Joint User Scheduling and Computing Resource Allocation Optimization in Asynchronous Mobile Edge Computing Networks

Yihan Cang, Ming Chen, Yijin Pan, Zhaohui Yang,
Ye Hu, Haijian Sun, and Mingzhe Chen

***Abstract*—In this paper, the problem of joint user scheduling and computing resource allocation in asynchronous mobile edge computing (MEC) networks is studied. In such networks, edge devices will offload their computational tasks to an MEC server, using the energy they harvest from this server. To get their tasks processed on time using the harvested energy, edge devices will strategically schedule their task offloading, and compete for the computational resource at the MEC server. Then, the MEC server will execute these tasks asynchronously based on the arrival of the tasks. This joint user scheduling, time and computation resource allocation problem is posed as an optimization framework whose goal is to find the optimal scheduling and allocation strategy that minimizes the energy consumption of these mobile computing tasks. To solve this mixed-integer non-linear programming problem, the general benders decomposition method is adopted which decomposes the original problem into a primal problem and a master problem. Specifically, the primal problem is related to computation resource and time slot allocation, of which the optimal closed-form solution is obtained. The master problem regarding discrete user scheduling variables is constructed by adding optimality cuts or feasibility cuts according to whether the primal problem is feasible, which is a standard mixed-integer linear programming problem and can be efficiently solved. By iteratively solving the primal problem and master problem, the optimal scheduling and resource allocation scheme is obtained. Simulation results demonstrate that the proposed asynchronous computing framework reduces 87.17% energy consumption compared with conventional synchronous computing counterpart.**

***Index Terms*—Mobile edge computing, asynchronous computing, user scheduling, wireless power transfer.**

The work of Ming Chen was supported by the key research and development plan projects of Jiangsu Province under grant BE2022316 & BE2022067-4, by Fundamental Research on Foreword Leading Technology of Jiangsu Province under grant BK20192002, and by the National Natural Science Foundation of China (NSFC) under grant 61960206005 & 61960206006. The work of Y. Pan was supported by Chongqing Natural Science Joint Fund Project under Grant No. CSTB2023NSCQ-LZX0121. (Corresponding author: Ming Chen).

Y. Cang, Ming Chen and Y. Pan are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mails: yhcang@seu.edu.cn, chenming@seu.edu.cn, panyj@seu.edu.cn). Ming Chen is also with the Purple Mountain Laboratories, Nanjing 211100, China.

Z. Yang is with College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and with International Joint Innovation Center, Zhejiang University, Haining 314400, China, and also with Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou 310027, China (e-mail: yang_zhaohui@zju.edu.cn).

Y. Hu is with the Department of Industrial and System Engineering, University of Miami, Coral Gables, FL, 33146, (e-mail: yehu@miami.edu).

H. Sun is with the School of Electrical and Computer Engineering, The University of Georgia, Athens, GA 30602 USA (e-mail: hsun@uga.edu).

M. Chen is with the Department of Electrical and Computer Engineering and Institute for Data Science and Computing, University of Miami, Coral Gables, FL, 33146, USA (e-mail: mingzhe.chen@miami.edu).

I. Introduction

Mobile edge computing (MEC) provides powerful computing ability to edge devices [1], [2]. Numerous works have investigated MEC systems from the perspective of resource allocation. In [3], computing offloading and service caching are jointly optimized in MEC-enabled smart grid to minimize system cost. Work [4] proposed a reverse auction-based offloading and resource allocation scheme in MEC. With the aid of machine learning, a multi-agent deep deterministic policy gradient (MADDPG) algorithm is designed to maximize energy efficiency in [5]. However, deploying computing resources at edge servers of a wireless network faces several challenges. First, due to limited energy of edge servers, they may not be able to provide sufficient computation resource according to devices' requirements [6], [7]. Second, executing all offloading tasks synchronously requires edge servers to wait the arrival of the task with maximum transmission delay which may not be efficient. Meanwhile, task scheduling sequence is nonnegligible in synchronous task offloading, which will also impact the network loads and task completion [8].

To address the first challenge, wireless power transfer (WPT) technology that exploits energy carried by radio frequency (RF) signals emerges [9]. Instead of using solar and wind sources, ambient RF signals can be a viable new source for energy scavenging. Harvesting energy from the environment provides perpetual energy supplies to wireless devices for tasks offloading [10]. Thus, WPT has been regarded as a promising paradigm for MEC scenarios. Combining WPT with MEC, the authors in [11] proposed a multi-user wireless-powered MEC framework aiming at minimizing the total energy consumption under latency constraints. In [12], considering binary computation offloading, the weighted sum computation rate of all wireless devices was maximized by optimizing computation mode selection and transmission time allocation. The work in [13] proposed a multiple intelligent reflecting surfaces (IRSs) assisted wireless powered MEC system, where the IRSs are deployed to assist both the downlink WPT from the access point (AP) to the wireless devices and the uplink computation offloading. However, the above works [11]–[17] assumed that all computational tasks offloaded by users will arrive at the server at the same time and then the server starts to process all tasks simultaneously, which is not efficient and even impractical due to users' dynamic computational task processing requests [18].

Currently, only a few existing works [18]–[23] optimized MEC networks under dynamic computation requests. The

work in [18] designed a Whittle index based offloading algorithm to maximize the long-term reward for asynchronous MEC networks where computational tasks arrive randomly. In [19], the authors studied the co-channel interference caused by asynchronous task uploading in NOMA MEC systems. The work in [20] investigated the energy efficient asynchronous task offloading for a MEC system where computational tasks with various latency requirements arrive at different time slots. Task scheduling problem for MEC systems with task interruptions and insertions was studied in [21]. However, the above works [18]–[21] that focused on the asynchronous task offloading neglected how the asynchronous task arrival affects the computation at the MEC server. The authors in [22] used a sequential computation method to solve the energy consumption minimization problem under asynchronous task arrivals. The work in [23] designed a computation strategy that only allows a task to be executed after the completion of the previous tasks. Yet, works in [22] and [23] are still constrained by their limited usage of the server computation capacity, and cannot act as resource efficient asynchronous task offloading solutions.

The sequential computation strategy [24] has shown to have the potential to improve the computation resource efficiency and task execution punctuality in an asynchronous MEC network. However, since the computation resource allocation at the server depends on the arrival of the offloaded tasks, the sequential scheduling of the tasks will inevitably affect system performance, which is a fact that has been wildly ignored [22], [25], [26].

The main contribution of this paper is a novel asynchronous MEC framework that jointly schedules tasks and allocates computation resource with optimized system energy efficiency. In brief, our key contributions include:

- We develop a novel framework to manage computation resource for the sequential computation in asynchronous MEC networks. In particular, we consider a MEC network in which the edge devices sequentially harvest energy for transmission, offload their computational tasks to a MEC server, and then compete for computation source at the server to get their tasks accomplished. To achieve the high energy efficient task execution, a policy needs to be designed for determining the optimal task scheduling sequence, time and computational resource allocation. We pose this joint scheduling and resource allocation problem in an optimization framework and seek to find the strategy which minimizes the energy consumption of the tasks.
- Then, a general benders decomposition (GBD) based algorithm is proposed to solve the formulated mixed-integer non-linear programming (MINLP) problem which is decomposed into a primal problem that allocates computation resource and time, and a master problem that schedules user tasks. By iteratively solving the primal problem and master problem, the optimal scheduling and resource allocation scheme is obtained.
- To show the effectiveness of the proposed algorithm, we prove that the optimal energy efficient scheduling and resource allocation scheme also optimizes the task punctuality. Our analytical results also show that the optimal allocation scheme for a given offloading task follows a specific pattern: the computation frequency allocated to each task remains constant initially, then gradually decreases before eventually reaching zero. Notably, all tasks experience a simultaneous decrease, the time of which is given in a closed form, in terms of their required central processing unit (CPU) cycles. Leveraging these identified properties, we introduce a computation resource allocation algorithm that offers a low-complexity solution.

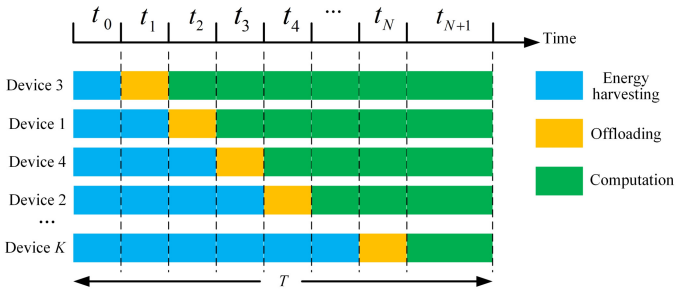


Fig. 1: An illustration about the flow chart of the ordered TDMA system with asynchronous computing.

Simulation results demonstrate that the proposed asynchronous computing framework reduces 87.87% energy consumption compared with conventional synchronous computing counterpart. Moreover, computational complexity of the proposed computation resource allocation algorithm is reduced by 100 times compared with conventional interior point method.

The rest of this paper is structured as follows. Section II elaborates system model and problem formulation. In Section III, we investigate the properties of asynchronous frequency allocation with given time allocation and user scheduling. The joint optimization of user scheduling, time allocation, and computation resource allocation is rendered in Section IV. Simulation results are presented in Section V. Finally, Section VI draws the conclusions.

II. System Model and Problem Formulation

Consider a MEC network consisting of one MEC server, and a set $\mathcal{K} = \{1, 2, \cdots, K\}$ of energy harvesting enabled edge devices. Within this network, each device $k \in \mathcal{K}$ needs to execute an A_k bits computational task, and will offload its computational task to the MEC server. As shown in Fig. 1, the devices need to first harvest energy from the server to enable such offloading. Then, using the time division multi-access (TDMA) technique, the devices need to schedule their offloading toward the MEC server. In other words, the computational tasks offloaded by devices will arrive at the MEC server asynchronously. To this end, the MEC server will process each device's computational task in an asynchronous manner. In particular, the server will process devices' computational tasks according to the time that it receives each of these computational tasks.

The server and edge devices must complete their computational tasks within a time period T which is divided

into $(K+2)$ time slots. The duration of each time slot $n \in \{0, \cdots, N+1\}$ is represented by Δt_n, with $N = K$. Each device uses one time slot to offload its computational task. Let $a_{k,n}$ be the index to indicate whether device k offloads its task to the server at time slot t_n. In particular, if device k uses time slot t_n to offload its computational task, we have $a_{k,n} = 1$; otherwise, $a_{k,n} = 0$. Since each device uses only one time slot and each time slot can only be allocated to one device, we have $\sum_{k=1}^{K} a_{k,n} = 1,\ \forall n \in \{1, \cdots, N\}$, and $\sum_{n=1}^{N} a_{k,n} = 1,\ \forall k \in \mathcal{K}$. Meanwhile, when $a_{k,n} = 1$, device k will harvest energy from time slot t_0 to t_{n-1}. Once task k arrives at the MEC server, i.e., at time slot slot t_{n+1}, the server will process this computational task.

