execution at the fog server, i.e., $s^{\mathsf{f}}_k = 1$; *Mode 2* - task execution at the cloud server, i.e., $s^{\mathsf{c}}_k = 1$. In addition, the fog computing resources are only required by the users in *Mode 1* and the backhaul resources are only used by the users in *Mode 2*. Considering these two modes, a three-step solution approach is proposed to verify the feasibility of sub-problem $(\mathcal{P}_{\mathcal{B}})$ as follows. In Step 1, the minimum required fog computing resource of every user is determined by assuming that it is in *Mode 1*. This step is fulfilled by solving sub-problem $(\mathcal{P}_3)_k$ for every user k, see Section III-D1. In Step 2, the minimum required backhaul rate for each user is optimized by assuming that it is in *Mode 2*. This step can be accomplished by solving sub-problem $(\mathcal{P}_4)_k$ for every user k, see Section III-D2. In Step 3, using the results obtained in the two previous steps, problem $(\mathcal{P}_{\mathsf{FV},\eta})$ is equivalently transformed to a mode-mapping problem, see Section III-D3.

1) Step 1 - Minimum Fog Computing Resources for User $k \in \mathcal{B}$: If the application of user k is executed at the fog server, the minimum fog computing resource required for this application, denoted as $f^{\mathsf{f},\mathsf{rq}}_k$, can be optimized based on the following sub-problem:

$$(\mathcal{P}_3)_k \ \min_{\Omega_{2,k}} \ f^{\mathsf{f}}_k \ \text{s.t.} \ s^{\mathsf{f}}_k{=}1, (\mathrm{C0})_k, (\mathrm{C1})_k, (\mathrm{C5})_k{-}(\mathrm{C7})_k, (\mathrm{C9})_k,$$

where $\Omega_{2,k} = \{\omega^{\mathsf{u}}_k,\ f^{\mathsf{u}}_k,\ f^{\mathsf{f}}_k,\ p_k,\ \rho_k\}$, $(\mathrm{C0})_k$, $(\mathrm{C1})_k$, $(\mathrm{C5})_k - (\mathrm{C7})_k$, and $(\mathrm{C9})_k$ denote the respective constraints of user k corresponding to (C0), (C1), (C5) − (C7), and (C9). In sub-problem $(\mathcal{P}_3)_k$, the WEDC function Ξ_k consists of posynomials and other terms involving $\log(1 + p_k\beta_{k,0})$. We can convert Ξ_k into a convex function via logarithmic transformation as follows. When $s^{\mathsf{f}}_k = 1$, all variables in set $\Omega_{2,k}$ must be positive to satisfy constraints (C0) and (C9); therefore, we can employ the following variable transformations: $\tilde{\omega}^{\mathsf{u}}_k = \log(\omega^{\mathsf{u}}_k)$, $\tilde{f}^{\mathsf{u}}_k = \log(f^{\mathsf{u}}_k)$, $\tilde{f}^{\mathsf{f}}_k = \log(f^{\mathsf{f}}_k)$, $\tilde{p}_k = \log(p_k)$, and $\tilde{\rho}_k = \log(\rho_k)$. With these transformations, the objective function and all constraints of $(P_3)_k$ except $(\mathrm{C0})_k$ and $(\mathrm{C9})_k$ are converted into a linear form while the total delay and the WEDC in $(\mathrm{C9})_k$ and $(\mathrm{C0})_k$ can be rewritten, respectively, as $T_k = \frac{b^{\mathsf{in}}_k \mathrm{e}^{-\tilde{\omega}^{\mathsf{u}}_k - \tilde{\rho}_k}}{\log\left(1+\beta_{k,0}\mathrm{e}^{\tilde{\rho}_k}\right)} + \mathcal{Q}_{k,1}$, and $\Xi_k = \frac{w^{\mathrm{E}}_k b^{\mathsf{in}}_k \left[\mathrm{e}^{\tilde{\rho}_k - \tilde{\omega}^{\mathsf{u}}_k} + p_{k,0}\mathrm{e}^{-\tilde{\omega}^{\mathsf{u}}_k}\right]}{\log\left(1+\beta_{k,0}\mathrm{e}^{\tilde{\rho}_k}\right)} + w^{\mathrm{E}}_k \alpha_k \mathcal{Q}_{k,2} + w^{\mathrm{T}}_k T_k$, where $\mathcal{Q}_{k,1} = \left(c_{k,0}+\gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{co}}_{k,3}\right)\mathrm{e}^{-\tilde{f}^{\mathsf{u}}_k} + \gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{co}}_{k,1}\mathrm{e}^{\left(-\tilde{f}^{\mathsf{u}}_k + \gamma^{\mathsf{co}}_{k,2}\tilde{\omega}^{\mathsf{u}}_k\right)} + \left(c_{k,1}+\gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{de}}_{k,3}\right)\mathrm{e}^{-\tilde{f}^{\mathsf{f}}_k} + \gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{de}}_{k,1}\mathrm{e}^{\left(-\tilde{f}^{\mathsf{f}}_k + \gamma^{\mathsf{de}}_{k,2}\tilde{\omega}^{\mathsf{u}}_k\right)}$ and $\mathcal{Q}_{k,2} = \left(c_{k,0}+\gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{co}}_{k,3}\right)\mathrm{e}^{2\tilde{f}^{\mathsf{u}}_k} + \gamma^{\mathsf{u}}_{k,0}\gamma^{\mathsf{co}}_{k,1}\mathrm{e}^{\left(2\tilde{f}^{\mathsf{u}}_k + \gamma^{\mathsf{co}}_{k,2}\tilde{\omega}^{\mathsf{u}}_k\right)}$. The convexity of $(P_3)_k$ is formally stated in the following proposition.

Proposition 2: Sub-problem $(\mathcal{P}_3)_k$ is convex with respect to set $\tilde{\Omega}_{2,k} \cup \tilde{l}_k$, where $\tilde{l}_k = \tilde{\omega}^{\mathsf{u}}_k + \tilde{\rho}_k$ and $\tilde{\Omega}_{2,k} = \{\tilde{\omega}^{\mathsf{u}}_k, \tilde{f}^{\mathsf{u}}_k, \tilde{f}^{\mathsf{f}}_k, \tilde{p}_k, \tilde{\rho}_k\}$.

Proof: The proof is given in Appendix B. ∎

Based on Proposition 2, we can apply the interior point method to find the optimal solution $\tilde{\Omega}^{\star}_{2,k} = \{\tilde{\omega}^{\mathsf{u}\star}_k,\ \tilde{f}^{\mathsf{u}\star}_k,\ \tilde{f}^{\mathsf{f}\star}_k,\ \tilde{p}^{\star}_k,\ \tilde{\rho}^{\star}_k\}$ of $(\mathcal{P}_3)_k$ [35]. The original optimal solution $\Omega^{\star}_{2,k} = \{\omega^{\mathsf{u}\star}_k,\ f^{\mathsf{u}\star}_k,\ f^{\mathsf{f}\star}_k,\ p^{\star}_k,\ \rho^{\star}_k\}$ can then be obtained from $\tilde{\Omega}^{\star}_{2,k}$. If $(\mathcal{P}_3)_k$ is infeasible, we set $s^{\mathsf{f}}_k = 0$. It is noted that $f^{\mathsf{f}\star}_k$ is also the value of $f^{\mathsf{f},\mathsf{rq}}_k$.

2) Step 2 - Minimum Allocated Backhaul Resource for User $k \in \mathcal{B}$: If the application of user k is executed at the cloud server, the minimum backhaul capacity for transferring its application to the cloud server, denoted as d^{rq}_k, can be determined by solving the following sub-problem:

$$(\mathcal{P}_4)_k \quad \min_{\Omega_{2,k}\cup d_k \setminus f^{\mathsf{f}}_k} d_k \quad \text{s.t } s^{\mathsf{c}}_k{=}1, (\mathrm{C0})_k, (\mathrm{C1})_k, (\mathrm{C5})_k{-}(\mathrm{C7})_k, (\mathrm{C9})_k.$$

Similar to $(\mathcal{P}_3)_k$, $(\mathcal{P}_4)_k$ can be converted to a convex problem via logarithmic transformation; thus, we can find the optimal point d^{rq}_k. If $(\mathcal{P}_4)_k$ is infeasible, we set $s^{\mathsf{c}}_k = 0$.

3) Step 3 - Feasibility Verification: With the obtained values $f^{\mathsf{f},\mathsf{rq}}_k$ and d^{rq}_k, problem $(\mathcal{P}_{\mathsf{FV},\eta})$ can be transformed to

$$(\mathcal{P}_{\mathsf{FV},\eta}) \ \min_{\Omega_3} \ \mathcal{G}_{\mathcal{B},\eta}(\Omega_3) = \sum\nolimits_{k\in\mathcal{B}} (1 - s^{\mathsf{c}}_k) f^{\mathsf{f},\mathsf{rq}}_k \quad \text{s.t } (\mathrm{C3}, 4, 8) : \sum\nolimits_{k\in\mathcal{B}} s^{\mathsf{c}}_k d^{\mathsf{rq}}_k \leq D^{\mathsf{max}}, \ s^{\mathsf{c}}_k \in \{0,1\},$$

where $\Omega_3 = \{s^{\mathsf{c}}_k | k \in \mathcal{B}\}$ for a given η. In fact, $(\mathcal{P}_{\mathsf{FV},\eta})$ is a *"0-1 knapsack"* problem [36], which can be solved optimally and effectively using the CVX solver. If $G^{\star}_{\mathcal{B},\eta} \leq F^{\mathsf{f},\mathsf{max}}$, combining the set of all solutions of the $(\mathcal{P}_3)_k$'s, $(\mathcal{P}_4)_k$'s, and $(\mathcal{P}_{\mathsf{FV},\eta})$ yields a feasible solution of $(\mathcal{P}_{\mathcal{B}})$ for this value of η. Hence, $(\mathcal{P}_{\mathcal{B}})$ is feasible in such scenario. The feasibility verification of $(\mathcal{P}_{\mathcal{B}})$ is summarized in Algorithm 2.

Algorithm 2 Feasibility Verification of $(\mathcal{P}_{\mathcal{B}})$

1: Solve $(\mathcal{P}_3)_k$ to find $f^{\mathsf{f},\mathsf{rq}}_k, \forall k \in \mathcal{B}$, as in Section III-D1.
2: Solve $(\mathcal{P}_4)_k$ to find $d^{\mathsf{rq}}_k, \forall k \in \mathcal{B}$, as in Section III-D2.
3: **if** $\exists k$ such that $s^{\mathsf{f}}_k + s^{\mathsf{c}}_k = 0$ **then** Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible
4: **else** Solve $(\mathcal{P}_{\mathsf{FV},\eta})$ to find $G^{\star}_{\mathcal{B},\eta}$, as in Section III-D3.
5: **if** $G^{\star}_{\mathcal{B},\eta} < F^{\mathsf{f},\mathsf{max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}})$ is infeasible **end if**
6: **end if**

E. Optimal JCORA Algorithm to Solve $(\mathcal{P}_2)$

Based on the results presented in the previous sections, the solution of $(\mathcal{P}_2)$ can be found by employing Algorithm 1 and the $(\mathcal{P}_{\mathcal{B}})$ feasibility verification presented in Algorithm 2. The optimality of the obtained solution is formally stated in the following theorem.

Theorem 2: The integration of Algorithm 2 into Algorithm 1 yields the global optimum of MINLP $(\mathcal{P}_2)$.

