Delay-Aware Computation Offloading in NOMA MEC Under Differentiated Uploading Delay

Min Sheng, *Senior Member, IEEE*, Yanpeng Dai, *Student Member, IEEE*, Junyu Liu, *Member, IEEE*, Nan Cheng, *Member, IEEE*, Xuemin Shen, *Fellow, IEEE*, and Qinghai Yang

***Abstract*— In mobile edge computing (MEC), the computation offloading of massive users could cause the task uploading congestion to deteriorate the users' offloading delay. The non-orthogonal multiple access (NOMA) enabled MEC is envisioned to address this issue by allowing multiple users to simultaneously upload their tasks on one subchannel. However, the differentiated uploading delay of users may make task uploading completion inconsistent with NOMA decoding order, which complicates the co-channel interference and restricts NOMA to reducing the uploading delay. In this paper, we characterize the interaction between the differentiated uploading delay and co-channel interference for a pair of NOMA users. Furthermore, we propose a computation offloading scheme to reduce the users' average offloading delay by jointly optimizing offloading decision and resource allocation. Specifically, the proposed scheme first obtains the optimal power allocation based on the characterized interaction and the closed-form solution of computation resource allocation by convex programming. Then, the NOMA user pairing and offloading decision are iteratively determined by semidefinite relaxation and convex-concave procedure. Simulation results show that the proposed scheme effectively mitigates co-channel interference under differentiated uploading delay of users and outperforms in reducing the users' average offloading delay and increasing the number of users to offload tasks.**

***Index Terms*— Non-orthogonal multiple access (NOMA), edge computing, delay, multiple access interference, resource management.**

I. Introduction

THE dramatic advancement of the fifth generation (5G) cellular networks and Internet of Things are spawning many emerging compute-intensive and delay-sensitive applications for wireless users, such as mobile gaming, social networking, and virtual/augmented reality. In order to supply low–latency computing services, mobile edge computing (MEC) has been proposed to realize the cloud-like computing at the wireless network edge [1]–[3]. The edge node possesses the computation capability to execute the applications for users, avoiding the remote transmission to the cloud. Nevertheless, as the number of wireless devices rapidly increases, the restricted radio resources result in the congestion of the computation task uploading [4], [5]. This kind of congestion severely prolongs the task uploading delay and overall delay on finishing computation offloading, which has become the limiting factor for MEC. Therefore, it is crucial for low-latency MEC offloading to promote the capacity of task uploading under the limited bandwidth.

Recently, the non-orthogonal multiple access (NOMA) has been proposed as a key multiple access technique for 5G network to enhance network capacity [6]. The NOMA supports multiple users to share the same orthogonal spectrum resource and exploits the signal difference on power domain to distinguish different users by the successive interference cancellation (SIC). Compared to the orthogonal multiple access (OMA), the NOMA can rely on better spectrum efficiency and network connectivity to enhance the task uploading capacity for MEC. Therefore, the NOMA-enabled MEC is a promising technique in the future wireless network and has attracted a lot of attention [7]–[12]. Ding *et al.* in [7] demonstrated the superiority of NOMA-enabled MEC on the energy efficiency and the delay on computation offloading through performance analysis. To further guarantee the low-latency computation offloading, Wu *et al.* in [10], Wang *et al.* in [11], and Song *et al.* in [12] proposed the computation offloading schemes for NOMA-enabled MEC in different scenarios. However, although the SIC decoding is applied, the co-channel interference still partially exists between NOMA users. Furthermore, in practical MEC system, there exists the difference between task uploading delay of users, due to the heterogeneous input-data size of tasks and channel conditions. The differentiated uploading delay can vary the co-channel interference between NOMA users during the process of task uploading. In turn, the co-channel interference also affects the task uploading delay of NOMA users. Therefore, it is necessary for NOMA-enabled MEC system to characterize the interaction between the task uploading delay and co-channel interference of NOMA users.

Due to the differentiated uploading delays of users, the scheduling of computation offloading becomes very

Manuscript received May 19, 2019; revised November 1, 2019 and January 14, 2020; accepted January 15, 2020. Date of publication January 28, 2020; date of current version April 9, 2020. This work was supported in part by the Natural Science Foundation of China under Grant 61725103, Grant 61701363, Grant 91638202, and Grant 61931005, in part by the Shaanxi Innovation Group under Grant 2017KCT-30-03, in part by the Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-04-04, in part by the China Postdoctoral Science Foundation under Grant 2018T111017, and in part by the National S&T Major Project of China under Grant 2018ZX03001014-004. The associate editor coordinating the review of this article and approving it for publication was D. So. *(Corresponding author: Yanpeng Dai.)*

Min Sheng, Yanpeng Dai, Junyu Liu, Nan Cheng, and Qinghai Yang are with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: msheng@mail.xidian.edu.cn; daiyanpeng@stu.xidian.edu.cn; junyuliu@xidian.edu.cn; nancheng@xidian.edu.cn; qhyang@xidian.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

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.2020.2968426

challenging in NOMA-enabled MEC. First, the co-channel interference between NOMA users is determined by both SIC decoding order and uploading completion order of users. Specifically, if the uploading completion order is consistent with the SIC decoding order, the co-channel interference between NOMA users does not vary during the process of task uploading. Otherwise, the co-channel interference between NOMA users will be varied during the process of task uploading. Thus, the interference coordination for NOMA-enabled MEC is more complicated than that in NOMA cellular uplink system [13]–[16]. Second, in a typical NOMA system with multiple subchannels, the NOMA user pairing and subchannel assignment directly affect the achievable rate of NOMA users for task uploading. Thus, it is necessary for NOMA-enabled MEC to jointly design both of them. Furthermore, the offloading decision should consider the differentiated uploading delays of users to accurately evaluate the latency on task offloading to make the right decision between the local computing and edge computing. Therefore, it is required to jointly consider the offloading decision, NOMA user pairing, and resource allocation under the differentiated uploading delays of users.

In this paper, we investigate the computation offloading problem for multi-carrier NOMA enabled MEC under the consideration of the differentiated uploading delays of users. Specifically, we first characterize the impact of the differentiated uploading delay on the co-channel interference between a pair of NOMA users. Based on this, an optimal power allocation algorithm is then proposed which coordinates the co-channel interference to minimize the uploading delay of both NOMA users. Finally, an average offloading delay minimization problem is formulated which jointly considers the offloading decision, NOMA user pairing, subchannel assignment and computation resource allocation of edge server. Since the original problem is a binary quadratic programming problem, a computation offloading scheme based on the semidefinite relaxation (SDR) and convex-concave procedure (CCP) is proposed. The main contributions of our paper are summarized as follows.

- We characterize the interaction between the differentiated uploading delay and the co-channel interference between NOMA users to further investigate the effectiveness of interference management on reducing average offloading delay of users.
- We derive a sufficient condition that if the uploading completion order of users is inconsistent with their SIC decoding order, reducing co-channel interference cannot decrease the uploading delays of both NOMA users. Based on this, the optimal power allocation is achieved.
- Through analyzing the Karush–Kuhn–Tucker (KKT) conditions, the close-form expression of optimal computation resource allocation is obtained to minimize the task execution delay of the edge server.
- We propose a computation offloading scheme which first adopts SDR to achieve a lower bound of average offloading delay and then exploits it to iteratively obtain the offloading decision and NOMA user pairing. The results show that the proposed scheme can converges to a near-optimal solution.

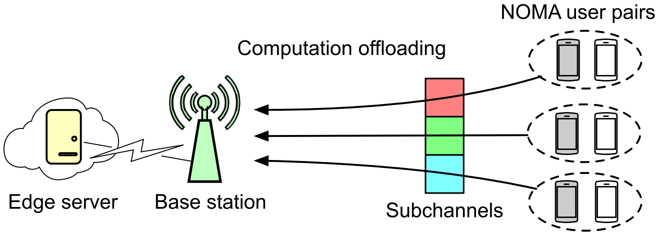


Fig. 1. Multi-carrier NOMA-enabled computation offloading for MEC.

The remainder of the paper is organized as follows. In Section II, we introduce the network scenario, offloading modes, and problem formulation. The problem decomposition and the proposed optimization-theory based scheme are presented in Section III. In Section IV, we design a matching-theory based algorithm with low complexity. Simulation results are given in Section V. Finally, we conclude this paper in Section VI.

II. System Model and Problem Formulation

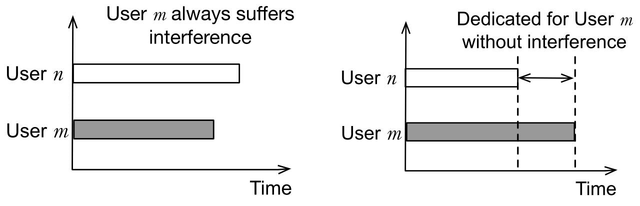
In this section, we first introduce the network scenario, computation model, and NOMA transmission model for computation offloading. Then, we formally give the formulation for the users' average delay minimization problem.

A. Network Scenario and Computation Model

As shown in Fig. 1, we consider the computation offloading scenario for MEC in a scheduling period, where there are one base station (BS), multiple users, denoted by $\mathcal{U} = \{1, \ldots, M\}$, and available subchannels on spectrum bandwidth, denoted by $\mathcal{S} = \{1, \ldots, K\}$. Specifically, the BS is equipped with one edge server to provide computing service. Each user has one compute-intensive task to be finished. For user m's task, s_m denotes the input-data size in bits and w_m denotes the workload in CPU cycles. The NOMA technique is adopted as multiple access scheme to realize that a pair of users upload their tasks on the same subchannel via power domain multiplexing. We consider that the BS can obtain perfect channel state information of all users and be informed the knowledge about the data sizes and workloads of tasks by application service provider [17], [18]. Based on this information, the BS schedules the computation offloading for users which have made the service request and been admitted at the current scheduling period [19], [20].

In this work, we adopt the binary offloading as the offloading policy [1], [11]. Under the binary offloading, the users select only one computing approach between local computing and edge computing. Hence, the delay of a user on finishing the task equals to the local execution delay or the task offloading delay which includes the uploading delay and the edge execution delay.

If selecting local computing, the user executes the task by its local device. Let f_m^{1} denote the computational rate in CPU cycles/s of user m. The execution delay of user m's task in


User m always suffers interference
User n
User m
Time
Dedicated for User m without interference
User n
User m
Time

(a) Case 1

(b) Case 2

Fig. 2. Two cases of task uploading for both NOMA users.

local is expressed as $t_m^{\mathrm{l}} = \frac{w_m}{f_m^{\mathrm{l}}}$. If selecting edge computing, the user is first assigned with one subchannel to upload the task to BS, and then the edge server executes all offloaded tasks in parallel. Let f_m^{e} denote the computational rate of edge server assigned to user m. The execution delay of user m's task by edge server is expressed as $t_m^{\mathrm{e}} = \frac{w_m}{f_m^{\mathrm{e}}}$.

Besides the execution delay at edge server, we should consider the task uploading delay which is influenced by the co-channel interference. Hence, the task uploading in NOMA system become complicated and different from that in OMA system.

B. NOMA Enabled Task Uploading

Let $h_{m,k}$ denote the channel power gain of subchannel k between BS and user m. In NOMA uplink system, the BS adopts the SIC to successively decode the users' signals according to the different power level of the received signals. As [21], [22], the SIC decoding order is the descending order of channel power gain. Once a signal is decoded successfully, it will be subtracted from the superposed signal. Thus, user m receives the co-channel interference from users n, if $h_{m,k} > h_{n,k}$. Considering the practical complexity of SIC, we assume that each subchannel is reused by at most two users [16], [23]–[26].