The task computation process of the server and a device jointly completing a computational task k consists of three stages: 1) energy harvesting, 2) task offloading, and 3) remote computing. Next, we first introduce the process of the energy harvesting, task offloading, and remote computing stages. Then, the problem formulation is given.

A. Energy Harvesting Model

The path loss model is given by $\bar{h}_k = A\left(\frac{c}{4\pi f_c d_k}\right)^{\ell}$, where A represents antenna gain, c denotes the speed of light, f_c is the carrier frequency, ℓ denotes the path-loss factor, and d_k represents the distance between device k and the server [27]. The instant channel gain between device k and server denoted by h_k, follows an i.i.d. Rician distribution with line-of-sight (LoS) link gain equal to $\gamma\bar{h}_k$, where γ is Rician factor. If device k offloads its task at time slot t_n (i.e., $a_{k,n} = 1$), the harvested energy of device k is $E_k^H = \sum_{i=0}^{n-1} \Delta t_i h_k \eta P_0$, where η is the energy harvesting efficiency of each device, which is assumed to be equal for all devices [28]. P_0 denotes the transmit power of the server. Since each device has only a single time slot for task offloading (i.e., there exists only one $n \in \mathcal{N}$ such that $a_{k,n} = 1$ for a certain device k), the energy harvested by device k can be reformulated by $E_k^H = \sum_{n=1}^{K}\sum_{i=0}^{n-1} a_{k,n}\Delta t_i h_k \eta P_0\ (\forall k \in \mathcal{K})$.

B. Tasks Offloading Model

Based on the monomial offloading power model [20], [29], the transmit power of device k at its offloading time slot t_n is

$$p_{k,n} = \frac{\lambda(r_{k,n})^3}{h_k} = \frac{\lambda(A_k)^3}{h_k(\Delta t_n)^3}, \forall k \in \mathcal{K}, \forall n \in \{1, \cdots, N\}, \tag{1}$$

where $r_{k,n} = A_k/\Delta t_n$ is the transmission rate, $\lambda > 0$ is the energy coefficient related to the bandwidth and the noise power, and the order 3 is the monomial order associated with coding scheme. Since the transmit power of devices comes from harvested energy, we have $\sum_{n=1}^{N} a_{k,n}\Delta t_n p_{k,n} \le E_k^H$ $(\forall k \in \mathcal{K})$.

C. Computing Model

The MEC server is equipped with multiple CPUs such that the computational tasks offloaded from different devices can be executed in parallel [30]. Let I_k be the computation intensity of task k in terms of CPU cycles per bit. As shown in Fig. 2, to sufficiently utilize asynchronous computing, the computation resource of the server will be reallocated to the computational tasks offloaded from devices at each time slot from t_2 to t_{N+1}. Intuitively, the first uploading task can occupy the whole computation capacity of the server before the second offloading task arrives, while all the tasks compete for computation resource at time slot t_{N+1}. At an arbitrary time slot $m \in \{2, \cdots, N+1\}$, $f_{n,m}, \forall n \in \{1, \cdots, m-1\}$ is set to be the computation resource allocated to the task that arrives at the server at time slot t_n. Given these definitions, we have $\sum_{n=1}^{m-1} f_{n,m} \le F_{\max}\ (\forall m = 2, \cdots, N+1)$, where $F_{\max}$ represents the maximum computation capacity of the MEC server. To complete the task computation for each device k, we have $\sum_{n=1}^{N}\sum_{m=n+1}^{N+1} a_{k,n} f_{n,m}\Delta t_m \ge F_k$ $(\forall k \in \mathcal{K})$, where $F_k = A_k I_k$ represents the computation cycles of device k. Besides, the energy consumption of the MEC server for all tasks computation can be formulated by $E_{MEC} = \sum_{n=1}^{N}\sum_{m=n+1}^{N+1} \kappa f_{n,m}^3 \Delta t_m$, where κ denotes the energy coefficient of the MEC server.

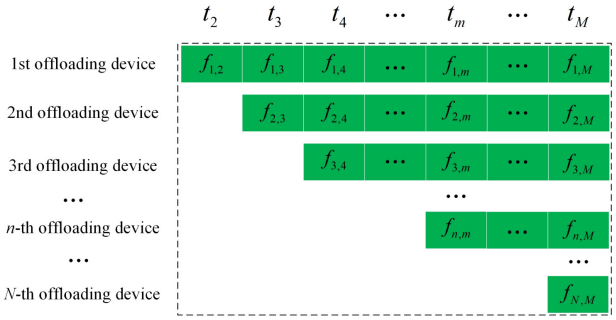


Fig. 2: Asynchronous computation resource allocation.

D. Problem Formulation

Our goal is to minimize the MEC server's energy consumption of completing the tasks offloaded by all devices, which is formulated as an optimization problem as

$$\min_{\Delta \boldsymbol{t}, \boldsymbol{A}, \boldsymbol{f}} \sum_{n=1}^{N}\sum_{m=n+1}^{N+1} \kappa f_{n,m}^3 \Delta t_m, \tag{2}$$

$$\text{s.t.} \sum_{n=1}^{N} a_{k,n}\frac{\lambda(A_k)^3}{h_k(\Delta t_n)^2} \le \sum_{n=1}^{N}\sum_{i=0}^{n-1} a_{k,n}\Delta t_i h_k \eta P_0, \forall k \in \mathcal{K}, \tag{2a}$$

$$\sum_{n=1}^{m-1} f_{n,m} \le F_{\max}, \quad \forall m = 2, \cdots, N+1, \tag{2b}$$

$$\sum_{n=1}^{N}\sum_{m=n+1}^{N+1} a_{k,n} f_{n,m}\Delta t_m \ge A_k I_k, \quad \forall k \in \mathcal{K}, \tag{2c}$$

$$\sum_{i=0}^{N+1} \Delta t_i \le T, \tag{2d}$$

$$\sum_{k=1}^{K} a_{k,n} = 1, \quad \forall n \in \{1, \cdots, N\}, \tag{2e}$$

$$\sum_{n=1}^{N} a_{k,n} = 1, \quad \forall k \in \mathcal{K}, \tag{2f}$$

$$a_{k,n} \in \{0,1\}, \quad \forall k \in \mathcal{K}, \forall n \in \{1,\cdots,N\}, \tag{2g}$$

where $\Delta \boldsymbol{t} = [\Delta t_0, \cdots, \Delta t_{N+1}]^T$, $\boldsymbol{f} = (f_{n,m})_{\forall n \in \{1,\cdots,N\}, m \in \{n+1,\cdots,N+1\}}$, and $\boldsymbol{A} = (a_{k,n})_{K \times N}$. In (2), (2a) is energy consumption causality constraint; (2b) represents a computational resource allocation constraint; (2c) ensures the completion of task computing; (2d) implies that the execution time of all devices should be less than T; (2e)-(2g) are user scheduling constraints. Since the discrete user scheduling variables $a_{k,n}$ and continuous resource allocation variables Δt_i, $f_{n,m}$ are highly coupled, problem (2) is a standard MINLP problem which is difficult to solve. To handle this issue, we first analyze the optimal computation resource allocation with given user scheduling and time allocation in Section III, based on which an efficient low-complexity computation frequency optimization algorithm is proposed. Finally, in Section IV, we propose a GBD-based algorithm to jointly optimize user scheduling and resource allocation so as to solve problem (2). [1]

III. Analysis and Algorithm of the Optimal Computation Resource Allocation

In this section, we first analyze the properties of the optimal computation resource allocation, and then a low-complexity computation resource allocation algorithm is accordingly proposed. For ease of notation, we use F_n $(\forall n \in \{1,\cdots,N\})$ to represent the computation cycles to complete the task that arrives at the server with order n. With given time slot allocation vector $\Delta \boldsymbol{t}$ and user scheduling matrix $\boldsymbol{A}$, problem (2) is simplified as follows:

$$\min_{\{f_{n,m}\}} \sum_{n=1}^{N} \sum_{m=n+1}^{N+1} \kappa f_{n,m}^3 \Delta t_m, \tag{3}$$

$$s.t. \sum_{n=1}^{m-1} f_{n,m} \le F_{\max}, \quad \forall m = 2,\cdots,N+1, \tag{3a}$$

$$\sum_{m=n+1}^{N+1} f_{n,m} \Delta t_m \ge F_n, \quad \forall n \in \{1,\cdots,N\}, \tag{3b}$$

$$f_{n,m} \ge 0, \forall n \in \{1,\cdots,N\}, \forall m = n+1,\cdots,N+1. \tag{3c}$$

Before solving problem (3), we provide the feasibility condition as follows.

Proposition 1. Problem (3) is feasible if and only if $F_{\max} \ge \max_{n \in \{1,\cdots,N\}} \frac{\sum_{i=n}^{N} F_i}{\sum_{i=n+1}^{N+1} \Delta t_{i+1}}$.

[1]For multi-server edge computing systems, new indicator variables can be introduced to denote the association between tasks and servers. Then the energy minimization problem can be formulated as a MINLP problem containing two kinds of binary optimization variables for task-server association and scheduling sequence, respectively. Despite being more complex, the problem can be solved efficiently using conventional MINLP methods such as convex relaxation and branch-and-bound, or latest approach using machine learning (see e.g., [27]). It is worth noting that with given task-server association, the proposed algorithm in this work is still applicable to scheduling and resource allocation optimization for each server. The detailed transmission protocol and algorithm procedure are left for future works.

Proof. Please refer to Appendix A. □

Denote $\{\alpha_m\}$, $\{\beta_n\}$, and $\{\gamma_{n,m}\}$ as the non-negative Lagrangian multipliers associated with the maximum frequency constraints (3a), task computation completion constraints (3b) and non-negative frequency constraints (3c), respectively. The optimal computation resource allocation is given by the following proposition.

