Energy-Efficient Resource Allocation for Latency-Sensitive Mobile Edge Computing

Xihan Chen, Yunlong Cai, Liyan Li, Minjian Zhao, Benoit Champagne, and Lajos Hanzo

***Abstract*—Resource allocation algorithms are conceived for minimizing the energy consumption of multiuser mobile edge computing (MEC) systems operating in the face of interference channels, and where mobile users can offload their latency-sensitive tasks to the mobile edge server via a base station (BS). Latency-sensitive applications that benefit from MEC services can be divided into two major classes: 1) applications requiring uninterrupted execution and that cannot be fragmented and therefore require full offloading (FO); 2) applications which can benefit from fractional or partial offloading (PO). For each class of applications, we first formulate a joint optimization problem where the aim is to minimize the overall energy consumption across the sub-network subject to latency, transmission quality, computational budget and transmit power constraints. The proposed optimization problems are nonconvex, tightly coupled, and consequently challenging to solve. By exploiting binary relaxation, smooth approximation and auxiliary variables, we convert these problems into more tractable forms and subsequently develop novel algorithms based on the concave-convex procedure (CCCP) to solve them. Furthermore, by incorporating a measure of user priority, a reduced-complexity solution is proposed for the FO scheme. The benefits of our energy-efficient resource allocation algorithms for latency-sensitive MEC are demonstrated through simulations.**

***Index Terms*—Mobile edge computing, Resource allocation, Full offloading scheme, Partial offloading scheme, CCCP.**

I. Introduction

During the last decade, the explosive growth in the number, type and functionality of smart mobile devices has spurred the development of new mobile services [1]. A distinguishing feature of these services - such as augmented reality, real-time image recognition and natural language processing - is their computation-intensive and latency-sensitive nature, which poses very stringent requirements on both the computational and radio resources. In effect, the limited computational capability and battery life of mobile devices cannot guarantee the quality of experience (QoE) anticipated by the end users. To overcome these limitations, the telecommunication industry is increasingly turning towards mobile edge computing (MEC), a new network architecture that supports cloud computing along with compelling Internet services at the network edge [2]. This architecture has the potential of significantly reducing latency, whilst avoiding congestion and prolonging the battery charge life of mobile devices. This is achieved by the provision of ample computational and storage resources at the network edge, in order to support demanding computation-intensive and latency-critical applications, hence freeing from these tasks the resource-limited mobile devices [3]–[5].

A wide variety of latency-sensitive[1] applications can benefit from MEC services, with distinctive characteristics that call for different resource allocation and offloading mechanisms [3], [4], [7]. Indeed, the design objectives for various classes of latency-sensitive applications can be quite different. For example, the image compression applications focus on computation efficiency, while interactive applications such as mobile games emphasize the stability of operation [8]. In practice, due to hardware and software limitations, latency-sensitive applications can be divided into two main classes, i.e.:continuous-execution and data-partitioning-oriented applications [9]. In continuous-execution applications, it is not known in advance how long the application will take to be executed. Interactive applications such as mobile gaming, virtual reality, etc. belong to this class [10]. These applications must not be fragmented and hence they either have to be executed without interruption by user, or completely offloaded to the MEC server, which is termed as full offloading (FO). In data-partitioning-oriented applications, the amount of data to be processed is known in advance and therefore, fragmentation is allowed. In this case, one can take advantage of parallel processing, where a portion of the data is processed at the MEC server, while the remaining portion is processed locally by the user device, which is termed as partial-offloading (PO).

A. Prior Work

Considerable research efforts have been dedicated to the study of centralized resource allocation strategies for FO MEC systems. In [11] and [12], the authors derived the optimal resource allocation solution for a single-user MEC offloading system having multiple elastic tasks, with the goal of minimizing the average execution latency of all tasks under power constraints. In [13], game-theoretic decentralized computation offloading algorithms were proposed for wireless multi-user MEC systems. Sardellitti *et al.* [14] considered the joint optimization of radio and computational resources for computation offloading in a dense deployment scenario, in the presence of intercell interference. In order to reduce the

X. Chen, Y. Cai, L. Li and M. Zhao are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: chenxihan@zju.edu.cn; ylcai@zju.edu.cn; liyan_li@zju.edu.cn; mjzhao@zju.edu.cn).

B. Champagne is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: benoit.champagne@mcgill.ca).

L. Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/Noo4558/1, EP/PO34284/1, COALESCE, of the Royal Society's Global Challenges Research Fund Grant as well as of the European Research Council's Advanced Fellow Grant QuantCom.

[1] Latency sensitive applications are characterized by their bounded end-to-end delay requirements [6].

signalling overhead encountered in FO MEC systems, decentralized algorithms [15] have also been proposed. Wang *et al.* [16] conceived a decentralized algorithm based on the alternating direction multiplier method (ADMM) for computation offloading, resource allocation and Internet content caching optimization in heterogeneous MEC-aided wireless networks. Sergio *et al.* [17] invoked MEC-aided computation offloading techniques for millimeter wave (mmWave) communications and tackled the intermittent nature of mmWave propagation by relying on multiple links. Wang *et al.* [18] exploited the multi-antenna non-orthogonal multiple access (NOMA) technique for computation offloading in order to enable multiple users to share the allotted spectrum in the most effective manner. The authors of [19], [20] integrated the MEC system with wireless power transfer (WPT) techniques to provide numerous low-power wireless devices with enhanced computation capability and sustainable energy supply.

Recently, several methods have also been proposed to improve the performance of PO-based MEC systems. You *et al.* [21] investigated the optimal resource and offloading decision policy by minimizing the weighted sum of mobile energy consumption under a computation latency constraint in a multiuser time-division multiple access (TDMA)/orthogonal frequency-division multiple access (OFDMA) MEC system. Chen *et al.* [22] struck a trade-off between the execution delay and the network's energy consumption by proposing a joint cooperative computation and interactive communication framework for relay-assisted MEC systems. Ranadheera *et al.* [23] formulated the computational offloading problem as a minority game and developed a novel distributed server activation mechanism. They also addressed the randomness in both channel quality and user requests, and analyzed the statistical characteristics of the offloading delay. The authors of [24]–[26] combined MEC-based computation offloading techniques with unmaned aeiral vehicle (UAV) communications to provide high-quality services with reduced cost and high maneuverability. Ren *et al.* [27] investigated the latency-minimization problem in a multi-user TDMA MEC system relying on joint communication and computation resource allocation. Wang *et al.* [7] investigated partial computation offloading in conjunction with dynamic voltage and frequency scaling in MEC systems by considering the energy versus latency minimization problem.

B. Challenges and Contributions

To the best of our knowledge, the following challenges have not been well investigated in the literature on MEC: (i) How should we construct the multiuser MEC system model in the presence of interference channels, whilst exploiting the characteristics of different types of latency-sensitive applications? (ii) What is the optimal resource allocation strategy that minimizes the system's energy consumption under the constraints of latency, transmission quality, computational budget and transmit power? (iii) How can we alleviate the performance bottleneck caused by the channel state information (CSI) overhead and computational complexity issue due to the scalability in a large-scale mobile network? In this work, we aim to shed more light on these key issues by explicitly formulating the energy-efficient resource allocation problem for latency sensitive applications in multi-user MEC systems, and conceiving computationally efficient solution approaches for both the FO and PO schemes. The formulated optimization problems contain highly coupled nonconvex objective functions and constraints, which are difficult to handle. In particular, in the FO scheme, the presence of discrete binary constraints on the allocation variables makes the problem more strenuous. As for the PO scheme, the presence of nonconvex constraints involving the l_0-norm constitutes a different, yet no less simpler issue.

Our main contributions in addressing the above challenges are summarized as follows:

1) We construct the system model of multiuser MEC under interference channels for both continuous-execution applications (FO scheme) and data-partitioning-oriented applications (PO scheme). We then formulate the corresponding constrained energy consumption minimization (ECM) problems for efficient resource allocation, which we refer to as the ECM-FO and ECM-PO problems.
2) By applying a series of suitable transformations involving auxiliary variables and relaxation techniques, we first recast these challenging optimization problems into equivalent but more tractable forms. Specifically, we transform the binary variables into continuous ones in the FO scheme and approximate the nonconvex l_0-norm with smooth functions in the PO scheme.
3) We then propose new algorithms for the resultant problems based on the concave-convex procedure (CCCP) for handling the highly coupled terms and jointly optimize the MEC system parameters. In addition, we develop an ADMM-based algorithm which can be implemented in a *distributed* fashion to mitigate the effects caused by the CSI overhead and high complexity. Furthermore, by incorporating user priority parameters into the system model and invoking bisection search, a sub-optimal algorithm with reduced-complexity is conceived for the FO scheme.
4) We evaluate the performance of the proposed algorithms using in-depth simulations with diverse system parameter configurations. The results clearly demonstrate the convergence behavior of the new algorithms and the effect of various parameters on the system performance, whilst providing useful insights into the benefits of MEC for low-latency applications.

The rest of this paper is organized as follows. Section II presents the system model of a multiuser MEC system under interference channels. Section III formulates the ECM-FO and the ECM-PO problems for the continuous-execution and data-partitioning-oriented classes of applications, respectively. Section IV proposes an efficient CCCP-based FO (CFO) algorithm as well as a simplified algorithm for solving the ECM-FO problem. Section V proposes the CCCP-based PO (CPO) algorithm for solving the ECM-PO problem. Section VI discusses the ADMM-based distributed implementation of our proposed algorithms. Section VII presents the simulation results. The paper is concluded in Section VIII.

II. System model

We consider a multiuser MEC system as shown in Fig. 1, which consists of a multi-antenna BS connected to a MEC server via a high-speed optical link and N single-antenna mobile wireless users. The MEC server, which is located at the network edge, is equipped with abundant computing capabilities and core-network communication resources, allowing the mobile users to further benefit from cloud computing and Internet services. We assume that the mobile users have computationally intensive and delay sensitive tasks to be completed with the assistance of the MEC server. To this end, each user can offload some or all of its computational tasks to the MEC server via the BS, whilst executing the remaining tasks locally (i.e., on the mobile device). We consider the quasi-static scenario, where the mobile users remain unchanged during a computation offloading period, but may change across successive periods. For simplicity, we assume flat fading channels between the mobile users and the BS. Since the offloading decision of each mobile user is strongly influenced by the available channel conditions and computing resources at the MEC server, we next introduce details of the communication and computation models. For convenience, we list the key notations of this paper in Table I.

TABLE I: Summary of key notations

Notation	Description
N	Number of single-antenna mobile wireless users
W	System bandwidth
L_i	Size of the tasks before computation for user i
J_i	Average number of CPU cycles required to process each bit for user i
K	Coefficient depending on chip architecture
α_i	Offloading parameter for user i
P_i^{tr}	Transmit power for user i
P_i^l	Power consumption for local computation at user i
P_i^{idle}	Power consumption at user i in standby mode
$P_{i,\max}^t$	Transmit power budget for user i
f_i^l	Local CPUs computational speed for user i
f_i^c	Portion of MEC server's CPU resources allocated to user i
f_c	Available computational budget at the MEC server
$f_{\max}^l$	Limitation on the CPU speed for user i on the local user side
$f_{\max}^c$	Limitation on the CPU speed for user i on the MEC server side
Γ_i	SINR for user i
γ_i	SINR threshold for user i
R_i	Achievable transmission rate for user i
t_i^l	Potential execution time for the local computation at user i
t_i^c	Potential execution time of the MEC server for user i
t_i^{tr}	Potential transmission time of user i during the offloading period
$t_{i,F}^{\text{idle}}$	Duration of standby mode at user i for the FO scheme
$t_{i,P}^{\text{idle}}$	Duration of standby mode at user i for the PO scheme
t_i^d	Deadline for user i's task during the overall MEC phase
E_i^l	Energy consumption for the local computation at user i
$E_{i,F}^{\text{idle}}$	Energy consumption at user i in standby mode for the FO scheme
$E_{i,P}^{\text{idle}}$	Energy consumption at user i in standby mode for the PO scheme
E_i^{tr}	Transmission energy consumption of user i during the offloading period
E_i^F	Overall energy consumption of user i for the FO scheme
E_i^P	Overall energy consumption of user i for the PO scheme
θ	Measure of the offloading priority given by the network to user i
p	Parameter controlling the smoothness of approximation

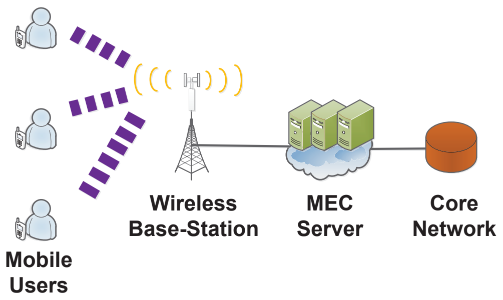


Fig. 1: MEC system model

A. Communication model

We denote α_i as the offloading parameter for user i. In the FO scheme, $\alpha_i \in \{0,1\}$ is a binary variable serving as the offloading indicator for user i, i.e.: $\alpha_i = 1$ if user i fully offloads its computational tasks to MEC server and $\alpha_i = 0$ if he/she executes all these tasks locally. In the PO scheme, $0 \leq \alpha_i \leq 1$ is a continuous variable which gives the fraction of the computation tasks of user i that is offloaded to the MEC server, while $(1-\alpha_i)$ gives the fraction of the tasks executed locally.