To investigate NOMA enabled task uploading, we consider a scenario where user m and user n are allocated subchannel k, and $h_{m,k} > h_{n,k}$. As such, the achievable rates of user m and n on task uploading are respectively expressed as

$$R_{m,k}^{\mathrm{up},1} = B\log_2\left(1 + \frac{p_m h_{m,k}}{\sigma^2 + p_n h_{n,k}}\right) \tag{1}$$

$$R_{n,k}^{\mathrm{up},2} = B\log_2\left(1 + \frac{p_n h_{n,k}}{\sigma^2}\right) \tag{2}$$

where p_m and p_n are the transmit powers of user m and user n, respectively. B is the subchannel bandwidth and σ^2 is the additive white gaussian noise power. Thus, the uploading delay of user n is expressed as $t_{n,k}^{\mathrm{up},2} = \frac{s_n}{R_{n,k}^{\mathrm{up},2}}$.

However, the existence of the co-channel interference is determined by task uploading delays of both users. As shown in Fig. 2, if user n finishes the task uploading earlier than user m, user m will acquire a dedicated time duration to upload the remaining task and suffer no more co-channel interference. Otherwise, user m always suffer the co-channel interference during the task uploading process and its rate keeps constant. Therefore, the differentiated uploading delay leads to two cases for user m's task uploading as follows.

- *Case 1.* If user m can finish task uploading earlier than user n, user m always receives the co-channel interference from user n and keeps constant transmission rate, as Fig. 2a shows. The uploading delay of user m is given by $t_{m,k}^{\mathrm{up}} = \frac{s_m}{R_{m,k}^{\mathrm{up},1}}$.
- *Case 2.* If user m cannot finish task uploading in $t_{n,k}^{\mathrm{up},2}$, user m has a dedicated time duration to upload remain input data, as Fig. 2b shows. In this time duration, the co-channel interference does not exist. The uploading delay of user m is expressed as

$$t_{m,k}^{\mathrm{up}} = t_{n,k}^{\mathrm{up},2} + \frac{s_m - t_{n,k}^{\mathrm{up}} R_{m,k}^{\mathrm{up},1}}{R_{m,k}^{\mathrm{up},2}} \tag{3}$$

where $R_{m,k}^{\mathrm{up},2} = \log_2\left(1 + \frac{p_m h_{m,k}}{\sigma^2}\right)$ is the rate of user m on subchannel k without co-channel interference.

Thus, the uploading delay of user m is finally expressed as

$$t_{m,k}^{\mathrm{up},1} = \min\left\{\frac{s_m}{R_{m,k}^{\mathrm{up},1}}, t_{n,k}^{\mathrm{up},2}\right\} + \max\left\{0, \frac{s_m - t_{n,k}^{\mathrm{up},2} R_{m,k}^{\mathrm{up},1}}{R_{m,k}^{\mathrm{up},2}}\right\}. \tag{4}$$

The sum uploading delay of user m and user n using subchannel k is given by

$$t_{m,n,k}^{\mathrm{up}} = \begin{cases} t_{m,k}^{\mathrm{up},1} + t_{n,k}^{\mathrm{up},2}, & m \neq n, h_{n,k} < h_{m,k}, \\ t_{n,k}^{\mathrm{up},1} + t_{m,k}^{\mathrm{up},2}, & m \neq n, h_{n,k} > h_{m,k}, \\ t_{m,k}^{\mathrm{up},2}, & m = n. \end{cases} \tag{5}$$

The sum execution delay of user m and user n by the edge server is given by

$$t_{m,n}^{\mathrm{e}} = \begin{cases} t_m^{\mathrm{e}} + t_n^{\mathrm{e}}, & m \neq n, \\ t_m^{\mathrm{e}}, & m = n. \end{cases} \tag{6}$$

The case of $m = n$ in (5) means that subchannel k is used only by one user, and meanwhile $t_{m,n,k}^{\mathrm{up}}$ denotes the uploading delay of this user. Similarly, when $m = n$, $t_{m,n}^{\mathrm{e}}$ in (6) just includes the execution delay of one user at edge server.

C. Problem Formulation

In order to improve the general user experience in MEC system, we investigate the average overall delay of users on finishing tasks minimization (ADM) problem. It jointly considers the offloading decision, subchannel assignment, power control, and computation resource allocation. The average overall delay of users equals to the ratio of the sum of all users' delay on finishing tasks to the number of users, which reflects the general level of the delay for one user to finish the task at current scheduling period [27]. Let $\mathbf{X} = \left\{x_{m,n}^k \mid m \leqslant n, \forall m, n \in \mathcal{U}, \forall k \in \mathcal{S}\right\}$ denote the offloading decision and subchannel assignment variables. $x_{m,n}^k = 1$ if a pair of user, user m and user n, are assigned subchannel k to offload their tasks, and $x_{m,n}^k = 0$ otherwise. $x_{m,m}^k = 1$ indicates that user m is exclusively allocated subchannel k. Let $\mathbf{P} = \{p_m \mid m \in \mathcal{U}\}$ denotes the transmit power variables, and $\mathbf{F} = \{f_m^{\mathrm{e}} \mid m \in \mathcal{U}\}$ denotes the computation resource variables. We formulate the ADM problem as follows.

$$\begin{aligned}
\min_{\mathbf{P},\mathbf{X},\mathbf{F}} \ & \frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{n=m}^{M} x_{m,n}^{k}\left(t_{m,n,k}^{\mathrm{up}}+t_{m,n}^{\mathrm{e}}\right) \\
& +\frac{1}{M}\sum_{m=1}^{M}\left(1-\sum_{k=1}^{K}\left(\sum_{n=m+1}^{M}x_{m,n}^{k}+\sum_{n=1}^{m}x_{n,m}^{k}\right)\right)t_m^{\mathrm{l}} \\
\text{s.t. } \mathrm{C1}: & \sum_{m=1}^{M}\sum_{n=m}^{M}x_{m,n}^{k}\leqslant 1, \quad \forall k\in\mathcal{S} \\
\mathrm{C2}: & \sum_{k=1}^{K}\left(\sum_{n=m+1}^{M}x_{m,n}^{k}+\sum_{n=1}^{m}x_{n,m}^{k}\right)\leqslant 1, \quad \forall m\in\mathcal{U} \\
\mathrm{C3}: & 0\leqslant p_m\leqslant P_{\max}, \quad \forall m\in\mathcal{U} \\
\mathrm{C4}: & \sum_{m=1}^{M}f_m^{\mathrm{e}}\leqslant F, \quad \forall m\in\mathcal{U} \\
\mathrm{C5}: & f_m^{\mathrm{e}}\geqslant 0, \quad \forall m\in\mathcal{U} \\
\mathrm{C6}: & x_{m,n}^{k}\in\{0,1\}, \quad \forall m,n\in\mathcal{U}, \ \forall k\in\mathcal{S}
\end{aligned}$$

where $P_{\max}$ is the maximum transmit power for the users, and F is the total computational rate of edge server. C1 indicates that each subchannel is assigned to at most two users. C2 ensures that each user is allocated at most one subchannel. C3 restricts the maximum transmit power for all users. C4 and C5 denotes the total computational rate assigned to users cannot exceed F.

Due to the binary variable $\mathbf{X}$ and non-convex objective function, ADM problem is a mixed-integer and non-convex optimization problem. The subchannel assignment and power control can determine which case the users work in, and the power allocation scheme should be individually designed in two cases. Moreover, the computation resource allocation is influenced by the radio resource allocation. In general, this kind of problem is NP-hard, and there is not systematic approach to solve it. Therefore, we focus on designing an efficient algorithm.

III. Proposed Optimization-Theory Based Algorithm for ADM Problem

In this section, we decompose ADM problem into power allocation subproblem, computation resource allocation subproblem, and subchannel assignment subproblem. Specifically, the subchannel assignment subproblem is a master subproblem which can affect the solution of other two subproblems. The power allocation subproblem and computation resource allocation subproblem are independent of each other. We first propose the power allocation algorithm and computation resource algorithm, when the subchannel assignment is given. Then, we use the SDR and CCP method to solve subchannel assignment subproblem. Finally, the near-optimal solution of ADM problem can be obtained.

A. Uploading Power Allocation

The power allocations on different subchannels are independent and can be finished by a parallel approach. In this subsection, we first explore a criterion to determine whether two users on the same subchannel should work in Case 1 or Case 2 for minimizing their uploading delays. Then, we design an algorithm to find the optimal power allocation for each case. Without loss of generality, it is assumed that user m and user n are allocated subchannel k and their channel power gains meet $h_{m,k} > h_{n,k}$.

At first, we investigate how the co-channel interference affects the uploading delays of user m and user n, when they work in Case 2. Under current p_m and p_n, we give the following theorem.

Theorem 1: The decrease of user n's transmit power, $\bar{p}_n \triangleq p_n - \Delta p$, Δp>0, can result in the increase of user m's transmission rate, $\bar{R}_{m,k}^{\mathrm{up},1} = R_{m,k}^{\mathrm{up},1} + \Delta R_m$, and the increase of user n's uploading delay, $\bar{t}_{n,k}^{\mathrm{up},2} = t_{n,k}^{\mathrm{up},2} + \Delta t$. We can get the following formula

$$\lim_{\Delta p\to 0^+} R_{m,k}^{\mathrm{up},1}t_{n,k}^{\mathrm{up},2} + R_{m,k}^{\mathrm{up},2}\Delta t - \bar{R}_{m,k}^{\mathrm{up},1}\bar{t}_{n,k}^{\mathrm{up}} > 0. \tag{7}$$

Proof: See Appendix A. ■

Theorem 1 demonstrates that reducing p_n makes user m transmit less data bits. In other words, in Case 2, reducing p_n can prolong the uploading delays of both user m and user n. Since user m cannot cause any interference to user n, p_m should equal to $P_{\max}$ for minimizing the uploading delay of both user m and user n, which is described as the following lemma.

Lemma 1: The necessary condition for user m and user n achieving minimum uploading delay is $p_m = P_{\max}$.

Then, we give the following theorem about the sufficient condition such that user m and user n should work in Case 2 rather than Case 1.

Theorem 2: The sufficient condition for user m and user n achieving less uploading delay in Case 2 than that in Case 1 is that when $p_m = p_n = P_{\max}$, $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} < s_m$.

Proof: We can prove the theorem from two aspects. On the one hand, because Theorem 1 proves that the reduction of p_n increases the uploading delays of both user m and user n, we cannot reduce p_n when $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} < s_m$. On the other hand, Lemma 1 indicates that p_m should equal to $P_{\max}$. Thus, when $p_m = p_n = P_{\max}$, if $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} < s_m$, we should not do any adjustment to p_m and p_n, such that user m and user n should work in Case 2. ■

Based on Theorem 2, we can know that when $p_m = p_n = P_{\max}$, if $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} \geqslant s_m$, only p_n can be adjusted. Hence, user m and user n should work in Case 1 rather than Case 2. The following corollary can be given.

Corollary 1: The sufficient condition for user m and user n working in Case 1 is that $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} \geqslant s_m$ when $p_m = p_n = P_{\max}$.