Proof: Algorithm 2 verifies the feasibility of $(\mathcal{P}_{\mathcal{B}})$ for any given value of $\eta_{\mathcal{B}} = \eta$. Therefore, if Algorithm 1 employs Algorithm 2, $(\mathcal{P}_2)$ is solved optimally. Note that after convergence, the optimal variables are given by the optimal solution of $(\mathcal{P}_3)_k$ if $s^{\mathsf{f}}_k = 1$ or $(\mathcal{P}_4)_k$ if $s^{\mathsf{c}}_k = 1$ where the values of the s^{f}_k's and s^{c}_k's are the outcomes of $(\mathcal{P}_{\mathsf{FV},\eta})$. ∎

F. Complexity Analysis

We analyze the computational complexity of the JCORA algorithm (Algorithm 2 is integrated into Algorithm 1) in terms of the required number of arithmetic operations. In Algorithm 1, the while-loop for the bisection search of η requires $\log_2(\frac{\eta^{\max}-\eta^{\min}}{\epsilon})$ iterations. To verify the feasibility of $(\mathcal{P}_{\mathcal{B}})$ for a given η, the convex problems $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ can be solved by using the interior point method with complexity $\mathcal{O}(m_1^{1/2}(m_1+m_2)m_2^2)$, where m_1 is the number of equality constraints, m_2 represents the number of variables [37], and $\mathcal{O}$ denotes the big-O notation. It can be verified that $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ have the same complexity. On the other hand, the knapsack problem $(\mathcal{P}_{\mathsf{FV},\eta})$ for $|\mathcal{B}|$ users can be solved by Algorithm 2 in pseudo-polynomial time with complexity $\mathcal{O}(\nu_1|\mathcal{B}|)$, where ν_1 is determined by the coefficients in $(\mathcal{P}_{\mathsf{FV},\eta})$ [36]. Moreover, $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ can be solved independently for all users $k \in \mathcal{B}$; therefore, the complexity of each bisection search step can be expressed as $|\mathcal{B}|\mathcal{O}((\mathcal{P}_3)_k)+|\mathcal{B}|\mathcal{O}((\mathcal{P}_4)_k)+\mathcal{O}(\mathcal{P}_{\mathsf{FV},\eta}) = \mathcal{O}(\nu_2|\mathcal{B}|)$, where $\nu_2 = \nu_1+2m_1^{1/2}(m_1+m_2)m_2^2$. Consequently, the overall complexity of the JCORA algorithm is $\mathcal{O}(\log_2(\frac{\eta^{\max}-\eta^{\min}}{\epsilon})\nu_2\ K)$, i.e., $|\mathcal{B}| \le K$.

IV. DC at Both Mobile Users and Fog Server

We now consider the more general case where the fog server also performs DC before transmitting the compressed data over the backhaul link to the cloud server. This design option can further enhance the performance for systems with a congested backhaul link. The backhaul compression ratio is defined as $\omega_k^{\mathsf{f}} = b_k^{\mathsf{in}}/b_k^{\mathsf{out,f}}$ where $b_k^{\mathsf{out,f}}$ stands for the number of bits transmitted over the backhaul link. Note that if $b_k^{\mathsf{out,f}} = b_k^{\mathsf{out,u}}$, then no DC is employed at the fog server, which corresponds to the design in Section III. Hence, *Mode 2* in Section III-D1 is equivalent to the scenario that the task is executed at the cloud server without DC at the fog server. However, the fog server can re-compress the data before transmitting it to the cloud server for processing, which is referred to as *Mode 3* in the following. Denote s_k^{m} as the binary variable indicating whether or not DC is performed at the fog server for user k ($s_k^{\mathsf{m}} = 1$ for DC, and $s_k^{\mathsf{m}} = 0$, otherwise). Then, we have $s_k^{\mathsf{f}} = 1$ if user k is in *Mode 1*; $s_k^{\mathsf{c}} = 1$ if user k is in *Mode 2*; $s_k^{\mathsf{m}} = 1$ if user k is in *Mode 3*. In this general case, constraints (C3) and (C4) can be rewritten as ($\check{\text{C}}$3): $s_k^{\mathsf{u}}, s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}} \in \{0,1\}, \forall k \in \mathcal{K}$ and ($\check{\text{C}}$4): $s_k^{\mathsf{u}} + s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + s_k^{\mathsf{m}} = 1, \forall k \in \mathcal{K}$.

Then, the computational load for compression and the output data corresponding to *Mode 3* can be modeled as $c_k^{\mathsf{co,f}} = \gamma_{k,0}^{\mathsf{f}}\big[\gamma_{k,1}^{\mathsf{co,f}}(\omega_k^{\mathsf{f}})^{\gamma_{k,2}^{\mathsf{co,f}}} + \gamma_{k,3}^{\mathsf{co,f}}\big]$ and $b_k^{\mathsf{out,f}} = b_k^{\mathsf{in}}/\omega_k^{\mathsf{f}}$, respectively, where $\gamma_{k,0}^{\mathsf{f}}, \gamma_{k,1}^{\mathsf{co,f}}, \gamma_{k,3}^{\mathsf{co,f}} \in \mathbb{R}_+$ are positive numbers. Here, we have additional constraints for the compression ratio at the fog server as ($\check{\text{C}}$10): $\omega_k^{\mathsf{f}} \in [\omega_k^{\mathsf{f,min}}, \omega_k^{\mathsf{f,max}}],\ \forall k \in \mathcal{K}$. Then, the total computational load for user k at the fog server becomes $\check{c}_k^{\mathsf{f}} = s_k^{\mathsf{f}}\big(c_{k,1} + c_k^{\mathsf{de,u}}\big) + s_k^{\mathsf{m}}(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})$, and the computing time at the fog server is $\check{t}_{1,k}^{\mathsf{f}} = \check{c}_k^{\mathsf{f}}/f_k^{\mathsf{f}}$. Moreover, the transmission time incurred by offloading the data of user k from the fog server to the cloud server can be rewritten as $\check{t}_{2,k}^{\mathsf{f}} = \big(s_k^{\mathsf{f}} b_k^{\mathsf{out,u}} + s_k^{\mathsf{m}} b_k^{\mathsf{out,f}}\big)/d_k$. Then, the total delay for completing the computation task of user k is given by $\check{T}_k = t_{1,k}^{\mathsf{u}} + t_{2,k}^{\mathsf{u}} + \check{t}_{1,k}^{\mathsf{f}} + \check{t}_{2,k}^{\mathsf{f}} + (s_k^{\mathsf{c}} + s_k^{\mathsf{m}})T^{\mathsf{c}}$, and the WEDC becomes $\check{\Xi}_k = w_k^{\mathsf{T}}\check{T}_k + w_k^{\mathsf{E}}\xi_k$. Then, constraint (C9) is rewritten as ($\check{\text{C}}$9): $\check{T}_k \le T_k^{\max}$.

With the additional variables s_k^{m} and $\omega_k^{\mathsf{f}}, \forall k \in \mathcal{B}$, the extended versions of problems $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ can be stated, respectively, as

$$(\mathcal{P}_1^{\mathsf{ext}}) \quad \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\}} \max_k\ \check{\Xi}_k$$
$$\text{s.t } (\text{C1}), (\text{C2}), (\text{C5})-(\text{C8}), (\check{\text{C}}3), (\check{\text{C}}4), (\check{\text{C}}9), (\check{\text{C}}10).$$
$$(\mathcal{P}_2^{\mathsf{ext}}) \quad \min_{\Omega_1 \cup_k \{s_k^{\mathsf{m}}, \omega_k^{\mathsf{f}}\} \cup \eta}\ \eta \text{ s.t } (\check{\text{C}}0) : \check{\Xi}_k \le \eta,$$
$$(\text{C1}), (\text{C2}), (\text{C5})-(\text{C8}), (\check{\text{C}}3), (\check{\text{C}}4), (\check{\text{C}}9), (\check{\text{C}}10).$$

The main challenge for solving the extended problem in comparison to the original one comes from the users in *Mode 3*. These users require both fog computing and backhaul resources. To solve the extended problem, we employ the general solution approach presented in Section III but modify the feasibility verification for $(\mathcal{P}_{\mathcal{B}})$. In particular, Algorithm 1 is used to determine sets $\mathcal{A}$ and $\mathcal{B}$ for a given η and we update η using the bisection search method. The results in Theorem 1 are still applicable for the extended problem. In the following, we propose several techniques for dealing with *Mode 3* and verify the feasibility of user classification for a given η in *Step 4* of Algorithm 1.

For a given η, $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is obtained by adding ($\check{\text{C}}$10) to $(\mathcal{P}_{\mathcal{B}})$ and replacing Ξ_k and T_k by $\check{\Xi}_k$ and $\check{T}_k$, respectively. To verify the feasibility of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$, a similar three-step solution approach as for $(\mathcal{P}_{\mathcal{B}})$ is employed. In Steps 1 and 2, $f_k^{\mathsf{f,rq}}$ and d_k^{rq} which correspond to the users in *Mode 1* and *2* are optimized by solving $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$ as in Sections III-D1 and III-D2, respectively. In Step 3, we first investigate the network resources required by the users in *Mode 3*, modify problem $(\mathcal{P}_{\mathsf{FV},\eta})$ to adapt it to the extended problem, and solve that problem to verify the feasibility. Three different methods for this extended problem will be proposed as follows.

In the first approach, we represent $f_k^{\mathsf{f,rq}}$ of user k in *Mode 3* as a function of d_k by employing a piece-wise linear approximation (PLA) method. Based on this approximation, we transform $(\mathcal{P}_{\mathsf{FV},\eta})$ into a standard mixed-integer linear programming (MILP) problem, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$, which can be solved effectively by using the CVX solver. In the other two approaches, we directly deal with the modified problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ without approximating $f_k^{\mathsf{f,rq}}$ of user k in *Mode 3*. To cope with this challenging MINLP problem, we first reduce the optimization variable set by exploiting some useful relations among the variables. Then, two algorithms are proposed to solve the resulting problem for the remaining variables. One algorithm is based on a one-dimensional search for the Lagrangian multiplier, see Section IV-B1, while the other algorithm iteratively updates the Lagrangian multiplier, see Section IV-B2. The relationship between the (sub)problems when solving $(\mathcal{P}_1^{\mathsf{ext}})$ is illustrated in Fig. 4.

A. Piece-wise Linear Approximation Based Algorithm (PLA)

After determining the minimum computing and backhaul resources, $f_k^{\mathsf{f,rq}}$ and d_k^{rq}, required in *Modes 1* and *2*,

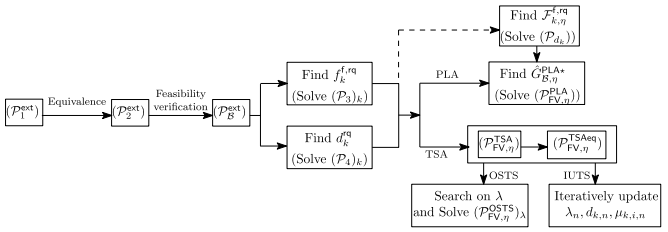

Fig. 4. Relationship between the (sub)problems when solving $(\mathcal{P}_1^{\mathsf{ext}})$.

respectively, one can set $d_k \in (0, d_k^{\mathsf{rq}})$ for the users in *Mode 3*. We now study the relationship between f_k^{f} and d_k in *Mode 3* where user k demands both fog computing resources for re-compression and backhaul capacity resources. Towards this end, we determine the required fog computing resources for a given $d_k \in (0, d_k^{\mathsf{rq}})$ by solving the following problem:

$$(\mathcal{P}_{d_k}) \quad \min_{\Omega_{2,k}\cup\{\omega_k^{\mathsf{f}}\}} \ f_k^{\mathsf{f}}$$
$$\text{s.t } s_k^{\mathsf{m}}{=}1, (\check{\mathrm{C}}0)_k, (\mathrm{C}1)_k, (\mathrm{C}5)_k - (\mathrm{C}7)_k, (\check{\mathrm{C}}9)_k, (\check{\mathrm{C}}10)_k.$$