Proposition 2. Given the optimal $\{\alpha_m^*\}$, $\{\beta_n^*\}$, the optimal solution of problem (3) is given by

$$f_{n,m}^* = \sqrt{\left[\frac{\beta_n^*}{3\kappa} - \frac{\alpha_m^*}{3\kappa \Delta t_m}\right]^+}, \forall n \in \mathcal{K}, \forall m \in \{2,\cdots,K+1\}. \tag{4}$$

Proof. Since (4) can be effectively obtained by solving Karush-Kuhn-Tucker (KKT) and Slater conditions, the proofs is omitted here. □

According to Propostion 2, we can use the sub-gradient method to obtain the optimal $\{\alpha_m^*\}$ and $\{\beta_n^*\}$ so as to acquire the optimal computation resource allocation. To further reduce the computational complexity and provide some design insights, the properties of the optimal solution of problem (3) are summarized in the following theorem.

Theorem 3. Denote $\digamma(i) = \sum_{n=1}^{i-2} \frac{F_n}{\sum_{m=n+1}^{K+1} \Delta t_m} + \frac{\sum_{n=i-1}^{K} F_n}{\sum_{m=i}^{K+1} \Delta t_m}$ $(2 \le i \le K+1)$. The optimal computation resource has the following properties:

1) The optimal solution of problem (3) satisfies $f_{n,n+1}^* = \cdots = f_{n,i}^* > \cdots > f_{n,j}^* = \cdots = f_{n,K+1}^* = 0$, $(n+1 \le i < j \le K+1)$, where t_i is referred as "transition point".
2) The optimal $\{\alpha_m^*\}$ satisfies $0 = \frac{\alpha_2^*}{\Delta t_2} = \cdots = \frac{\alpha_i^*}{\Delta t_i} < \frac{\alpha_{i+1}^*}{\Delta t_{i+1}} \cdots < \frac{\alpha_{K+1}^*}{\Delta t_{K+1}}$.
3) The transition point is t_i $(3 \le i \le K+1)$ if and only if $\digamma(i-1) \le F_{\max} < \digamma(i)$.

Proof. The proofs of 1), 2) and 3) are provided in Appendix B, C, D, respectively. □

According to property 1) in Theorem 3, the optimal frequency allocation scheme for a certain offloading task always follows a specific pattern: the frequency allocated to each device remains constant initially, then gradually decreases and eventually reaches zero. This property motivates us to deduce the condition $f_{n,i}^* > f_{n,i+1}^*$. The property 2) in Theorem 3 implies that the computation resource of the server is redundant at time slots from t_2 to t_i, while the maximum computation resource $F_{\max}$ is utilized at slots from t_{i+1} to t_{K+1}.

According to property 1) in Theorem 3, unless $f_{n,n+1}^* = \cdots = f_{n,K+1}^*$ $(\forall n \in \mathcal{K})$, there always exists a special time slot $t_\varkappa$ we called "*transition point*" such that $f_{n,n+1}^* = \cdots = f_{n,\varkappa-1}^* > f_{n,\varkappa}^* \ge \cdots \ge f_{n,K+1}^*$ $(3 \le \varkappa \le K+1)$. The transition point indicates the number of time slots that the computation resource remains the same. The computation resource decreases for all tasks at the transition point. The method to find out the transition point when it exists is given by property 3) in Theorem 3.

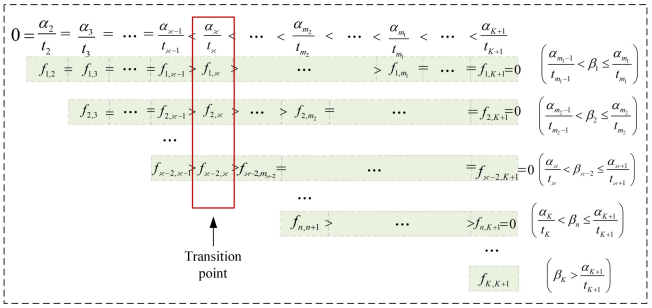


Fig. 3: Properties of the optimal computation resource allocation.

Property 3) in Theorem 3 also shows that the transition point is impacted by the computation ability of the server. We can directly determine the transition point $t_{\varkappa}$ utilizing property 3) in Theorem 3 without the need of solving problem (3). After determining the transition point, we have $\alpha_m^* = 0$ $(2 \leq m \leq \varkappa - 1)$ according to property 2) in Theorem 3.

Fig. 3 depicts an illustration of properties in the optimal computation resource allocation. As can be seen, before the transition point $t_{\varkappa}$, the optimal $\alpha_m^* = 0$ and $f_{n,m}^*$ keeps unchanged as $m = 2, \cdots, \varkappa - 1$. Based on Theorem 3, a low-complexity algorithm is proposed in Algorithm 1. First, we check the feasibility of problem (3) according to Proposition 1. Then, we determine the transition point $t_{\varkappa}$ based on property 3) in Theorem 3. If there is no transition points, which means that the computation resource of server is abundant, we can directly obtain the optimal solution $f_{n,n+1}^* = \cdots = f_{n,K+1}^* = \frac{F_n}{\sum_{m=n+1}^{K+1} \Delta t_m}$ for $n = 1, \cdots, K$; otherwise, we obtain the transition point $t_{\varkappa}$ and have $\alpha_m^* = 0$ for $m = 2, \cdots, \varkappa - 1$. Hence, we only need to find out the optimal α_m^* $(m = \varkappa, \cdots, K+1)$ and β_n^* $(n = 1, \cdots, K)$. Note that with given β_n^* $(n = 1, \cdots, K)$, we can obtain the optimal α_m^* by solving the following $(K - \varkappa + 2)$ equalities

$$G(m, \alpha_m) \triangleq \sum_{n=1}^{m-1} \sqrt{\left[\frac{\beta_n}{3\kappa} - \frac{\alpha_m}{3\kappa \Delta t_m}\right]^+} = F_{\max}, \quad (m = \varkappa, \cdots, K+1), \tag{5}$$

since the maximum frequency is utilized at time slots from $t_{\varkappa}$ to t_{K+1}. Since $G(m, \alpha_m)$ decreases with respect to α_m, the bisection method is adopted. It should be noticed that $G(m, \alpha_m)$ achieves the maximum value of $\sum_{n=1}^{m-1} \sqrt{\frac{\beta_n}{3\kappa}}$ when $\alpha_m = 0$ and the minimum value of 0 when $\frac{\alpha_m}{\Delta t_m} \geq \max_{n=1,\cdots,m-1} \beta_n$. Therefore, the upper bound of $\frac{\alpha_m}{\Delta t_m}$ is set as $\frac{\alpha_m^{ub}}{\Delta t_m} = \max_{n=1,\cdots,m-1} \beta_n$. For the lower bound, we set $\frac{\alpha_m^{lb}}{\Delta t_m} = \frac{\alpha_{m-1}}{\Delta t_{m-1}}$ according to property 2) in Theorem 3. After obtaining α_m for $m = \varkappa, \cdots, K+1$, β_n is updated by a sub-gradient method [31], where ϕ_n is the dynamically chosen step-size. Through repeating Steps 5 to 13 until the objective of (3) converges, we can obtain the optimal α_m^* for $m = 2, \cdots, K+1$ and β_n^* for $n = 1, \cdots, K$.

The complexity of Algorithm 1 is $\mathcal{O}\left(\frac{(K+2-\varkappa)}{\sqrt{\epsilon_1}} \log_2(\frac{1}{\epsilon_0})\right)$, where ϵ_0 denotes the accuracy of the bisection method and ϵ_1 is the accuracy of the objective of problem (3). Compared with the complexity of $\mathcal{O}\left(\left(K^2 + K\right)^{3.5}\right)$ by the interior point method, the complexity of the proposed algorithm is significantly reduced. Moreover, when $\varkappa$ is large, the complexity can be further reduced since more numbers of α_m^* are zeros.

Algorithm 1: Optimal Computation Frequency Allocation Algorithm

1 If $F_{\max} \geq \max_{n \in \{1,\cdots,K\}} \frac{\sum_{i=n}^{K} F_i}{\sum_{i=n+1}^{K+1} \Delta t_{i+1}}$, go to Step 2; otherwise, problem (3) is infeasible.
2 According to Theorem 3, if there is no transition point, the optimal solution is given by $f_{n,n+1}^* = \cdots = f_{n,K+1}^* = \frac{F_n}{\sum_{m=n+1}^{K+1} \Delta t_m}$ for $n = 1, \cdots, K$; otherwise, obtain transition point $t_{\varkappa}$ and let $\alpha_m^* = 0$ for $m = 2, \cdots, \varkappa - 1$.
3 Initialize $\beta_n = \left(\frac{F_n}{\sum_{m=n+1}^{K+1} \Delta t_m}\right)^2$ for $n = 1, \cdots, K$ and required precision ϵ_0.
4 **repeat**
5 **for** $m = \varkappa, \cdots, K+1$ **do**
6 Let $\frac{\alpha_m^{lb}}{\Delta t_m} = \frac{\alpha_{m-1}}{\Delta t_{m-1}}$ and $\frac{\alpha_m^{ub}}{\Delta t_m} = \max_{n=1,\cdots,m-1} \beta_n$.
7 **while** $\frac{\alpha_m^{ub}}{\Delta t_m} - \frac{\alpha_m^{lb}}{\Delta t_m} > \epsilon_0$ **do**
8 Set $\frac{\alpha_m}{\Delta t_m} \leftarrow (\frac{\alpha_m^{lb}}{\Delta t_m} + \frac{\alpha_m^{ub}}{\Delta t_m})/2$.
9 Calculate $f_{n,m} = \sqrt{\left[\frac{\beta_n}{3\kappa} - \frac{\alpha_m}{3\kappa \Delta t_m}\right]^+}$.
10 If $\sum_{n=1}^{m-1} f_{n,m} > F_{\max}$, let $\frac{\alpha_m^{lb}}{t_m} \leftarrow \frac{\alpha_m}{\Delta t_m}$; otherwise, let $\frac{\alpha_m^{ub}}{t_m} \leftarrow \frac{\alpha_m}{\Delta t_m}$.
11 **end**
12 **end**
13 Update $\beta_n \leftarrow \left[\beta_n + \phi_n \left(F_n - \sum_{m=n+1}^{K+1} f_{n,m} \Delta t_m\right)\right]^+$ for $n = 1, \cdots, K$.
14 **until** the objective of (3) converges;
15 Output the optimal $\{f_{n,m}^*\}$.

IV. Joint User Scheduling and Resource Allocation Algorithm

In this section, we employ the GBD method to solve problem (2). The core idea of GBD method is decomposing the original MINLP problem into a primal problem related to continuous variables and a master problem associated with integer variables, which are iteratively solved[2]. Specifically, for problem (2), the primal problem is a joint communication and computation resource optimization problem with fixed user scheduling. The master problem optimizes user scheduling by utilizing the optimal solutions and dual variables of the primal problem. Next, we describe the detailed procedures.