Therefore, we can get a clear criterion. The criterion is that when $p_m = p_n = P_{\max}$, user m and user n should work in Case 1 if $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} \geqslant s_m$, and user m and user n should work in Case 2 if $t_{n,k}^{\mathrm{up},2}R_{m,k}^{\mathrm{up},1} < s_m$. For Case 2, it is obvious that the optimal power allocation is $p_m = p_n = P_{\max}$. For Case 1, the co-channel interference should be carefully managed for minimizing $t_{m,n,k}^{\mathrm{up}}$. Thus, we formulate the power allocation problem in Case 1 as follows.

$$\begin{aligned}&\min_{p_m,p_n} t^{\mathrm{up}}_{m,n,k}=t^{\mathrm{up},1}_{m,k}+t^{\mathrm{up},2}_{n,k}\\&\text{s.t. } p_m,p_n\in(0,P_{\max}]\end{aligned}\tag{8}$$

where $t^{\mathrm{up},1}_{m,k}=\frac{s_m}{R^{\mathrm{up},1}_{m,k}}$. Problem (8) is a typical non-convex optimization problem. To effectively tackle this problem, we introduce two variable substitutions, i.e., $t^{\mathrm{up},1}_{m,k}=\frac{1}{\tau_m}$ and $t^{\mathrm{up},2}_{n,k}=\frac{1}{\tau_n}$. Then, Problem (8) can be transformed into

$$\begin{aligned}&\min_{\tau_m,\tau_n}\frac{1}{\tau_m}+\frac{1}{\tau_n}\\&\text{s.t. } \mathrm{C7}:\frac{\sigma^2}{h_{n,k}}\left(2^{\frac{1}{B}s_n\tau_n}-1\right)\leqslant P_{\max}\\&\quad\ \ \mathrm{C8}:\frac{\sigma^2}{h_{m,k}}\left(2^{\frac{1}{B}s_m\tau_m}-1\right)2^{\frac{1}{B}s_n\tau_n}\leqslant P_{\max}\\&\quad\ \ \mathrm{C9}:\tau_m>0,\ \mathrm{C10}:\tau_n>0.\end{aligned}\tag{9}$$

Because of Lemma 1, two sides of C8 is equal, i.e., $\tau_m=\frac{B}{s_m}\log_2\left(1+\frac{P_{\max}h_{m,k}}{\sigma^2 2^{\frac{1}{B}s_n\tau_n}}\right)$. Problem (9) is reformulated as follows.

$$\begin{aligned}&\min_{\tau_n} h(\tau_n)=\frac{s_m\ln 2}{B\ln\left(1+\frac{P_{\max}h_{m,k}}{\sigma^2}2^{-\frac{1}{B}s_n\tau_n}\right)}+\frac{1}{\tau_n}\\&\text{s.t. } \mathrm{C7},\mathrm{C10}.\end{aligned}\tag{10}$$

For $h(\tau_n)$, we have the following important property.

Theorem 3: The function $h(\tau_n)$ is a unimodal function, because the function $h'(\tau_n)$ is monotonically increasing. The function $h'(\tau_n)$ is given by

$$h'(\tau_n)=-\frac{1}{\tau_n^2}+\frac{s_m s_n A\ln^2 2 g(\tau_n)}{B^2[1+Ag(\tau_n)]\ln^2[1+Ag(\tau_n)]}\tag{11}$$

where $g(\tau_n)=2^{-\frac{1}{B}s_n\tau_n}$ and $A=\frac{P_{\max}h_{m,k}}{\sigma^2}$.

Proof: See Appendix B. ■

According to Theorem 3, we can determine the optimal power allocation for Case 1. Specifically, from C7 and C10, we can get $\tau_n\in(0,\tau_n^{\max}]$, where $\tau_n^{\max}=\frac{B}{s_n}\log_2\left(1+\frac{P_{\max}h_{n,k}}{\sigma^2}\right)$. It is easily known that $\lim_{\tau_n\to 0^+}h(\tau_n)=-\infty$. If $h'(\tau_n^{\max})<0$, $h(\tau_n)$ is monotonically decreasing. In this case, the optimal solution $\tau_n^*=\tau_n^{\max}$. If $h'(\tau_n^{\max})>0$, $h(\tau_n)$ is first monotonically decreasing and then monotonically increasing. We can use bisection search [28] to find τ_n^* which meets $h'(\tau_n^*)=0$, and $h(\tau_n^*)$ is the minimum value. Based on the value of τ_n^*, we get the optimal solutions of p_m and p_n. We conclude the power allocation algorithm in Algorithm 1. The complexity of Algorithm 1 equals to $O(\log\mathit{\Delta})$, which is determined by the bisection search. $\mathit{\Delta}$ is the number of equal intervals in $(0,\tau_n^{\max}]$.

B. Computation Resource Allocation

When the subchannel assignment is determined, the computation resource allocation is irrelevant to power allocation. Let $\mathcal{U}^{\mathrm{e}}$ denote the set of users which are scheduled to offload their tasks to edge server. The computation resource allocation subproblem can be formulated as follows.

Algorithm 1 Power Allocation Algorithm

```
Initialization
• Input subchannel $k$, user $m$, and user $n$ with $h_{m,k} > h_{n,k}$.
• Set $p_m = 0$ and $p_n = 0$.
if subchannel $k$ is occupied by only one user $m \in \mathcal{U}$ or not occupied by any user then
    Set $p_m = P_{\max}$ or do not need any allocation.
else
    if $t^{\mathrm{up},2}_{n,k} R^{\mathrm{up},1}_{m,k} < s_m, p_m = p_n = P_{\max}$ then
        Set $p_m = P_{\max}$ and $p_n = P_{\max}$.
    else
        if $h'(\tau_n^{\max}) > 0$ then
            By exploiting the monotonically increasing property of $h'(\tau_n)$ when $\tau_n \in (0, \tau_n^{\max}]$, we use the bisection search to find $\tau_n^* \in (0, \tau_n^{\max}]$ such that $h'(\tau_n^*) = 0$.
        else
            By exploiting the monotonically decreasing property of $h(\tau_n)$ when $\tau_n \in (0, \tau_n^{\max}]$, we get $\tau_n^* = \tau_n^{\max}$.
        end if
    end if
    Calculate $\tau_m^* = \frac{B}{s_m}\log_2\left(1+\frac{P_{\max}h_{m,k}}{\sigma^2 2^{\frac{1}{B}s_n\tau_n^*}}\right)$.
    Calculate $p_m = \frac{\sigma^2}{h_{m,k}}\left(2^{\frac{1}{B}s_m\tau_m^*}-1\right)2^{\frac{1}{B}s_n\tau_n^*}$ and $p_n = \frac{\sigma^2}{h_{n,k}}\left(2^{\frac{1}{B}s_n\tau_n^*}-1\right)$.
end if
return $p_m$ and $p_n$.
```

$$\begin{aligned}&\min_{\mathbf{F}}\sum_{m\in\mathcal{U}^{\mathrm{e}}}t^{\mathrm{e}}_m\\&\text{s.t. } \mathrm{C4},\mathrm{C5}.\end{aligned}\tag{12}$$

Since Problem (12) is a typical convex problem, it can be directly solved by Lagrangian dual decomposition. We construct the Lagrangian function of Problem (12) by

$$L(\mathbf{F},\mu,\mathbf{v})=\sum_{m\in\mathcal{U}^{\mathrm{e}}}\frac{w_m}{f^{\mathrm{e}}_m}+\mu\left(\sum_{m\in\mathcal{U}^{\mathrm{e}}}f^{\mathrm{e}}_m-F\right)-\sum_{m\in\mathcal{U}^{\mathrm{e}}}v_m f^{\mathrm{e}}_m\tag{13}$$

where μ and $\mathbf{v}=\{v_m\mid m\in\mathcal{U}^{\mathrm{e}}\}$ are the Lagrangian multipliers of C4 and C5, respectively.

Since Problem (12) satisfies the Salter's condition, the strong duality holds between it and its dual problem such that the dual gap is zero. The KKT conditions specify that

$$\frac{\partial L}{\partial f^{\mathrm{e}}_m}=-\frac{w_m}{(f^{\mathrm{e}}_m)^2}+\mu-v_m=0\tag{14}$$

$$\mu\left(\sum_{m'\in\mathcal{U}^{\mathrm{e}}}f^{\mathrm{e}}_{m'}-F\right)=0\tag{15}$$

$$v_m f^{\mathrm{e}}_m=0\tag{16}$$

In order to minimize the overall execution delay, it must be $f^{\mathrm{e}}_m>0$, $\forall m\in\mathcal{U}^{\mathrm{e}}$, and hence $v_m=0$, $\forall m\in\mathcal{U}^{\mathrm{e}}$. From (14), we know that each user is assigned with $f_m^{\rm e} = \sqrt{\frac{w_m}{\mu}}$. Furthermore, when the optimal solution $f_m^{\rm e,*}$ is achieved, the computation resource should be used up, i.e., $\sum_{m'\in\mathcal{U}^{\rm e}} f_{m'}^{\rm e} - F = 0$. Therefore, we can get that $\mu = \left(\frac{1}{F}\sum_{m'\in\mathcal{U}^{\rm e}}\sqrt{w_{m'}}\right)^2$. The policy on computation resource allocation is given by

$$f_m^{\rm e} = \frac{F\sqrt{w_m}}{\sum\limits_{m'\in\mathcal{U}^{\rm e}}\sqrt{w_{m'}}}, \forall n \in \mathcal{U}^{\rm e}. \tag{17}$$

C. Offloading Decision and Subchannel Assignment

Based on the power allocation policy presented in Algorithm 1, we can directly determine the optimal solution $t_{m,n,k}^{\rm up,*}$ of the uploading delays. By the policy of the computation resource allocation, we can express the execution delay of user m's task by edge computing as

$$t_m^{\rm e,*} = \frac{\sqrt{w_m}}{F}\sum_{n=1}^{M}\sqrt{w_n}\sum_{k=1}^{K}\left(\sum_{l=1}^{n} x_{l,n}^k + \sum_{l=n+1}^{M} x_{n,l}^k\right), \quad \forall m\in\mathcal{U} \tag{18}$$

which can be used to calculate $t_{m,n}^{\rm e,*}$ according to 6. After getting $t_{m,n,k}^{\rm up,*}$ and $t_{m,n}^{\rm e,*}$, we can formulate the offloading decision and subchannel assignment subproblem as follows.

$$\begin{aligned}\min_{\mathbf{X}}\ & \frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{n=m}^{M} x_{m,n}^k\left(t_{m,n}^{\rm e,*} + t_{m,n,k}^{\rm up,*}\right)\\ & + \frac{1}{M}\sum_{m=1}^{M}\left[1 - \sum_{k=1}^{K}\left(\sum_{n=m+1}^{M} x_{m,n}^k + \sum_{n=1}^{m} x_{n,m}^k\right)\right] t_m^{\rm l}\\ \text{s.t.}\ & \text{C1, C2, C6}\end{aligned} \tag{19}$$

which is an integer quadratic programming problem. Although its optimal solution can be found by Branch-and-Bound algorithm [28], it has huge computational complexity, especially when the dimension of $\mathbf{X}$ is large. Thus, we propose an efficient algorithm based on the SDR.