Let $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ be the optimal solution of this problem, which can be obtained by employing the logarithmic transformations described in Section III-D1. However, finding a closed-form expression for $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ is not tractable. Hence, we propose to employ the *"Piece-wise Linear Approximation"* (PLA) method to divide the original domain into multiple small segments such that $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ can be approximated by a linear function in each segment. Suppose that the interval $[\epsilon_{\mathsf{d}}, d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}}]$ is divided into L segments of equal size, where ϵ_{d} is a very small number compared to d_k^{rq}, e.g, $\epsilon_{\mathsf{d}} = 1$. Specifically, the l^{th} segment corresponds to interval $[d_{k,l}, d_{k,l+1}]$, where $d_{k,l}{=}(d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L$ is a point such that $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ and the value of the approximated function at this point are equal. Then, we can approximate $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_k)$ as $\hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}(V_k, U_k){=}\sum_{l=0}^{L-1}(v_{k,l}A_{k,l}{+}u_{k,l}B_{k,l})$, where $A_{k,l} = \left(\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l+1}) - \mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})\right)/(d_{k,l+1} - d_{k,l})$, $B_{k,l} = \mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l}) - A_{k,l}d_{k,l}$, $V_k{=}\{v_{k,l}, l = 0, 1, \ldots, L-1\}$, $U_k{=}\{u_{k,l}, l = 0, 1, \ldots, L-1\}$, and continuous variable $v_{k,l}$ and binary variable $u_{k,l}$ satisfy the following constraints:

$$s_k^{\mathsf{m}} = \sum_{l=0}^{L-1} u_{k,l} \le 1, \quad \forall k{\in}\mathcal{B}, \tag{9}$$

$$u_{k,l}d_{k,l} \le v_{k,l} {\le} u_{k,l+1}d_{k,l+1}, \ \forall k{\in}\mathcal{B}, l{=}0, 1, \ldots, L{-}1. \tag{10}$$

Then, the allocated backhaul resources due to user k in *Mode 3* are rewritten as $s_k^{\mathsf{m}}d_k{=}\sum_{l=0}^{L-1}v_{k,l}$. Therefore, problem $(\mathcal{P}_{\mathsf{FV},\eta})$, which is used to determine the minimum total required fog computing resources for all users, is modified in this extended case as follows:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}}) \quad \min_{\check{\Omega}_3} \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{PLA}}\left(\check{\Omega}_3\right) = \sum\nolimits_{k\in\mathcal{B}}\left(s_k^{\mathsf{f}}f_k^{\mathsf{f,rq}} + \hat{\mathcal{F}}_{k,\eta}^{\mathsf{f,rq}}(V_k, U_k)\right)$$
$$\text{s.t. } (\check{\mathrm{C}}3)^{\mathsf{PLA}} : s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, u_{k,l} \in \{0,1\}, \quad \forall k, l;$$
$$(\check{\mathrm{C}}4)^{\mathsf{PLA}} : s_k^{\mathsf{f}} + s_k^{\mathsf{c}} + \sum\nolimits_{l=0}^{L-1} u_{k,l} = 1;$$
$$(\check{\mathrm{C}}8\mathrm{a})^{\mathsf{PLA}} : u_{k,l}d_{k,l} {\le} v_{k,l} {\le} u_{k,l+1}d_{k,l+1}, \quad \forall k, l;$$
$$(\check{\mathrm{C}}8\mathrm{b})^{\mathsf{PLA}} : \sum\nolimits_{k\in\mathcal{B}}\left(\sum\nolimits_{l=0}^{L-1} v_{k,l} + s_k^{\mathsf{c}}d_k^{\mathsf{rq}}\right) \le D^{\mathsf{max}},$$

where $\check{\Omega}_3 = \cup_{k\in\mathcal{B}}\left(s_k^{\mathsf{f}} \cup s_k^{\mathsf{c}} \cup U_k \cup V_k\right)$ and constraints $(\check{\mathrm{C}}3)^{\mathsf{PLA}}$, $(\check{\mathrm{C}}4)^{\mathsf{PLA}}$, and $(\check{\mathrm{C}}8\mathrm{a})^{\mathsf{PLA}}{-}(\check{\mathrm{C}}8\mathrm{b})^{\mathsf{PLA}}$ are the transformed constraints of original constraints $(\check{\mathrm{C}}3)$, $(\check{\mathrm{C}}4)$, and $(\mathrm{C}8)$, respectively. This transformed problem is an MILP problem, which can be solved effectively by using the CVX solver. The PLA based algorithm for verifying the feasibility of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is summarized in Algorithm 3, which can be integrated into Algorithm 1 to solve $(\mathcal{P}_2^{\mathsf{ext}})$. It is noted that if the value of $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ is unbounded for a given $d_{k,l}$, this infeasible point is removed when applying the PLA based algorithm.

Algorithm 3 PLA-based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **Initialize**: L, η
2: Compute $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all $k \in \mathcal{B}$ as in Step 1 and 2 of Algorithm 2.
3: Define $d_{k,l} = (d_k^{\mathsf{rq}} - \epsilon_{\mathsf{d}})l/L, \ \forall k \in \mathcal{B}, l = 0 : L$.
4: Compute $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$. **If** $\mathcal{F}_{k,\eta}^{\mathsf{f,rq}}(d_{k,l})$ is unbounded **then** Remove point $d_{k,l}$ **end if**.
5: Compute $A_{k,l}$, $B_{k,l}$, and then solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$ to get optimal value $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star}$ of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$.
6: **if** $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star} \le F^{\mathsf{f,max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible, **else** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible **end if**

B. Two-stage Solution Approach (TSA)

In this section, two two-stage algorithms are developed by exploiting the fact that the decompression computational load (and therefore, the associated energy consumption) is almost independent from the compression ratio as can be seen in Fig. 2. This implies that for a given η, the optimal values f_k^{u}, ω_k^{u}, p_k, and ρ_k for mobile user k are similar for both $s_k^{\mathsf{f}} = 1$ and $s_k^{\mathsf{c}} = 1$. Hence, in the first stage, after solving $(\mathcal{P}_3)_k$ and $(\mathcal{P}_4)_k$, $\forall k \in \mathcal{B}$, introduced in Section III, we can set these variables to the corresponding optimal solution of $(\mathcal{P}_3)_k$, denoted as $f_{k,1}^{\mathsf{u}\star}$, $\omega_{k,1}^{\mathsf{u}\star}$, $p_{k,1}^{\star}$, and $\rho_{k,1}^{\star}$. In the second stage, we find the remaining variables pertaining to the fog server $\Omega_4 = \cup_{k\in\mathcal{B}}\{s_k^{\mathsf{f}}, s_k^{\mathsf{c}}, s_k^{\mathsf{m}}, d_k, f_k^{\mathsf{f}}, \omega_k^{\mathsf{f}}\}$ by solving the following problem[8]:

$$(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}}) \quad \min_{\Omega_4} \ \hat{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{TSA}}(\Omega_4) = \sum\nolimits_{k\in\mathcal{B}}\left(s_k^{\mathsf{m}}f_k^{\mathsf{f}} + s_k^{\mathsf{f}}f_k^{\mathsf{f,rq}}\right)$$
$$\text{s.t. } (\check{\mathrm{C}}0\&9) : s_k^{\mathsf{m}}\left(\frac{b_k^{\mathsf{out,f}}}{d_k} + \frac{(c_k^{\mathsf{co,f}} + c_k^{\mathsf{de,u}})}{f_k^{\mathsf{f}}}\right) \le \nu_{k,0},$$
$$(\check{\mathrm{C}}8): \sum_{k\in\mathcal{B}}(s_k^{\mathsf{m}}d_k + s_k^{\mathsf{c}}d_k^{\mathsf{rq}}) \le D^{\mathsf{max}},$$
$$(\check{\mathrm{C}}3), (\check{\mathrm{C}}4), (\check{\mathrm{C}}10),$$

where $\nu_{k,0} = \min\{(\eta - \Xi_{k,1})/w_k^{\mathsf{T}}, T_k^{\mathsf{max}} - T_{k,1}\} + (c_{k,1}{+}c_k^{\mathsf{de}})/f_k^{\mathsf{f,rq}} - T^{\mathsf{c}}$, and $\Xi_{k,1}$ and $T_{k,1}$ are the optimal values of Ξ_k and T_k in $(\mathcal{P}_3)_k$, respectively; $(\check{\mathrm{C}}0\&9)$ is determined by the time delay constraint as $\check{T}_k \le \min(T_k^{\mathsf{max}}, (\eta - w_k^{\mathsf{E}}\xi_k)/w_k^{\mathsf{T}})$ which is equivalent to constraints $(\check{\mathrm{C}}0)$ and $(\check{\mathrm{C}}9)$. This constraint captures the fact that an application should be offloaded

[8]We note that by reducing the number of optimization variables in $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$, the complexity of the resulting algorithms for feasibility verification of $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is lower than that of the PLA based algorithm.

to the cloud server if the resulting WEDC is smaller than that achieved when the application is executed at the fog server and the delay constraint ($\check{\text{C}}9$) is not violated. Because $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ is a difficult MINLP problem, we tackle it by reducing the set of variables based on the results in the following three propositions. In particular, Propositions 3–5 are introduced to respectively rewrite variables f^{f}_k, ω^{f}_k, and d_k, for all k as functions of the remaining variables. Subsequently, two algorithms are proposed to solve for the remaining variables, one based on a one-dimensional search of the Lagrangian multiplier, and the other one based on an iterative update of the Lagrangian multiplier.

Proposition 3: For any value of d_k's satisfying ($\check{\text{C}}8$), the optimal solution of f^{f}_k in $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ can be determined as $f^{\mathsf{f}\star}_k = s^{\mathsf{m}}_k \frac{(c^{\mathsf{co,f}}_k + c^{\mathsf{de,u}}_k)}{\nu_{k,0} - b^{\mathsf{out,f}}_k/d_k} = s^{\mathsf{m}}_k \mathcal{H}_0\big(\omega^{\mathsf{f}}_k, d_k\big)$, *where* $\mathcal{H}_0\big(\omega^{\mathsf{f}}_k, d_k\big) = \frac{\omega^{\mathsf{f}}_k d_k \left[\tilde{\gamma}^{\mathsf{co,f}}_{k,1} (\omega^{\mathsf{f}}_k)^{\gamma^{\mathsf{co,f}}_{k,2}} + \tilde{\gamma}^{\mathsf{co,f}}_{k,3}\right]}{\nu_{k,0}\omega^{\mathsf{f}}_k d_k - b^{\mathsf{in}}_k}$, $\tilde{\gamma}^{\mathsf{co,f}}_{k,1} = \gamma^{\mathsf{f}}_{k,0}\gamma^{\mathsf{co,f}}_{k,1}$, *and* $\tilde{\gamma}^{\mathsf{co,f}}_{k,3} = \gamma^{\mathsf{f}}_{k,0}\gamma^{\mathsf{co,f}}_{k,3} + c^{\mathsf{de,u}}_k$.

Proof: When $s^{\mathsf{m}}_k = 1$, the left-hand side of ($\check{\text{C}}1\&9$) is inversely proportional to f^{f}_k; thus, f^{f}_k is minimized if users spend the maximum possible resources. ∎

Proposition 4: When $s^{\mathsf{m}}_k = 1$ and $d_k \geq \bar{d}_{k,1}$, the optimal value of ω^{f}_k, denoted as $\omega^{\mathsf{f}\star}_k$, is given as follows:

$$\omega^{\mathsf{f}\star}_k = \begin{cases} \omega^{\mathsf{max,f}}_k, & \text{if } \gamma^{\mathsf{co,f}}_{k,2} \leq 0 \cup \{\gamma^{\mathsf{co,f}}_{k,2} \geq 0, \bar{d}_{k,1} < d_k \leq \bar{d}_{k,2}\}, \\ \mathrm{inv}\big(\mathcal{H}_1(d_k)\big), & \text{if } \gamma^{\mathsf{co,f}}_{k,2} \geq 0, \bar{d}_{k,2} < d_k \leq \bar{d}_{k,3}, \\ \omega^{\mathsf{f,min}}_k, & \text{if } \gamma^{\mathsf{co,f}}_{k,2} \geq 0, d_k > \bar{d}_{k,3}, \end{cases} \tag{11}$$

where $\bar{d}_{k,1} = b^{\mathsf{in}}_k/(\nu_{k,0}\omega^{\mathsf{f}}_k)$, $\bar{d}_{k,2} = \mathcal{H}_1\big(\omega^{\mathsf{max,f}}_k\big)$, $\bar{d}_{k,3} = \mathcal{H}_1\big(\omega^{\mathsf{f,min}}_k\big)$, *and* $\mathrm{inv}\big(\mathcal{H}_1(d_k)\big)$ *is the value of* ω^{f}_k *for which* $\mathcal{H}_1\big(\omega^{\mathsf{f}}_k\big)$ *is equal to* d_k, *and* $\mathcal{H}_1\big(\omega^{\mathsf{f}}_k\big) \triangleq \frac{\tilde{\gamma}^{\mathsf{co,f}}_{k,1} b^{\mathsf{in}}_k (\gamma^{\mathsf{co,f}}_{k,2}+1) \big(\omega^{\mathsf{f}}_k\big)^{\gamma^{\mathsf{co,f}}_{k,2}} + \tilde{\gamma}^{\mathsf{co,f}}_{k,3} b^{\mathsf{in}}_k}{\tilde{\gamma}^{\mathsf{co,f}}_{k,1} \nu_{k,0} \gamma^{\mathsf{co,f}}_{k,2} \big(\omega^{\mathsf{f}}_k\big)^{\gamma^{\mathsf{co,f}}_{k,2}+1}}$.