A. Primal Problem

With given user scheduling $\boldsymbol{A}$, problem (2) is reduced to the following optimization problem:

$$\min_{\Delta \boldsymbol{t}, \boldsymbol{f}} \quad \sum_{n=1}^{K} \sum_{m=n+1}^{K+1} \kappa f_{n,m}^3 \Delta t_m, \tag{6}$$

$$s.t. \quad \frac{\lambda (A_{\pi_n})^3}{h_{\pi_n} (\Delta t_n)^2} \leq \sum_{i=0}^{n-1} \Delta t_i h_{\pi_n} \eta P_0, \quad \forall n \in \mathcal{K}, \tag{6a}$$

[2]Interested readers may refer to [29], [32]–[34] for details.

$$\sum_{m=n+1}^{K+1} f_{n,m}\Delta t_m \geq A_{\pi_n} I_{\pi_n}, \quad \forall n \in \mathcal{K}, \tag{6b}$$

$$(2b), (2d), \tag{6c}$$

where π_n denotes the index of the n-th offloading device, i.e., we have $\pi_n = k$ if $a_{k,n} = 1$. Since the user scheduling scheme $\boldsymbol{A}$ is known, the value of $\pi_n, (\forall n \in \mathcal{K})$ can be deduced and substituted into problem (2). Since problem (6) is non-convex due to the constraints (6a), (6b) and the objective, we introduce $x_{n,m} = f_{n,m}\Delta t_m (\forall n = 1, \cdots, K, \forall m = n+1, \cdots, K+1)$ to represent computation amounts of the n-th offloading task at time slot m. Hence, problem (6) is equivalent to

$$\min_{\Delta \boldsymbol{t}, \boldsymbol{x}} \quad \sum_{n=1}^{K}\sum_{m=n+1}^{K+1} \kappa \frac{(x_{n,m})^3}{(\Delta t_m)^2}, \tag{7}$$

$$s.t. \quad \sum_{n=1}^{m-1} x_{n,m} \leq F_{\max}\Delta t_m, \forall m = 2, \cdots, K+1, \tag{7a}$$

$$\sum_{m=n+1}^{K+1} x_{n,m} \geq A_{\pi_n} I_{\pi_n}, \quad \forall n \in \mathcal{K}, \tag{7b}$$

$$(2d), (6a), \tag{7c}$$

where $\boldsymbol{x} = (x_{n,m})_{\forall n \in \mathcal{K}, m \in \{n+1, \cdots, K+1\}}$ is the collections of $x_{n,m}$. It can be proved that problem (7) is convex utilizing the tricks of perspective function [35]. To further provide useful insights and reduce computation complexity, we utilize the block coordinate decent (BCD) method to iteratively optimize time allocation and computation resource. Since the low-complexity computation resource allocation algorithm with given time allocation has been provided in Algorithm 1, next we propose time allocation algorithm with fixed computation resource allocation.

The Lagrangian function of problem (7) with respect to $\Delta \boldsymbol{t}$ is given by

$$\begin{aligned}\mathcal{L} = &\sum_{n=1}^{K}\sum_{m=n+1}^{K+1} \kappa \frac{(x_{n,m})^3}{(\Delta t_m)^2} + \sum_{n=1}^{K} \rho_n \left(\frac{\lambda (A_{\pi_n})^3}{h_{\pi_n}(\Delta t_n)^2} - \sum_{i=0}^{n-1} \Delta t_i h_{\pi_n} \eta P_0 \right) \\ &+ \xi \left(\sum_{i=0}^{K+1} \Delta t_i - T \right) + \sum_{m=2}^{K+1} \omega_m \left(\sum_{n=1}^{m-1} x_{n,m} - F_{\max}\Delta t_m \right),\end{aligned} \tag{8}$$

where ρ_n, ω_m and ξ are dual variables related to constraints (6a), (7a) and (2d), respectively. Taking the derivative with respect to $\Delta \boldsymbol{t}$, we have

$$\frac{\partial \mathcal{L}}{\partial \Delta t_0^*} = -\sum_{n=1}^{K} \rho_n^* h_{\pi_n} \eta P_0 + \xi^* = 0, \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \Delta t_1^*} = -\rho_1^* \frac{2\lambda (A_{\pi_1})^3}{h_{\pi_1}(\Delta t_1^*)^3} - \sum_{n=2}^{K} \rho_n^* h_{\pi_n} \eta P_0 + \xi^* = 0, \tag{10}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Delta t_i^*} = &-\sum_{n=1}^{i-1} 2\kappa \frac{(x_{n,i})^3}{(\Delta t_i^*)^3} - \rho_i^* \frac{2\lambda (A_{\pi_i})^3}{h_{\pi_i}(\Delta t_i^*)^3} - \sum_{n=i+1}^{K} \rho_n^* h_{\pi_n} \eta P_0 \\ &+ \xi^* - \omega_i^* F_{\max} = 0, \quad (2 \leq i \leq K-1),\end{aligned} \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial \Delta t_K^*} = -\sum_{n=1}^{K-1} 2\kappa \frac{(x_{n,K})^3}{(\Delta t_K^*)^3} - \rho_K^* \frac{2\lambda (A_{\pi_K})^3}{h_{\pi_K}(\Delta t_K^*)^3} + \xi^* - \omega_K^* F_{\max} = 0, \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial \Delta t_{K+1}^*} = -\sum_{n=1}^{K} 2\kappa \frac{(x_{n,K+1})^3}{(\Delta t_{K+1}^*)^3} + \xi^* - \omega_{K+1}^* F_{\max} = 0. \tag{13}$$

Through solving above equations, the optimal Δt_i^* $(\forall i)$ is obtained in the following proposition.

Proposition 4. The optimal $\Delta \boldsymbol{t}^*$ is given by

$$\Delta t_0^* = T - \sum_{i=1}^{K+1} \Delta t_i^*, \tag{14}$$

$$\Delta t_i^* = \sqrt[3]{\frac{2\kappa h_{\pi_i} \sum_{n=1}^{i-1} (x_{n,i})^3 + 2\rho_i^* \lambda (A_{\pi_i})^3}{\xi^* h_{\pi_i} - \sum_{n=i+1}^{K} \rho_n^* (h_{\pi_n})^2 \eta P_0 - \omega_i^* h_{\pi_i} F_{\max}}}, \quad (\forall i \in \{2, \cdots, K-1\}), \tag{15}$$

$$\Delta t_K^* = \sqrt[3]{\frac{2\kappa h_{\pi_K} \sum_{n=1}^{K-1} (x_{n,K})^3 + 2\rho_K^* \lambda (A_{\pi_K})^3}{\xi^* h_{\pi_K} - \omega_K^* h_{\pi_K} F_{\max}}}, \tag{16}$$

$$\Delta t_{K+1}^* = \sqrt[3]{\frac{2\kappa \sum_{n=1}^{K} (x_{n,K+1})^3}{\xi^* - \omega_{K+1}^* F_{\max}}}, \tag{17}$$

and Δt_1^* is the null point of $\Psi(x)$, where $\Psi(x) = \left(T - \sum_{i=1}^{K+1} \Delta t_i\right) x^2 (h_{\pi_1})^2 \eta P_0 - \lambda (A_{\pi_1})^3$, $x \in \left(0, \frac{2}{3}\left(T - \sum_{i=2}^{K+1} \Delta t_i\right)\right]$.

Proof. The proof of Proposition 4 is provided in Appendix H. □

Through iteratively optimizing time allocation and computation resource allocation, we can obtain the optimal solution of primal problem (7). However, if problem (7) is infeasible, we formulate the corresponding ℓ_1-minimization problem as follows:

$$\min_{\Delta \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\zeta} > 0, \boldsymbol{\iota} > 0} \sum_{k=1}^{K} (\zeta_k + \iota_k), \tag{18}$$

$$s.t. \quad \sum_{n=1}^{K} a_{k,n} \frac{\lambda (A_k)^3}{h_k (\Delta t_n)^2} \leq \zeta_k + \sum_{n=1}^{K}\sum_{i=0}^{n-1} a_{k,n} \Delta t_i h_k \eta P_0, \quad \forall k \in \mathcal{K}, \tag{18a}$$

$$\iota_k + \sum_{n=1}^{K}\sum_{m=n+1}^{K+1} a_{k,n} x_{n,m} \geq A_k I_k, \forall k \in \mathcal{K}, \tag{18b}$$

$$(2d), (7a). \tag{18c}$$

Since problem (18) is convex and always feasible, we can use the interior point method to obtain the optimal solution and corresponding dual variables.

Furthermore, we can observe that the solution of primal problem always provides a performance upper bound for problem (2) since user scheduling is fixed. Then the upper bound is updated as $UB^{(j)} \leftarrow \min\{UB^{(j-1)}, f^{(j)}\}$, where $f^{(j)}$ denotes the objective value of primal problem (6). As can be seen, the upper bound is always non-increasing as iteration proceeds. Subsequently, we construct master problem using the solutions and dual variables of primal problem (7) and feasibility problem (18).