Problem (19) can be reformulated as an equivalent QCQP problem [29] by replacing the binary constraints with quadratic constraints that are expressed as

$$x_{m,n}^k\left(1 - x_{m,n}^k\right) = 0, \quad 0 \leqslant x_{m,n}^k \leqslant 1 \Leftrightarrow x_{m,n}^k \in \{0,1\}. \tag{20}$$

The reformulated problem is expressed as

$$\begin{aligned}\min_{\boldsymbol{x}}\ & O(\boldsymbol{x}) = \frac{1}{M}\left(\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \boldsymbol{r}^T\boldsymbol{x} + \sum_{m=1}^{M} t_m^{\rm l}\right)\\ \text{s.t.}\ & \text{C11}: \mathbf{A}_1\boldsymbol{x} \leqslant \mathbf{1}_K\\ & \text{C12}: \mathbf{A}_2\boldsymbol{x} \leqslant \mathbf{1}_M\\ & \text{C13}: \boldsymbol{e}_p^T\boldsymbol{x} - \boldsymbol{x}^T\text{diag}(\boldsymbol{e}_p)\boldsymbol{x} \leqslant 0, \quad p = 1,\ldots,UK\\ & \text{C14}: \ 0 \leqslant \boldsymbol{x} \leqslant 1\end{aligned} \tag{21}$$

where $U = \frac{1}{2}M(M+1)$. e_p is a $UK \times 1$ unit vector with the pth element being 1. $\text{diag}(\boldsymbol{e}_p)$ is a diagonal matrix of $\boldsymbol{e}_p$ starting in the upper left corner. The variable $\mathbf{X}$ is arranged as the vector $\boldsymbol{x} = \left(\boldsymbol{x}_{1,1}^T,\ldots,\boldsymbol{x}_{1,M}^T,\ldots,\boldsymbol{x}_{K,1}^T,\ldots,\boldsymbol{x}_{K,M}^T\right)^T$, where $\boldsymbol{x}_{k,m} = \left(x_{m,m}^k,\ldots,x_{m,M}^k\right)^T$. The matrix $\mathbf{Q} = \frac{1}{2}\left(\mathbf{Q}_1 + \mathbf{Q}_1^T\right)$. $\mathbf{Q}_1$ is expressed as

$$\mathbf{Q}_1 = \begin{bmatrix}\mathbf{Q}_0 & \mathbf{Q}_0 & \cdots & \mathbf{Q}_0\\ \mathbf{Q}_0 & \mathbf{Q}_0 & \cdots & \mathbf{Q}_0\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{Q}_0 & \mathbf{Q}_0 & \cdots & \mathbf{Q}_0\end{bmatrix} \tag{22}$$

which is a $K \times K$ block matrix and each block is matrix $\mathbf{Q}_0 = \frac{1}{F}\boldsymbol{q}\boldsymbol{q}^T$. In matrix $\mathbf{Q}_0$, the vector $\boldsymbol{q} = \left(q_{1,1},\ldots,q_{1,M},q_{2,2},\ldots,q_{2,M},\ldots,q_{M,M}\right)^T$, and $q_{m,n}, \forall\{m,n\}\in\mathcal{U}$ is expressed as

$$q_{m,n} = \begin{cases}\sqrt{w_m} + \sqrt{w_n}, & m < n,\\ \sqrt{w_m}, & m = n.\end{cases} \tag{23}$$

The vector $\boldsymbol{r} = \left(\boldsymbol{r}_{1,1}^T,\ldots,\boldsymbol{r}_{1,M}^T,\ldots,\boldsymbol{r}_{K,1}^T,\ldots,\boldsymbol{r}_{K,M}^T\right)^T$ and $\boldsymbol{r}_{k,m} = \left(r_{m,m}^k,\ldots,r_{m,M}^k\right)^T$, where $r_{m,n}^k, \forall\{m,n\}\in\mathcal{U}, k\in\mathcal{S}$ is expressed as

$$r_{m,n}^k = \begin{cases}t_{m,n,k}^{\rm up,*} - t_m^{\rm l} - t_n^{\rm l}, & m \neq n\\ t_{m,n,k}^{\rm up,*} - t_m^{\rm l}, & m = n.\end{cases} \tag{24}$$

The matrix $\mathbf{A}_1$ and $\mathbf{A}_2$ are respectively expressed as

$$\mathbf{A}_1 = \begin{bmatrix}\mathbf{1}_{1\times U} & \mathbf{0}_{1\times U} & \cdots & \mathbf{0}_{1\times U}\\ \mathbf{0}_{1\times U} & \mathbf{1}_{1\times U} & \cdots & \mathbf{0}_{1\times U}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0}_{1\times U} & \mathbf{0}_{1\times U} & \cdots & \mathbf{1}_{1\times U}\end{bmatrix}_{K\times KU} \tag{25}$$

$$\mathbf{A}_2 = \left[\mathbf{A}_{2,1},\ldots,\mathbf{A}_{2,k},\ldots,\mathbf{A}_{2,K}\right] \tag{26}$$

$$\mathbf{A}_{2,k} = \begin{bmatrix}\mathbf{1}_{1\times M} & \mathbf{0}_{1\times M-1} & \mathbf{0}_{1\times M-2} & \cdots & 0\\ \boldsymbol{u}_{1\times M}^2 & \mathbf{1}_{1\times M-1} & \mathbf{0}_{1\times M-2} & \cdots & 0\\ \boldsymbol{u}_{1\times M}^3 & \boldsymbol{u}_{1\times M-1}^2 & \mathbf{1}_{1\times M-2} & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & 0\\ \boldsymbol{u}_{1\times M}^M & \boldsymbol{u}_{1\times M-1}^{M-1} & \boldsymbol{u}_{1\times M-2}^{M-2} & \cdots & 1\end{bmatrix}_{M\times U}, \quad \forall k\in\mathcal{S} \tag{27}$$

where $\mathbf{1}_{1\times i}$ is $1\times i$ vector with all elements being 1, and $\boldsymbol{u}_{1\times i}^p$ is $1\times i$ unit vector with the pth element being 1. C11 and C12 are equivalent to C1 and C2, respectively. C13 and C14 replace C6. To tackle Problem (21), we define $\mathbf{Z} = \boldsymbol{x}\boldsymbol{x}^T$ and equivalently rewrite Problem (21) as

$$\begin{aligned}\min_{\mathbf{Z},\boldsymbol{x}}\ & \frac{1}{M}\left(\text{Tr}(\mathbf{QZ}) + \boldsymbol{r}^T\boldsymbol{x} + \sum_{m=1}^{M} t_m^{\rm l}\right)\\ \text{s.t.}\ & \text{C11, C12, C14}\\ & \text{C15}: \text{Tr}(\text{diag}(\boldsymbol{e}_p)\mathbf{Z}) - \boldsymbol{e}_p^T\boldsymbol{x} = 0, p = 1,\ldots,UK\\ & \text{C16}: \begin{bmatrix}\mathbf{Z} & \boldsymbol{x}\\ \boldsymbol{x}^{\rm T} & 1\end{bmatrix} \succeq 0\end{aligned} \tag{28}$$

where C15 is equivalent to C13. Because only C16 is nonconvex, we first solve the optimal solution $\mathbf{Z}^*$ of Problem (28) without C16 by the standard convex programming method [30]. If $\mathbf{Z}^*$ is semidefinite, we directly obtain the optimal solution x^* of Problem (21) according to $\mathbf{Z} = \boldsymbol{x}\boldsymbol{x}^T$. If $\mathbf{Z}^*$ is not semidefinite, we just get a lower bound of the optimal solution. The SDR is $\frac{2}{\pi}$-optimal solution for binary quadratic programming problem [29]. Hence, we use the

Gaussian randomization and CCP to design an algorithm to obtain an approximate solution.

Specifically, we generate L sampling points based on the multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with mean $\mu = \boldsymbol{x}^*$ and covariance $\Sigma = \mathbf{Z}^* - \boldsymbol{x}^* \boldsymbol{x}^{*T}$. Although there is no guarantee that all the sampling points are feasible, we can use them to apply CCP to find the approximate solution. The matrix $\mathbf{Q}$ can be decomposed into the difference of two positive semidefinite matrices

$$\mathbf{Q} = \mathbf{Q}_+ - \mathbf{Q}_-, \; \mathbf{Q}_+ \succeq 0, \mathbf{Q}_- \succeq 0 \tag{29}$$

where $\mathbf{Q}_+ = \mathbf{Q} - \eta \mathbf{I}$ and $\mathbf{Q}_- = -\eta \mathbf{I}$. η is the minimum eigenvalue of $\mathbf{Q}$ and $\mathbf{I}$ is the identity matrix with same dimension of $\mathbf{Q}$. The objective function of Problem (21) can be expressed as $O(\boldsymbol{x}) = \frac{1}{M}\left(\boldsymbol{x}^T \mathbf{Q}_+ \boldsymbol{x} + \boldsymbol{r}^T \boldsymbol{x} + \sum_{m=1}^{M} t_m^{\mathrm{l}} - \boldsymbol{x}^T \mathbf{Q}_- \boldsymbol{x}\right)$. Thus, Problem (21) can be rewritten as a difference-of-convex (DC) programming problem. The CCP is a powerful method to find a local optimum of DC programming problems. Considering the infeasibility of the sampling points, we adopt the penalty CCP. In order to perform the penalty CCP, we convexify $O(\boldsymbol{x})$ and C13 by linearizing their concave parts around the iteration point $\boldsymbol{x}_j$. Then, Problem (21) transforms into the following form

$$\begin{aligned} \min_{\boldsymbol{x}, \theta} \; & O_j(\boldsymbol{x}, \theta) = \frac{1}{M}\left[\boldsymbol{x}^T \mathbf{Q}_+ \boldsymbol{x} + \boldsymbol{r}^T \boldsymbol{x} + \sum_{m=1}^{M} t_m^{\mathrm{l}}\right] \\ & -\frac{1}{M}\left[\boldsymbol{x}_j^T \mathbf{Q}_- \boldsymbol{x}_j + \boldsymbol{x}_j^T \mathbf{Q}_-^T (\boldsymbol{x} - \boldsymbol{x}_j)\right] + \tau_j \sum_{p=1}^{UK} \theta_p \\ \text{s.t.} \; & \mathrm{C11}, \mathrm{C12}, 14 \\ & \mathrm{C17}: -\boldsymbol{x}_j^T \operatorname{diag}(\boldsymbol{e}_p) \boldsymbol{x}_j - \boldsymbol{x}_j^T \operatorname{diag}(\boldsymbol{e}_p)^T (\boldsymbol{x} - \boldsymbol{x}_j) \\ & + \boldsymbol{e}_p^T \boldsymbol{x} \leqslant \theta_p, \quad p = 1, \ldots, UK \\ & \mathrm{C18}: \theta_p \geqslant 0, \quad p = 1, \ldots, UK \end{aligned} \tag{30}$$

where τ_j and $\theta = \{\theta_p, p = 1, \ldots, UK\}$ are penalty factors. The penalty factors relax the original problem and penalize the violations for the constraints [29]. We use each sampling point as the initial point to perform the penalty CCP method. The detailed procedure of the proposed optimization-theory based computation offloading and resource allocation (OCORA) scheme is concluded in Algorithm 2. If the interior-point method is applied, the complexity of step 3 is $O\left(U^7 \varepsilon^2\right)$, and the complexity from steps 5 to 13 is $O\left(L\Gamma U^{3.5} K^{3.5} \varepsilon^2\right)$, where ε is the search step size. The complexity of OCORA scheme is $O\left(UK \log \Delta + U^7 K^7 \varepsilon^2\right)$.