Proof: The proof is given in Appendix C. ∎

Based on the results in Propositions 3 and 4, $(\mathcal{P}^{\mathsf{TSA}}_{\mathsf{FV},\eta})$ is equivalent to the following problem:

$$(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta}) \quad \min_{\tilde{\Omega}_4} \sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_k \mathcal{H}_0\big(\omega^{\mathsf{f}\star}_k, d_k\big) + s^{\mathsf{f}}_k f^{\mathsf{f,rq}}_k\right] \quad \text{s.t } (\check{\text{C}}3), (\check{\text{C}}4), (\check{\text{C}}8),$$

where $\tilde{\Omega}_4 = \cup_{k\in\mathcal{B}}\{s^{\mathsf{c}}_k, s^{\mathsf{f}}_k, s^{\mathsf{m}}_k, d_k\}$.

Proposition 5: The optimal value of d_k for $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$, denoted as $d^{\star}_k$, is given as follows:

$$d^{\star}_k = \begin{cases} 0, & \text{if } s^{\mathsf{f}\star}_k = 1, \\ d^{\mathsf{rq}}_k, & \text{if } s^{\mathsf{c}\star}_k = 1, \\ \left\{ d_{k,\lambda} \middle| \left(\frac{\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)}{\partial d_k}\big|_{d_k = d_{k,\lambda}} \right) + \lambda = 0 \right\}, & \text{otherwise}, \end{cases} \tag{12}$$

where λ is the Lagrange multiplier of constraint ($\check{\text{C}}8$).

Proof: The Lagrangian of problem $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ can be expressed as $\mathcal{L}(\tilde{\Omega}_4, \lambda) = \sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_k \mathcal{H}_0\big(\omega^{\mathsf{f}\star}_k, d_k\big) + s^{\mathsf{f}}_k f^{\mathsf{f,rq}}_k\right] + \lambda\big(\sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_k d_k + (1 - s^{\mathsf{f}}_k - s^{\mathsf{m}}_k) d^{\mathsf{rq}}_k\right] - D^{\mathsf{max}}\big)$. When $s^{\mathsf{m}\star}_k = 1$, the necessary conditions for the optimal solution $f^{\mathsf{f}\star}_k, d^{\star}_k$ can be obtained by setting the derivatives of $\mathcal{L}$ with respect to these variables equal to zero as follows:

$$\frac{\partial \mathcal{L}}{\partial d_k} = s^{\mathsf{m}}_k \Big(\frac{\partial \mathcal{H}_0\big(\omega^{\mathsf{f}\star}_k, d_k\big)}{\partial d_k} + \lambda \Big) = 0, \tag{13}$$

$$\lambda \left(\sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_k d_k + (1 - s^{\mathsf{f}}_k - s^{\mathsf{m}}_k) d^{\mathsf{rq}}_k\right] - D^{\mathsf{max}} \right) = 0. \tag{14}$$

Based on (13), it can be verified that $d^{\star}_k$ can be expressed as in (12). ∎

Lemma 2: The gradient $\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)/\partial d_k$ is a monotonically increasing function of d_k.

Proof: The proof is given in Appendix D . ∎

As can be verified, if $\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)/\partial d_k\big|_{d_k = \bar{d}_{k,1}} + \lambda > 0$, then $d^{\star}_k = d_{k,\lambda} = 0, s^{\mathsf{f}\star}_k = 1$ will be the optimal solution. When $s^{\mathsf{m}\star}_k = 1$, λ must be positive because $\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)/\partial d_k$ is negative for all d_k. With the results in Lemma 2, we can conclude that for a given λ, there exists at most one value of d_k satisfying $\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)/\partial d_k + \lambda = 0$. This means if the optimal λ is known, problem $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ can be solved effectively. Therefore, as described in the following, to solve $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$, we propose two algorithms: one is based on a one-dimensional search for λ, and the other one is based on iterative updating λ.

1) One-Dimensional λ-Search Based Two-Stage Algorithm (OSTS Alg.): For a given λ, suppose that $d_{k,\lambda}$ satisfies $\partial \mathcal{H}_0(\omega^{\mathsf{f}\star}_k, d_k)/\partial d_k\big|_{d_k = d_{k,\lambda}} + \lambda = 0$. By defining $f_{k,\lambda} = \mathcal{H}_0\big(\omega^{\mathsf{f}\star}_k, d_k\big)\big|_{d_k = d_{k,\lambda}}$, $\mu_{k,\lambda} = s^{\mathsf{m}}_k$, $\mu_{k,\lambda} = 1 - s^{\mathsf{c}}_k$, and $\mu_{k,\lambda} = s^{\mathsf{c}}_k(1 - x_k)$, we can find the optimal solution of $\cup_{k\in\mathcal{B}}\{s^{\mathsf{c}}_k, x_k, d_k\}$ by solving the following problem:

$$\begin{aligned} (\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda) = \min_{\cup_{k\in\mathcal{B}} s_{k,\lambda}} & \sum_{k\in\mathcal{B}} \left[s^{\mathsf{m}}_{k,\lambda} f_{k,\lambda} + s^{\mathsf{f}}_{k,\lambda} f^{\mathsf{f,rq}}_k\right] \\ \text{s.t.} \quad & (\check{\text{C}}8)_\lambda : \sum_{k\in\mathcal{B}} s^{\mathsf{m}}_{k,\lambda} d_{k,\lambda} + (1 - s^{\mathsf{f}}_{k,\lambda} \\ & - s^{\mathsf{m}}_{k,\lambda}) d^{\mathsf{rq}}_k \leq D^{\mathsf{max}}, \\ & s^{\mathsf{m}}_{k,\lambda}, s^{\mathsf{f}}_{k,\lambda} \in \{0, 1\}, \end{aligned}$$

where $s_{k,\lambda} = \{s^{\mathsf{f}}_{k,\lambda}, s^{\mathsf{m}}_{k,\lambda}\}$. The above transformed problem is an integer linear programming (ILP) problem, which can be solved effectively by CVX. Let $\tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$ be the optimum of $(\mathcal{P}^{\mathsf{OSTS}}_{\mathsf{FV},\eta})_\lambda$, then we can find the optimum of $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ as $\tilde{G}^{\mathsf{OSTS}\star}_{\mathcal{B},\eta} = \min_\lambda \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$. Moreover, it can be shown that when we increase λ, all $d_{k,\lambda}$ will decrease. Therefore, the maximum value of λ is λ^{max} satisfying $\mathcal{H}_0(\omega^{\mathsf{f}}_k, d_{k,\lambda^{\mathsf{max}}}) \geq f^{\mathsf{f,rq}}_k$, $\forall k \in \mathcal{B}$ and $\sum_{k\in\mathcal{B}} d_{k,\lambda^{\mathsf{max}}} \leq D^{\mathsf{max}}$. Note that we can stop the search process when there exists a λ such that $\tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda) \leq F^{\mathsf{f,max}}$. When the bisection search for η converges, we can find the optimum $\lambda^\star = \mathrm{argmin}_\lambda \tilde{\mathcal{G}}^{\mathsf{OSTS}}_{\mathcal{B},\eta}(\lambda)$, and the optimal variables $s^{\mathsf{m}\star}_k = s^{\mathsf{m}}_{k,\lambda^\star}$, $s^{\mathsf{f}\star}_k = s^{\mathsf{f}}_{k,\lambda^\star}$, $s^{\mathsf{c}\star}_k = 1 - s^{\mathsf{m}\star}_k - s^{\mathsf{f}\star}_k$, $f^{\mathsf{f}\star}_k = s^{\mathsf{m}}_{k,\lambda^\star} f_{k,\lambda^\star} + s^{\mathsf{f}}_{k,\lambda^\star} f^{\mathsf{f,rq}}_k$, and $d^{\star}_k = s^{\mathsf{m}}_{k,\lambda^\star} d_{k,\lambda^\star} + (1 - s^{\mathsf{f}}_{k,\lambda^\star} - s^{\mathsf{m}}_{k,\lambda^\star}) d^{\mathsf{rq}}_k, \forall k \in \mathcal{B}$. The OSTS algorithm for feasibility verification of $(\mathcal{P}^{\mathsf{ext}}_{\mathcal{B}})$ is summarized in Algorithm 4.

2) Iterative λ-Update Based Two-Stage Algorithm (IUTS Alg.): This method can solve $(\mathcal{P}^{\mathsf{TSAeq}}_{\mathsf{FV},\eta})$ with very low complexity via Lagrangian dual updates. Specifically, the

Algorithm 4 One-dimensional Search Based Feasibility Verification for $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$

1: **initialize**: Δ_λ, $\lambda = 0$, Assign $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is infeasible.
2: Define $f_k^{\mathsf{f,rq}}$ and d_k^{rq} for all k as in Step 2 and Step 3 of Algorithm 2.
3: **repeat**
4: Assign $\lambda = \lambda + \Delta_\lambda$. Compute $d_{k,\lambda}$ as in (12) and solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ to find $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda)$.
5: **if** $\tilde{\mathcal{G}}_{\mathcal{B},\eta}^{\mathsf{OSTS}}(\lambda) \leq F^{\mathsf{f,max}}$ **then** Return $(\mathcal{P}_{\mathcal{B}}^{\mathsf{ext}})$ is feasible; **break**
6: **end if**
7: **until** $\lambda = \lambda^{\mathsf{max}}$

dual function of $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSAeq}})$ can be defined as $\mathcal{G}^{\circ}(\lambda) = \min_{\tilde{\Omega}_4} \mathcal{L}(\tilde{\Omega}_4, \lambda)$, and the dual problem can be stated as

$$\max_{\lambda} \mathcal{G}^{\circ}(\lambda) \text{ s.t. } \lambda \geq 0. \tag{15}$$

Since the dual problem is always convex, $\mathcal{G}^{\circ}(\lambda)$ can be maximized by using the standard sub-gradient method where the dual variable λ is iteratively updated as follows: $\lambda_n = \Big[\lambda_{n-1} + \delta_n \big(\sum_{k\in\mathcal{B}} \left(s_{k,\lambda_{n-1}}^{\mathsf{m}} d_{k,\lambda_{n-1}} + s_{k,\lambda_{n-1}}^{\mathsf{c}} d_k^{\mathsf{rq}} \right) - D^{\mathsf{max}} \big) \Big]^+$, where n denotes the iteration index, δ_n represents the step size, and $[a]^+$ is defined as $\max(0, a)$. The sub-gradient method is guaranteed to converge to the optimal value of λ for an initial primal point Ω_4 if the step size δ_n is chosen appropriately, e.g., $\delta_n \to 0$ when $n \to \infty$, which is met by setting $\delta_n = 1/\sqrt{n}$.