In both FO and PO schemes, task offloading requires the transmission of binary data from the corresponding users to the BS. Considering the mutual interference caused by other users and the background noise, the signal-to-interference-plus-noise-ratio (SINR) of user i after demodulation at the BS is given by

$$\Gamma_i = \frac{\beta P_i^{\text{tr}} H_i}{\sigma^2 + \sum\limits_{j \neq i}^{N} P_j^{\text{tr}} H_j}, \tag{1}$$

where H_i represents the channel gain[2] for user i, P_i^{tr} denotes the transmit power for user i, σ^2 denotes the variance of the additive channel noise, and β is a factor that depends on the specific type of modulation and signal processing being implemented at the physical layer. Then the achievable transmission rate (in bits/s) for user i is given by

$$R_i = W \log_2(1 + \Gamma_i), \tag{2}$$

where W is the system bandwidth (in Hz).

B. Computation model

We characterize the overall computation tasks at user i by the pair (L_i, J_i), where L_i (in bits) is the size of the tasks before computation, and J_i is the average number of CPU cycles required to process each bit. Below, we further develop the computation model for the local device and MEC server.

1) Local computing: The computation tasks are executed locally on each mobile device. As in [7], we model the power consumption of user i as $P_i^l = K(f_i^l)^3$, where f_i^l and K are the local CPU's computational speed (in cycles per second) and a coefficient depending on chip architecture, respectively. The potential execution time for the local computations at user i is given by

$$t_i^l = (1-\alpha_i) J_i L_i / f_i^l, \tag{3}$$

while the corresponding energy consumption is given by

$$E_i^l = P_i^l t_i^l = (1-\alpha_i) K J_i L_i (f_i^l)^2. \tag{4}$$

2) Edge computing: We recall that the MEC server is equipped with much more powerful hardware than the mobile devices. Hence, when user i offloads some of its tasks to the MEC server, the latter allocate a portion of its CPU resources to this user, as represented by the cycle frequency

[2] This includes the combined effect of radio transmission between the mobile user and BS, as well as other processing gains within the BS receiver.

f_i^c. Therefore, the potential execution time of the MEC server for user i is given by

$$t_i^c = \alpha_i J_i L_i / f_i^c. \tag{5}$$

In addition, we need to consider the time delay and energy consumption due to data transmission. Compared to the number of input bits sent by a mobile user to the MEC server for processing, we assume that the number of output bits returned by the MEC to that user as a result of such computations is relatively small. Therefore, we neglect the delay and the energy consumption incurred by the transmission of the output bits and only consider that of the input bits sent by the mobile user to the MEC server via the BS [7] [11] [12]. The potential transmission time and the energy consumption of user i during the offloading period will be

$$t_i^{\text{tr}} = \alpha_i L_i / R_i, \tag{6}$$

where the data rate R_i is taken from (2), and

$$E_i^{\text{tr}} = P_i^{\text{tr}} t_i^{\text{tr}}, \tag{7}$$

respectively. Besides, the time delay and energy consumption of mobile users in standby mode, i.e. waiting for the MEC server execution, should be considered.

In the FO scheme, user i is in standby mode with power consumption P^{idle} while the MEC executes its offloaded computation tasks. As depicted in Fig. 2, the duration of standby mode for user i in the FO scheme is given by $t_{i,F}^{\text{idle}} \triangleq t_i^c$, and the corresponding energy consumption by this user is given by

$$E_{i,F}^{\text{idle}} \triangleq P_i^{\text{idle}} t_{i,F}^{\text{idle}}. \tag{8}$$

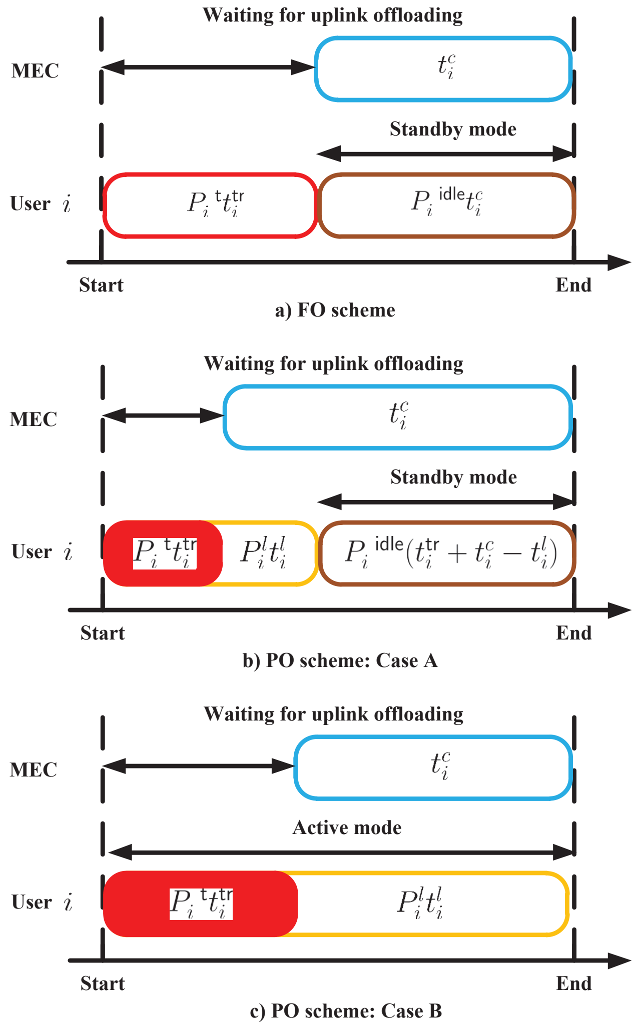
In the PO scheme, one can take advantage of parallel processing, where the process of local computing and offloading occur simultaneously. As depicted in Fig. 2, there exists two cases labeled as A and B. Case A corresponds to $t_i^l \leq t_i^c + t_i^{\text{tr}}$, where the duration of local processing is less than that of the data transmission plus edge computing. Here, the mobile user remains in standby mode until the data processing by the MEC server is completed. Case B corresponds to $t_i^l \geq t_i^c + t_i^{\text{tr}}$, where the duration of local processing exceeds that of data transmission plus edge computing, so that local device does not switch to standby mode. Therefore, the duration of the standby mode in the PO scheme is given by $t_{i,P}^{\text{idle}} \triangleq \max\{0, t_i^c + t_i^{\text{tr}} - t_i^l\}$. As in [21], [27], we assume that the performance of local computing remains unchanged during the offloading phase, and consequently, the energy consumption at mobile user i in standby mode is given by

$$E_{i,P}^{\text{idle}} \triangleq P_i^{\text{idle}} t_{i,P}^{\text{idle}}. \tag{9}$$

Finally, the overall energy consumption of a given user i during the overall MEC computing phase for the FO and PO schemes, respectively, can be expressed as

$$E_i^F = E_i^l + E_i^{\text{tr}} + E_{i,F}^{\text{idle}}, \tag{10}$$

$$E_i^P = E_i^l + E_i^{\text{tr}} + E_{i,P}^{\text{idle}}. \tag{11}$$


Waiting for uplink offloading
MEC
t_i^c
Standby mode
User i
$P_i^{\text{t}} t_i^{\text{tr}}$
$P_i^{\text{idle}} t_i^c$
Start
End
Waiting for uplink offloading
MEC
t_i^c
Standby mode
User i
$P_i^{\text{t}} t_i^{\text{tr}}$
$P_i^l t_i^l$
$P_i^{\text{idle}}(t_i^{\text{tr}} + t_i^c - t_i^l)$
Start
End
Waiting for uplink offloading
MEC
t_i^c
Active mode
User i
$P_i^{\text{t}} t_i^{\text{tr}}$
$P_i^l t_i^l$
Start
End

a) FO scheme

b) PO scheme: Case A

c) PO scheme: Case B

Fig. 2: The timeline of different offloading schemes.

III. Problem Formulation

In this section, we first consider the issue of achieving energy-efficient resource allocation for the multiuser MEC system. For convenience, let us define the following parameter vectors over which optimization will take place: $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$, $\mathbf{f}^l = [f_1^l, \ldots, f_N^l]^T$, $\mathbf{f}^c = [f_1^c, \ldots, f_N^c]^T$ and $\mathbf{P}^t = [P_1^{\text{tr}}, \ldots, P_N^{\text{tr}}]^T$. According to the communication and computation models presented in Section II, we can find that the offloading parameter $\boldsymbol{\alpha}$, the local computation capacity $\mathbf{f}^l$ and the computation capacity $\mathbf{f}^c$ at the MEC server, as well as the transmit power $\mathbf{P}^t$ of the mobile users are tightly coupled. If too many mobile users simultaneously choose to offload their tasks to the MEC server, this may lead to severe multiuser interference and scarcity of computational resources at the MEC server, which further reduce the transmission rate and incur long execution delays on the MEC server side. In turn, the mobile users will need to consume more energy in order to complete their latency-sensitive applications within a predefined time limit or deadline. In this case, it would be more beneficial for the mobile users to implement a larger fraction of their tasks locally. Based on such considerations, it appears necessary to perform *joint* optimization of the offloading parameters and computation resources as well as the transmit powers, in order to minimize the energy consumption for completing latency-sensitive applications. Below, we formulate the corresponding optimization problems in mathematical terms for the different types of applications, i.e., FO versus PO schemes.

A. Energy consumption minimization for the FO scheme

The energy consumption minimization problem based upon the FO scheme (ECM-FO) can be formulated as follows[3]:

$$\min_{\{\boldsymbol{\alpha},\mathbf{f}^c,\mathbf{f}^l,\mathbf{P}^t\}} \sum_{i=1}^{N} E_i^F(\boldsymbol{\alpha},\mathbf{f}^c,\mathbf{f}^l,\mathbf{P}^t), \tag{12a}$$

$$\text{s.t.} \quad (t_i^c + t_i^{\text{tr}}) + t_i^l \leq t_i^d, \tag{12b}$$

$$\Gamma_i \geq \alpha_i \gamma_i, \tag{12c}$$

$$\sum_{i=1}^{N} \alpha_i f_i^c \leq f_c, \tag{12d}$$

$$0 \leq P_i^t \leq P_{i,\max}^t, \tag{12e}$$

$$0 \leq f_i^c \leq f_{i,max}^c, \tag{12f}$$

$$0 \leq f_i^l \leq f_{i,\max}^l, \tag{12g}$$

$$\alpha_i \in \{0,1\}. \tag{12h}$$

In this formulation, constraint (12b) guarantees that the processing delay of user i's task during the overall MEC phase is less than the required deadline t_i^d. Constraint (12c) is added to guarantee the reliability of offloading transmission, where γ_i denotes the SINR threshold of user i. When user i offloads some of its tasks to the MEC server (i.e., $\alpha_i = 1$), the SINR of user i is required to be above a target γ_i, such that the achievable transmission rate of user i is no smaller than a given positive target. Consequently, the transmission time of user i during the offloading period is not too long. In this way, user i's tasks can be processed before the deadline t_i^d. In contrast, there is no need to impose quality of service (QoS) constraints on the offloading transmission when the computation tasks of user i are only executed locally on each mobile device (i.e., $\alpha_i = 0$). Constraint (12d) reflects the fact that the available computational budget at the MEC server is limited by f_c. Constraint (12e) represents the limitation in the transmit power budget for the offloading phase. Constraints (12f) and (12g) denote the limitations on the CPU speed for each user on the MEC server side and the local user side, respectively. Finally, (12h) reflects the fact that in the FO scheme, the offloading parameter α_i is binary.