IV. Proposed Matching-Theory Based Algorithm for ADM Problem

To further reduce the computational complexity, we propose a low-complexity algorithm for ADM problem based on matching theory [31]–[33]. Given the power allocation and the policy of computation resource allocation, the offloading decision and subchannel assignment can be reformulated as a many-to-one bipartite matching problem with externalities among the users.

Algorithm 2 Optimization-Theory Based Computation Offloading and Resource Allocation (OCORA) Scheme

```
1: Calculate $t_{m,n,k}^{\mathrm{up},*}, \forall \{m,n\} \in \mathcal{U}, \forall k \in \mathcal{S}$ by performing Algorithm 1.
2: Construct matrix $\mathbf{Q}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{Q}_-$ and $\mathbf{Q}_+$.
3: Solve Problem (28) to get $\mathbf{Z}^*$.
4: Generate the sampling points $\mathcal{X} = \{\boldsymbol{x}_1^{\mathrm{s}}, \ldots, \boldsymbol{x}_L^{\mathrm{s}}\}$ based on $\mathcal{N}(\mu, \Sigma)$.
5: for $l = 1:1:L$ do
6:   Set $j = 0$, $\boldsymbol{x}_0 = \boldsymbol{x}_l^{\mathrm{s}}$, $\tau_0 = 1$, $O_0 = +\infty$, and $\mu = 1.1$.
7:   repeat
8:     $j = j + 1$.
9:     Solve Problem (30) to get $O_j$ and $\boldsymbol{x}_j$ by linearing concave parts around $\boldsymbol{x}_{j-1}$.
10:    $\tau_j = \mu \tau_{j-1}$ and $e = |O_j - O_{j-1}|$.
11:  until $e \leqslant 1e-4$ or $j \geqslant j_{\max}$.
12:  $O_{\mathrm{opt}}^l = O_j$ and $\boldsymbol{x}_{\mathrm{opt}}^l = \boldsymbol{x}_j$.
13: end for
14: $\boldsymbol{x}_{\mathrm{opt}} = \arg\min_{\boldsymbol{x}_{\mathrm{opt}}^l} O_{\mathrm{opt}}^l$ and $O_{\mathrm{opt}} = \min_l O_{\mathrm{opt}}^l$.
15: return $O_{\mathrm{opt}}$, $\boldsymbol{x}_{\mathrm{opt}}$, and all uploading power variables.
```

A. Many-to-One Matching Model for ADM Problem

$\mathcal{U}$ and $\mathcal{S}$ are two mutually disjoint sets. In NOMA enabled task uploading, each subchannel can be assigned with at most two users simultaneously, and each user is allowed to access for at most one subchannel. This problem is to match the users to the subchannels aiming to minimize the average overall delay, where the users and the subchannels affect each other. Hence, ADM problem is a many-to-one matching problem, which has been studied extensively in economics and game theory [32], [34].

Definition 1: In the many-to-one matching model, μ is a function from the set $\mathcal{U} \cup \mathcal{S}$ into the set of all subsets of $\mathcal{U} \cup \mathcal{S} \cup \{0\}$ such that $\forall m \in \mathcal{U}$ and $\forall k \in \mathcal{S}$

1) $|\mu(m)| = 1$ *for each user* $m \in \mathcal{U} \cup \mathcal{O}$*;*
2) $|\mu(k)| \leqslant q_k$ *for each subchannel* $k \in \mathcal{S}$ *and* $|\mu(0)| \leqslant M$ *for subchannel 0;*
3) $\mu(m) \in \mathcal{S} \cup \{0\}$ *if and only if* $k \in \mu(m), \forall m \in \mathcal{U} \cup \mathcal{O}, \forall k \in \mathcal{S} \cup \{0\}$*;*
4) $k \in \mu(m) \Leftrightarrow m \in \mu(k), \forall m \in \mathcal{U} \cup \mathcal{O}, \forall k \in \mathcal{S} \cup \{0\}$.

The positive integer q_k denotes *quota*, which indicates the maximum number of users that can be matched with each subchannel. In this work, $q_k = 2, \forall k \in \mathcal{S}$. If the user is matched with subchannel 0, it is allocated to execute the task locally. The set $\mathcal{O}$ includes the "holes" which use up all available vacancies of subchannels. The meanings of μ are different when the parameters are different. For user $m \in \mathcal{U}$, $\mu(m)$ means its matched subchannel. For subchannel $k \in \mathcal{S}$, $\mu(k)$ gives the set of matched users. Due to the co-channel interference and computation resource competition between the users, each subchannel's preference depends not only on the users whom it support, but also on the users which the other subchannels support. Similarly, each user's preference is related to not only the matched subchannel but also all other subchannels. Therefore, we conclude that ADM problem is a many-to-one matching problem with the externalities [31], [32], [34].

To model the externalities, we define the preference value for the user on subchannel k as follows. If user m is the one with high channel gain on subchannel k, its preference value is expressed as $v_m(k) = t^{\mathrm{up},1}_{m,k} + t^{\mathrm{e}}_{m}$. If user m is that one with low channel gain on subchannel k, its preference value is $v_m(k) = t^{\mathrm{up},2}_{m,k} + t^{\mathrm{e}}_{m}$. If user m is matched with subchannel 0, its preference value by local computing is expressed as $v_m(0) = t^{\mathrm{l}}_{m}$. Then, we define the preference value of subchannel k as $v_k(\mu(k) = \{m,n\}) = t^{\mathrm{up}}_{m,n,k} + t^{\mathrm{e}}_{m,n}$, $\forall \{m,n\} \in \mathcal{M}$.

Each user has a preference relation $\succ_m, \forall m \in \mathcal{U}$, and each subchannel k has a strict preference relation $\succ_k, \forall k \in \mathcal{S} \cup \{0\}$. For user m and any two subchannels k and k', the preference relation of two matchings μ and μ' where $\mu(m) = k$ and $\mu'(m) = k'$ is expressed as

$$(k, \mu) \succ_m (k', \mu') \Leftrightarrow v_m(k) < v_m(k') \tag{31}$$

which indicates that user m prefers subchannel k of matching scheme μ rather than subchannel k' of matching scheme μ', since user m can obtain the lower delay on subchannel k than that on subchannel k'. For two sets of users $\mathcal{M}$ and $\mathcal{M}'$, we express the preference relation of subchannel k on two matchings μ and μ' where $\mu(k) = \mathcal{M}$ and $\mu'(k) = \mathcal{M}'$ as

$$(\mathcal{M}, \mu) \succ_k (\mathcal{M}', \mu') \Leftrightarrow v_k(\mathcal{M}) < v_{k'}(\mathcal{M}') \tag{32}$$

which means subchannel k prefers $\mathcal{M}$ to $\mathcal{M}'$ if subchannel k can get the lower delay from $\mathcal{M}$.

Since the externalities can affect the preference relation of each subchannel and user, it is hard to define a strong stability for this problem. Hence, the two-sided exchange stability is defined to tackle the externalities [34]. To achieve the exchange stability, we firstly define the swap matching μ^n_m as

$$\mu^n_m = \{\mu \setminus \{(m,k),(n,j)\} \cup \{(n,k),(m,j)\}\} \tag{33}$$

where $\mu(m) = k$ and $\mu(n) = j$. μ^n_m means that user m and user n exchange the subchannels with each other while keeping all other users' allocation. Note that one of the users in swap operation can be a "hole" to allow a user to move the vacancies of a subchannel. Then, we give the definition of swap-blocking pair.

Definition 2: In a matching μ, user m and user n can be a swap-blocking pair if and only if

1) $\forall i \in \{m,n,k,j\}, v_i(\mu^n_m(i)) \geqslant v_i(\mu(i))$,
2) $\exists i \in \{m,n,k,j\}, v_i(\mu^n_m(i)) > v_i(\mu(i))$,

where $k = \mu(m)$ and $j = \mu(n)$ are the subchannels.

The swap-blocking pair ensures that if a swap matching is approved, the preference values of all involved users and subchannels will not increase, and at least one user or one subchannel will decrease. Thus, we can check every pair of the users (or one user and a hole) whether they form a swap-blocking pair or not. If they form a candidate of swap-blocking pair, they can operate the swap matching by exchanging their matched subchannels to reduce their overall delays. Through a series of swap matchings, we finally obtain the two-sided exchange-stable matching which is defined as follows.

Definition 3: A matching μ is two-sided exchange-stable if and only if there does not exist a swap-blocking pair.

B. Proposed Algorithm Based on Many-to-One Matching

To find a two-sided exchange-stable matching for our considered problem, we propose a matching-theory based computation offloading and resource allocation (MCORA) algorithm, which is presented in Algorithm 3. To initialize the matching, we randomly match each user to one subchannel $k \in \mathcal{S} \cup \{0\}$. Under the initial matching μ_0, each user's overall delay is calculated according to Algorithm 1 and (18). In swap matching phase, each user keeps finding other users to form the swap-blocking pair to iteratively reduce its overall delay. Specifically, for two users or one user and one hole, we utilize Algorithm 3 and (18) to the preference values of the swap matching. If a swap-blocking pair is found, the swap matching must be approved. The swap matching phase will stop until there is not any swap-blocking pair. Because there is not any swap-blocking pair, the final matching μ^* of MCORA algorithm achieves the two-sided exchange stability.

The MCORA algorithm converges within a limited number of iterations. Since the number of users and subchannels are limited, it indicates that the number of swap operations is finite [26], [34]. Furthermore, according to Definition 2, we find that the average overall delay decreases after each approved swap matching. As the average overall delay is close to the lower bound, the swap matching will not be found anymore. Thus, the MCORA algorithm can converge to a two-sided exchange-stable matching with limited iterations.

V. Simulation Results

We present the simulation results to evaluate the performance of proposed algorithms for multi-carrier NOMA-enabled MEC system. We consider that the users are uniformly distributed in a single cell network. The spectrum bandwidth consists of multiple adjacent subchannels, and the interval between any two adjacent subchannels is 20 kHz [35]. The user's input-data size is uniformly distributed in [200, $s_{\max}$] in kbits. For most of the applications, the task workload correlates to input-data size. Here, we consider $w_m = \beta s_m, \forall m \in \mathcal{M}$ and $\beta = 1000$ cycles/bit [1]. The user's device computational rate is uniformly distributed in [0, $f^{\mathrm{l}}_{\max}$] in Gcycles/s. We summarize the simulation parameters in Table I. In this simulation, we compare the proposed algorithms with the scheme in [36] for the OMA based system. Moreover, the performances of four benchmark algorithms have been evaluated for comparison.