For a given λ_n, we can determine the primal variable $d_{k,\lambda_n} = \text{inv}(\mathcal{H}_2(\lambda_n))$. For given λ_n and d_{k,λ_n}, the primal problem becomes a linear program in $s_{k,\lambda_n}, \forall k \in \mathcal{B}$, which can be solved effectively by using standard linear optimization techniques. Moreover, the vertices in this problem are the points where the $s_{k,\lambda_n}^{\mathsf{m}}$'s, $s_{k,\lambda_n}^{\mathsf{f}}$'s, and $s_{k,\lambda_n}^{\mathsf{c}}$'s are either 0 or 1. Thus, *solving the relaxed problem will also return binary values* 0 *or* 1. However, once the $s_{k,\lambda_n}^{\mathsf{m}}$'s, $s_{k,\lambda_n}^{\mathsf{f}}$'s, and $s_{k,\lambda_n}^{\mathsf{c}}$'s take values of 0 or 1, the decision on the application execution location (fog or cloud) may be trapped at a local optimal solution such that the required fog computing resources cannot be updated to improve the solution. To overcome this critical issue, the gradient projection method can be adopted to slowly update variables $s_{k,\lambda_n}^{\mathsf{m}}$'s, $s_{k,\lambda_n}^{\mathsf{f}}$'s, and $s_{k,\lambda_n}^{\mathsf{c}}$'s as $\mathbf{s}_k^{(n+1)} = \mathbb{P}_{\Phi_k}\left(\mathbf{s}_k^{(n)} - \check{\delta}\nabla\mathbf{s}_k^{(n)}\right)$, where $\mathbf{s}_k^{(n)} = [s_{k,\lambda_n}^{\mathsf{m}}, s_{k,\lambda_n}^{\mathsf{f}}, s_{k,\lambda_n}^{\mathsf{c}}]$, $\check{\delta}$ is the step size, $\nabla\mathbf{s}_k^{(n)} = [\mathcal{H}_0(\omega_k^{\mathsf{f}\star}, d_{k,\lambda_n}) + \lambda_n d_{k,\lambda_n}, \lambda_n f_k^{\mathsf{f,rq}}, \lambda_n d_k^{\mathsf{rq}}]$, and $\mathbb{P}_{\Phi_k}(.)$ is the projection onto the set $\Phi_k = \left\{\mathbf{s}_k | \mathbf{s}_k \geq 0,\ s_{k,\lambda_n}^{\mathsf{f}} + s_{k,\lambda_n}^{\mathsf{c}} + s_{k,\lambda_n}^{\mathsf{m}} \leq 1\right\}$. Finally, it can be verified that this iterative mechanism always converges [38].

C. Complexity Analysis

The overall complexity of the PLA algorithm for solving the extended problem $(\mathcal{P}_2^{\mathsf{ext}})$ is $\log_2(\frac{\eta^{\mathsf{max}} - \eta^{\mathsf{min}}}{\epsilon}) \left(K\mathcal{O}((\mathcal{P}_3)_k) + LK\mathcal{O}((\mathcal{P}_4)_k) + \mathcal{O}(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}}) \right)$, i.e., $|\mathcal{B}| \leq K$. Moreover, for given $\mathcal{B}$, $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{PLA}})$ is an NP-hard problem, solving it via an optimal exhaustive search entails a complexity of $\mathcal{O}(2^{(L+1)^{|\mathcal{B}|}})$, which is upper bounded by $\mathcal{O}(2^{(L+1)^K})$.

The proposed two-stage IUTS and OSTS algorithms for solving the extended problem have an overall complexity of $\log_2(\frac{\eta^{\mathsf{max}} - \eta^{\mathsf{min}}}{\epsilon}) \left(K\mathcal{O}((\mathcal{P}_3)_k) + K\mathcal{O}((\mathcal{P}_4)_k) + \mathcal{O}(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}}) \right)$, i.e., $|\mathcal{B}| \leq K$. In Section IV-B1, for given $\mathcal{B}$, problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ can be transformed to a standard knapsack problem as in [36], while the optimal d_k and ω_k can be computed directly for a given value of λ. Therefore, the complexity of Algorithm 4 to solve $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ by the OSTS method is $\mathcal{O}(\frac{\lambda^{\mathsf{max}}}{\Delta_\lambda}\nu_3|2\mathcal{B}|)$, where ν_3 is determined by the coefficients in $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{OSTS}})_\lambda$ [36]. For the IUTS algorithm presented in Section IV-B2, we can directly update λ_n, d_{k,λ_n}, μ_{k,i,λ_n}, $\forall i, k, n$; which means that $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ has a complexity of $\mathcal{O}(N|\mathcal{B}|)$, where N is the number of iterations. We note that $\mathcal{O}((\mathcal{P}_3)_k)$ and $\mathcal{O}((\mathcal{P}_4)_k)$ are given in Section III-F.

V. Numerical Results

A. Simulation Setup

We consider a hierarchical fog-cloud system consisting of $K = 10$ users (except for Fig. 9) where the users are randomly distributed in the cell coverage area with a radius of 800 m and the BS is located at the cell center. The simulation parameters provided in Table I are adopted, unless specified otherwise. Particularly, the path-loss is calculated as $\beta_k(\text{dB}) = 128.1 + 37.6\log_{10}(dist_k)$, where $dist_k$ is the geographical distance between user k and the BS (in km) [39]. We further set the beamforming gain as $M_0 = 5$, the maximum transmission bandwidth as $\rho_k^{\mathsf{max}} = 1$ MHz, and the noise power density as $\sigma_{\mathsf{bs}} = 1.381 \times 10^{-23} \times 290 \times 10^{0.9}$ W/Hz [40]. All users are assumed to have the same maximum clock speed of 2.4 GHz, a maximum transmit power of $P_k^{\mathsf{max}} = 0.22$ W, and the circuit power consumption per Hz is set to $p_{k,0} = 22$ nW/Hz. We assume that the number of transmission bits incurred to support computation offloading b_k^{in} is the same for all users.

Moreover, the computation demands of the 10 users $\{c_1, c_2, \ldots, c_9, c_{10}\}$ are set randomly in the range $1.8 - 2.4$ Gcycles while the maximum delay time is to $T_k^{\mathsf{max}} = 1$ second, the non-offloadable load is $c_{k,0} = 0.1\ c_k$, and the offloadable load is $c_{k,1} = 0.9\ c_k$ for all users. We also set the energy coefficient as $\alpha_k = 0.1 \times 10^{-27}$ and the computing time at the cloud server as $T^{\mathsf{c}} = T_k^{\mathsf{max}}/5$. For the DC algorithm, we set the parameters according to the top-left sub-figure in Fig. 2 as follows: $\gamma_{k,1}^{\mathsf{co}} = 0.03 \times 2.6^{32.28}$, $\gamma_{k,2}^{\mathsf{co}} = 32.28$, $\gamma_{k,3}^{\mathsf{co}} = 0.3$, $\gamma_{k,1}^{\mathsf{de}} = 0.115$, $\gamma_{k,2}^{\mathsf{de}} = -0.9179$, $\gamma_{k,3}^{\mathsf{de}} = 0.046, \forall k$, $\omega_k^{\mathsf{u,min}} = 2.3$, and $\omega_k^{\mathsf{u,max}} = 2.9$. The energy and delay weights are chosen so that $w_k^{\mathsf{E}} + w_k^{\mathsf{T}} = 1, \forall k$. Simulation results are obtained by averaging over 100 realizations of the random locations of the users. Finally, for all figures, we set the raw data size as $b_k^{\mathsf{in}} = 4$ Mbits (except for Figs. 5, 7 and 9), $w_k^{\mathsf{E}} = 2w_k^{\mathsf{T}}, \forall k$ (except for Fig. 8), the maximum fog computing resource as $F^{\mathsf{f,max}} = 15$ GHz, the maximum backhaul capacity as $D^{\mathsf{max}} = 20$ Mbps (except for Figs. 7 and 8), and $\kappa = 50$ (except for Figs. 5 and 6), where κ captures

TABLE I
SIMULATION PARAMETER SETTINGS

Parameter	Setting
Path loss, β_k	$128.1+37.6\log_{10}(dist_k(\text{km}))$
Cell radius	800 meters
Noise power density, σ_{bs}	3.18×10^{-20} W/Hz
Number of users K	10
Beamforming gain M_0	5
Max. transmission bandwidth $\rho_k^{\max}$	1 MHz
Max. delay time $T_k^{\max}$	1 second
Max. clock speed $F_k^{\max}$	2.4 GHz
Max. transmit power $P_k^{\max}$	0.22 W
Circuit power consumption per Hz	$p_{k,0} = 22$ nW/Hz
User computation demand	$c_k \in [1.8 - 2.4]$ Gcycles
Offloadable load	$c_{k,1} = 0.9c_k$
Energy coefficient α_k	0.1×10^{-27}
Time T^{c}	0.2 second
Raw data size b_k^{in}	4 Mbits
Max. fog computing resource $F^{\text{f,max}}$	15 GHz
Max. backhaul capacity $D^{\max}$	20 Mbps
User compression ratio range: ω_k^{u}	$[2.3, 2.9]$
Coefficient κ	50
Fog compression ratio range: ω_k^{f}	$[3.4, 11.2]$

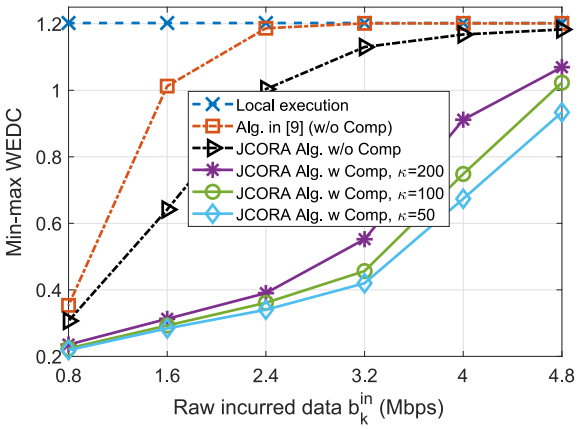

Fig. 5. Min-max WEDC vs. b_k^{in}.

the relationship between $\gamma_{k,0}^{\text{u}}$ in (1) and the raw data size as $\gamma_{k,0}^{\text{u}} = \kappa b_k^{\text{in}}$ [41].

In practice, a fog server can support more powerful DC algorithms compared to the users. This implies that the compression ratio for the fog server is much larger than that for the users. Therefore, when the fog server decompresses and re-compresses data, we set the parameters according to the top-middle sub-figure in Fig. 2 as follows: $\gamma_{k,1}^{\text{co,f}} = 0.076, \gamma_{k,2}^{\text{co,f}} = 0.7116, \gamma_{k,3}^{\text{co,f}} = 0.5794$, $\omega_k^{\text{f,min}} = 3.4$, and $\omega_k^{\text{f,max}} = 11.2$. The step size is set as $\check{\delta} = 0.1$. For the proposed algorithms presented in Section III and Section IV, numerical results are shown in Figs. 5–9 and Figs. 10–12, respectively.

B. Results for DC at Only Mobile Users

In Fig. 5, we show the significant benefits of DC for computation offloading where the min-max WEDC (called WEDC for brevity) vs. b_k^{in} is plotted for six different schemes: the 'Local-execution' scheme in which all users' applications are executed locally; the 'Alg. in [9] (w/o Comp)' scheme in which the benchmark algorithm in [9] is applied with

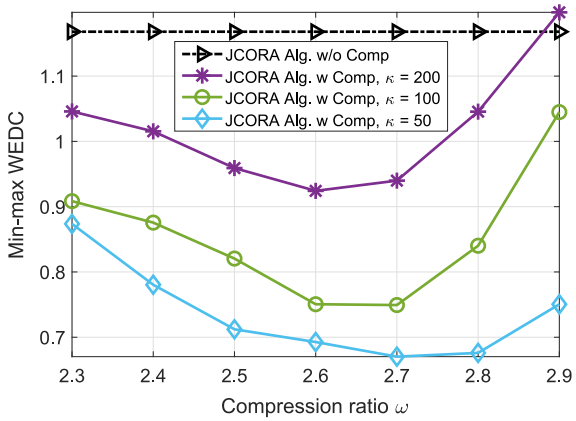

Fig. 6. Min-max WEDC vs. compression ratio.

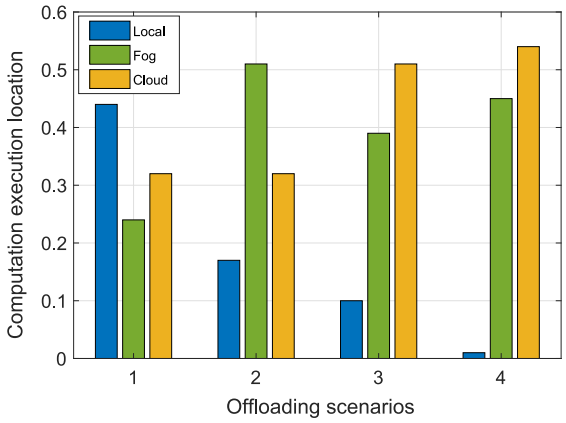

Fig. 7. User, fog, and cloud computational load processing.