B. Energy consumption minimization for the PO scheme

When the mobile users have data-partitioning-oriented applications to execute, it is preferable to employ the PO scheme to take advantage of parallel processing. For each user, a portion of the standing tasks is processed at the MEC server side while the remaining portion is processed at the local user side. We note several differences between the PO and FO schemes as follows: (i) the total processing delay does not simply depend on the local computing time or remote execution time [4]; (ii) to deal with these changes, we define the total processing delay as the maximum between the local computing and remote execution times, and make use of the l_0-"norm" to effectively characterize the resource occupancy. Thus, the energy consumption minimization problem for the PO scheme (ECM-PO) can be formulated as:

$$\min_{\{\boldsymbol{\alpha},\mathbf{f}^c,\mathbf{f}^l,\mathbf{P}^t\}} \sum_{i=1}^{N} E_i^P(\boldsymbol{\alpha},\mathbf{f}^c,\mathbf{f}^l,\mathbf{P}^t), \tag{13a}$$

$$\text{s.t.} \quad \max\{t_i^c + t_i^{\text{tr}}, t_i^l\} \leq t_i^d, \tag{13b}$$

$$\Gamma_i \geq |\alpha_i|_0 \gamma_i, \tag{13c}$$

$$\sum_{i=1}^{N} |\alpha_i|_0 f_i^c \leq f_c, \tag{13d}$$

$$\alpha_i \in [0,1], \tag{13e}$$

$$(12e) - (12g). \tag{13f}$$

Different from the ECM-FO problem, constraint (13b) guarantees that the parallel processing delay of users i's tasks during the overall MEC phase is less than the deadline t_i^d. Besides, (13e) reflects the fact that an arbitrary fraction of the computation tasks for each user can be off-loaded to the MEC, i.e., α_i is no longer a binary variable. In addition, the l_0-norm is introduced in constraints (13c) and (13d), where $|x|_0 = 1$ if $x \neq 0$ and $|x|_0 = 0$ otherwise.

Based on the ECM-FO and ECM-PO problem formulations, we will develop optimization approaches for devising energy-efficient resource allocation policies for the multiuser MEC system under interference channels. The main flow of ideas in developing these policies is illustrated in Fig. 3 while their precise details will be explained in the following sections.

IV. The Proposed full-offloading algorithm

In this section, we first analyze the structure of the ECM-FO problem (12) and elaborate on the challenges facing its solution. We then present the proposed CCCP-based FO algorithm, which is derived by first transforming the ECM-FO problem into a more tractable form and applying CCCP along with DC programming. At last, by incorporating a measure of user priority to improve the performance, a simplified algorithm is proposed that allows a further reduction of the computational complexity.

A. Challenges surrounding the ECM-FO problem

Solving problem (12) is quite challenging because the optimization variables are highly coupled in the nonconvex objective function and constraints. Besides, the user offloading indicator α_i is a discrete binary variable, which makes the feasible set nonconvex. Hence, we are faced with a mixed-integer linear programming (MILP) problem, which is usually considered as NP-hard. In principle, the standard exhaustive search for obtaining the optimal $\boldsymbol{\alpha}$ entails a complexity factor of $O(2^N)$, which represents an unacceptable computation overhead for large N since, for each permutation, the optimal transmit power and computational resource allocation

[3] For constraints involving user index i, it is implicitly assumed that the constraint must apply $\forall i \in \{1,...,N\}$.

[4] In the sequel, the term "remote execution time" shall refer to the transmission time plus the edge computing time.

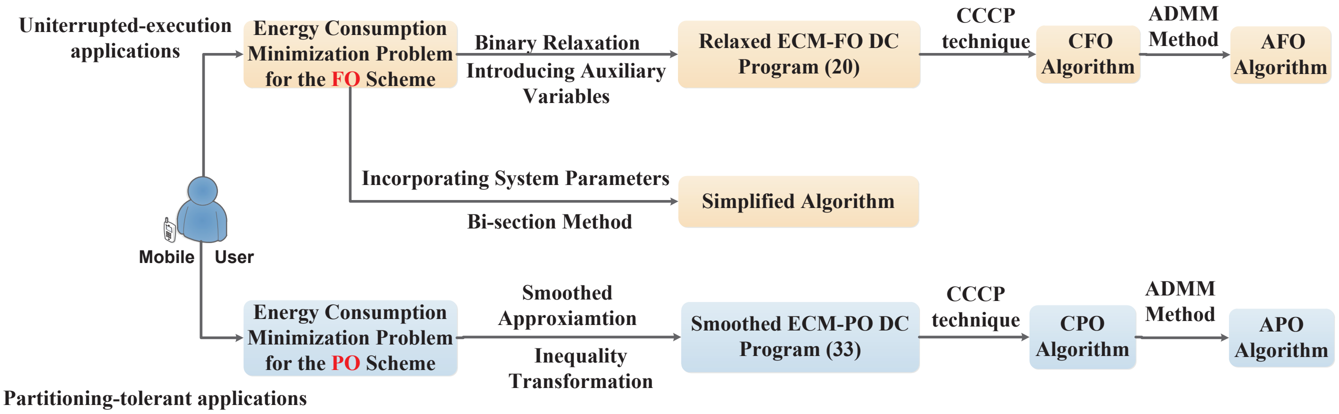

Uniterrupted-execution applications
Energy Consumption Minimization Problem for the FO Scheme
Binary Relaxation
Introducing Auxiliary Variables
Relaxed ECM-FO DC Program (20)
CCCP technique
CFO Algorithm
ADMM Method
AFO Algorithm
Incorporating System Parameters
Bi-section Method
Simplified Algorithm
Mobile
User
Energy Consumption Minimization Problem for the PO Scheme
Smoothed Approxiamtion
Inequality Transformation
Smoothed ECM-PO DC Program (33)
CCCP technique
CPO Algorithm
ADMM Method
APO Algorithm
Partitioning-tolerant applications

Fig. 3: Development of energy-efficient resource allocation policies for latency-sensitive MEC systems.

still need to be optimized. Hence, instead of employing an exhaustive search, we will propose an algorithm based on DC programming and the CCCP that significantly speeds up computations. For a brief review of these methods of optimization, the reader can consult Appendix A.

B. The CCCP-based FO algorithm

In the following, we develop an efficient CCCP-based FO (CFO) algorithm for the optimization of $\boldsymbol{\alpha}, \mathbf{f}^c$ and $\mathbf{P}^t$ in the non-convex ECM-FO problem (12). To this end, we first transform the problem into a more tractable form and then develop an efficient CCCP algorithm for its solution. In particular, in order to approximate the transformed problem as a convex one, an equivalent DC program is formulated.

1) Problem transformation: Before applying CCCP to the original problem (12), a suitable transformation for the later is necessary. The transformation consists of two steps, i.e.: introducing auxiliary variables and applying binary relaxation.

In order to transform the nonconvex feasible set of problem (12) into a convex set, we first introduce a number of auxiliary variables that will facilitate considerable simplifications of the problem. Firstly, we decompose the tangled constraint (12b) into multiple simpler constraints by introducing auxiliary variables : $z_{i,1}, z_{i,2}, z_{i,3}, R_i, \Delta_i, \varphi_i, s_i$ and u_i, so that we can transform the problem (12) into the following equivalent form:

$$\min_{\mathcal{X}} \sum_{i=1}^{N} \overline{E_i^F}(\mathcal{X}) \tag{14a}$$

$$\text{s.t. } z_{i,1} + z_{i,2} + z_{i,3} \le t_i^d, \tag{14b}$$

$$\alpha_i J_i L_i / f_i^c \le z_{i,1}, \tag{14c}$$

$$\alpha_i L_i / R_i \le z_{i,2}, \tag{14d}$$

$$(1 - \alpha_i) J_i L_i / f_i^l \le z_{i,3}, \tag{14e}$$

$$R_i \le W \log_2(1 + 1/\varphi_i), \tag{14f}$$

$$1/\varphi_i \le \Delta_i, \tag{14g}$$

$$\beta P_i^{\text{tr}} H_i / (\sigma^2 + \sum_{j \neq i}^{N} P_j^{\text{tr}} H_j) \le 1/\varphi_i, \tag{14h}$$

$$\alpha_i / f_i^c \le s_i, \tag{14i}$$

$$\alpha_i P_i^{\text{tr}} / R_i \le u_i, \tag{14j}$$

$$(12\text{d}) - (12\text{h}). \tag{14k}$$

where the modified objective function is given by

$$\overline{E_i^F}(\mathcal{X}) = E_i^l + J_i L_i P_i^{\text{idle}} s_i + L_i u_i, \tag{15}$$

with $\mathcal{X} = \{\alpha_i, f_i^c, f_i^l, R_i, \Gamma_i, \varphi_i, z_{i,1}, z_{i,2}, z_{i,3}, P_i^{\text{tr}}, s_i, u_i\}_{i=1}^N$ denoting the final set of optimization variables in the FO scheme. The equivalence between probelm (12) and (14) is proven in Appendix B. It is worth noting that the term $1/\varphi_i$ in constraint (14f) simplifies the subsequent application of the CCCP and avoids the use of successive convex approximations of logarithmic functions in toolbox CVX [30].

We next relax the domain of the discrete integer parameters $\alpha_i \in \{0, 1\}$ into a closed connected subset of the real axis, i.e, $\alpha_i \in [0, 1]$, In this context, continuous variable α_i may be interpreted as the offloading tendency of user i, that is: user i tends to offload when α_i is close to 1; while the same user tends to compute locally when α_i is close to 0. Following the introduction of auxiliary variables and binary relaxation, problem (12b) can be expressed as

$$\min_{\mathcal{X}} \sum_{i=1}^{N} \overline{E_i^F}(\mathcal{X}) \tag{16a}$$

$$\text{s.t. } \alpha_i \in [0,1], \tag{16b}$$

$$(12\text{d}) - (12\text{g}), (14\text{b}) - (14\text{j}). \tag{16c}$$

2) CCCP-based algorithm: Even though we converted (12) into the more tractable problem (16), the latter remains to solve due to the nonconvexity and coupling of constraints. The special bi-linear structure of these coupling constraints, which actually occurs in a variety of joint design problems, can be conveniently handled by means of the following lemma.

Lemma 1: The bi-linear funciton $g(x_1, x_2) = x_1 x_2$ can be expressed as the following DC program:

$$g_i(x_1, x_2) = \frac{1}{2}(x_1 + x_2)^2 - \frac{1}{2}(x_1^2 + x_2^2). \tag{17}$$

Lemma 1 provides a simple way of handling bi-linear coupled constraints via the introduction of an equivalent DC program. We note that bi-linear constraints occur in many design and optimization problems of various natures in science and engineering, and as such, *Lemma 1* can be applied to the solution of these other problems as well.

To make efficient use of the CCCP, we convert the problem (16) into a general form of DC programs with the aid of *Lemma 1*. The equivalent DC program is given by

$$\min_{\mathcal{X}} \sum_{i=1}^{N} \overline{E_i^F}(\mathcal{X}) \tag{18a}$$

$$\text{s.t. } 2\alpha_i J_i L_i + (f_i^c)^2 + z_{i,1}^2 - (f_i^c + z_{i,1})^2 \leq 0, \tag{18b}$$

$$2\alpha_i L_i + {R_i}^2 + z_{i,2}^2 - (R_i + z_{i,2})^2 \leq 0, \tag{18c}$$

$$2(1-\alpha_i) J_i L_i + (f_i^l)^2 + z_{i,3}^2 - (f_i^l + z_{i,3})^2 \leq 0, \tag{18d}$$

$$2(\sigma^2 + \sum_{j \neq i}^{N} P_j^{\text{tr}} H_j) + {\varphi_i}^2 + (P_i^{\text{tr}} H_i)^2 - (\varphi_i + P_i^{\text{tr}} H_i)^2 \leq 0, \tag{18e}$$

$$z_{i,1} + z_{i,2} + z_{i,3} \leq t_i^d, \tag{18f}$$

$$\begin{aligned} &(\alpha_i \gamma + \sigma^2 + \sum_{j \neq i}^{N} P_j^{\text{tr}} H_j)^2 - (\alpha_i \gamma)^2 - (\sigma^2 + \sum_{j \neq i}^{N} P_j^{\text{tr}} H_j)^2 \\ &- 2P_i^{\text{tr}} H_i \leq 0, \end{aligned} \tag{18g}$$

$$\sum_{i=1}^{N} (\alpha_i + f_i^c)^2 - (\alpha_i^2 + (f_i^c)^2) \leq 2f_c, \tag{18h}$$

$$2\alpha_i + (f_i^c)^2 + s_i^2 - (f_i^c + s_i)^2 \leq 0, \tag{18i}$$

$$(\alpha_i + P_i^{\text{tr}})^2 + R_i^2 + u_i^2 - \alpha_i^2 - (P_i^{\text{tr}})^2 - (R_i + u_i)^2 \leq 0, \tag{18j}$$

$$(12\text{e}), (12\text{f}), (14\text{f}), (16\text{b}). \tag{18k}$$

Based on the CCCP concept, we approximate the nonconvex part in both the constraints and the objective function with the aid of linearization. For example, focusing on (14f), we obtain

$$R_i - W \log_2(1 + \frac{1}{\hat{\varphi}_i}) + \frac{W(\varphi_i - \hat{\varphi}_i)}{\hat{\varphi}_i^2 + \hat{\varphi}_i} \leq 0, \tag{19}$$

where $\hat{\varphi}_i$ represent the current point of variable φ_i.