- Exhaustive search (ES) scheme: The scheme searches all possible results of offloading decision and subchannel assignment and performs proposed uploading power allocation algorithm and computation resource allocation policy to achieve the globally optimal solution.
- Differentiation-neglected (Diff-NG) scheme: The scheme ignores the differentiated uploading delays of NOMA

Algorithm 3 Matching-Theory Based Computation Offloading and Resource Allocation (MCORA) Algorithm

```
Initialization
• Initial matching $\mu = \mu_0$: The users are randomly matched to the subchannels $k \in \mathcal{S} \cup \{0\}$ while ensuring $|\mu(k)| \leqslant q_k, \forall k \in \mathcal{S}$.
• Calculate the preference values under $\mu_m^n$ by Algorithm 1 and Equation (18).
$flag = 1$.
while $flag = 1$ do
  $flag = 0$.
  for each $m \in \mathcal{U}$ do
    for each $n \in \mathcal{U} \cup \mathcal{O} \setminus m$ do
      Exploit Algorithm 1 and (18) to calculate the preference values of involved users and subchannels under $\mu_m^n$.
      if user $m$ and $n$ form a swap-blocking pair then
        Set $flag = 1$ and update $\mu = \mu_m^n$.
        break
      end if
    end for
    if $flag = 1$ then
      break
    end if
  end for
end while
return a stable matching $\mu$
```

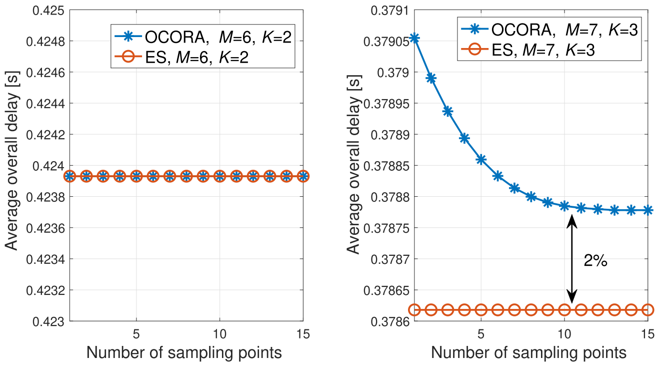
TABLE I
SIMULATION PARAMETERS

Parameters	Values
Cell radius	300 m
Maximum power, $P_{\max}$	24 dBm
Subchannel bandwidth, B	180 kHz [9], [37]
Noise power, σ^2	-174 dBm/Hz $\times B$
Path loss model	$37.6 \log_{10}(d[\mathrm{km}]) + 128.1$
Multiple-path fading	Exponential distribution with unit mean
Shadowing	Log-normal distribution with standard deviation of 8 dB

users and considers the co-channel interference always exists during the process of task uploading [10], [12].

- Maximum power (Max-power) scheme: The scheme allocates the users maximum transmit power, and its other procedures are same with OCORA algorithm.
- ECA-NOMA scheme [9]: The scheme allows all users to offload tasks and allocates them the subchannels in a greedy approach. The power allocation algorithm and computation resource allocation algorithm are similar with this paper.

A. Feasibility of Proposed Algorithms



(a) (b)

Fig. 3. Average overall delay versus the number of sampling points L ($s_{\max} = 800$ kbits, $F = 24$ Gcycles/s, $f_{\max}^1 = 1$ Gcycles/s).

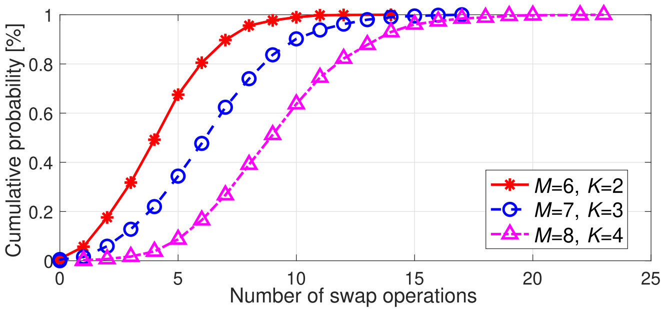


Fig. 4. Convergence of MCORA scheme.

Fig. 3 shows the performance comparison between OCORA scheme and ES scheme. We can see that with the increase of the number of the sampling points L, the OCORA scheme gradually reduces the performance gap against the optimal solution obtained by ES scheme. Fig. 3a shows when $M = 6$ and $K = 2$, the OCORA scheme achieves the same performance with ES scheme. Fig. 3b shows that when $M = 7$ and $K = 3$, the performance gap is less than 2% between ES and OCORA within 15 sampling points. This is because more sampling points leads to higher probability to find a better solution. Fig. 4 shows the cumulative probability of the number of swap operations for MCORA. We can see that the MCORA scheme converges after a limited number of iterations, and the required number of iterations depends on M and K.

B. Impact of the Computational Rate on Network Performance

Fig. 5 shows the average overall delay versus the edge computational rate F of different schemes. It is seen that with the increase of F, the average overall delay decreases. We can see that two proposed algorithms achieve lower average overall delay than Diff-NG scheme and Max-power scheme. Furthermore, with the growth of F, the gaps between them are enlarged. This is because that the OCORA and MCORA schemes consider the difference of uploading delay between users to allocate the user's transmit power for different cases. Since the user pairing and subchannel assignment are efficiently solved, the proposed algorithms obtain better average overall delay than ECA-NOMA.

Fig. 6 shows that the average overall delay versus maximum device computational rate $f_{\max}^1$ of different schemes. We can see that with the increase of $f_{\max}^1$, the average overall delay is

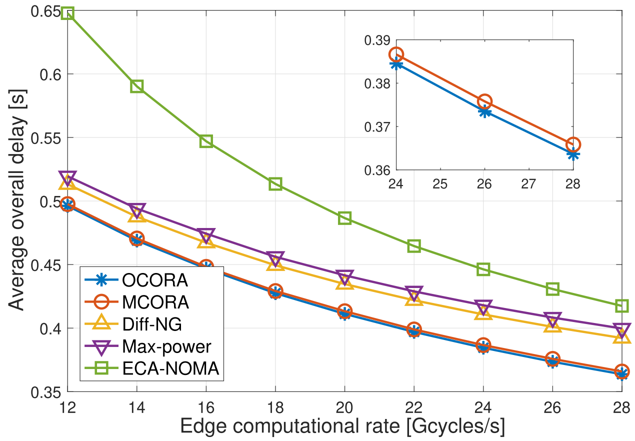


Fig. 5. Average overall delay versus edge computational rate F ($M = 10$, $K = 5$, $s_{\max} = 800$ kbits, $f^{l}_{\max} = 1$ Gcycles/s).

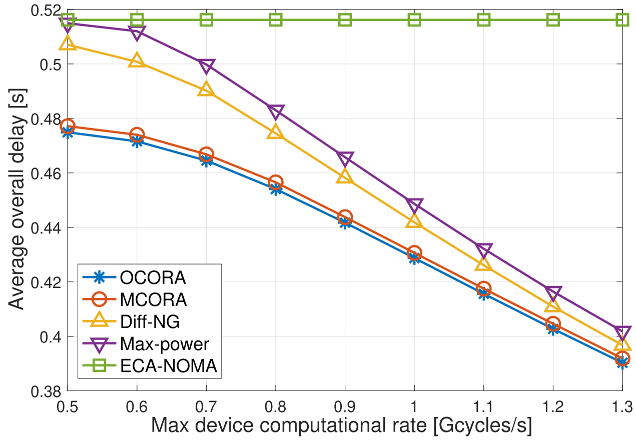


Fig. 6. Average overall delay versus maximum device computational rate $f^{l}_{\max}$ ($M = 10$, $K = 5$, $s_{\max} = 800$ kbits, $F = 18$ Gcycles/s).

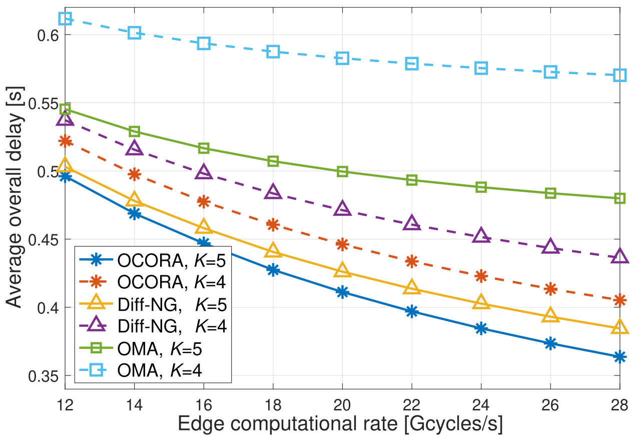


Fig. 7. Average overall delay versus F with the different number of subchannels K ($M = 10$, $s_{\max} = 800$ kbits, $f^{l}_{\max} = 1$ Gcycles/s).

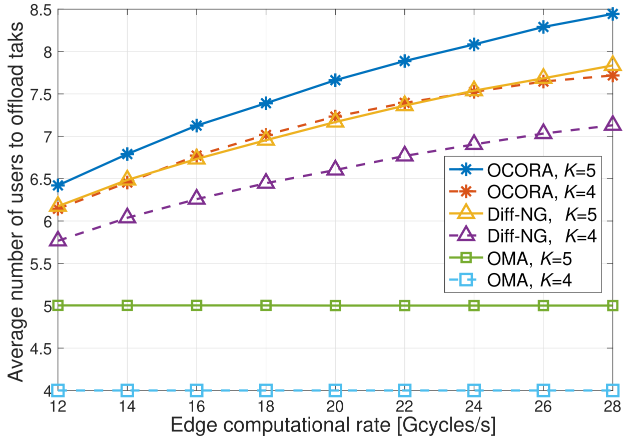


Fig. 8. Average number of users to offload task versus F with different K ($M = 10$, $s_{\max} = 800$ kbits, $f^{l}_{\max} = 1$ Gcycles/s).

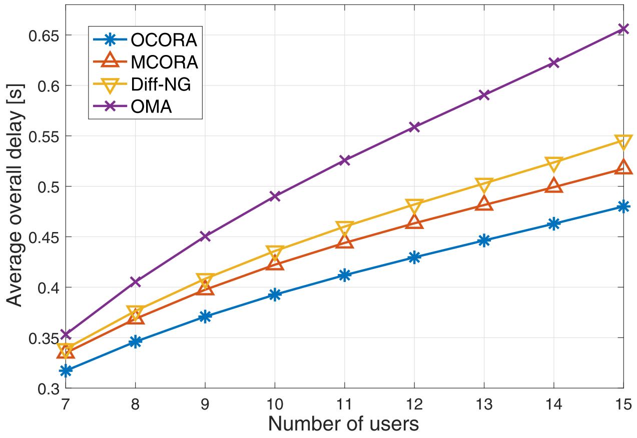


Fig. 9. Average overall delay versus the number of users M ($K = 5$, $s_{\max} = 800$ kbits, $F = 26$ Gcycles/s, $f^{l}_{\max} = 1$ Gcycles/s).

reduced. Since K is sufficient for all users to offload the tasks, the ECA-NOMA scheme decides all users to offload their tasks such that its average overall delay is constant. It is seen that two proposed schemes achieve better performance than other schemes. However, the gaps between different schemes are gradually diminished except ECA-NOMA scheme. This is because that the higher $f^{l}_{\max}$ leads to that more users are decided to conduct the local computing.

C. Impact of the Spectrum Resource on Network Performance

Fig. 7 shows that the average overall delay with the different number of subchannels K. From the results, we can see that under given F, the increase of K can reduce the average overall delay. The reason is that more subchannels provide higher transmission capacity to reduce the task uploading delay of each user. Compared with the OMA based scheme, the NOMA based scheme can significantly reduce the average overall delay on computation offloading. Fig. 8 shows that the average number of users to offload tasks with different K. We can see that the increase of K can improve the number of users to offload tasks in NOMA system, and the OCORA scheme can obtain more users to offload tasks than other schemes. The reasons come from two aspects. The first one is that larger K can provide higher the transmission capacity to offload more user's tasks. The second one is that the OCORA scheme efficiently exploits the edge server's computational resources to provide low offloading delay for more users to offload tasks.