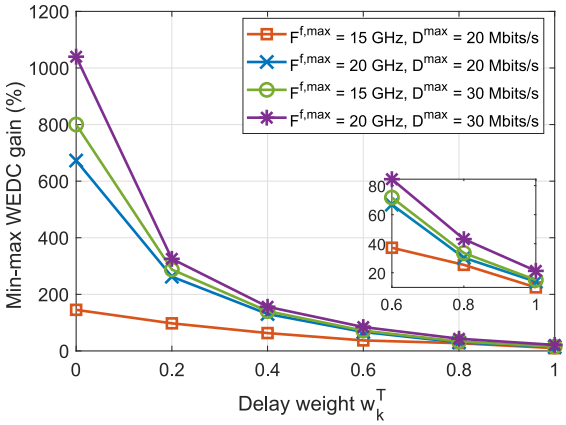

Fig. 8. Min-max WEDC gain vs. delay weight.

$\omega_k^{\text{u}} = 1, \forall k$, and no DC[9]; the 'JCORA Alg. w/o Comp' in which the proposed JCORA algorithm is applied with $\omega_k^{\text{u}} = 1, \forall k$, and no DC (the other variables are optimized as in the JCORA algorithm); and three other instances of the proposed JCORA algorithm with DC and three different values of $\kappa = 50, 100, 200$ ($\kappa = \gamma_{k,0}^{\text{u}}/b_k^{\text{in}}$). To guarantee a fair comparison between the 'Alg. in [9] (w/o Comp)' scheme and

[9]As discussed in Section I, this paper provides the first study of joint DC and computation offloading in hierarchical fog-cloud systems. Therefore, the recent work [9] on computation offloading in hierarchical fog-cloud systems, which does not exploit DC, is selected as benchmark for performance comparison.

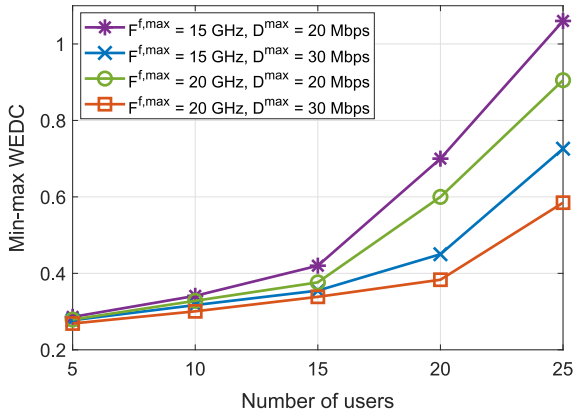

Fig. 9. Min-max WEDC vs. number of users.

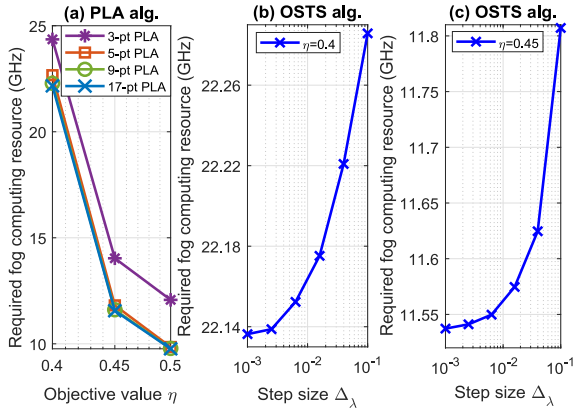

Fig. 10. Accuracy of proposed PLA and OSTS algs.

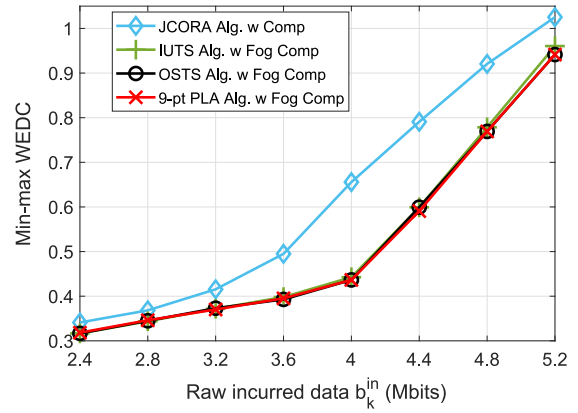

Fig. 11. Min-max WEDC in general design scenario.

our proposed schemes, we also apply MIMO and optimize the offloading decision and the allocation of the fog computing resources, transmit power, bandwidth, and local CPU clock speed for the 'Alg. in [9] (w/o Comp)' scheme. In addition, for the remaining variable d_k, we allocate the backhaul capacity equally to the users that offload their tasks to the cloud server.

As can be observed from Fig. 5, computation offloading can greatly improve the WEDC when there are sufficient radio and computing resources to support the offloading (e.g., the incurred amount of data is not too large). Specifically, computation offloading even without DC can result in a significant reduction of the WEDC compared to local

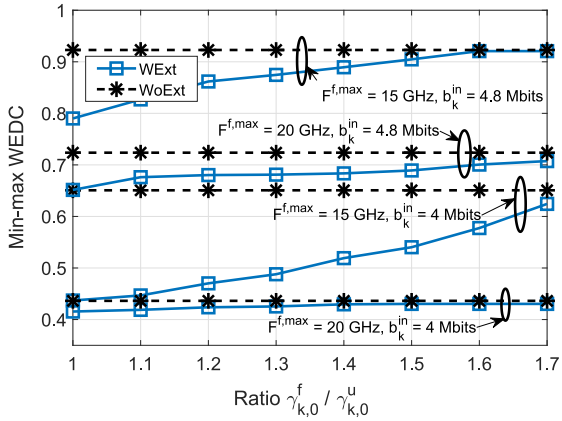

Fig. 12. Min-max WEDC vs. $\gamma^{\mathrm{f}}_{k,0}/\gamma^{\mathrm{u}}_{k,0}$.

execution, especially when the incurred amount of data b^{in}_k is small such that the constrained radio resources do not limit performance. Furthermore, even without exploiting DC, our proposed algorithm (JCORA Alg. w/o Comp) results in a much better performance than the algorithm proposed in [9]. This is because our proposed design jointly optimizes the offloading decisions and the computing and radio resource allocation, while in [9], the offloading decisions are found nearly independent of the computing and radio resource allocation. In particular, the semidefinite relaxation technique employed in [9] may not always guarantee the rank-1 condition for the optimized matrix. Joint optimization of DC, computation offloading, and resource allocation can lead to a significant further reduction of the WEDC for a larger range of b^{in}_k (e.g., when $b^{\mathrm{in}}_k = 2.4$ Mbps, the min-max WEDC is reduced by up to 65%). However, the energy and time consumed for (de)compression also affect the achievable min-max WEDC, and their impact tends to become stronger for larger $\gamma^{\mathrm{u}}_{k,0}$ and when the available radio resource is more limited.

In Fig. 6, we investigate the impact of the compression ratio on the min-max WEDC for the JCOCA scheme with and without DC for different values of $\omega^{\mathrm{u}}_k = \omega, \forall k$ (i.e., the compression ratio ω^{u}_k is fixed while the remaining variables are optimized as in the JCOCA scheme). As can be seen, there is an optimal ω that achieves the minimum WEDC. Moreover, the optimal value of ω tends to decrease for increasing computational load because the optimal compression ratio has to efficiently balance the demand on the radio and computing resources. In fact, for the right choice of ω, the *"JCORA Alg. w Comp"* scheme greatly outperforms the *"JCORA Alg. w/o Comp"* scheme. Moreover, this figure shows that for the optimal ω, 29% reduction in the min-max WEDC can be achieved compared to the worst choice of ω.

Fig. 7 shows the computational loads processed locally as well as in the fog and cloud servers when $b^{\mathrm{in}}_k = 4.8$ Mbits for four different scenarios: 1) $F^{\mathrm{f,max}} = 15$ GHz, $D^{\mathrm{max}} = 20$ Mbps; 2) $F^{\mathrm{f,max}} = 20$ GHz, $D^{\mathrm{max}} = 20$ Mbps; 3) $F^{\mathrm{f,max}} = 15$ GHz, $D^{\mathrm{max}} = 30$ Mbps; and 4) $F^{\mathrm{f,max}} = 20$ GHz, $D^{\mathrm{max}} = 30$ Mbps. The results shown in Fig. 7 suggest that more of the users' computational load should be offloaded and executed at the fog and cloud servers if sufficient resources to support the offloading process are available. Particularly, nearly all users offload their computation tasks in Scenario 4, while in

Scenario 1, about half of the users offload their computation demand.

In Fig. 8, we show the min-max WEDC gain due to DC as a function of the delay weight w_k^T. The min-max WEDC gain is computed as $\frac{\eta^{\mathsf{NoComp}\star}-\eta^{\mathsf{Comp}\star}}{\eta^{\mathsf{Comp}\star}} \times 100$ (%) where $\eta^{\mathsf{Comp}\star}$ and $\eta^{\mathsf{NoComp}\star}$ denote the optimal min-max WEDCs with and without DC under the JCORA framework. When energy saving is the only concern for the mobile devices ($w_k^{\mathsf{T}} = 0, w_k^{\mathsf{E}} = 1$), this figure confirms that JCORA with DC can save more than 170% of energy compared with JCORA without DC even for the scenario with $F^{\mathsf{f,max}} = 15$ GHz and $D^{\mathsf{max}} = 20$ Mbps. The min-max WEDC gain decreases when we focus more on latency (i.e., for higher delay weight w_k^T). Moreover, for $w_k^{\mathsf{T}} = 1$, DC results in a 15% reduction of the execution delay for $F^{\mathsf{f,max}} = 15$ GHz, $D^{\mathsf{max}} = 20$ Mbps, and about 25% delay reduction for $F^{\mathsf{f,max}} = 20$ GHz, $D^{\mathsf{max}} = 30$ Mbps. In Fig. 9, we show the min-max WEDC vs. the number of users in the system for $b_k^{\mathsf{in}} = 2.4$ Mbps, $\forall k$. When there are more users that may offload their computational loads to the fog and cloud servers, the available resources that can be allocated to each user become smaller; therefore, the min-max WEDC increases. However, the proposed JCORA scheme still achieves the optimal performance in the multi-user hierarchical fog-cloud system.

C. Results for DC at Both Mobile Users and Fog Server

To evaluate the system performance when DC is performed at both the mobile users and the fog server, we consider the following parameter setting: $\gamma_{k,0}^{\mathsf{f}} = \gamma_{k,0}^{\mathsf{u}}$ (except for Fig. 12), $F^{\mathsf{f,max}} = 15$ GHz, and $D^{\mathsf{max}} = 20$ Mbps. In Fig. 10, we show the required computing resources for the proposed PLA and OSTS algorithms when solving the extended problem. In Fig. 10-(a), 'n-pt PLA' corresponds to the n-point PLA method. In the PLA method, when the number of points used to approximate the actual function is sufficiently large, the difference between the actual and approximated functions becomes negligible. As shown in Fig 10-(a), there is only a small difference in the required fog computing resources when the number of points increases from 5 to 9. In addition, these required resources are nearly identical for both the 9-point and 17-point curves. Therefore, we use '9-pt PLA' as a benchmark method to evaluate the performance of the OSTS and IUTS algorithms. The middle and right sub-figures illustrate the accuracy of the OSTS algorithm in solving problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ vs. the step size Δ_λ. Specifically, these figures show that the value of $G_{\mathcal{B},\eta}^{\mathsf{OSTS}\star}$ becomes stable when Δ_λ is about 5×10^{-3}. Moreover, the value of $G_{\mathcal{B},\eta}^{\mathsf{OSTS}\star}$ achieved with the OSTS algorithm at $\Delta_\lambda = 5 \times 10^{-3}$ is almost the same as the value of $\hat{G}_{\mathcal{B},\eta}^{\mathsf{PLA}\star}$ achieved with '17-pt PLA', which means that the approximated problem $(\mathcal{P}_{\mathsf{FV},\eta}^{\mathsf{TSA}})$ can be used to find a close-to-optimal solution of the extended problem. Besides, the difference in $G_{\mathcal{B},\eta}^{\mathsf{OSTS}\star}$ for $\Delta_\lambda = 0.1$ and $\Delta_\lambda = 0.001$ is less than 2%, which means that a large step size ($\Delta_\lambda = 0.1$) can be used to make the OSTS algorithm converge quickly while still guaranteeing good system performance.