Finally, by invoking the CCCP (see Appendix A and especially eq. (46)), an iterative algorithm is obtained for the solution of (18)) whereby at the current iteration, only the simplified convex optimization problem below needs to be solved:

$$\min_{\mathcal{X}} \sum_{i=1}^{N} \overline{E_i^F}(\mathcal{X}) \tag{20a}$$

$$\begin{aligned} \text{s.t. } &2\alpha_i J_i L_i + (f_i^c)^2 + z_{i,1}^2 - (\hat{f}_i^c + \hat{z}_{i,1})^2 \\ &- 2(\hat{f}_i^c + \hat{z}_{i,1})(f_i^c + z_{i,1} - \hat{f}_i^c - \hat{z}_{i,1}) \leq 0, \end{aligned} \tag{20b}$$

$$\begin{aligned} &2\alpha_i L_i + {R_i}^2 + z_{i,2}^2 - (R_i + z_{i,2})^2 \\ &- 2(\hat{R}_i + \hat{z}_{i,2})(R_i + z_{i,2} - \hat{R}_i + \hat{z}_{i,2}) \leq 0, \end{aligned} \tag{20c}$$

$$\begin{aligned} &2(1-\alpha_i) J_i L_i + (f_i^l)^2 + z_{i,3}^2 - (\hat{f}_i^l + \hat{z}_{i,3})^2 \\ &- 2(\hat{f}_i^l + \hat{z}_{i,3})(f_i^l + z_{i,3} - \hat{f}_i^l + \hat{z}_{i,3}) \leq 0, \end{aligned} \tag{20d}$$

$$\begin{aligned} &2(\sigma^2 + \sum_{j,j \neq i}^{N} P_j^{\text{tr}} H_j) + {\varphi_i}^2 + (P_i^{\text{tr}} H_i)^2 - (\hat{\varphi}_i + \hat{P}_i^{\text{tr}} H_i)^2 - 2(\hat{\varphi}_i \\ &+ \hat{P}_i^{\text{tr}} H_i)(\varphi_i - \hat{\varphi}_i) - 2H_i(\hat{\varphi}_i + \hat{P}_i^{\text{tr}} H_i)(P_i^{\text{tr}} - \hat{P}_i^{\text{tr}}) \leq 0, \end{aligned} \tag{20e}$$

$$\begin{aligned} &(\alpha_i \gamma + \sigma^2 + \sum_{j \neq i}^{N} P_j^{\text{tr}} H_j)^2 - (\hat{\alpha}_i \gamma)^2 - (\sigma^2 + \sum_{j,j \neq i}^{N} P_j^{\text{tr}} H_j)^2 - 2P_i^{\text{tr}} H_i \\ &- 2\gamma \hat{\alpha}_i (\alpha_i - \hat{\alpha}_i) - 2\sum_{j,j \neq i}^{N} H_j (\sigma^2 + \sum_{j \neq i}^{N} \hat{P}_j^{\text{tr}} H_j)(P_j^{\text{tr}} - \hat{P}_j^{\text{tr}}) \leq 0, \end{aligned} \tag{20f}$$

$$\begin{aligned} &\sum_{i=1}^{N} (\alpha_i + f_i^c)^2 - \hat{\alpha}_i^2 - (\hat{f}_i^c)^2 - 2\hat{\alpha}_i(\alpha_i - \hat{\alpha}_i) \\ &- 2\hat{f}_i^c (f_i^c - \hat{f}_i^c) \leq 2f_c, \end{aligned} \tag{20g}$$

$$\begin{aligned} &2\alpha_i + (f_i^c)^2 + s_i^2 - (f_i^c + s_i)^2 - 2(\hat{f}_i^c + \hat{s}_i)(f_i^c + s_i \\ &- \hat{f}_i^c - \hat{s}_i) \leq 0, \end{aligned} \tag{20h}$$

$$\begin{aligned} &(\alpha_i + P_i^{\text{tr}})^2 + R_i^2 + u_i^2 - \hat{\alpha}_i^2 - (\hat{P}_i^{\text{tr}})^2 - 2(\hat{\alpha}_i)(\alpha_i - \hat{\alpha}_i) - 2\hat{P}_i^{\text{tr}}(P_i^{\text{tr}} \\ &- \hat{P}_i^{\text{tr}}) - (\hat{R}_i + \hat{u}_i)^2 - 2(\hat{R}_i + \hat{u}_i)(R_i + u_i - \hat{R}_i - \hat{u}_i) \leq 0, \end{aligned} \tag{20i}$$

$$(12\text{e}), (12\text{f}), (16\text{b}), (18\text{f}), (19). \tag{20j}$$

where $\hat{\mathcal{X}} = \{\hat{\alpha}_i, \hat{f}_i^c, \hat{f}_i^l, \hat{R}_i, \hat{\Delta}_i, \hat{\varphi}_i, \hat{z}_{i,1}, \hat{z}_{i,2}, \hat{z}_{i,3}, \hat{P}_i^{\text{tr}}, \hat{s}_i, \hat{u}_i\}_{i=1}^{N}$ represents the current point of the set of optimization variables in the FO scheme.

Problem (20) can be efficiently solved by the convex programming toolbox CVX [30]. The implementation of the proposed CFO algorithm is summarized in Table 1. Repeated application of the CCCP iteration will eventually lead to a stationary solution of problem (20) [31]. We can show that the limit point of the iterates generated by the proposed CFO algorithm also satisfies the KKT conditions of the DC program (18) and thus converges to a local optimal solution of problem (12). The proof is similar to that of *Lemma 2* and *Theorem 1* in [32], and we therefore omit the details. Denoting by I_1 the number of iterations required by the CCCP algorithm, it can be shown that the complexity of the CFO algorithm is $O(I_1\sqrt{8N+1}(4N^2 + 84N + 14))$, where the factor $\sqrt{8N+1}(4N^2 + 84N + 14)$ comes from the application of a generic interior-point method for solving the convex optimization at each iteration [33].

Algorithm 1 Proposed CCCP-based FO (CFO) iterative algorithm

```
1. **Initialization**: Define the tolerance of accuracy $\delta_1$ and the maximum number of iteration $N_{\text{max}}$. Initialize the algorithm with a feasible point $\mathcal{X}^0$. Set the iteration number $t = 0$.
2. **Repeat**
    – Solve the convex optimization problem (20) with the affine approximation, and assign the solution to $\mathcal{X}^{t+1}$.
    – Update the iteration number: $t \leftarrow t + 1$
3. **Until** $|\overline{E^F}(\mathcal{X}^{t+1}) - \overline{E^F}(\mathcal{X}^t)| \leq \delta_1$ or reaching the max iteration number.
```

C. Simplified algorithm

In the previous subsection, we have proposed the CFO algorithm for solving the ECM-FO problem. While this new algorithm achieves near optimal performance, it still incurs a high computational burden because the CCCP calls for the solution of an iterative sequence of complex convex optimization problems, where the computation bottelneck results from the SINR related constraints and DC programs. Here, motivated by these considerations, a simplified algorithm is proposed for solving the ECM-FO problem. With the double goal of improving system performance and reducing design complexity, we introduce a new set of system parameters, i.e.

$$\theta_i = H_i/(J_i L_i f_i^l), \ i \in \{1, \dots, N\}. \tag{21}$$

Specifically, θ_i will be used here as a measure of the priority given by the network to user i in offloading its tasks to the MEC. That is, a user with a larger value of parameter θ_i will have a higher priority to offload their jobs. Considering (21), a user with a higher channel power gain H_i, smaller job size $J_i L_i$, or smaller local computation capacity will be given a higher priority to offload [34]. In the sequel, to simplify presentation, we shall assume that the users have been indexed by increasing order of priority, i.e. $\theta_1 \leq \theta_2 \leq \dots \leq \theta_N$ to determine the offloading users[5].

Here, we adopt a simple offloading procedure derived from the bi-section method and making use of the priority parameters. Specifically, once a user has been selected for offloading, it will adopt the maximum transmit power. Meanwhile, the computation resources at the MEC server are equally allocated to the selected users. Let $N_{\mathrm{m}ax}$ be the maximum number of offloading mobile users such that problem (12) remains feasible. To find $N_{\mathrm{m}ax}$ in an efficient bisection manner, we have to solve the following convex feasibility problem,

$$\begin{aligned} \mathbf{F1}(i): \ & \text{Find } U^{[i]} \\ & \text{s.t. (12b)} - \text{(12d)}, \text{(12f)}, \text{(12g)}, \end{aligned} \tag{22}$$

where $U^{[i]} = \{i + 1, ..., N\}$ represents the set of selected users. That is, the last $N - i$ mobile users, sorted by their θ_i values, are chosen for offloading their tasks via the BS to the MEC server. We note that if problem $\mathbf{F1}(i)$ is feasible, then a feasible solution exists to $\mathbf{F1}(j)$ for all $j \leq i$. Therefore, determining the largest $N = N_{\text{max}}$ that results in a feasible solution of the problem $\mathbf{F1}(i)$ can be accomplished by solving no more than $O(1+[\log(1+\text{N})])$ such feasibility problems via the bisection search. The simplified algorithm resulting from these considerations is summarized in Table 2.

Algorithm 2 Simplified algorithm

```
1. **Initialization**: Obtain the parameters $H_i, L_i, f_i^l, P_{i,max}^t, P_i^{\text{idle}}$ and sort the users in ascending order of priority: $\theta_1 \leq \theta_2 \leq ... \leq \theta_K$. Initialize $N_{\text{low}} = 0, N_{\text{up}} = 0, i = 0$.
2. **Repeat**
    – set $i \leftarrow [\frac{N_{\text{low}} + N_{\text{up}}}{2}]$.
    – Solve the feasibility problem $\mathbf{F1}(i)$: If it is feasible, set $N_{\text{low}} = i$; otherwise, set $N_{\text{up}} = i$.
3. **Until** $N_{\text{up}} - N_{\text{low}} = 1$, obtain $N_{\text{ max}} = N_{\text{up}}$ and obtain the offloading user set $U^\star = \{N_{\text{max}} + 1, ..., N\}$.
```

V. Proposed partial-offloading algorithm

In this section, we first show that the ECM-PO problem can be converted into a DC program after approximating various l_0-norms with smooth functions and applying adequate inequality transformations [29]. Finally, a CCCP-based PO (CPO) algorithm is proposed to efficiently solve the resulting problem.

A. Problem transformation

In addition to the challenges previously faced in the solution of problem (12) in the previous section, especially highly coupled nonconvex objective functions and constraints, the presence of multiple non-convex l_0-norms in (13) further complicates its solution. Before applying CCCP to problem (13), a suitable transformation consisting of two main steps is necessary, i.e.: smoothed l_0-norm approximation and introduction of auxiliary variables.

In this work, we approximate the discontinuous l_0-norm in constraints (13c) and (13d) with a smooth surrogate function [35]. While several such functions have been proposed [36], here we make use of the following approximation:

$$u_p(x) = 1 - e^{(-|x|/p)}, \ p > 0, \tag{23}$$

where $p > 0$ is a parameter controlling the smoothness of approximation. In particular, the use of a smaller p leads to a better approximation but reduced smoothness for $u_p(x)$ and vice versa. This surrogate function has the additional property of providing a lower bound of the l_0-norm function, which is preferred for faster convergence [36]. Before applying the smooth approximation $|x|_0 \approx u_p(x)$ to the constraint (13d), we first decompose the latter into several simpler constraints by introducing auxiliary variables $\{\mu_i\}_{i=1}^N$:

$$f_i^c \leq \mu_i + (1 - |\alpha_i|_0) f_{\max}^c, \tag{24}$$

$$\sum_{i=1}^N \mu_i \leq f_c, \tag{25}$$

Combining (24) and (25), (13d) can be expressed as:

$$\sum_{i=1}^N f_i^c \leq f_c + f_{\max}^c \sum_{i=1}^N (1 - |\alpha_i|_0). \tag{26}$$

[5] This is always possible since the various parameters entering the definition of θ_i in (27) are assumed known.