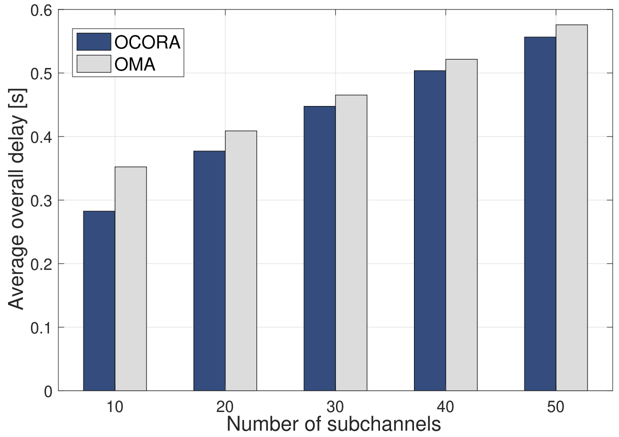


Fig. 10. Average overall delay under different K and M ($s_{\max} = 800$ kbits, $F = 64$ Gcycles/s, $f^1_{\max} = 1$ Gcycles/s).

To evaluate the performance of proposed schemes in the subchannel-limited scenario, we plot the average overall delay of different schemes versus the number of users M, as shown in Fig. 9. With the growth of M, the subchannel competition among the users become fierce. Different from the existing schemes, the proposed OCORA scheme and MCORA scheme take into account the difference of uploading delay between users to efficiently restrain the co-channel interference. Thus, the competition among the users can be effectively dealt with such that lower the average overall delay is achieved.

Furthermore, we consider a scenario with large M and K. Specifically, K is set as 10, 20, 30, 40, and 50. Corresponding to K, M is set as 15, 30, 45, 60, and 75. For instance, when $K = 50$ and $M = 75$, the MEC system has 10MHz spectrum bandwidth for all the users to offload the tasks. Fig. 10 shows the average overall delay varying with K and M. We can see that as the increase of K and M, although the performance gap between OCORA scheme and OMA scheme gradually reduces, the OCORA scheme still outperforms the OMA scheme. This is because the limited F cannot afford the task offloading for all users. With the increase of M, more users is allocated to conduct the local computing. Hence, the performance gap between the OCORA scheme and OMA scheme is diminished. Due to the high spectrum efficiency of NOMA, the proposed OCORA scheme achieves lower average overall delay than OMA scheme.

D. Impact of the Input-Data Size on Network Performance

We adopt two specific scenarios to show the necessity of considering the difference of the uploading delays between users. In such scenarios, the channel model only includes the path loss.

The first specific scenario includes one subchannel and two users. For user 1 and user 2, the link length is 150 m and 50 m, respectively. The computational rate is set as $f^1_1 = f^1_2 = 0.2$ Gcycles/s and $F = 24$ Gcycles/s. The input-data size of user 1's task s_1 is 300 kbits. Fig. 11 shows the average overall delay of user 1 and user 2 varying with the input-data size of user 2's task s_2. The Case 1 only scheme only uses the power allocation method of Case 1 to perform their

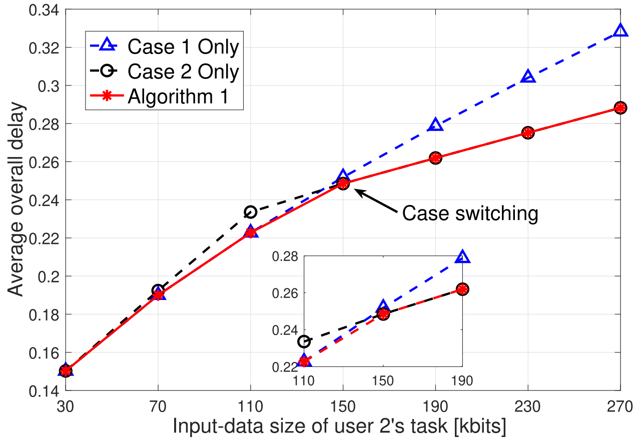


Fig. 11. Average overall delay versus input-data size of user 2's task.

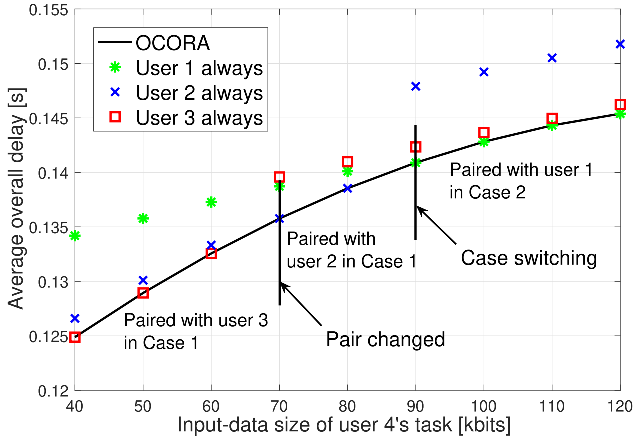


Fig. 12. Average overall delay versus input-data size of user 4's task.

task uploading, and the Case 2 only scheme only uses the power allocation method of Case 2. We can see that since the proposed power allocation algorithm (Algorithm 1) jointly considers the Case 1 and Case 2, it achieves lower average overall delay than other two schemes. Furthermore, from the results, we can see that there are two phases for Algorithm 1. With the increase of s_2, user 1 and user 2 first work in Case 1, and then switch to work in Case 2. This is because when s_2 is small, user 2 can finish task uploading earlier than user 1. Thus, they should work in Case 1 to achieve minimum uploading delay, which validates Corollary 1. As s_2 increase to a certain value, user 2 cannot upload total input data in the uploading delay of user 1. They switch to Case 2 to minimize uploading delay, which validates Theorem 2.

The second specific scenario includes four users and three subchannels. User 1, user 2, and user 3 occupy different subchannels. From user 1 to user 4, the link length is 75 m, 100 m, 100 m, and 50 m, respectively. The computational rate is set as $f^1_1 = f^1_2 = f^1_3 = f^1_4 = 0.2$ Gcycles/s, and $F = 24$ Gcycles/s. From user 1 to user 3, the input-data sizes of their tasks are set as $s_1 = 300$ kbits, $s_2 = 400$ kbits, and $s_3 = 300$ kbits. Fig. 12 shows the average overall delay of four users varying with the input-data size of user 4's task s_4. We can see that there are three phases in OCORA scheme. First, when s_4 is small, user 4 is paired with user 3 and works in Case 1. This is because s_3 is smaller than s_1 and s_2, and the co-channel interference between user 3 and user 4 is least. Second, since the growth of s_4 makes user 4 not finish task uploading earlier than user 3, user 4 is allocated to pair with user 2 and works in Case 1. This is because user 4 still can upload total input data within the uploading delay of user 2 such that the pairing of user 2 and user 4 can further optimize the average overall delay. Finally, due to large s_4, user 4 works in Case 2 and is paired with user 1. This is because that user 4 cannot finish the task uploading earlier than any other users and only works in Case 2. In Case 2, the NOMA user pair can achieve lower uploading delay, if user 4 obtains longer dedicated time duration for a task uploading. Hence, even though the co-channel interference is severe, user 4 and user 1 are allocated to be paired for minimum average overall delay. Therefore, it can be seen that the OCORA scheme achieves lower average overall delay than other schemes (User 1/2/3 always schemes) which always pair user 4 with user 1/2/3. Furthermore, because user 4 has best channel condition and much smaller input-data than other users, the OCORA scheme always pairs user 4 with other users, and other three users are allocated with different subchannels.

VI. Conclusion

This paper has investigated the computation offloading in multi-carrier NOMA-enabled MEC under the differentiated uploading delay. First, we have characterized the interaction between the differentiated uploading delay and co-channel interference for a pair of NOMA users. Second, the optimal power allocation algorithm has been proposed that coordinates the co-channel interference according to the difference of uploading delay between NOMA users. Finally, the OCORA and MCORA schemes have been proposed to reduce the average overall delay of users in finishing tasks by jointly scheduling the offloading decision and NOMA transmission. Simulation results have been provided to validate the necessity for considering the differentiated uploading delay, and to demonstrate the advantage of our proposed scheme. In future work, we will consider the partial offloading in multi-carrier NOMA-enabled MEC system, especially in ultra dense networks with multiple heterogeneous BSs.

Appendix

A. Proof of Theorem 1

We first consider the case that p_n decreases. Let $\bar{p}_n = p_n - \Delta p$ and $\Delta p \to 0^+$. The uploading rate of user m and user n is respectively expressed as

$$\begin{aligned}
&\bar{R}_{m,k}^{\mathrm{up},1}\\
&= R_{m,k}^{\mathrm{up},1} - \frac{\partial R_{m,k}^{\mathrm{up},1}}{\partial p_n}\Delta p_n \qquad (34)\\
&= R_{m,k}^{\mathrm{up},1} + \frac{p_m h_{m,k} h_{n,k}}{\sigma^2 + p_n h_{n,k} + p_m h_{m,k}} \frac{B\ln 2}{\sigma^2 + p_n h_{n,k}}\Delta p_n \qquad (35)\\
&= R_{m,k}^{\mathrm{up},1} + \Delta R_m \qquad (36)
\end{aligned}$$

$$\begin{aligned}
&\bar{R}_{n,k}^{\mathrm{up},2}\\
&= R_{n,k}^{\mathrm{up},2} - \frac{\partial R_{n,k}^{\mathrm{up},2}}{\partial p_n}\Delta p_n \qquad (37)\\
&= R_{n,k}^{\mathrm{up},2} - \frac{h_{n,k} B\ln 2}{\sigma^2 + p_n h_{n,k}}\Delta p_n \qquad (38)\\
&= R_{n,k}^{\mathrm{up},2} - \Delta R_n. \qquad (39)
\end{aligned}$$

The uploading delay of user n is expressed as

$$\begin{aligned}
&\bar{t}_{n,k}^{\mathrm{up},2}\\
&= \frac{s_n}{\bar{R}_{n,k}^{\mathrm{up},2}} = \frac{s_n}{R_{n,k}^{\mathrm{up},2}}\left(1 - \frac{h_{n,k}}{R_{n,k}^{\mathrm{up},2}} \frac{B\ln 2}{\sigma^2 + p_n h_{n,k}}\Delta p\right)^{-1} \qquad (40)\\
&= \frac{s_n}{R_{n,k}^{\mathrm{up},2}}\left(1 + \frac{h_{n,k}}{R_{n,k}^{\mathrm{up},2}} \frac{B\ln 2}{\sigma^2 + p_n h_{n,k}}\Delta p\right) \qquad (41)\\
&= \frac{s_n}{R_{n,k}^{\mathrm{up},2}} + \frac{s_n}{R_{n,k}^{\mathrm{up},2}} \frac{h_{n,k} B\ln 2}{R_{n,k}^{\mathrm{up},2}\left(\sigma^2 + p_n h_{n,k}\right)}\Delta p \qquad (42)\\
&= t_{n,k}^{\mathrm{up},2} + \Delta t_n \qquad (43)
\end{aligned}$$

where from (40) to (41), we use an equivalent infinitesimal replacement, i.e., $(1+ax)^b - 1 \sim abx$. In this case, the uploaded data of user m in $\bar{t}_{n,k}^{\mathrm{up},2}$ is expressed as

$$\begin{aligned}
&s_m^{\mathrm{up},1}\\
&= \bar{R}_{m,k}^{\mathrm{up},1} t_{n,k}^{\mathrm{up},2} = \left(R_{m,k}^{\mathrm{up},1} + \Delta R_m\right)\left(t_{n,k}^{\mathrm{up},2} + \Delta t_n\right) \qquad (44)\\
&= R_{m,k}^{\mathrm{up},1} t_{n,k}^{\mathrm{up},2} + R_{m,k}^{\mathrm{up},1}\Delta t_n + t_{n,k}^{\mathrm{up},2}\Delta R_m + \Delta R_m \Delta t_n \qquad (45)\\
&\approx R_{m,k}^{\mathrm{up},1} t_{n,k}^{\mathrm{up},2} + R_{m,k}^{\mathrm{up},1}\Delta t_n + t_{n,k}^{\mathrm{up},2}\Delta R_m \qquad (46)
\end{aligned}$$

where we omit $\Delta R_m \Delta t_n$, since it is the product of two infinitesimals.