The benefits of data re-compression at the fog are shown in Fig. 11 where we plot the min-max WEDC vs. b_k^{in} for four different schemes: the 'JCORA Alg. w Comp' scheme in which data are compressed only at the users while the three remaining schemes correspond to the proposed algorithms for the extended case. In particular, '9-pt PLA Alg. w Fog Comp', 'OSTS Alg. w Fog Comp', and 'IUTS Alg. w Fog Comp' correspond to the 9-point PLA, OSTS, and IUTS algorithms, respectively, which perform compression at both the users and the fog server. For $b_k^{\mathsf{in}} = 4$ Mbits, an additional min-max WEDC reduction of 35% can be achieved by performing DC at both the users and the fog server. Moreover, the required radio resources decrease with decreasing b_k^{in}; therefore, the gain is reduced due to the decreasing demand for data transmission. When b_k^{in} increases, the main bottleneck for computation offloading are the limited radio resources available to support data transmissions between the users and the fog server; therefore, the gain due to data re-compression at the fog server becomes less significant. This figure also confirms that the '9-pt PLA', 'OSTS', and 'IUTS' schemes achieve almost the same min-max WEDC.

In Fig. 12, we plot the min-max WEDC vs. the ratio between the maximum computational loads (in CPU cycles) required to compress data at the fog server ($\gamma_{k,0}^{\mathsf{f}}$) and the user ($\gamma_{k,0}^{\mathsf{u}}$) for different values of $F^{\mathsf{f,max}}$ and b_k^{in}. The 'WoExt' and 'WExt' correspond to the JCORA and OSTS algorithms presented in Sections III and IV, respectively. This figure shows that DC at the fog server can bring additional performance benefits, especially in scenarios with limited fog computing resources (i.e., $F^{\mathsf{f,max}} = 15$ GHz). As the compression ratio adopted at the fog server could be much larger than that at the users, a better performance can be obtained by applying DC at both the users and the fog server when $\gamma_{k,0}^{\mathsf{f}}$ is not much larger than $\gamma_{k,0}^{\mathsf{u}}$. Otherwise, if the cost due to data re-compression becomes larger, the benefits of adopting *Mode 3* are less significant (i.e., for $\gamma_{k,0}^{\mathsf{f}} = 1.7\gamma_{k,0}^{\mathsf{u}}$).

VI. Conclusion

In this paper, we have proposed novel and efficient algorithms for joint DC and computation offloading in hierarchical fog-cloud systems which minimize the weighted energy and delay cost while maintaining user fairness. Specifically, we have considered the cases where DC is leveraged at only the mobile users and at both the mobile users and the fog server, respectively. Numerical results have confirmed the significant performance gains of the proposed algorithms compared to conventional schemes not using DC. Particularly, the following key observations can be drawn from our numerical studies: 1) Joint DC and computation offloading can result in min-max WEDC reductions of up to 65% compared to optimal computation offloading without DC; 2) the proposed JCORA scheme can efficiently distribute the computational load among the mobile users, the fog server, and the cloud server and exploits the available system resources in an optimal manner; 3) when energy saving is the only concern for the mobile users, the JCORA scheme can achieve an energy saving gain of up to a few hundred percent compared to optimal computation offloading without DC; and 4) an additional min-max WEDC reduction of up to 35% can be achieved by further employing DC at the fog server. In future work, we plan to

extend our designs to multi-task offloading and systems with multiple fog servers.

APPENDIX A
PROOF OF THEOREM 1

Assume that $(\mathcal{A}', \mathcal{B}')$ is an optimal classification corresponding to the optimum value $\eta^\star$. Due to Statement 2 in Lemma 1 and Proposition 1, we have the following results:

$$\max(\eta_{\mathcal{A}'}, \eta_{\mathcal{B}'}) = \eta^\star, \quad \eta_{\mathcal{A}'} = \max_{k\in\mathcal{A}'} \eta_k^{\mathsf{lo}}. \tag{16}$$

If there is no user k in $\mathcal{B}$ whose η_k^{lo} is less than or equal to $\eta^\star$, we can conclude that $(\mathcal{A}', \mathcal{B}') \equiv (\mathcal{A}^\star, \mathcal{B}^\star)$. Then, $(\mathcal{A}^\star, \mathcal{B}^\star)$ must be an optimal classification.

Conversely, if there exists a user k in $\mathcal{B}$ such that $\eta_k^{\mathsf{lo}} \leq \eta^\star$, we will prove that the user classification determined in Theorem 1 is also an optimal classification. Let $\mathcal{C} = \{k \in \mathcal{B}' | \eta_k^{\mathsf{lo}} \leq \eta^\star\}$. Then, it is easy to see that $\mathcal{A}^\star = \mathcal{A}' \cup \mathcal{C}$ and $\mathcal{B}^\star = \mathcal{B}'/\mathcal{C}$. According to the definition of $\mathcal{C}$, (16), and the result in Proposition 1, we have $\eta_{\mathcal{A}^\star} \leq \eta^\star$. In addition, since $\mathcal{B}^\star \subset \mathcal{B}'$, because of Statement 3 in Lemma 1, we can conclude that $\eta_{\mathcal{B}^\star} \leq \eta_{\mathcal{B}'} \leq \eta^\star$. Using these results, we can conclude that $(\mathcal{A}^\star, \mathcal{B}^\star)$ is an optimal classification.

APPENDIX B
PROOF OF PROPOSITION 2

Functions $\mathcal{Q}_{k,1}$ and $\mathcal{Q}_{k,2}$ are sums of exponential terms with positive coefficients; therefore, they are convex with respect to the variables in set $\tilde{\Omega}_{2,k}$ as proven in [35]. On the other hand, the first term of the WEDC and the total delay can be represented via function $\mathcal{H}(\tilde{p}_k, y_k) = \frac{a_{k,0}\mathrm{e}^{a_{k,1}\tilde{p}_k + a_{k,2}y_k}}{\log\left(1+\beta_{k,0}\mathrm{e}^{\tilde{p}_k}\right)}$, where $y_k \in \{\tilde{\omega}_k^{\mathsf{u}}, \tilde{\rho}_k, \tilde{l}_k\}$, $a_{k,0} > 0$, $a_{k,1} = \{0, 1\}$, and $\beta_{k,0}\mathrm{e}^{\tilde{p}_k} > 0$ due to the required positive data rate when users decide to offload their computational load.

Now, we will show that $\mathcal{H}(\tilde{p}_k, y_k)$ is a convex function of $\tilde{p}_k$ and y_k. Firstly, $\mathcal{H}(\tilde{p}_k, y_k)$ is convex with respect to y_k. Now, we need to prove that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 \geq 0$ and the determinant $|H(\tilde{p}_k, y_k)| > 0$, where $H(\tilde{p}_k, y)$ is the Hessian matrix of $\mathcal{H}(\tilde{p}_k, y_k)$.

Because we have $u_k = \beta_{k,0}\mathrm{e}^{\tilde{p}_k} > 0$ and the fact that $\log(1 + u_k) < u_k, \forall u_k > 0$, it can be verified that $|H(\tilde{p}_k, y)| = \frac{a_{k,0}a_{k,2}^2\beta_{k,0}[u_k - \log(1+u_k)]\mathrm{e}^{(2a_{k,1}+1)\tilde{p}_k + 2\ a_{k,2}y}}{(1+u_k)^2\log^4(1+u_k)} > 0$. In addition, we have

$$\frac{\partial^2\mathcal{H}(\tilde{p}_k, y_k)}{\partial\tilde{p}_k^2} = \begin{cases} \dfrac{u_k[2u_k - \log(1+u_k)]}{(1+u_k)^2\log^3(1+u_k)}, & \text{if } a_{k,1} = 0, \\ \dfrac{a_{k,0}\mathrm{e}^{a_{k,2}y}\mathrm{e}^{\tilde{p}_k}\mathcal{H}_{\mathrm{a}}(u_k)}{(1+u_k)^2\log^3(1+u_k)}, & \text{if } a_{k,1} = 1, \end{cases} \tag{17}$$

where $\mathcal{H}_{\mathrm{a}}(u_k) = (1 + u_k)^2\log^2(1 + u_k) + 2u_k^2 - (3u_k + 2u_k^2)\log(1 + u_k)$. From (17), it can be verified that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 > 0, \forall u_k > 0$ when $a_{k,1} = 0$. For the case with $a_{k,1} = 1$, since $\mathcal{H}_{\mathrm{a}}(u_k)$ is a quadratic function of $\log(1 + u_k)$, the discriminant of $\mathcal{H}_{\mathrm{a}}(u_k)$ is $u_k^2\left[2 - \left(1 + 2u_k\right)^2\right]$, which leads to $\mathcal{H}_{\mathrm{a}}(u_k) = (1+u_k)^2\prod_{j=\{-1,1\}}\left(\log(1+u_k) - u_{k,j}\right)$ if $u_k \leq \frac{\sqrt{2}-1}{2}$, where $u_{k,j} = \frac{u_k(3+2u_k) + ju_k\sqrt{2-(1+2u_k)^2}}{2(1+u_k)^2}$, $j = \{-1, 1\}$. Otherwise, $\mathcal{H}_{\mathrm{a}}(u_k)$ will be positive. Using again $\log(1 + u_k) < u_k, \forall u_k > 0$, we have $u_{k,\{1\}} - \log(1 + u_k) \geq u_{k,\{-1\}} - \log(1 + u_k) \geq u_{k,\{-1\}} - u_k > 0, \forall u_k > 0$. This implies that $\mathcal{H}_{\mathrm{a}}(u_k) > 0, \forall u_k > 0$, and we can conclude that $\partial^2\mathcal{H}(\tilde{p}_k, y_k)/\partial\tilde{p}_k^2 > 0$ as shown in (17). As $\mathcal{H}(\tilde{p}_k, y_k)$ is a convex function, Ξ_k and T_k are also convex. Furthermore, $(\mathrm{C6})_k$ can be easily transformed to a linear constraint as $\tilde{\rho}_k + \tilde{p}_k \leq \log(P_k^{\mathsf{max}})$, while $(\mathrm{C1})_k$, $(\mathrm{C5})_k$, and $(\mathrm{C7})_k$ can be converted to box constraints for $\tilde{f}_k^{\mathsf{u}}$, $\tilde{\omega}_k^{\mathsf{u}}$, and $\tilde{\rho}_k$, respectively. Therefore, $(\mathcal{P}_3)_k$ is a convex optimization problem with respect to $\tilde{\Omega}_{2,k} \cup \tilde{l}_k$.

APPENDIX C
PROOF OF PROPOSITION 4

We have the derivative $\partial\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)/\partial\omega_k^{\mathsf{f}} = \mathcal{H}_3(\omega_k^{\mathsf{f}}, d_k)/(\nu_{k,0}\omega_k^{\mathsf{f}}d_k - b_k^{\mathsf{in}})^2$, where $\mathcal{H}_3(\omega_k^{\mathsf{f}}, d_k) = d_k\left[-\tilde{\gamma}_{k,1}^{\mathsf{co,f}}b_k^{\mathsf{in}}(\gamma_{k,2}^{\mathsf{co,f}} + 1)\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} - \tilde{\gamma}_{k,3}^{\mathsf{co,f}}b_k^{\mathsf{in}} + \tilde{\gamma}_{k,1}^{\mathsf{co,f}}\nu_{k,0}\gamma_{k,2}^{\mathsf{co,f}}d_k\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}+1}\right]$. As $\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)$ is positive when $s_k^{\mathsf{m}} = 1$, it implies that $\nu_{k,0}\omega_k^{\mathsf{f}}d_k > b_k^{\mathsf{in}}$. Therefore, we can infer that $\mathcal{H}_3(\omega^{\mathsf{f}}, d_k) \leq -\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\left(\omega^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}}b_k^{\mathsf{in}}d_k - \tilde{\gamma}_{k,3}^{\mathsf{co,f}}b_k^{\mathsf{in}}d_k < 0, \forall\omega^{\mathsf{f}}, d_k$ if $\gamma_{k,2}^{\mathsf{co,f}} \leq 0$. Hence, $\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)$ achieves its minimal value at $\omega_k^{\mathsf{f}\star} = \omega_k^{\mathsf{max,f}}$ when $\gamma_{k,2}^{\mathsf{co,f}} \leq 0$. When $\gamma_{k,2}^{\mathsf{co,f}} > 0$, it can be verified that $\mathcal{H}_3\left(\omega_k^{\mathsf{f}\star}, d_k\right) = 0$ if and only if $d_k = \mathcal{H}_1\left(\omega_k^{\mathsf{f}\star}\right)$. On the other hand, the derivative of $\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)$ is $\frac{\partial\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)}{\partial\omega_k^{\mathsf{f}}} = -\frac{(\gamma_{k,2}^{\mathsf{co,f}}+1)}{\gamma_{k,2}^{\mathsf{co,f}}}\frac{b_k^{\mathsf{in}}(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}}))}{\nu_{k,0}\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}+2}} < 0$. So, $\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)$ is a monotonically decreasing function with respect to ω_k^{f}. Therefore, $\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)$ is minimized if $\omega_k^{\mathsf{f}} = \omega_k^{\mathsf{f}\star}$ satisfies (11).