Now, using the smooth approximation (23) for the l_0-norm, we can replace (13c) and (13d) by the following:

$$\sum_{i=1}^{N} f_i^c \le f_c + f_{\max}^{\mathrm{c}} \sum_{i=1}^{N} e^{-\alpha_i/p}, \tag{27}$$

$$\Gamma_i \ge G_i \ge (1-e^{-\alpha_i/p})\gamma_i, \tag{28}$$

where G_i is an auxiliary variable. As for the other coupled constraints, we can simplify them by proceeding as in Section IV-B2. Ultimately, problem (13) can be converted into the following form:

$$\min_{\mathcal{Y}} \sum_{i=1}^{N} \overline{E_i^P}(\mathcal{Y}) \tag{29a}$$

$$\text{s.t. } \alpha_i J_i L_i / f_i^c \le z_{i,1}, \tag{29b}$$

$$\alpha_i L_i / R_i \le z_{i,2}, \tag{29c}$$

$$R_i \le W \log_2(1+1/\varphi_i), \forall i, \tag{29d}$$

$$2(\sigma^2 + \sum_{j\neq i}^{N} P_j^{\mathrm{tr}} H_j) + {\varphi_i}^2 + (P_i^{\mathrm{tr}} H_i)^2 - (\varphi_i + P_i^{\mathrm{tr}} H_i)^2 \le 0, \tag{29e}$$

$$z_{i,1} + z_{i,2} \le t_i^d, \tag{29f}$$

$$(1-\alpha_i) J_i L_i / f_i^l \le z_{i,3}, \tag{29g}$$

$$z_{i,3} \le t_i^d, \tag{29h}$$

$$\begin{aligned}&(G_i + \sigma^2 + \sum_{j\neq i}^{N} P_j^{\mathrm{tr}} H_j)^2 - G_i^2 - (\sigma^2 + \sum_{j\neq i}^{N} P_j^{\mathrm{tr}} H_j)^2 \\ &- 2P_i^t H_i \le 0,\end{aligned} \tag{29i}$$

$$G_i \ge (1-e^{-\alpha_i/p})\gamma_i, \tag{29j}$$

$$0 \le z_{i,1} + z_{i,2} - z_{i,3} \le s_i, \tag{29k}$$

$$\alpha_i P_i^{\mathrm{tr}} / R_i \le u_i, \tag{29l}$$

$$(13\text{e}), (13\text{f}), (27). \tag{29m}$$

In (29a), the modified objective function $\overline{E_i^P}(\mathcal{Y})$ for user i is given by

$$\overline{E_i^P}(\mathcal{Y}) = E_i^l + P_i^{\mathrm{idle}} s_i + L_i u_i, \tag{30}$$

where $\mathcal{Y} = \{\alpha_i, f_i^c, f_i^l, R_i, \varphi_i, z_{i,1}, z_{i,2}, z_{i,3}, G_i, P_i^{\mathrm{tr}}, s_i, u_i\}_{i=1}^N$ represents the set of optimization variables in the PO scheme.

B. The CCCP-based PO algorithm

To efficiently apply the CCCP technique, we first convert problem (29) into a general form of DC program by means of *Lemma 1*, i.e.:

$$\min_{\mathcal{Y}} \sum_{i=1}^{N} \overline{E_i^P}(\mathcal{Y}) \tag{31a}$$

$$\text{s.t. } 2\alpha_i J_i L_i + f_{i,c}^2 + z_{i,1}^2 - (f_{i,c} + z_{i,1})^2 \le 0, \tag{31b}$$

$$2\alpha_i L_i + {R_i}^2 + z_{i,2}^2 - (R_i + z_{i,2})^2 \le 0, \tag{31c}$$

$$2(1-\alpha_i) J_i L_i + (f_i^l)^2 + (z_{i,3})^2 - (f_i^l + z_{i,3})^2 \le 0, \tag{31d}$$

$$(\alpha_i + P_i^{\mathrm{tr}})^2 + R_i^2 + u_i^2 - \alpha_i^2 - (P_i^{\mathrm{tr}})^2 - (R_i + u_i)^2 \le 0, \tag{31e}$$

$$(29\text{d}) - (29\text{f}), (29\text{h}) - (29\text{m}) \tag{31f}$$

To efficiently solve problem (31) while decreasing the objective value, we then linearize the nonconvex part in both constraints and the objective function to tackle the nonconvexity. Let us focus on constraint (29e) as an example. By linearizing the nonconvex term $-(\varphi_i + P_i^{\mathrm{tr}} H_i)^2$ around the current point $\{\hat{P}_i^{\mathrm{tr}}, \hat{\varphi}_i\}$, we approximate (29e) as the following constraint

$$\begin{aligned}&2(\sigma^2 + \sum_{j,j\neq i}^{N} P_j^{\mathrm{tr}} H_j) + {\varphi_i}^2 + (P_i^{\mathrm{tr}} H_i)^2 - (\hat{\varphi}_i + \hat{P}_i^{\mathrm{tr}} H_i)^2 - 2(\hat{\varphi}_i \\ &+ \hat{P}_i^{\mathrm{tr}} H_i)(\varphi_i - \hat{\varphi}_i) - 2H_i(\hat{\varphi}_i + \hat{P}_i^{\mathrm{tr}} H_i)(P_i^{\mathrm{tr}} - \hat{P}_i^{\mathrm{tr}}) \le 0.\end{aligned} \tag{32}$$

Hence, with the aid of CCCP concepts, problem (31) can be reformulated as the following convex optimization problem:

$$\min_{\hat{\mathcal{Y}}} \sum_{i=1}^{N} \overline{E_i^P}(\hat{\mathcal{Y}}) \tag{33a}$$

$$\begin{aligned}\text{s.t. } &(G_i + \sigma^2 + \sum_{j\neq i}^{N} P_j^{\mathrm{tr}} H_j)^2 - \hat{G}_i^2 - (\sigma^2 + \sum_{j\neq i}^{N} \hat{P}_j^{\mathrm{tr}} H_j)^2 - 2\hat{G}_i(G_i \\ &- \hat{G}_i) - 2\sum_{j\neq i}^{N} H_j(\sigma^2 + \sum_{j\neq i}^{N} \hat{P}_j^{\mathrm{tr}} H_j)(P_j^{\mathrm{tr}} - \hat{P}_j^{\mathrm{tr}}) - 2P_i^{\mathrm{tr}} H_i \le 0,\end{aligned} \tag{33b}$$

$$\gamma + \frac{\gamma e^{-\hat{\alpha}_i/p}}{p}(\alpha_i - \hat{\alpha}_i) - G_i - \gamma e^{-\hat{\alpha}_i/p}, \tag{33c}$$

$$\sum_{i=1}^{N} f_i^c - f_c + \frac{f_{\max}^{\mathrm{c}}}{p} \sum_{i=1}^{N} e^{-\hat{\alpha}_i/p}(\alpha_i - \hat{\alpha}_i) \le 0, \tag{33d}$$

$$(13\text{e})\text{–}(13\text{f}), (19), (20\text{b})\text{–}(20\text{e}), (20\text{i}), (29\text{f}), (29\text{h}), (29\text{k}), (32), \tag{33e}$$

where $\hat{\mathcal{Y}} = \{\hat{\alpha}_i, \hat{f}_i^c, \hat{f}_i^l, \hat{R}_i, \hat{\varphi}_i, \hat{z}_{i,1}, \hat{z}_{i,2}, \hat{z}_{i,3}, \hat{G}_i, \hat{P}_i^{\mathrm{tr}}, \hat{s}_i, \hat{u}_i\}_{i=1}^N$ represents the values of the optimization variables in the PO scheme at the current iteration. Similar to the CFO algorithm in Section IV-B2, this problem can be efficiently solved by employing the convex programming toolbox CVX [30], and the convergence is guaranteed. The proposed CPO algorithm is summarized in Table 3.

Algorithm 3 Proposed CCCP-based PO (CPO) iterative algorithm

```
1. Initialization: Define the tolerance of accuracy $\delta_2$ and the maximum number of iterations $N_{\max}$. Initialize the algorithm with a feasible point $\mathcal{Y}^0$. Set the iteration number $t = 0$.
2. Repeat
     – Solve the convex optimization problem (33), and assign the solution to $\mathcal{Y}^{t+1}$.
     – Update the iteration number: $t \leftarrow t + 1$
3. Until $|\overline{E^P}(\mathcal{Y}^{t+1}) - \overline{E^P}(\mathcal{Y}^t)| \le \delta_2$ or reaching the maximum number of iterations.
```

VI. ADMM-based Distributed Implementation

In the previous sections, we proposed the CFO and CPO algorithms for solving the ECM-FO and ECM-PO problems in a centralized manner, respectively. In practice, the computational complexity of these algorithms increases rapidly with the number of users, while their implementation requires full CSI knowledge at BS.

To alleviate these issues, it is of interest to design a distributed implementation method, which enables the BS and all users within the system to simultaneously participate in the computation. In this section, we develop extensions of the proposed CFO and CPO algorithms based on the ADMM, which can be implemented in a distributed manner and significantly reduce the peak computational complexity compared to a centralized implementation.

A. *Proposed distributed algorithms*

The main obstacle in designing a distributed algorithm lies in solving the approximated convex problem (20), which is part of the CCCP. In the following, we focus on the FO scenario to expose our proposed ADMM-based approach to the derivation of a distributed algorithm; however, a similar approach can be applied to obtain a distributed algorithm for the PO scenario.

Step 1 (Problem decomposition): To enable each user to participate in the computation, their optimization variables should be separable so that the original problem can be decomposed into N independent subproblems to be solved individually among the users. However, the optimization variables in $\mathcal{X}$ are highly coupled in the constraints of problem (20), which renders such a decomposition impractical.

To overcome this difficulty, we first define the optimization variables in $\mathcal{X}$ as the global optimization variables, and then introduce local copies of the global variables in $\mathcal{X}$ at each user. Specifically, we define $\tilde{\mathcal{X}} = \{\tilde{\mathcal{X}}_n\}_{n=1}^N$ with $\tilde{\mathcal{X}}_n = \{\tilde{\alpha}_{n,i},\tilde{f}^c_{n,i},\tilde{f}^l_{n,i},\tilde{R}_{n,i},\tilde{\Gamma}_{n,i},\tilde{\varphi}_{n,i},\tilde{z}^n_{i,1},\ \tilde{z}^n_{i,2},\tilde{z}^n_{i,3},\tilde{P}^{\text{tr}}_{n,i},\tilde{s}_{n,i},\tilde{u}_{n,i}\}_{i=1}^N$ as the local copy of $\hat{\mathcal{X}}$ for user n, where following consensus constraints apply: $\tilde{\alpha}_{n,i} = \alpha_i$, $\tilde{f}^c_{n,i} = f^c_i$, $\tilde{f}^l_{n,i} = f^l_i$, $\tilde{R}_{n,i} = R_i$, $\tilde{\Gamma}_{n,i} = \Gamma_i$, $\tilde{\varphi}_{n,i} = \varphi_i$, $\tilde{z}^n_{i,1} = z^n_{i,1}$, $\tilde{z}^n_{i,2} = z^n_{i,2}$, $\tilde{z}^n_{i,3} = z^n_{i,3}$, $\tilde{P}^{\text{tr}}_{n,i} = P^{\text{tr}}_i$, $\tilde{s}_{n,i} = s_i$, $\tilde{u}_{n,i} = u_i, \forall i, n$.