B. Master Problem

At each iteration, optimality cut or feasibility cut are added to master problem depending on whether the primal problem is feasible. Denote $\mathcal{J}_1$ and $\mathcal{J}_2$ as the set of iteration indexes indicating the primal problem is feasible and infeasible, respectively. Specifically, the optimality cut for each $j \in \mathcal{J}_1$ of feasible iterations is defined as

$$\theta(\boldsymbol{A}, \boldsymbol{\rho}^{(j)}, \boldsymbol{\beta}^{(j)}) = \sum_{n=1}^{K} \sum_{m=n+1}^{K+1} \kappa \frac{\left(x_{n,m}^{(j)}\right)^3}{\left(\Delta t_m^{(j)}\right)^2} + \sum_{k=1}^{K} \rho_k^{(j)} \left(\sum_{n=1}^{K} a_{k,n} \frac{\lambda(A_k)^3}{h_k \left(\Delta t_n^{(j)}\right)^2} - \sum_{n=1}^{K} \sum_{i=0}^{n-1} a_{k,n} \Delta t_i^{(j)} h_k \eta P_0 \right) + \sum_{k=1}^{K} \beta_k^{(j)} \left(A_k I_k - \sum_{n=1}^{K} \sum_{m=n+1}^{K+1} a_{k,n} x_{n,m}^{(j)} \right), \tag{19}$$

where $\rho_k^{(j)}$ and $\beta_k^{(j)}$ represent the dual variables related to primal problem at the j-th iteration, $x_{n,m}^{(j)}$ and $\Delta t_m^{(j)}$ denote the solution of primal problem at the j-th iteration. The terms irrelavant to $\boldsymbol{A}$ are omitted based on complementary slackness theorem [31]. Similarly, the feasibility cut for each $j \in \mathcal{J}_2$ of infeasible iterations is defined as

$$\hat{\theta}(\boldsymbol{A}, \hat{\boldsymbol{\rho}}^{(j)}, \hat{\boldsymbol{\beta}}^{(j)}) = \sum_{k=1}^{K} \hat{\beta}_k^{(j)} \left(A_k I_k - \sum_{n=1}^{K} \sum_{m=n+1}^{K+1} a_{k,n} \hat{x}_{n,m}^{(j)} \right) + \sum_{k=1}^{K} \hat{\rho}_k^{(j)} \left(\sum_{n=1}^{K} a_{k,n} \frac{\lambda(A_k)^3}{h_k \left(\Delta \hat{t}_n^{(j)}\right)^2} - \sum_{n=1}^{K} \sum_{i=0}^{n-1} a_{k,n} \Delta \hat{t}_i^{(j)} h_k \eta P_0 \right), \tag{20}$$

where $\hat{\rho}_k^{(j)}$ and $\hat{\beta}_k^{(j)}$ represent the dual variables related to feasibility problem at the j-th iteration, $\hat{x}_{n,m}^{(j)}$ and $\Delta \hat{t}_m^{(j)}$ denote the solution of feasibility problem at the j-th iteration. Therefore, master problem is formulated as

$$\min_{\boldsymbol{A}, \psi} \quad \psi, \tag{21}$$

$$s.t. \quad \theta(\boldsymbol{A}, \boldsymbol{\rho}^{(j)}, \boldsymbol{\beta}^{(j)}) \leq \psi, \quad \forall j \in \mathcal{J}_1, \tag{21a}$$

$$\hat{\theta}(\boldsymbol{A}, \hat{\boldsymbol{\rho}}^{(j)}, \hat{\boldsymbol{\beta}}^{(j)}) \leq 0, \quad \forall j \in \mathcal{J}_2, \tag{21b}$$

$$(2e) - (6g). \tag{21c}$$

In particular, (21a) and (21b) denote the set of hyperplanes spanned by the optimality cut and feasibility cut from the first to the j-th iteration, respectively. The two different types of cuts are exploited to reduce the search region for the global optimal solution [36]. Master problem (21) is a standard mixed-integer linear programming (MILP) problem, which can be solved by numerical solvers such as Gurobi [37] and Mosek [38]. Since master problem is the relaxing problem of MINLP problem (2), solving master problem provides a performance lower bound for problem (2). The lower bound is given by $LB^{(j)} \leftarrow \psi$. Since at each iteration, an additional cut (optimality cut or feasibility cut) is added to master problem which narrows the feasible zone, the lower bound is always non-decreasing. As a consequence, the performance upper bound obtained by primal problem and the performance lower bound obtained by the master problem are non-increasing and non-decreasing w.r.t. the iteration index, respectively. As a result, the performance upper bound and the performance lower bound go to converge [29]. Therefore, through iteratively solving primal problem and master problem, we can obtain the optimal solution when the upper bound and lower bound are sufficiently close [33], [36]. The detailed algorithm is summarized in Algorithm 2.

Algorithm 2: Joint User Scheduling and Resource Allocation Algorithm

Initialize arbitrary feasible user scheduling $\boldsymbol{A}^{(j)}$, and set $j = 1$, $UB = +\infty$, $LB = -\infty$, $\mathcal{J}_1 = \mathcal{J}_2 = \emptyset$;
repeat
  if problem (7) is feasible **then**
    repeat
      Obtain the optimal computation resource $\boldsymbol{f}^{(j)}$ and dual variable $\beta_k^{(j)}$ according to Algorithm 1;
      Obtain the optimal time allcation $\Delta \boldsymbol{t}^{(j)}$ and dual variable $\rho_k^{(j)}$;
    until the objective of (7) converges;
    Update UB and $\mathcal{J}_1$;
  else
    Solve feasibility problem (18) and update $\mathcal{J}_2$;
    Obtain the corresponding optimal solution $\hat{\boldsymbol{x}}^{(j)}$ and $\Delta \hat{\boldsymbol{t}}^{(j)}$ as well as dual variables $\hat{\rho}_k^{(j)}$ and $\hat{\beta}_K^{(j)}$;
  end
  Solve master problem (21) by adding optimality cuts (19) and feasibility cuts (20);
  Set $j \leftarrow j + 1$;
  Update $\boldsymbol{A}^{(j)}$ and LB;
until UB and LB are sufficiently close;
Output the optimal $\boldsymbol{f}^*$, $\Delta \boldsymbol{t}^*$ and $\boldsymbol{A}^*$.

C. Complexity Analysis

The complexity of solving problem (2) by Algorithm 2 lies in solving the primal problem, feasibility problem, and master problem at each iteration. For primal problem, where we iteratively update time allocation variables and frequency variables. The frequency optimization method is given in Algorithm 1, whose complexity is $\mathcal{O}\left(\frac{K+2-\varkappa}{\sqrt{\epsilon_1}} \log_2(1/\epsilon_0)\right)$ as analyzed in Section III. The time allocation optimization is according to Proposition 4, whose complexity is estimated as $\mathcal{O}\left(K \log_2(T)\right)$. Therefore, the total complexity of solving primal problem is $\mathcal{O}\left(\frac{K+2-\varkappa}{\sqrt{\epsilon_1}} \log_2(1/\epsilon_0) K \log_2(T) L_1\right)$, where L_1 denotes the iteration number in the primal problems. For the feasibility problem, the complexity is given by $\mathcal{O}\left(\left(\frac{(K+1)K}{2} + 3K + 1\right)^{3.5}\right)$ by the interior point method. For the master problem, the computational complexity is $\mathcal{O}\left(2^K\right)$ by the Branch and Bound (BnB) method [39].

V. Simulations

In this section, we perform simulations to validate the proposed scheme and algorithm. There are $K = 10$ devices around the server. The task size A_k and computation intensity

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3358237

obey uniform distribution on $[10, 50]$ Kbits and $[500, 1500]$ cycles/bit, respectively. The transmit power of BS is $P_0 = 3$ W. The energy coefficient of the MEC server and energy conversion factor of devices are set as $\kappa = 10^{-26}$ and $\eta = 0.51$. We set the energy constant of transmission $\lambda = 10^{-25}$. Furthermore, the maximum computation resource is $F_{\max} = 1$ GHz and the allowable delay is $T = 1$ second. In channel model, we set antenna gain $A = 3$, carrier frequency $f_c = 915$ MHz, path-loss factor $\ell = 3$, speed of light $c = 3 \times 10^8$ m/s, and the Rician factor is $\gamma = 0.3$. The following benchmarking schemes are provided:

- *JSORA [26]*: The joint sensing-and-offloading resource allocation algorithm, where the allocated frequency for each device keeps unchanged during its computation duration, i.e., $f_{n,n+1} = \cdots = f_{n,K+1}$ $(\forall n \in \mathcal{K})$.
- *JCCRM-Sync [11]–[17]*: The joint communication and computation resource management algorithm, where the computing of server will not begin until all tasks are received, which is adopted by most of the literature.
- *Random scheduling scheme [22], [25]*: We randomly set the user scheduling for offloading.
- *Exhaustive search*: We randomly choose multiple initial points for Algorithm 2 and select the smallest result as output. The results of exhaustive search method can be regarded as global optimal solutions.

Besides, all accuracies used in the simulations are set as 10^{-5} for fairness.

Fig. 4 demonstrates the energy consumption performance comparisons between different schemes under different numbers of devices. We can observe that the energy consumption of the proposed scheme as well as benchmark schemes increases with the number of devices getting large. This is because that devices have to compete for fixed communication and computation resource. As the number of devices increases, the average transmission time and computation time of each device get small, thus average computation resource becomes large. Therefore, larger energy consumption of server is required in order to finish devices' tasks within the required delay. Moreover, as can be seen in Fig. 4, the gap between the proposed algorithm and exhaustive search scheme is small. This indicates that the proposed algorithm achieves close-to-optimal solutions. Compared with JSORA scheme, random scheduling scheme and JCCRM-Sync scheme, the proposed scheme achieves 30.88%, 19.51%, 87.87% energy reductions, respectively. This can be explained by that the proposed algorithm can take full advantage of the flexibility of asynchronous computing and user scheduling. Particularly, compared with the proposed scheme, JCCRM-Sync scheme wastes the idle computation resource from time slots t_2 to t_K. Similarly, JSORA scheme can not make full use of computation resource from time slots t_2 to t_K. Hence, its performance is better than JCCRM-Sync but worse than the proposed scheme. Additionally, random scheduling, as most of the existing literature does, can not utilize the heterogeneity of tasks size and computation intensity well in MEC networks.

In Fig. 5, we depict the energy consumption curves of different schemes versus the maximum allowable delay. As

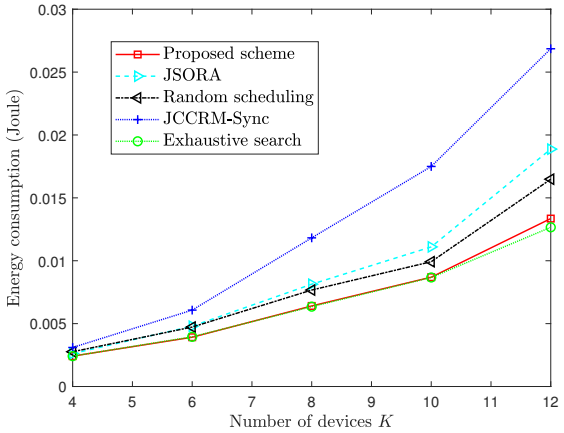


Fig. 4: Energy consumption versus number of devices.