Then, we consider the case that p_n keeps constant. In this case, the uploaded data of user m in $\bar{t}_{n,k}^{\mathrm{up},2}$ is expressed as $s_m^{\mathrm{up},2} = R_{m,k}^{\mathrm{up},1} t_{n,k}^{\mathrm{up},2} + R_{m,k}^{\mathrm{up},2}\Delta t_n$.

The difference of $s_m^{\mathrm{up},1}$ and $s_m^{\mathrm{up},2}$ can be expressed as (47), shown at the bottom of this page. Then, we can deduce the formula given by

$$\begin{aligned}
&\frac{\log_2\left[\frac{(\sigma^2+a)(\sigma^2+b)}{\sigma^2(\sigma^2+a+b)}\right]}{\log_2\left(\frac{\sigma^2+b}{\sigma^2}\right)} - \frac{a}{\sigma^2+a+b}\\
&\qquad > \frac{\log_2\left[\frac{(\sigma^2+a)(\sigma^2+b)}{\sigma^2(\sigma^2+a+b)}\right]}{\log_2\left(\frac{\sigma^2+b}{\sigma^2}\right)} - \frac{\sigma^2+a}{\sigma^2+a+b} = \frac{\log_2 \alpha}{\log_2 \beta} - \frac{\alpha}{\beta}
\end{aligned} \qquad (48)$$

$$s_m^{\mathrm{up},2} - s_m^{\mathrm{up},1} = \left\{\frac{\log_2\left[\frac{\left(\sigma^2+p_m h_{m,k}\right)\left(\sigma^2+p_n h_{n,k}\right)}{\sigma^2\left(\sigma^2+p_m h_{m,k}+p_n h_{n,k}\right)}\right]}{\log_2\left(\frac{\sigma^2+p_n h_{n,k}}{\sigma^2}\right)} - \frac{p_m h_{m,k}}{\sigma^2+p_m h_{m,k}+p_n h_{n,k}}\right\} \frac{s_n}{R_{n,k}^{\mathrm{up},2}} \frac{h_{n,k} B\ln 2}{p_n h_{n,k}+\sigma^2}\Delta p \qquad (47)$$

where $a = p_m h_{m,k}$ and $b = p_n h_{n,k}$. Since $\beta > \alpha > 1$ and the gradient of $\log_2(x), x > 1$ is less than 1, $\frac{\log_2 \alpha}{\log_2 \beta} \geqslant \frac{\alpha}{\beta}$. Therefore, $s_m^{\mathrm{up},2} - s_m^{\mathrm{up},1} > 0$. Now, we finish the proof of Theorem 1.

B. Proof of Theorem 3

The second derivative of $h(\tau_n)$ can be expressed as

$$h''(\tau_n) = \frac{2}{\tau_n^3} + \frac{s_m s_n A \ln^2 2}{B^2} \frac{\partial G}{\partial g} \frac{\partial g}{\partial \tau_n} \tag{49}$$

$$\frac{\partial G}{\partial g} = \frac{\ln(1+Ag) - 2Ag}{(1+Ag)^2 \ln^3(1+Ag)} \tag{50}$$

$$\frac{\partial g}{\partial \tau_n} = -\frac{s_n}{B} 2^{-\frac{1}{B} s_n \tau_n} \ln 2 \tag{51}$$

where $G = \frac{g}{(1+Ag)\ln^2(1+Ag)}$. It is obvious that $\frac{\partial G}{\partial g} < 0$ and $\frac{\partial g}{\partial \tau_n} < 0$. Therefore, $h''(\tau_n) > 0$ and $h'(\tau_n)$ is monotonically increasing.

From C7 to C10, we can determine that $\tau_n \in (0, \tau_n^{\max}], \tau_n^{\max} = \frac{B}{s_n} \log_2\left(1 + \frac{P_{\max} h_{n,k}}{\sigma^2}\right)$. When $\tau_n \to 0^+$, $\lim_{\tau_n \to 0^+} h'(\tau_n) = -\infty$. If $h'(\tau_n^{\max}) \leqslant 0$, $h(\tau_n)$ is monotonically decreasing. If $h'(\tau_n^{\max}) > 0$, $h(\tau_n)$ is first monotonically decreasing and then monotonically increasing. As a result, we prove that $h(\tau_n)$ is a unimodal function.

REFERENCES

[1] 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.

[2] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[3] N. Cheng *et al.*, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[4] J. Liu, M. Sheng, L. Liu, and J. Li, "Interference management in ultra-dense networks: Challenges and approaches," *IEEE Netw.*, vol. 31, no. 6, pp. 70–77, Nov. 2017.

[5] N. Javaid, A. Sher, H. Nasir, and N. Guizani, "Intelligence in IoT-based 5G networks: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 94–100, Oct. 2018.

[6] O. Kucur, G. K. Kurt, M. Z. Shakir, and I. S. Ansari, "Nonorthogonal multiple access for 5G and beyond," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–2, Jul. 2018.

[7] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[8] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[9] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018.

[10] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[11] 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. 2019.

[12] Z. Song, Y. Liu, and X. Sun, "Joint radio and computational resource allocation for NOMA-based mobile edge computing in heterogeneous networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2559–2562, Dec. 2018.

[13] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7244–7257, Nov. 2016.

[14] L. P. Qian, Y. Wu, H. Zhou, and X. Shen, "Dynamic cell association for non-orthogonal multiple-access V2S networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2342–2356, Oct. 2017.

[15] R. Ruby, S. Zhong, H. Yang, and K. Wu, "Enhanced uplink resource allocation in non-orthogonal multiple access systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1432–1444, Mar. 2018.

[16] J. Zhao, Y. Liu, K. K. Chai, A. Nallanathan, Y. Chen, and Z. Han, "Spectrum allocation and power control for non-orthogonal multiple access in HetNets," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5825–5837, Sep. 2017.

[17] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ESTI, France, ESTI White Paper 11, Sep. 2015. [Online]. Available: https://infotech.report/Resources/Whitepapers/f205849d-0109-4de3-8c47-be52f4e4fb27_etsi_wp11_mec_a_key_technology_towards_5g.pdf

[18] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[19] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[20] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[21] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji, and Z. Shi, "Optimal SIC ordering and computation resource allocation in MEC-aware NOMA NB-IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2806–2816, Apr. 2019.

[22] Y. Dai, M. Sheng, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Joint mode selection and resource allocation for D2D-enabled NOMA cellular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6721–6733, Jul. 2019.

[23] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

[24] D. Zhai and J. Du, "Spectrum efficient resource management for multi-carrier-based NOMA networks: A graph-based method," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 388–391, Jun. 2018.

[25] Z. Xiao, L. Zhu, J. Choi, P. Xia, and X.-G. Xia, "Joint power allocation and beamforming for non-orthogonal multiple access (NOMA) in 5G millimeter wave communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 2961–2974, May 2018.

[26] J. Zhao, Y. Liu, K. K. Chai, Y. Chen, and M. Elkashlan, "Joint subchannel and power allocation for NOMA enhanced D2D communications," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 5081–5094, Nov. 2017.

[27] 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.

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

[29] J. Park and S. Boyd, "General heuristics for nonconvex quadratically constrained quadratic programming," 2017, *arXiv:1703.07870*. [Online]. Available: https://arxiv.org/abs/1703.07870

[30] M. Grant, S. Boyd, and Y. Ye. (2008). *CVX: MATLAB Software for Disciplined Convex Programming*. [Online]. Available: https://cvxr.com/cvx/

[31] H. Sasaki and M. Toda, "Two-sided matching problems with externalities," *J. Econ. Theory*, vol. 70, no. 1, pp. 93–108, Jul. 1996.

[32] E. Bodine-Baron, C. Lee, A. Chong, B. Hassibi, and A. Wierman, "Peer effects and stability in matching markets," in *Proc. Int. Symp. Algorithmic Game Theory*. Berlin, Germany: Springer, 2011, pp. 117–129.

[33] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[34] K. Bando, "Many-to-one matching markets with externalities among firms," *J. Math. Econ.*, vol. 48, no. 1, pp. 14–20, Jan. 2012.

[35] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) Radio Transmission and Reception (Release 11)*, document TS 36.104 V11.12.0, 3GPP, Jul. 2015.

[36] M. Li, S. Yang, Z. Zhang, J. Ren, and G. Yu, "Joint subcarrier and power allocation for OFDMA based mobile edge computing system," in *Proc. IEEE PIMRC*, Oct. 2017, pp. 1–6.

[37] 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.

Min Sheng (Senior Member, IEEE) received the M.S. and Ph.D. degrees in communication and information systems from Xidian University, China, in 2000 and 2004, respectively. She is currently a Full Professor and the Director with the State Key Laboratory of Integrated Service Networks, Xidian University. Her general research interests include mobile ad hoc networks, 5G mobile communication systems, and satellite communications networks. She was awarded as a Distinguished Young Researcher from NSFC and Changjiang Scholar from Ministry of Education, China.

Yanpeng Dai (Student Member, IEEE) received the B.Eng. degree in communication engineering from Shandong Normal University, China, in 2014. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Service Networks, Xidian University, China. He was a Visiting Student with the University of Waterloo, Canada, from September 2017 to September 2018. His research interest includes resource management in heterogeneous wireless networks.

Junyu Liu (Member, IEEE) received the B.Eng. and Ph.D. degrees in communication and information systems from Xidian University, China, in 2007 and 2016, respectively. He is currently a Lecturer and a Post-Doctoral Researcher with the State Key Laboratory of Integrated Service Networks, Institute of Information and Science, Xidian University. His research interests include interference management and performance evaluation of wireless heterogeneous networks and ultradense wireless networks.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He worked as a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2018. He is currently a Professor with the School of Telecommunication, Xidian University, China. His research interests include performance analysis, MAC, opportunistic communication for vehicular networks, unmanned aerial vehicles, and artificial intelligence for wireless networks.

Xuemin (Sherman) Shen (Fellow, IEEE) received the B.Sc. degree from Dalian Maritime University, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, USA, in 1987 and 1990, respectively, all in electrical engineering. He is currently a University Professor and an Associate Chair for Graduate Studies with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He is currently an Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL and the Vice President of the Publications of IEEE Communications Society.

Qinghai Yang received the B.S. degree in communication engineering from the Shandong University of Technology, China in 1998, the M.S. degree in information and communication systems from Xidian University, China, in 2001, and the Ph.D. degree in communication engineering from Inha University, South Korea, in 2007, with the University-President Award. From 2007 to 2008, he was a Research Fellow with UWB-ITRC, South Korea. Since 2008, he has been a Full Professor with Xidian University. His current research interest lies in the fields of autonomic communication and content delivery networks.