With the above notations, problem (20) can be equivalently reformulated as

$$\min_{\tilde{\mathcal{X}}} \sum_{n=1}^{N} \overline{E^F_n}(\tilde{\mathcal{X}}_n) \tag{34a}$$

$$\text{s.t. } 2\tilde{\alpha}_{n,i}J_iL_i + \left(\tilde{f}^c_{n,i}\right)^2 + z^2_{i,1} - (\hat{f}^c_{n,i} + \hat{z}^n_{i,1})^2 - 2(\hat{f}^c_{n,i} + \hat{z}^n_{i,1})(\tilde{f}^c_{n,i} + \tilde{z}^n_{i,1} - \hat{f}^c_{n,i} - \hat{z}^n_{i,1}) \leq 0, \tag{34b}$$

$$2\tilde{\alpha}_{n,i}L_i + \tilde{R}^2_{n,i} + (\tilde{z}^n_{i,2})^2 - (\tilde{R}_{n,i} + \tilde{z}^n_{i,2})^2 - 2(\hat{R}_{n,i} + \hat{z}^n_{i,2})(\tilde{R}^n_i + \tilde{z}^n_{i,2} - \hat{R}_{n,i} + \hat{z}^n_{i,2}) \leq 0, \tag{34c}$$

$$2(1 - \tilde{\alpha}_{n,i})J_iL_i + \left(\tilde{f}^l_{n,i}\right)^2 + (\tilde{z}^n_{i,3})^2 - (\hat{f}^l_{n,i} + \hat{z}^n_{i,3})^2 - 2(\hat{f}^l_{n,i} + \hat{z}^n_{i,3})(\tilde{f}^l_{n,i} + \tilde{z}^n_{i,3} - \hat{f}^l_{n,i} + \hat{z}^n_{i,3}) \leq 0 \tag{34d}$$

$$2(\sigma^2 + \sum_{j,j\neq i}^{N} \tilde{P}^{\text{tr}}_{n,j}H_j) + \tilde{\varphi}^2_{n,i} - (\hat{\varphi}_{n,i} + \hat{P}^{\text{tr}}_{n,i}H_i)^2 + (\tilde{P}^{\text{tr}}_{n,i}H_i)^2 - 2(\hat{\varphi}_{n,i} + \hat{P}^{\text{tr}}_{n,i}H_i)(\tilde{\varphi}_{n,i} - \hat{\varphi}_{n,i}) - 2H_i(\hat{\varphi}_{n,i} + \hat{P}^{\text{tr}}_{n,i}H_i)(\tilde{P}^{\text{tr}}_{n,i} - \hat{P}^{\text{tr}}_{n,i}) \leq 0, \tag{34e}$$

$$(\tilde{\alpha}_{n,i}\gamma + \sigma^2 + \sum_{j\neq i}^{N} \tilde{P}^{\text{tr}}_{n,j}H_j)^2 - (\sigma^2 + \sum_{j,j\neq i}^{N} \tilde{P}^{\text{tr}}_{n,j}H_j)^2 - (\hat{\alpha}_{n,i}\gamma)^2 - 2\tilde{P}^{\text{tr}}_{n,i}H_i - 2\gamma\hat{\alpha_{n,i}}(\tilde{\alpha}_{n,i} - \hat{\alpha}_{n,i}) - 2\sum_{j,j\neq i}^{N} H_j(\sigma^2 + \sum_{j\neq i}^{N} \hat{P}^{\text{tr}}_{n,j}H_j)(\tilde{P}^{\text{tr}}_{n,j} - \hat{P}^{\text{tr}}_{n,j}) \leq 0, \tag{34f}$$

$$\sum_{i=1}^{N}(\tilde{\alpha}_{n,i} + \tilde{f}^c_{n,i})^2 - \hat{\alpha}^2_{n,i} - \left(\hat{f}^c_{n,i}\right)^2 - 2\hat{\alpha}_{n,i}(\tilde{\alpha}_{n,i} - \hat{\alpha}_{n,i}) - 2\hat{f}^c_{n,i}(\tilde{f}^c_{n,i} - \hat{f}^c_{n,i}) \leq 2f_c, \tag{34g}$$

$$2\tilde{\alpha}_{n,i} + \left(\tilde{f}^c_{n,i}\right)^2 + \tilde{s}^2_{n,i} - (\tilde{f}^c_{n,i} + \tilde{s}_{n,i})^2 - 2(\hat{f}^c_{n,i} + \hat{s}_{n,i}) \times (\tilde{f}^c_{n,i} + \tilde{s}_{n,i} - \hat{f}^c_{n,i} - \hat{s}_{n,i}) \leq 0, \tag{34h}$$

$$\tilde{R}_{n,i} - W\log_2(1 + \frac{1}{\hat{\varphi}_{n,i}}) + \frac{W(\tilde{\varphi}_{n,i} - \hat{\varphi}_{n,i})}{\hat{\varphi}^2_{n,i} + \hat{\varphi}_{n,i}} \leq 0, \tag{34i}$$

$$(\tilde{\alpha}_{n,i} + \tilde{P}^{\text{tr}}_{n,i})^2 + \tilde{R}^2_{n,i} + \tilde{u}^2_{n,i} - 2\hat{P}^{\text{tr}}_{n,i}(\tilde{P}^{\text{tr}}_{n,i} - \hat{P}^{\text{tr}}_{n,i}) - \hat{\alpha}^2_{n,i} - (\hat{R}_{n,i} + \hat{u}_{n,i})^2 - 2(\hat{\alpha}_{n,i})(\tilde{\alpha}_{n,i} - \hat{\alpha}_{n,i}) - (\hat{P}^{\text{tr}}_{n,i})^2 - 2(\hat{R}_{n,i} + \hat{u}_{n,i})(\tilde{R}_{n,i} + \tilde{u}_{n,i} - \hat{R}_{n,i} - \hat{u}_{n,i}) \leq 0, \tag{34j}$$

$$0 \leq \tilde{f}^c_{n,i} \leq f^c_{i,\max}, 0 \leq \tilde{f}^l_{n,i} \leq f^l_{i,\max}, \tilde{\alpha}_i \in [0,1], \tag{34k}$$

$$0 \leq \tilde{P}^t_{n,i} \leq P^t_{i,max}, \tilde{z}^n_{i,1} + \tilde{z}^n_{i,2} + \tilde{z}^n_{i,3} \leq \tilde{t}^d_{n,i}, \tag{34l}$$

$$\tilde{P}^{\text{tr}}_{n,i} = P^{\text{tr}}_i, \tilde{f}^c_{n,i} = f^c_i, \tilde{f}^l_{n,i} = f^l_i, \tilde{R}_{n,i} = R_i, \tag{34m}$$

$$\tilde{\varphi}_{n,i} = \varphi_i, \tilde{z}^n_{i,1} = z^n_{i,1}, \tilde{z}^n_{i,2} = z^n_{i,2}, \tilde{z}^n_{i,3} = z^n_{i,3}, \tag{34n}$$

$$\tilde{s}_{n,i} = s_i, \tilde{u}_{n,i} = u_i, \tilde{\Gamma}_{n,i} = \Gamma_i, \tilde{\alpha}_{n,i} = \alpha_i. \tag{34o}$$

We define Ω_n as the feasible set of the local copy $\tilde{\mathcal{X}}_n$ for user n, whose constituent variables satisfy all the above constraints, except for the consensus ones in (34m)-(34o). Note that the objective function for user i in problem (34), i.e., $\overline{E^F_n}(\tilde{\mathcal{X}}_n)$, is now decoupled with respect to the other users in the system.

Step 2 (ADMM iterative update equation): By penalizing and dualizing the consensus constraints into a global objective function (34a), we obtain the following augmented Lagrangian problem:

$$\begin{aligned} \min_{\{\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n\}_{n=1}^N} \quad & \mathcal{L}_\rho(\mathcal{X},\{\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n\}_{n=1}^N) \\ \text{s.t.} \quad & \tilde{\mathcal{X}}_n \in \Omega_n, \forall n = 1,\cdots,N, \end{aligned} \tag{35}$$

where $\rho \in \mathbb{R}_+$ is the penalty parameter, $\boldsymbol{\lambda}_n = [\boldsymbol{\lambda}_{1,n}^T,\cdots,\boldsymbol{\lambda}_{12,n}^T]^T$ denotes the Lagrange multipliers corresponding to the consensus constraints in problem (34) for each user, and

$$\begin{aligned} &\mathcal{L}_\rho(\mathcal{X},\{\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n\}_{n=1}^N) \\ &= \sum_{n=1}^N \overline{E_n^F}(\tilde{\mathcal{X}}_n) + \frac{\rho}{2}\sum_{n=1}^N\sum_{i=1}^N \{|\tilde{\alpha}_{n,i} - \alpha_i + \frac{\lambda_{1,n,i}}{\rho}|^2 + |\tilde{f}^c_{n,i} - f^c_i \\ &+ \frac{\lambda_{2,n,i}}{\rho}|^2 + |\tilde{f}^l_{n,i} - f^l_i + \frac{\lambda_{3,n,i}}{\rho}|^2 + |\tilde{R}_{n,i} - R_i + \frac{\lambda_{4,n,i}}{\rho}|^2 \\ &+ |\tilde{\varphi}_{n,i} - \varphi_i + \frac{\lambda_{5,n,i}}{\rho}|^2 + |\tilde{z}^n_{i,1} - z^n_{i,1} + \frac{\lambda_{6,n,i}}{\rho}|^2 + |\tilde{z}^n_{i,2} - z^n_{i,2} \\ &+ \frac{\lambda_{7,n,i}}{\rho}|^2 + |\tilde{z}^n_{i,3} - z^n_{i,3} + \frac{\lambda_{8,n,i}}{\rho}|^2 + |\tilde{P}^{\text{tr}}_{n,i} - P^{\text{tr}}_i + \frac{\lambda_{9,n,i}}{\rho}|^2 \\ &+ |\tilde{s}_{n,i} - s_i + \frac{\lambda_{10,n,i}}{\rho}|^2 + |\tilde{u}_{n,i} - u_i + \frac{\lambda_{11,n,i}}{\rho}|^2 \\ &+ |\tilde{\Gamma}_{n,i} - \Gamma_i + \frac{\lambda_{12,n,i}}{\rho}|^2\} = \sum_{n=1}^N \mathcal{L}_{n,\rho}(\mathcal{X},\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n). \end{aligned}$$

The overall update procedure of the proposed ADMM-based distributed algorithm for solving problem (20) are summarized in Table 4. Below, we further elaborate the main steps of this procedure.

a) Optimization of local copies $\{\tilde{\mathcal{X}}_n\}_{n=1}^N$: Observing that local copies $\{\tilde{\mathcal{X}}_n\}_{n=1}^N$ are now separable from each other in problem (35), we can decompose the latter into N independent subproblems, each of which can be solved at the user end:

$$\begin{aligned} \min_{\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n} \quad & \mathcal{L}_{n,\rho}(\mathcal{X},\tilde{\mathcal{X}}_n,\boldsymbol{\lambda}_n) \\ \text{s.t.} \quad & \tilde{\mathcal{X}}_n \in \Omega_n. \end{aligned} \tag{36}$$

This problem, which is a convex problem due to its quadratic objective function and convex feasible set Ω_n, can be solved by using the CVX programming toolbox. After the computation, each user sends the updating solution to the BS.

b) Optimization of global variables $\{\mathcal{X}\}$: It follow from (35) that all subproblems with respect to each one of global variables are unconstrained quadratic problems, which can be efficiently solved by applying the first-order optimality condition. Thus, by using the corresponding local copies from the users, the global variables can be updated at the BS relying on (37)-(40), which is shown at the bottom of this page.

c) Adjustment of Lagrange multipliers $\boldsymbol{\lambda}$: After the BS has received the current local copies of the updated local variables from all users, the Lagrange multipliers are adjusted according to (41)-(44), displayed at the bottom of this page. Once the computation of the global variables and Lagrange multipliers is completed, the BS feeds back the updated results to each user.

To sum up, the overall distributed ADMM-based FO (AFO) algorithm consists of two embedded loops, where the outer loop follows the CCCP steps shown in Table 1 and the inner loop performs the ADMM-based algorithm shown in Table 4.

Algorithm 4 Proposed distributed ADMM-based FO (AFO) algorithm for solving problem (20)

```
1. Initialization: Define the tolerance of accuracy $\delta_4$ and the maximum number of iterations $N_{\max}$. Initialize the algorithm with feasible global variables $\mathcal{X}^0$ and local copies $\tilde{\mathcal{X}}^0$. Set the iteration number $t = 0$ and the penalty parameter $\rho$.
2. Repeat
     – Solve problem (36). Update the local copies $\tilde{\mathcal{X}}_n^t$ at each user.
     – Update the global variables $\mathcal{X}^t$ according to (37)-(40) at the BS.
     – Update the Lagrangian multipliers $\boldsymbol{\lambda}^t$ according to (41)-(44) at the BS.
     – Update the iteration index: $t \leftarrow t + 1$.
3. Until the difference between successive values of the objective function in (36) is less than $\delta_4$ or the maximum iteration number is reached.
```