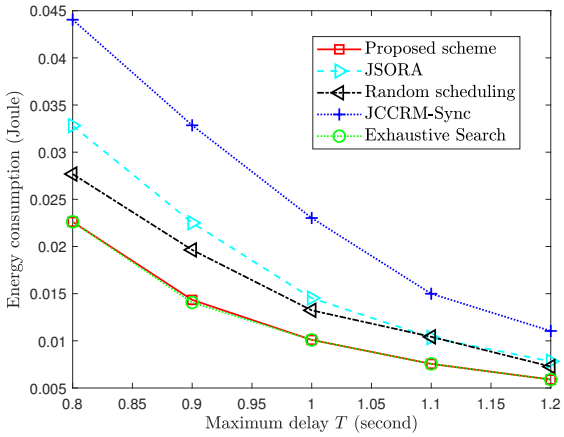
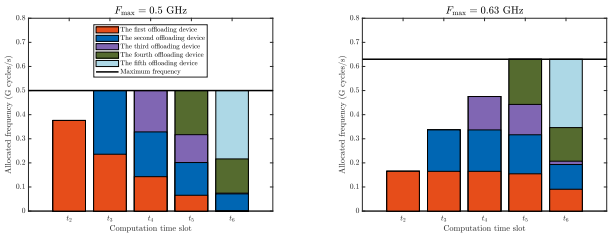


Fig. 5: Energy consumption versus maximum delay.

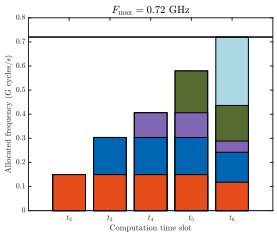
can be seen, the energy consumption of all schemes decreases as the maximum delay becomes large. This is because as delay gets large, the server has more time to finish tasks. Thus, the fewer computation resource is allowable. Hence, energy consumption can be reduced. From Fig. 5, it can be verified that the proposed algorithm outperforms benchmarking schemes in terms of energy consumption in the considered region of delay, especially in resource-scarce scenarios. This phenomenon can be observed in Fig. 4 and Fig. 5 that the difference in energy consumption between the proposed algorithm and benchmark schemes gets small when resource is abundant. This is because the flaws of benchmark schemes compared with the proposed algorithm can be appropriately compensated by utilizing additional sources. Furthermore, it should be noticed that JSORA scheme is equivalent to the proposed scheme when computation resource is abundant according to Theorem 3.

Fig. 6 illustrates a specific case of the allocated frequency of each device at each computation slot under different maximum computation frequencies when $K = 5$. It can be seen that as the maximum frequency $F_{\max}$ becomes large, the transition point is gradually postponed, and finally no transition point exists when computation resource is sufficiently large which is in accordance with Theorem 3. Specifically, for each subfigure,

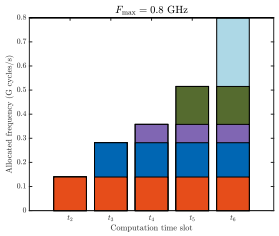


(a) Transition point: t_3

(b) Transition point: t_5



(c) Transition point: t_6



(d) No transition point

Fig. 6: Illustrations of different transition points under different maximum frequencies.

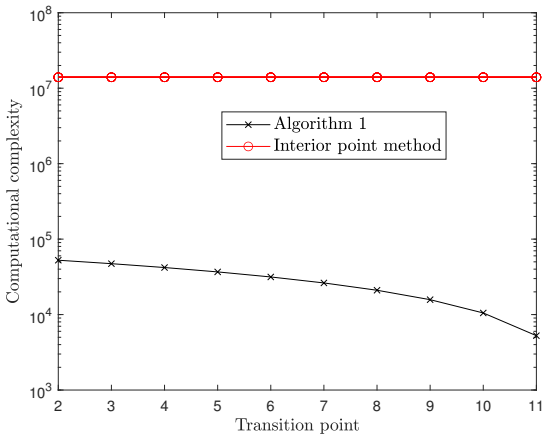


Fig. 7: Comparisons of computational complexity with $K = 10$.

we can find that before the transition point, the allocated frequency for each device being computed remains unchanged and the maximum frequency constraints do not work. From the transition point to the end, the allocated frequency for each device becomes small and the maximum frequency of the server is used. This verifies Theorem 3.

Fig. 7 depicts the computational complexity comparisons between the proposed Algorithm 1 and the interior point method under different transition points. As can be seen, the computational complexity of Algorithm 1 is significantly reduced compared with the interior point method, by more than 100 times on average. As the transition point becomes larger, the complexity further decreases. For example, when the transition point $\varkappa = 11$, the complexity of Algorithm 1 is reduced by 1000 times. This is because the proposed computation resource allocation algorithm fully utilizes the properties in Theorem 3 to reduce algorithm complexity, especially when the computation resource of the server is abundant.

To test the compatibility of the proposed algorithm under different task scale differences, the energy consumption versus minimum task size $A_{\min}$ is shown in Fig. 8, where task size obeys uniform distribution on $[A_{\min}, A_{\max}]$ with fixed mean value $\frac{A_{\min}+A_{\max}}{2} = 30$ Kbits. With a large $A_{\min}$, the task scale difference is small. In Fig. 8, the proposed scheme and exhaustive search scheme achieve nearly the same performance, and outperform other schemes. One can observe that the energy consumption increases as task scale difference gets large. This can be explained by that the resources have to be tilted towards the devices with large task sizes, thus resulting in more energy consumption.

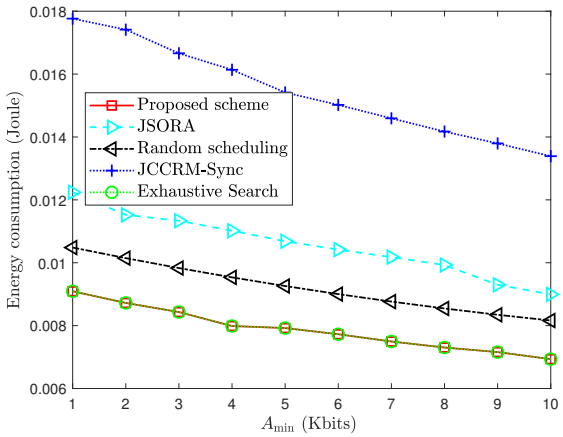


Fig. 8: Energy consumption versus minimum task size.

VI. Conclusion

In this paper, we have investigated a joint user scheduling and resource optimization framework for MEC networks with asynchronous computing. An optimization problem of joint user scheduling, communication and computation resource management has been solved aiming to minimize the energy consumption of server under the delay constraint. Simulations verified that the proposed algorithm yields significant performance gains compared with benchmark schemes. This work establishes a new principle of asynchronous computing and verifies the superiority over its synchronous counterpart. For future works, we will generalize the proposed asynchronous computing framework to heterogeneous task deadlines scenarios so as to further activate its potential. As another direction, the extension to online algorithm design and accommodate new coming devices deserve further investigation.

Appendix A
Proof of Proposition 1

The feasibility problem of (3) can be constructed as

$$\min_{f_{n,m}} \quad \max_{m\in\{2,\cdots,K+1\}} \sum_{n=1}^{m-1} f_{n,m}, \tag{A.1}$$

$$s.t. \quad \sum_{m=n+1}^{K+1} f_{n,m}\Delta t_m \geq F_n, \quad \forall n \in \mathcal{K}, \tag{A.1a}$$

$$f_{n,m} \geq 0, \quad \forall n \in \mathcal{K}, \forall m = n+1, \cdots, K+1. \tag{A.1b}$$

If the optimal objective of problem (A.1) is less than or equal to $F_{\max}$, problem (3) is feasible; otherwise, it is infeasible. Subsequently, we analyze the optimal solution of problem (A.1). First, when $K=1$, i.e., there exists only one task, the optimal objective of problem (A.1) is $\frac{F_K}{\Delta t_{K+1}}$. When $K=2$, we consider two cases: 1) If $\frac{F_1}{\Delta t_2} \leq \frac{F_2}{\Delta t_3}$, this indicates that the optimal scheme is computing task 2 after task 1 is finished. Therefore, the optimal solution is $f_{1,2}=\frac{F_1}{\Delta t_2}$ and $f_{2,3}=\frac{F_2}{\Delta t_3}$. Hence, the optimal objective is $\frac{F_2}{\Delta t_3}$. 2) If $\frac{F_1}{\Delta t_2} > \frac{F_2}{\Delta t_3}$, this implies that part of task 1 can be processed in parallel with task 2. Hence, the optimal solution is given by $f_{1,2}=f_{1,3}+f_{2,3}=\frac{F_1+F_2}{\Delta t_2+\Delta t_3}$. Since $\frac{F_1+F_2}{\Delta t_2+\Delta t_3} > \frac{F_2}{\Delta t_3}$, the optimal objective is $\frac{F_1+F_2}{\Delta t_2+\Delta t_3}$. In conclusion, when $K=2$, the optimal objective is $\max\{\frac{F_1+F_2}{\Delta t_2+\Delta t_3}, \frac{F_2}{\Delta t_3}\}$. Similarly, by recursion, we can deduce that when there exist K TD, the optimal solution is $\max\{\frac{F_1+\cdots+F_K}{\Delta t_2+\cdots+\Delta t_{K+1}}, \frac{F_2+\cdots+F_K}{\Delta t_3+\cdots+\Delta t_{K+1}}, \cdots, \frac{F_K}{\Delta t_{K+1}}\}$. That completes the proof. ■

APPENDIX B
PROOF OF PROPERTY 1) IN THEOREM 3

Before that, we give the following two corollaries to facilitate the proof.

Corollary 5. [Row property] The optimal computation resource of each task is non-increasing during its computation period, i.e., $f^*_{n,n+1} \geq f^*_{n,n+2} \geq \cdots \geq f^*_{n,K+1}$ $(\forall n \in \mathcal{K})$.

Proof. Please refer to Appendix E. □

Corollary 6. [Column property] Denote the sum computation cycles of the n-th offloading device in time slots t_m and t_{m+1} by F_n for $m=3,\cdots,K$ and $n=1,\cdots,m-1$. If $F_n>0$ holds for all $n=1,\cdots,m-1$, the optimal frequency shifts $\delta^*_n \triangleq f^*_{n,m}-f^*_{n,m+1}$ $(n=1,\cdots,m-1)$ are either all zeros or all positive, i.e., have the coincident zero or positive characteristics.