APPENDIX D
PROOF OF LEMMA 2

First, it can be verified that $\frac{\partial\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)}{\partial d_k} = -\frac{b_k^{\mathsf{in}}\omega_k^{\mathsf{f}}\left[\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}}\right]}{\left(\nu_{k,0}\left(\omega_k^{\mathsf{f}}\right)d_k - b_k^{\mathsf{in}}\right)^2} = \mathcal{H}_2\left(\omega_k^{\mathsf{f}}, d_k\right)$. As $\partial\mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)/\partial\omega_k^{\mathsf{f}} < 0$ for all ω_k^{f}, $\omega_k^{\mathsf{f}\star}$ will not increase when $d_k > \bar{d}_{k,1}$ increases. When $\gamma_{k,2}^{\mathsf{co,f}} \leq 0$, $\omega_k^{\mathsf{f}\star} = \omega_k^{\mathsf{max,f}}$ as proved in Proposition 4. Therefore, $\mathcal{H}_2\left(\omega_k^{\mathsf{max,f}}, d_k\right)$ increases with respect to d_k. When $\gamma_{k,2}^{\mathsf{co,f}} > 0$, we will show that $\mathcal{H}_2\left(\omega_{k,1}^{\mathsf{f}\star}, d_k\right)\big|_{d_k = d_{k,1}} < \mathcal{H}_2\left(\omega_{k,2}^{\mathsf{f}\star}, d_k\right)\big|_{d_k = d_{k,2}}$, where $\bar{d}_{k,1} < d_{k,1} < d_{k,2}$ and $\omega_{k,i}^{\mathsf{f}\star}$ denotes the optimal value of ω_k^{f} when d_k is equal to $d_{k,i}$, for $i = 1, 2$.

Indeed, when ω_k^{f} is fixed, $\mathcal{H}_2\left(\omega_k^{\mathsf{f}}, d_k\right)$ is an increasing function of d_k. The second derivative of $\mathcal{H}_0\left(\omega_k^{\mathsf{f}}, d_k\right)$ when substituting $d_k = \mathcal{H}_1\left(\omega_k^{\mathsf{f}}\right)$ is given as $\frac{\partial\mathcal{H}_2\left(\omega_k^{\mathsf{f}}, d_k\right)}{\partial\omega_k^{\mathsf{f}}} = -\frac{\mathcal{H}_4\left(\omega_k^{\mathsf{f}}\right)}{\left(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}}\right)^2}$, where $\mathcal{H}_4\left(\omega_k^{\mathsf{f}}\right) = (\tilde{\gamma}_{k,1}^{\mathsf{co,f}})^2\left(\gamma_{k,2}^{\mathsf{co,f}}\right)^2\left(\omega_k^{\mathsf{f}}\right)^{2\gamma_{k,2}^{\mathsf{co,f}}}\left(\tilde{\gamma}_{k,1}^{\mathsf{co,f}}\left(\gamma_{k,2}^{\mathsf{co,f}}+1\right)\left(\omega_k^{\mathsf{f}}\right)^{\gamma_{k,2}^{\mathsf{co,f}}} + \tilde{\gamma}_{k,3}^{\mathsf{co,f}}\left(2\gamma_{k,2}^{\mathsf{co,f}}+1\right)\right) > 0$, for all ω_k^{f} when $\gamma_{k,2}^{\mathsf{co,f}} > 0$. Thus, it can be concluded that $\mathcal{H}_2\left(\omega_k^{\mathsf{f}}, d_k\right)$ is a decreasing function of ω_k^{f}. Furthermore, the optimal solution $\omega_k^{\mathsf{f}\star}$ monotonically decreases as d_k increases as shown in (11); hence, $\omega_{k,1}^{\mathsf{f}\star} \geq \omega_{k,2}^{\mathsf{f}\star}$. Therefore, we

have $\mathcal{H}_2\left(\omega_{k,1}^{\mathrm{f}\star}, d_k\right)\big|_{d_k=d_{k,1}} \leq \mathcal{H}_2\left(\omega_{k,2}^{\mathrm{f}\star}, d_k\right)\big|_{d_k=d_{k,1}} < \mathcal{H}_2\left(\omega_{k,2}^{\mathrm{f}\star}, d_k\right)\big|_{d_k=d_{k,2}}$.

REFERENCES

[1] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[2] T. T. Nguyen and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[3] T. T. Nguyen and B. L. Le, "Joint computation offloading and resource allocation in cloud based wireless HetNets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[4] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[5] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[6] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[7] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[8] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246–3257, Aug. 2018.

[9] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[10] M. Liu, Y. Mao, and S. Leng, "Cooperative fog-cloud computing enhanced by full-duplex communications," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 2044–2047, Oct. 2018.

[11] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for IoT," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 245–254, Apr. 2017.

[12] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Rate-distortion balanced data compression for wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 12, pp. 5072–5083, Jun. 2016.

[13] W. Zhang, Y. Wen, Y. J. Zhang, F. Liu, and R. Fan, "Mobile cloud computing with voltage scaling and data compression," in *Proc. IEEE Workshop. SPAWC*, Jul. 2017, pp. 1–5.

[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[15] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2017.

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

[17] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[18] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[19] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2018.

[20] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[21] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[22] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[23] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[24] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[25] T. T. Nguyen, L. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. Services Comput.*, to be published.

[26] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[27] M. Powell. *The Canterbury Corpus*. Accessed: Jan. 2001. [Online]. Available: http://corpus.canterbury.ac.nz/descriptions/#cantrbry

[28] Canadian Museum of Nature: Exploring our Natural Future. *Wallpaper*. Accessed: Dec. 2015. [Online]. Available: https://nature.ca/en/explore-nature/blogs-videos-more/wallpaper

[29] T. T. Nguyen, V. N. Ha, L. Le, and R. Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," Mar. 2019, *arXiv:1903.08566*. [Online]. Available: https://arxiv.org/abs/1903.08566

[30] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[31] CloudSigma. *Are We Stealing From You? Understanding CPU Steal Time in the Cloud*. Accessed: Nov. 2016. [Online]. Available: https://www.cloudsigma.com/understanding-cpu-steal-time-in-the-cloud/

[32] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[33] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 4, pp. 787–802, Dec. 2017.

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

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. PressDec. , 2004.

[36] S. Martello, *Knapsack Problems: Algorithms and Computer Implementations*. Hoboken, NJ, USA: Wiley, 1990.

[37] T. D. Hoang, L. B. Le, and T. Le-Ngoc, "Energy-efficient resource allocation for D2D communications in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 6972–6986, Sep. 2016.

[38] M. Sanjabi, M. Razaviyayn, and Z.-Q. Luo, "Optimal joint base station assignment and beamforming for heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1950–1961, Apr. 2014.

[39] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*, document 3GPP-TR- 36.814, 2010.

[40] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[41] Y. Yu, B. Krishnamachari, and V. P. Kumar, *Information Processing and Routing in Wireless Sensor Networks*. Singapore: World Scientific, 2006.

Ti Ti Nguyen (S'16) received the B.Eng. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, an addendum degree from Telecom Bretagne, France, in 2013, and the M.Eng. degree in embedded system from the University of Rennes 1, France, in 2015. He is currently pursuing the Ph.D. degree with the Institut National de la Recherche Scientifique–Énergie, Matériaux et Télécommunications (INRS-EMT), Université du Québec, Montréal, Québec, Canada. His current research interests include mobile edge computing, radio resource management, 5G new radio, and AI for wireless communications.

Vu Nguyen Ha (S'11–M'17) received the B.Eng. degree from the French Training Program for Excellent Engineers in Vietnam (PFIEV), Ho Chi Minh City University of Technology (HCMUT), Vietnam, an addendum degree from de École Nationale Supérieure des Télécommunications de Bretagne–Groupe des École des Télécommunications, Bretagne, France, in 2007, and the Ph.D. degree from the Institut National de la Recherche Scientifique–Énergie, Matériaux et Télécommunications (INRS-EMT), Université du Québec, Montréal, Québec, Canada, in 2017. From 2008 to 2011, he was a Research Assistant with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. He is currently a Post-Doctoral Fellow with École Polytechnique de Montréal, Montréal. His research interests include radio resource management and emerging enabling technologies for 5G wireless systems with special emphasis on heterogeneous small-cell networks, cloud RAN, massive MIMO communications, mmWave, and mobile edge computing. He is currently a recipient of the FRQNT Post-Doctoral Fellowship for International Researcher (PBEEE).

Long Bao Le (S'04–M'07–SM'12) received the B.Eng. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree in telecommunications from the Asian Institute of Technology, Thailand, in 2002, and the Ph.D. degree in electrical engineering from the University of Manitoba, Canada, in 2007. He was a Post-Doctoral Researcher with the Massachusetts Institute of Technology from 2008 to 2010 and the University of Waterloo from 2007 to 2008. Since 2010, he has been with the Institut National de la Recherche Scientifique (INRS), Université du Québec, Montréal, QC, Canada, where he is currently an Associate Professor. He is a coauthor of the books *Radio Resource Management in Multi-Tier Cellular Wireless Networks* (Wiley, 2013) and *Radio Resource Management in Wireless Networks: An Engineering Approach* (Cambridge University Press, 2017). His current research interests include smartgrids, radio resource management, and emerging enabling technologies for 5G and beyond wireless systems.

Dr. Le is a member of the editorial board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as a Technical Program Committee Chair/Co-Chair for different symposiums/tracks of several IEEE conferences, including IEEE WCNC, IEEE VTC, and IEEE PIMRC. He was an Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS from 2011 to 2016.

Robert Schober (S'98–M'01–SM'08–F'10) received the Diploma (University) and Ph.D. degrees in electrical engineering from the Friedrich-Alexander University of Erlangen–Nuremberg (FAU), Germany, in 1997 and 2000, respectively. From 2002 to 2011, he was a Professor and a Canada Research Chair with The University of British Columbia (UBC), Vancouver, Canada. Since January 2012, he has been an Alexander von Humboldt Professor and the Chair of digital communication with FAU. His research interests fall into the broad areas of communication theory, wireless communications, and statistical signal processing.

Dr. Schober is a fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He received several awards for his work, including the 2002 Heinz Maier Leibnitz Award of the German Science Foundation (DFG), the 2004 Innovations Award of the Vodafone Foundation for Research in Mobile Communications, a 2006 UBC Killam Research Prize, a 2007 Wilhelm Friedrich Bessel Research Award of the Alexander von Humboldt Foundation, the 2008 Charles McDowell Award for Excellence in Research from UBC, a 2011 Alexander von Humboldt Professorship, a 2012 NSERC E.W.R. Stacie Fellowship, and a 2017 Wireless Communications Recognition Award by the IEEE Wireless Communications Technical Committee. From 2012 to 2015, he served as an Editor-in-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS. He currently serves as the Chair of the ComSoc Fellow Evaluation Committee, a member of the Editorial Board of the PROCEEDINGS OF THE IEEE, and a ComSoc Director of Journals. Since 2017, he has been listed as a Highly Cited Researcher by the Web of Science.