B. Convergence and computational complexity

Due to general properties of the ADMM optimization framework [37], the proposed ADMM-based algorithm can

$$\alpha_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{\alpha}_{n,i} + \frac{\lambda_{1,n,i}}{\rho}\right), f^c_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{f}^c_{n,i} + \frac{\lambda_{2,n,i}}{\rho}\right), f^l_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{f}^l_{n,i} + \frac{\lambda_{3,n,i}}{\rho}\right), \tag{37}$$

$$R_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{R}_{n,i} + \frac{\lambda_{4,n,i}}{\rho}\right), \varphi_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{\varphi}_{n,i} + \frac{\lambda_{5,n,i}}{\rho}\right), z^n_{i,1} = \frac{1}{N}\sum_{n=1}^N\left(\tilde{z}^n_{i,1} + \frac{\lambda_{6,n,i}}{\rho}\right), \tag{38}$$

$$z^n_{i,2} = \frac{1}{N}\sum_{n=1}^N\left(\tilde{z}^n_{i,2} + \frac{\lambda_{7,n,i}}{\rho}\right), z^n_{i,3} = \frac{1}{N}\sum_{n=1}^N\left(\tilde{z}^n_{i,3} + \frac{\lambda_{8,n,i}}{\rho}\right), P^{\text{tr}}_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{P}^{\text{tr}}_{n,i} + \frac{\lambda_{9,n,i}}{\rho}\right), \tag{39}$$

$$s_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{s}_{n,i} + \frac{\lambda_{10,n,i}}{\rho}\right), u_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{u}_{n,i} + \frac{\lambda_{11,n,i}}{\rho}\right), \Gamma_i = \frac{1}{N}\sum_{n=1}^N\left(\tilde{\Gamma}_{n,i} + \frac{\lambda_{12,n,i}}{\rho}\right). \tag{40}$$

$$\lambda_{1,n,i} = \lambda_{1,n,i} + \rho(\tilde{\alpha}_{n,i} - \alpha_i), \lambda_{2,n,i} = \lambda_{2,n,i} + \rho(\tilde{f}^c_{n,i} - f^c_i), \lambda_{3,n,i} = \lambda_{3,n,i} + \rho(\tilde{f}^l_{n,i} - f^l_i), \tag{41}$$

$$\lambda_{4,n,i} = \lambda_{4,n,i} + \rho(\tilde{R}_{n,i} - R_i), \lambda_{5,n,i} = \lambda_{5,n,i} + \rho(\tilde{\varphi}_{n,i} - \varphi_i), \lambda_{6,n,i} = \lambda_{6,n,i} + \rho(\tilde{z}^n_{i,1} - z^n_{i,1}), \tag{42}$$

$$\lambda_{7,n,i} = \lambda_{7,n,i} + \rho(\tilde{z}^n_{i,2} - z^n_{i,2}), \lambda_{8,n,i} = \lambda_{8,n,i} + \rho(\tilde{z}^n_{i,3} - z^n_{i,3}), \lambda_{9,n,i} = \lambda_{9,n,i} + \rho(\tilde{P}^{\text{tr}}_{n,i} - P^{\text{tr}}_i), \tag{43}$$

$$\lambda_{10,n,i} = \lambda_{10,n,i} + \rho(\tilde{s}_{n,i} - s_i), \lambda_{11,n,i} = \lambda_{11,n,i} + \rho(\tilde{u}_{n,i} - u_i), \lambda_{12,n,i} = \lambda_{12,n,i} + \rho(\tilde{\Gamma}_{n,i} - \Gamma_i). \tag{44}$$

globally solve the approximated convex problem (20). Thus, it is readily seen that the proposed overall distributed algorithm can converge to a stationary point of problem (12).

Each inner iteration of the distributed AFO algorithm is divided into three steps: updating the local copies, updating the global variables, and updating the Lagrange multipliers. When updating the local copies, the computational complexity is dominated by the use of a generic interior-point method in toolbox CVX, which needs $O(N)$ flops. When updating the global variables, the solution of the corresponding quadratic problems can be obtained in $O(1)$ flops. At last, updating of Lagrange multiplier can be achieved in $O(1)$ flops. Based on this analysis, the total computation complexity of the proposed AFO algorithm is given by $O(I_3N)$, where I_3 denotes the number of required iterations by the proposed distributed algorithm. In comparison with its centralized counterpart, the proposed distributed algorithm has much lower complexity, which is amenable to large-scale multiuser scenarios.

VII. Simulation Results

In this section, we use Monte Carlo simulations to demonstrate the benefits of the proposed resource allocation algorithms for MEC in terms total energy consumption. For all the simulation results, unless specified otherwise, we consider a MEC system with N mobile users distributed randomly within a circular area with radius $r = 0.2$km [39]. The radio bandwidth available for transmission from the mobile users to the BS is $W = 20$MHz [40]. As in [38], the corresponding channel coefficients are generated as normalized, independent Rayleigh fading components with distance-dependent path loss, modeled as $\text{PL} = 20\log_{10}(d) + 112.45$dB, where d is the distance between the mobile users and the BS in kilometers. The target SINR of each mobile user is set to $\gamma = 10$ dB. In addition, the mobile users' standby power and maximum transmission power are set to $P^{\text{idle}} = 0.5$W and $P^t_{\max} = 1$W, respectively. Delay-sensitive applications are characterized by their bounded end-to-end delay requirements [7]. As in [6], [8], the delay tolerance of each task is set to $t^d = 0.15$s. For each task of mobile user i, L_i follows the uniform distribution over $[1 \times 10^5, 5 \times 10^5]$ (bits), and the workload is set to $J_i = 18000$ CPU cycles per bit [41]. Furthermore, the power consumption coefficient for the given chip architecture is set as $K = 10^{-24}(\text{Watt} \times \text{s}^3)$ [21]. Besides, the computational budget of the MEC server and the local user are set to $f^c_{\max} = 1600$ MHz and $f^l_{\max} = 400$MHz, respectively. For convenience, all simulation parameters are listed in Table II.

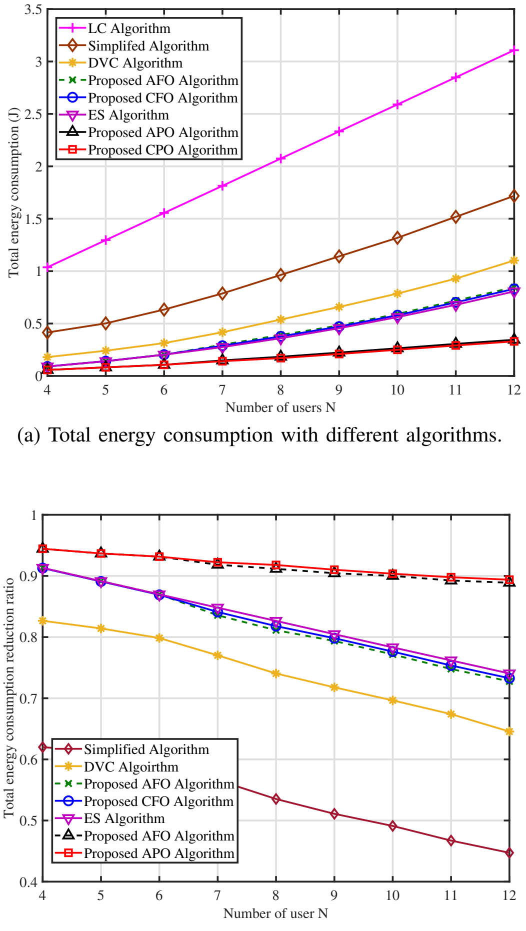
TABLE II: Simulation parameters

Parameters	Value
Number of mobile users N	4-12
Radius of coverage area r	0.2km
Radio bandwidth between mobile users and BS W	20MHz
Distance-dependent path loss PL	$20\log(d) + 112.45$dB
Target SINR γ	10dB
Maximum transmission power $P^t_{\max}$	1W
Standby power of mobile user P^{idle}	0.5W
Delay tolerance t^d	0.15s
Size of each task before computation L_i	$[1 \times 10^5, 5 \times 10^5]$ bits
CPU cycles required to process per bit J_i	18000 cycles / bit
Coefficient depending on chip architecture K	10^{-24} Watt $\times \text{s}^3$,
Computational budget of the MEC server $f^c_{\max}$	1600 MHz
Computational budget of the local user $f^l_{\max}$	400 MHz

In our simulations, we compare the energy consumption performance of the proposed algorithms to three alternative baseline algorithms, as follows:

- Local Computing (LC) algorithm: For any type of applications in the multiuser MEC system, the tasks of each user are only executed locally without offloading, i.e., on the corresponding user device.
- Dynamic Computation Offloading (DVC) algorithm [7]: Existing joint offloading decision and resource allocation algorithm designed for partial offloading scheme.
- Exhaustive-Search (ES) algorithm: For continuous-execution applications, the user selection indicators for offloading are obtained by exhaustive-search and the remaining variables are optimized by the CCCP algorithm.

While the ES algorithm exhibits very high computational complexity, it provides an upper bound on performance for continuous-execution applications in multiuser MEC systems and therefore serves as a useful benchmark.



(a) Total energy consumption with different algorithms.

(b) Total energy consumption reduction ratio over the LC algorithm.

Fig. 4: Total energy performance versus the number of users for the different allocation algorithms.

In Fig. 4, we plot the total energy performance versus the number of users N for the algorithms under comparison. In particular, the total energy consumption reduction ratio over the LC algorithm is defined as $\Upsilon_{\text{alg}} \triangleq \frac{E_{\text{LC}} - E_{\text{alg}}}{E_{\text{LC}}}$, where E_{alg} and E_{LC} are respectively the total energy consumption

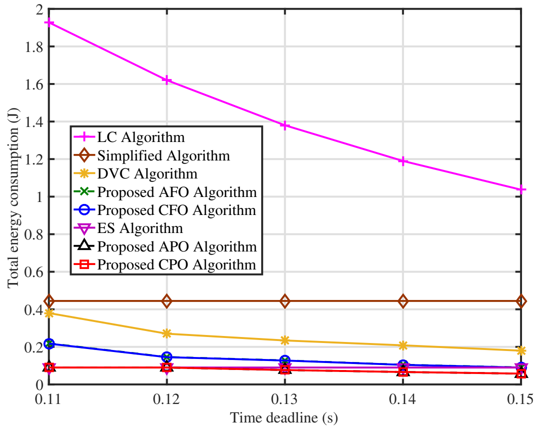


Fig. 5: Total energy consumption versus delay tolerance for the different allocation algorithms.

achieved by the algorithm analyzed and the LC algorithm. For continuous-execution applications, it can be observed that the minimum total energy consumption in a multiuser MEC system is achieved by the ES algorithm, followed by the proposed CFO algorithm, where the performance gap between the ES and the CFO algorithms is extremely small. Furthermore, the proposed CFO algorithm achieves significant gains over the LC, simplified, and DVC algorithms, which demonstrates the importance of joint optimization. As the number of users increases, due to the power control, the gap between the proposed CFO algorithm and these other algorithms is enlarged. Although the performance of the simplified algorithm is not as good as that of the CFO algorithm, the former is still very promising due to its lower computational complexity. Moreover, it can be seen that the CPO algorithm outperforms all the other competing schemes. This is due to the fact that the CPO algorithm can take advantage of parallel processing in handling data-partitioning-oriented applications, which can reduce the total energy consumption under specific delay tolerance. Finally, we should note that both AFO and APO algorithm can achieve near-optimal performance in a distributed manner when compared to the CFO algorithm and CPO algorithm, respectively.

In Fig. 5, we plot the total energy consumption versus the delay tolerance t_i^d for the different algorithms under comparison. When the delay tolerance grows, the total energy consumption decreases, and the performance gap between the ES and the proposed CFO algorithms becomes smaller. The results also show that compared with the LC algorithm, the CFO algorithm can significantly reduce the total energy consumption, by about 90% over the considered range of t_i^d. Here again, it is seen that the energy consumptions of the CPO and ES algorithms nearly coincide over the considered range of t_i^d values. Indeed, because the computation capacity of the MEC server is much larger than that of mobile users, when the delay tolerance is stringent, the mobile users tend to offload more of their applications to the MEC server for shortening latency, even if these applications are data-partitioning-oriented.

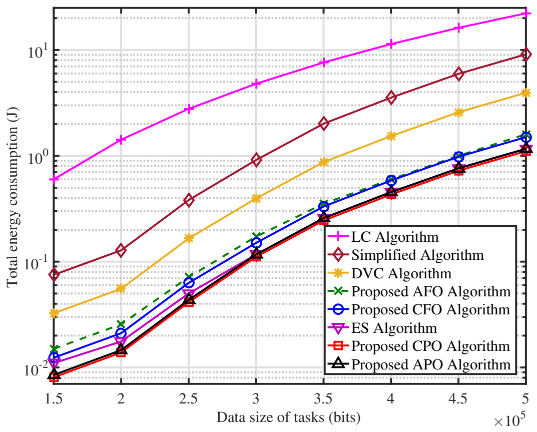


Fig. 6: Total energy consumption versus the data size of tasks for the different allocation algorithms.

In Fig. 6, we plot the total energy consumption versus the data size of the different tasks L_i for the different algorithms under comparison. Note that different from other simulations, L_i is deterministic and does not follow the uniform distribution as defined before. For convenience in comparison, we set the size of tasks for different users equal. We can see that as the data size of tasks grows, the total energy consumption gradually increases. For continuous-execution tasks, it is also observed that the total energy consumption achieved with the proposed CFO algorithm is lower than that achieved by the other algorithms except the ES algorithm. However, the performance of the CPO algorithm coincides with that of the ES algorithm when the data size of tasks is large. Hence, it appears that for tasks with large sizes, processing them at the MEC server is preferable from an optimum resource allocation perspective.