Proof. Please refer to Appendix F. □

First, applying the KKT conditions gives

$$3\kappa(f^*_{n,m})^2\Delta t_m+\alpha_m-\beta_n\Delta t_m-\gamma_{n,m}=0, \tag{B.1}$$

$$\alpha_m\left(\sum_{n=1}^{m-1} f^*_{n,m}-F_{\max}\right)=0, \tag{B.2}$$

$$\beta_n\left(F_n-\sum_{m=n+1}^{K+1} f^*_{n,m}\Delta t_m\right)=0, \tag{B.3}$$

$$\gamma_{n,m}f^*_{n,m}=0, \quad \forall n\in\mathcal{K}, \tag{B.4}$$

$$\alpha_m\geq 0, \beta_n\geq 0, \gamma_{n,m}\geq 0. \tag{B.5}$$

Based on (B.1), we obtain that

$$f^*_{n,m}=\sqrt{\frac{\beta_n}{3\kappa}+\frac{\gamma_{n,m}-\alpha_m}{3\kappa\Delta t_m}}. \tag{B.6}$$

In case of $f_{n,m}>0$ $(\forall n\in\mathcal{K},\forall m=n+1,\cdots,K+1)$, we have $\gamma_{n,m}=0$ according to (B.4). Furthermore, $\beta_n>0$ is derived from (B.6). According to Corollary 5, the optimal solution satisfies $f^*_{n,m}\geq f^*_{n,m+1}$. Thus, we have $\frac{\alpha_m}{\Delta t_m}\leq\frac{\alpha_{m+1}}{\Delta t_{m+1}}$. Assume that there exists a certain $i\in\{3,\cdots,K\}$ such that $f^*_{n,i}=f^*_{n,i+1}$. We have $\frac{\alpha_i}{\Delta t_i}=\frac{\alpha_{i+1}}{\Delta t_{i+1}}$. If $\frac{\alpha_i}{\Delta t_i}=\frac{\alpha_{i+1}}{\Delta t_{i+1}}>0$, i.e., $\alpha_i>0$ and $\alpha_{i+1}>0$, we should have $\sum_{n=1}^{i-1}f_{n,i}=\sum_{n=1}^{i}f_{n,i+1}=F_{\max}$ according to (B.2). Furthermore, due to $f_{n,m}>0$ $(\forall n\in\mathcal{K},\forall m\in=n+1,\cdots,K+1)$, the computation cycles $f_{n,i}\Delta t_i+f_{n,i+1}\Delta t_{i+1}>0$ $(\forall n\in 1,\cdots,i-1)$. According to Corollary 6, we have $f_{n,i}=f_{n,i+1}$ $(n=1,\cdots,i-1)$. Hence, it can be derived that $f_{i,i+1}=0$ which contradicts that $f_{i,i+1}$ is positive. Therefore, we have $\alpha_i=\alpha_{i+1}=0$, i.e., $\frac{\alpha_i}{\Delta t_i}=\frac{\alpha_{i+1}}{\Delta t_{i+1}}=0$. Since $\frac{\alpha_2}{\Delta t_2}\leq\cdots\leq\frac{\alpha_i}{\Delta t_i}$ and $\alpha_2,\cdots,\alpha_{i-1}\geq 0$, we can further obtain that $\alpha_2=\cdots=\alpha_{i+1}=0$. This indicates that if there exists a certain $i\in\{3,\cdots,K\}$ such that $f^*_{n,i}=f^*_{n,i+1}>0$, we have $f^*_{n,n+1}=f^*_{n,n+2}=\cdots=f^*_{n,i+1}$.

Additionally, if there exists a certain $i<j\leq K+1$ such that $f^*_{n,j}=0$, we can deduce that $f^*_{n,j}=f^*_{n,j+1}=\cdots=f^*_{n,K+1}=0$ since $f^*_{n,j}\geq f^*_{n,j+1}\geq\cdots\geq f^*_{n,K+1}$.

Combing the above two cases, we complete the proof. ■

APPENDIX C
PROOF OF PROPERTY 2) IN THEOREM 3

According to Appendix B, we can obtain that $\alpha^*_2=\cdots=\alpha^*_i=0$. Moreover, since $f^*_{n-1,n}>f^*_{n-1,n+1}\geq 0$ $(i\leq n\leq K)$, we have $\sqrt{\frac{\beta_{n-1}}{3\kappa}+\frac{\gamma_{n-1,n}-\alpha_n}{3\kappa\Delta t_n}}>\sqrt{\frac{\beta_n-1}{3\kappa}+\frac{\gamma_{n-1,n+1}-\alpha_{n+1}}{3\kappa\Delta t_{n+1}}}$ according to (B.6). Due to that $\gamma_{n-1,n}=0$ and $\gamma_{n-1,n+1}>0$, we can deduce that $-\frac{\alpha_n}{\Delta t_n}>\frac{\gamma_{n-1,n+1}-\alpha_{n+1}}{\Delta t_{n+1}}>-\frac{\alpha_{n+1}}{\Delta t_{n+1}}$, i.e., $\frac{\alpha_n}{\Delta t_n}<\frac{\alpha_{n+1}}{\Delta t_{n+1}}$ $(i\leq n\leq K)$. Overall, we can conclude that $0=\frac{\alpha^*_2}{\Delta t_2}=\cdots=\frac{\alpha^*_i}{\Delta t_i}<\frac{\alpha^*_{i+1}}{\Delta t_{i+1}}\cdots<\frac{\alpha^*_{K+1}}{\Delta t_{K+1}}$. ■

APPENDIX D
PROOF OF PROPERTY 3) IN THEOREM 3

We first prove the "only if" part. According to property 1) in Theorem 3, if t_i is the transition point, we have $f^*_{n,n+1}=\cdots=f^*_{n,i-1}>f^*_{n,i}\cdots>f^*_{n,j}=\cdots=f^*_{n,K+1}=0$ $(n+1\leq i<j\leq K+1)$ and $\sum_{n=1}^{m-1}f_{n,m}=F_{\max}$ $(i\leq m\leq K+1)$. Since $\sum_{m=n+1}^{K+1}f^*_{n,m}\Delta t_m=F_n$, we can obtain that $\sum_{m=i}^{K+1}f^*_{n,m}\Delta t_m<\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$ $(1\leq n\leq i-2)$. Thus, we have $(\sum_{m=i}^{K+1}\Delta t_m)F_{\max}=\sum_{n=1}^{i-2}\sum_{m=i}^{K+1}f^*_{n,m}\Delta t_m+\sum_{n=i-1}^{K}\sum_{m=n+1}^{K+1}f^*_{n,m}\Delta t_m<\sum_{n=1}^{i-2}\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}+\sum_{n=i-1}^{K}F_n$, i.e., $F_{\max}<F(i)$.

Similarly, if t_{i-1} is the transition point, we have $F_{\max}<F(i-1)$. Since $F(i)-F(i-1)=\frac{F_{i-2}}{\sum_{m=i-1}^{K+1}\Delta t_m}+\frac{\sum_{n=i-1}^{K}F_n}{\sum_{m=i}^{K+1}\Delta t_m}-\frac{\sum_{n=i-2}^{K}F_n}{\sum_{m=i-1}^{K+1}\Delta t_m}=\frac{\sum_{n=i-1}^{K}F_n}{\sum_{m=i}^{K+1}\Delta t_m}-\frac{\sum_{n=i-1}^{K}F_n}{\sum_{m=i-1}^{K+1}\Delta t_m}>0$, we should have $F(i-1)\leq F_{\max}<F(i)$.

For the "if" part, if $F_{\max}<F(i)$, we can deduce that $(\sum_{m=i}^{K+1}\Delta t_m)F_{\max}<\sum_{n=1}^{i-2}\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}+\sum_{n=i-1}^{K}F_n$. Moreover, since $\sum_{n=1}^{i-2}\sum_{m=i}^{K+1}f^*_{n,m}\Delta t_m+\sum_{n=i-1}^{K}F_n\leq(\sum_{m=i}^{K+1}\Delta t_m)F_{\max}$, we have $\sum_{n=1}^{i-2}\sum_{m=i}^{K+1}f^*_{n,m}\Delta t_m+\sum_{n=i-1}^{K}F_n<\sum_{n=1}^{i-2}\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}+\sum_{n=i-1}^{K}F_n$, i.e., $\sum_{n=1}^{i-2}\sum_{m=i}^{K+1}f^*_{n,m}\Delta t_m<\sum_{n=1}^{i-2}\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$. Since

$f^*_{n,n+1} \geq \cdots \geq f^*_{n,i-1} \geq \cdots \geq f^*_{n,K+1}$ $(n+1 \leq i < K+1)$, we can obtain that $\sum_{m=i}^{K+1} f^*_{n,m}\Delta t_m \leq \frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$ $(1 \leq n \leq i-2)$. Therefore, to let $\sum_{n=1}^{i-2}\sum_{m=i}^{K+1} f^*_{n,m}\Delta t_m < \sum_{n=1}^{i-2}\frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$ hold, we should have $\sum_{m=i}^{K+1} f^*_{n,m}\Delta t_m < \frac{F_n(\sum_{m=i}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$ $(1 \leq n \leq i-2)$. Further, it can be deduced that $\sum_{m=i-1}^{K+1} f^*_{n,m}\Delta t_m < \frac{F_n(\sum_{m=i-1}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m}$ $(1 \leq n \leq i-3)$.

Additionally, since $\digamma(i-1) \leq F_{\max}$, we can deduce that $\sum_{n=1}^{i-3}\frac{F_n(\sum_{m=i-1}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m} + \sum_{n=i-2}^{K} F_n \leq (\sum_{m=i-1}^{K+1}\Delta t_m)F_{\max}$. Thus, we have $\sum_{n=1}^{i-3}\sum_{m=i-1}^{K+1} f^*_{n,m}\Delta t_m + \sum_{n=i-2}^{K} F_n < \sum_{n=1}^{i-3}\frac{F_n(\sum_{m=i-1}^{K+1}\Delta t_m)}{\sum_{m=n+1}^{K+1}\Delta t_m} + \sum_{n=i-2}^{K} F_n \leq (\sum_{m=i-1}^{K+1}\Delta t_m)F_{\max}$, which indicates that the computation resource is abundant from t_{i-1} to t_{K+1}. Therefore, we can deduce that $f^*_{n,n+1} = \cdots = f^*_{n,i-1}$ $(n \leq i-2)$. Assume that $t_{\tilde{i}}$ $(\tilde{i} > i)$ is the transition point. We have $\digamma(\tilde{i}-1) \leq F_{\max}$. Since $\digamma(\tilde{i}-1) - \digamma(i) \geq 0$, $F_{\max}$ is infeasible, which breaks the assumption. Therefore, we can conclude that t_i is the transition point.

Combining the proofs of "if" and "only if" part, we complete the proof. ■

Appendix E
Proof of Corollary 5

To find out the optimal computation resource allocation scheme, we first investigate the property of the most energy-efficient scheme without the maximum frequency restriction in Lemma 7, whose proof is provided in Appendix G.