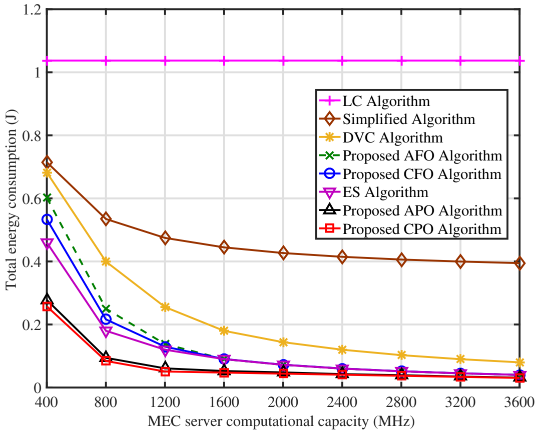


Fig. 7: Total energy consumption versus the MEC server computation capacity for the different allocation algorithms.

In Fig. 7, we plot the total energy consumption versus the MEC server computation capacity $f_{\max}^c$. For continuous-execution applications, the proposed CFO algorithm achieves a near-optimal performance, i.e. very close to the upper bound on performance provided by the ES algorithm, and significantly outperforms all the competing algorithms. Moreover, it is observed that the proposed CPO algorithm can achieve a significant gain over the CFO algorithm when the computational capacity of the MEC server is limited. In other words, by leveraging the parallel processing of data-partitioning-oriented applications, the proposed CPO algorithm provides more flexibility for the resource allocation scheme as well as more refined control on the multiuser interference management in the MEC system. Finally, when the computational capacity of the MEC server is sufficient, the energy consumptions of the CFO, CPO and ES algorithms converge to a common value, where the system performance is constrained by the use of the radio resource. This reveals a fundamental design principle for MEC systems: once the system operation is constrained by the available radio resource, there is no need to deploy additional computational resources.

VIII. Conclusion

In this paper, we have investigated an energy-efficient resource allocation problem for a multiuser MEC system under interfering channels and have formulated specific optimization problems for two different types of applications, specifically, the ECM-FO problem for full offloading of continuous-execution tasks and the ECM-PO problem for partial offloading of data-partitioning-oriented tasks. In order to handle binary variables as well as highly coupled nonconvex terms in the objective functions and nonconvex constraints of the ECM-FO problem, we have introduced auxiliary variables and applied linearization to transform the original problem into a more tractable form. We then proposed the CFO algorithm to find local stationary solutions. In addition, a simplified resource allocation algorithm with reduced computational complexity was proposed for the FO case by introducing a suitable measure of user priority. To solve the ECM-PO problem, we resorted smooth functional approximation of the l_0-norm constraints and employed various algebraic transformations, which in turn lead to the CPO algorithm. To alleviate the performance bottleneck caused by the CSI overhead and high complexity, due to the scalability of large-scale mobile users, we develop a distributed implementation method for both CFO and CPO algorithm. Simulation results demonstrate that, for continuous-execution applications in a multiuser MEC system, the energy consumption performance of the CFO algorithm is very close to the best performance achieved from an exhaustive search. For data-partitioning-oriented applications in a multiuser MEC system, the proposed CPO algorithm achieves significant gains over the benchmark schemes by leveraging the parallel processing.

Appendix A
Brief Review Of CCCP Method

DC programming deals with optimization problems involving objective and constraint functions with each represented as a difference of two convex functions [28]. A general form of DC programming problems can be expressed as follows:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) - g_0(\mathbf{x}) \tag{45}$$
$$f_k(\mathbf{x}) - g_k(\mathbf{x}) \le 0, \text{for } k \in \{1, \dots, K\},$$

where f_k and g_k for $k = 0, 1, ..., K$, are all convex functions, and K is the number of constraints.

However, a DC program is not convex unless the functions g_i are affine, and is difficult to solve in general. The CCCP is a heuristic algorithm to find a local optimal solution of DC programs [29]. Its main idea is to convexify the problem by replacing the concave part in each DC function, which is $g_k(\mathbf{x})$, by its first order Taylor series expansion around the current estimated value of $\boldsymbol{x}$, and in this way, successively solve a sequence of convex problems in an iterative manner, starting with an initial feasible point $\mathbf{x}_0$, i.e., $f_k(\mathbf{x}_0) - g_k(\mathbf{x}_0) \le 0$. At iteration t, it solves the following convex subproblem:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) - g_0(\mathbf{x}^t) - \nabla g_0(\mathbf{x}^t)^T(\mathbf{x} - \mathbf{x}^t) \tag{46}$$
$$f_k(\mathbf{x}) - g_k(\mathbf{x}^t) - \nabla g_k(\mathbf{x}^t)^T(\mathbf{x} - \mathbf{x}^t) \le 0, \forall k,$$

where $\mathbf{x}^t$ is the optimal solution obtained from the previous iteration. The objective function values decrease monotonically and thus the limit point of the iterates generated by the CCCP method will converge to the KKT solution of the original problem [31].

Appendix B
Proof of equivalence between (12) and (14)

Let us first introduce variables $z_{i,1}$, $z_{i,2}$ and $z_{i,3}$ as the upper bound of t_i^c, $t_i^{\rm tr}$ and t_i^l , respectively, $\forall i$. Thus, the constraint (12b) can be equivalently expressed as

$$z_{i,1} + z_{i,2} + z_{i,3} \le t_i^d, \tag{47}$$
$$\alpha_i J_i L_i / f_i^c \le z_{i,1}, \tag{48}$$
$$\frac{\alpha_i L_i}{W \log_2\left(1 + \frac{\beta P_i^{\rm tr} H_i}{\sigma^2 + \sum_{j \neq i}^{N} P_j^{\rm tr} H_j}\right)} \le z_{i,2}, \tag{49}$$
$$(1 - \alpha_i) J_i L_i / f_i^l \le z_{i,3}. \tag{50}$$

Due to the fractional form of the Γ_i expressions (see (1)), constraints (49) remain difficult to tackle. In the following, we convert them into equivalent yet tractable forms. By introducing auxiliary variable φ_i, Δ_i, s_i and u_i as the upper bound of $\frac{\sigma^2 + \sum_{j \neq i}^{N} P_j^{\rm tr} H_j}{\beta P_i^{\rm tr} H_i}$, $1/\varphi_i$, $\alpha_i / f_i^{\rm c}$, and $\alpha_i P_i^{\rm tr} / R_i$, respectively, and R_i as the lower bound of $W \log_2(1 + 1/\varphi_i)$, problem (12) can be equivalently converted as

$$\min_{\mathcal{X}} \sum_{i=1}^{N} \overline{E_i^F}(\mathcal{X}) \tag{51a}$$
$$\text{s.t. } z_{i,1} + z_{i,2} + z_{i,3} \le t_i^d, \tag{51b}$$
$$\alpha_i J_i L_i / f_i^c \le z_{i,1}, \tag{51c}$$
$$\alpha_i L_i / R_i \le z_{i,2}, \tag{51d}$$
$$(1 - \alpha_i) J_i L_i / f_i^l \le z_{i,3}, \tag{51e}$$
$$R_i \le W \log_2(1 + 1/\varphi_i), \tag{51f}$$
$$1/\varphi_i \le \Delta_i, \tag{51g}$$
$$\beta P_i^{\rm tr} H_i / (\sigma^2 + \sum_{j \neq i}^{N} P_j^{\rm tr} H_j) \le 1/\varphi_i, \tag{51h}$$
$$\alpha_i / f_i^{\rm c} \le s_i, \tag{51i}$$
$$\alpha_i P_i^{\rm tr} / R_i \le u_i, \tag{51j}$$
$$(12\text{c}) - (12\text{h}). \tag{51k}$$

This completes the proof.

References

[1] CISCO, "Cisco visual networking index: global mobile data traffic forecast update, 2016-2021 white paper," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 2017.

[2] European Telecommunications Standards Institute (ETSI), "Mobile-edge-computing-Introductory technical white paper," Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge computing - introductory technical white paper v1.

[3] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129-140, 2013.

[4] 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, 4th Quart., 2017.

[5] L. Wang, M. L. Guan, Y. T. Ai, Y. Y. Chen, B. L. Jiao, L. Hanzo, "Beamforming aided NOMA expedites collaborative multiuser computational off-loading," *IEEE Trans. Veh. Technol.*, to appear.

[6] S. Chimmanee, "PACS metric based on regression for evaluating end-to-end QoS capability over the Internet for telemedicine," *in Proc. IEEE ICOIN*, Bangkok, Thailand, Jan. 2013, pp. 359-364.

[7] Y. Wang, M. Sheng, X. Wang, L. Wang, J. Li, "Mobile-edge computing: partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268-4282, Oct. 2016.

[8] D. T. T. Nga, M.-G. Kim, and M. Kang, "Delay-guaranteed energy saving algorithm for the delay-sensitive applications in IEEE 802.16e systems," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1339-1347, Nov. 2007.

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

[10] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 870-883, Jun. 2013.

[11] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.

[12] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451-1455.

[13] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans.* Networking, vol. 24, no. 5, pp. 2795-2808, Oct. 2015.

[14] S. Sardellitti et al., "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Info. Process. Over Networks,* vol. 1, no. 2, pp. 89-103, Jun. 2015.

[15] T. Q. Dinh, Q. D. La, T. Q. S. Quek, H. D. Shin, " Distributed Learning for Computation Offloading in Mobile Edge Computing," *IEEE Trans. Commun.,* to appear.

[16] C. Wang, C. Liang, F. Richard Yu, Q. Chen and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924-4938, Aug. 2017.

[17] S. Barbarossa, E. Ceci, M. Merluzzi and E. Calvanese, "Enabling effective Mobile Edge Computing using millimeterwave links," in *IEEE International Communication Conference*, Paris, May 21, 2017.

[18] F. Wang, J. Xiu, 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, Nov. 2018.

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

[20] 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, Dec. 2018.

[21] C. You, K. Huang, H. Chae, 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.

[22] X. Chen, Q. Shi, Y. Cai and M. Zhao, "Joint cooperative computation and interactive communication for relay-Assisted mobile edge computing,"[Online]. Available: arXiv.org/abs/1710.11420.

[23] S. Ranadheera, S. Maghsudi and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game,"[Online]. Available:arXiv.org/abs/1710.05499.

[24] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile edge computation offloading: energy-efficient resource management," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7590-7605, Nov. 2018.

[25] F. Zhou, Y. Wu, R. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927-1941, Sep. 2018.

[26] L. Fan, W. Yan, X. Chen, Z. Chen, and Q. Shi, "An energy efficient design for UAV communication with mobile edge computing," *China Communications*, vol. 16, no. 1, pp. 26-36, Mar. 2019.

[27] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading." [Online]. Available: arxiv.org/pdf/1704.00163.pdf

[28] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915-936, 2003.

[29] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118-6131, Sep. 2016.

[30] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, Sep. 2012.

[31] G. R. Lanckriet, and B. K. Sriperumbudur, "On the convergence of the concave-convex procedure," *Advances in Neural Inf. Process. Systems.*, 2009.

[32] Y. Cheng and M. Pesavento, "Joint optimization of source power allocation and distributed relay beamforming in multiuser peer-topeer relay networks," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2962-2973, Jun. 2012.

[33] K. Wang, A. So, T. H. Chang, W. K. Ma, and C. Y. Chi, "Outage constrained robust transmit optimization for multiuser MISO downlinks: Tractable approximations by conic optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5690-5705, Nov. 2014.

[34] Y. Shi, J. Cheng, J. Zhang, B. Bai, W. Chen, and K. B. Letaief, "Smoothed Lp-minimization for green cloud-RAN with user admission control," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, Apr. 2016.

[35] E. Candes, M. Wakin, S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *J Fourier Anal Appl.*, vol. 14, no. 5, Oct 2008.

[36] J. Song, P. Babu, and D. P. Palomar, "Sparse generalized eigenvalue problem via smooth optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1627-1642, 2015.

[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn*, vol. 3, no. 1, pp. 1-122, 2011.

[38] Y. Yu et al., "Joint subcarrier and CPU time allocation for mobile edge computing," *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016.

[39] A. Liu, X. Chen, W. Yu, V. Lau, and M. Zhao, "Two-timescale hybrid compression and forward for massive MIMO aided C-RAN," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2484-2498, May 2019.

[40] X. Chen, A. Liu, Y. Cai, V. Lau, and M. Zhao, "Randomized two-timescale hybrid precoding for downlink multicell massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4152-4167, Jul. 2019.

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

[42] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC)*, San Francisco, CA, Mar. 2017.