Lemma 7. Regardless of Δt_m, Δt_{m+1} and with given computation cycles F_n in time slots t_m and t_{m+1}, scheme $f^*_{n,m} = f^*_{n,m+1}$ consumes the least energy among all the solutions satisfying $f_{n,m}\Delta t_m + f_{n,m+1}\Delta t_{m+1} = F_n$.

For Corollary 5, we first prove that $f^*_{1,2} \geq f^*_{1,3}$ with given computation cycle F_1 and $f_{2,3}$. Denote the sum computation cycles in t_2 and t_3 of the first offloading device as F_1. Through relaxing the maximum computation resource constraint, the energy consumption is the least when $f_{1,2} = f_{1,3} = \frac{F_1}{\Delta t_2+\Delta t_3}$ according to Lemma 7. Since constraint (3a) should be satisfied, we have

$$f_{1,2} \leq F_{\max}, \quad f_{1,3} + f_{2,3} \leq F_{\max}. \tag{E.1}$$

We consider the following two cases; otherwise, $f_{1,2}$ and $f_{1,3}$ have no feasible solution with the given F_1.

Case 1: $\frac{F_1}{\Delta t_2+\Delta t_3} + f_{2,3} \leq F_{\max}$. In this case, we can deduce that $f_{1,2} = f_{1,3} = \frac{F_1}{\Delta t_2+\Delta t_3}$ satisfies (E.1). Since $f_{1,2} = f_{1,3} = \frac{F_1}{\Delta t_2+\Delta t_3}$ is the most energy efficient solution, the optimal solution in this case is $f^*_{1,2} = f^*_{1,3} = \frac{F_1}{\Delta t_2+\Delta t_3}$.

Case 2: $\frac{F_1}{\Delta t_2+\Delta t_3} + f_{2,3} > F_{\max}$. Obviously, $f_{1,2} = f_{1,3} = \frac{F_1}{\Delta t_2+\Delta t_3}$ is infeasible in this case. We then prove that $f_{1,2} < f_{1,3}$ is also impossible. Since $f_{1,2}\Delta t_2 + f_{1,3}\Delta t_3 = F_1$, $f_{1,2}$ increases as $f_{1,3}$ decreases. If $f_{1,2} < f_{1,3}$, we can deduce that $f_{1,3} > \frac{F_1}{\Delta t_2+\Delta t_3}$. Therefore, we have $f_{1,3} + f_{2,3} > \frac{F_1}{\Delta t_2+\Delta t_3} + f_{2,3} > F_{\max}$ which violates the maximum frequency constraint. As a consequence, the optimal solution is $f^*_{1,2} > f^*_{1,3}$. According to Lemma 7, the energy consumption E increases with $\delta_1 = f_{1,2} - f_{1,3}$ in the considered region $0 < \delta_1 \leq F_1/\Delta t_2$. Therefore, in order to achieve the fewest energy consumption, we should let δ_1 as small as possible. Hence, we can obtain that $f^*_{1,3} = F_{\max} - f_{2,3}$ and $f^*_{1,2} = \frac{F_1 - f^*_{1,3}\Delta t_3}{\Delta t_2}$. The corresponding optimal $\delta^*_1 = \frac{F_1 - (F_{\max}-f_{2,3})(\Delta t_2+\Delta t_3)}{\Delta t_2}$.

Summarizing the above two cases, we can obtain that $f^*_{1,2} \geq f^*_{1,3}$. Subsequently, we prove that in t_m and t_{m+1} $(m = 3, \cdots, K)$, we always have $f^*_{n,m} \geq f^*_{n,m+1}$ for all $n = 1, \cdots, m-1$.

Denote the sum computation cycles of the n-th offloading device in time slots t_m and t_{m+1} by F_n, i.e.,

$$\begin{cases} f_{1,m}\Delta t_m + f_{1,m+1}\Delta t_{m+1} = F_1, \\ \quad\cdots \\ f_{n,m}\Delta t_m + f_{n,m+1}\Delta t_{m+1} = F_n, \\ \quad\cdots \\ f_{m-1,m}\Delta t_m + f_{m-1,m+1}\Delta t_{m+1} = F_{m-1}. \end{cases} \tag{E.2}$$

According to Lemma 7, when $f_{n,m} = f_{n,m+1} = \frac{F_n}{\Delta t_m+\Delta t_{m+1}}$ for all $n = 1, \cdots, m-1$, the minimum energy consumption of the n-th offloading device can be achieved, thus the total energy consumption is minimum. Moreover, the following constraints should be satisfied:

$$\sum_{n=1}^{m-1} f_{n,m} \leq F_{\max}, \quad \sum_{n=1}^{m} f_{n,m+1} \leq F_{\max}. \tag{E.3}$$

We consider two cases.

Case 1: $\sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} \leq F_{\max}$. In this case, we can deduce that $f_{n,m} = f_{n,m+1} = \frac{F_n}{\Delta t_m+\Delta t_{m+1}}$ for all $n = 1, \cdots, m-1$ satisfies (E.3). Thus, the optimal solution in this case is $f^*_{n,m} = f^*_{n,m+1} = \frac{F_n}{\Delta t_m+\Delta t_{m+1}}$ for all $n = 1, \cdots, m-1$.

Case 2: $\sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} > F_{\max}$. Obviously, $f_{n,m} = f_{n,m+1} = \frac{F_n}{\Delta t_m+\Delta t_{m+1}}$ is infeasible in this case. We then prove that $\sum_{n=1}^{m-1} f^*_{n,m} > \sum_{n=1}^{m-1} f^*_{n,m+1}$ by contradiction. By summing all the equalities in (E.2), we have $\Delta t_m\sum_{n=1}^{m-1} f_{n,m} + \Delta t_{m+1}\sum_{n=1}^{m-1} f_{n,m+1} = \sum_{n=1}^{m-1} F_n$. Thus, $\sum_{n=1}^{m-1} f_{n,m}$ is negatively correlated with $\sum_{n=1}^{m-1} f_{n,m+1}$. If $\sum_{n=1}^{m-1} f_{n,m} < \sum_{n=1}^{m-1} f_{n,m+1}$, it can be inferred that $\sum_{n=1}^{m-1} f_{n,m+1} > \sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}}$. We can further have $\sum_{n=1}^{m-1} f_{n,m+1} + f_{m,m+1} > \sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} > F_{max}$ which violates constraint (E.3). Similarly, if $\sum_{n=1}^{m-1} f_{n,m} = \sum_{n=1}^{m-1} f_{n,m+1}$, we can obtain that $\sum_{n=1}^{m-1} f_{n,m} = \sum_{n=1}^{m-1} f_{n,m+1} = \sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}}$. Hence, we have $\sum_{n=1}^{m-1} f_{n,m+1} + f_{m,m+1} = \sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} > F_{max}$ which breaks constraint (E.3). As a consequence, the optimal solution in this case satisfies $\sum_{n=1}^{m-1} f^*_{n,m} > \sum_{n=1}^{m-1} f^*_{n,m+1}$.

Next, we prove that $f^*_{n,m} \geq f^*_{n,m+1}$ for all $n = 1, \cdots, m-1$. With given Δt_m and Δt_{m+1}, we denote the energy consumption of the n-th offloading device by $E_n(F_n, \delta_n)$, where

$\delta_n = f_{n,m} - f_{n,m+1}$. According to (G.2), $E_n(F_n, \delta_n)$ is expressed by

$$E_n(F_n,\delta_n) = \kappa \frac{(F_n+\delta_n \Delta t_{m+1})^3}{(\Delta t_m + \Delta t_{m+1})^3}\Delta t_m + \kappa \frac{(F_n-\delta_n \Delta t_m)^3}{(\Delta t_m + \Delta t_{m+1})^3}\Delta t_{m+1}, (n=1,\cdots,m-1), \quad \text{(E.4)}$$

which decreases when $-F_n/\Delta t_{m+1} \le \delta_n \le 0$ while increases when $0 < \delta_n \le F_n/\Delta t_m$.

Furthermore, since $\sum_{n=1}^{m-1} f_{n,m+1} \le F_{\max} - f_{m,m+1}$, we have

$$\begin{aligned}&\sum_{n=1}^{m-1} f_{n,m} - \sum_{n=1}^{m-1} f_{n,m+1} \\ &\ge \frac{\sum_{n=1}^{m-1} F_n - \Delta t_{m+1}\sum_{n=1}^{m-1} f_{n,m+1}}{\Delta t_m} - \sum_{n=1}^{m-1} f_{n,m+1} \\ &\ge \frac{\sum_{n=1}^{m-1} F_n - (\Delta t_m + \Delta t_{m+1})(F_{\max} - f_{m,m+1})}{\Delta t_m}. \quad \text{(E.5)}\end{aligned}$$

According to (E.5), we have $\sum_{n=1}^{m-1}\delta_n \ge \frac{\sum_{n=1}^{m-1} F_n - (\Delta t_m+\Delta t_{m+1})(F_{\max}-f_{m,m+1})}{\Delta t_m} \triangleq \Omega$. Next, we utilize contradiction to prove that $\delta_n^* \ge 0$ for all $n = 1,\cdots,m-1$. Assume in the optimal solution δ_n^* there exists a certain $\delta_n < 0$. We can suitably decrease other positive δ_n^* and increase the negative $\delta_n < 0$ to zero while keeping $\sum_{n=1}^{m-1}\delta_n$ unchanged. In this case, the total energy consumption is effectively reduced, which contradicts the optimality. That completes the proof of $\delta_n^* \ge 0$, i.e., $f_{n,m}^* \ge f_{n,m+1}^*$ for all $n = 1,\cdots,m-1$.

In summary, since we have proven $f_{1,2}^* \ge f_{1,3}^*$ and $f_{n,m}^* \ge f_{n,m+1}^*$ for $m = 3,\cdots,K$ and $n = 1,\cdots,m-1$, we can deduce Corollary 5. ■

Appendix F
Proof of Corollary 6

In case of $\sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} \le F_{\max}$, the optimal δ_n^* $(n = 1,\cdots,m-1)$ are all zeros according to Corollary 5. Therefore, we only need to justify the case of $\sum_{n=1}^{m-1}\frac{F_n}{\Delta t_m+\Delta t_{m+1}} + f_{m,m+1} > F_{\max}$. In this case, we first prove that the optimal $\sum_{n=1}^{m-1}\delta_n^* = \Omega$. Assume that the optimal $\sum_{n=1}^{m-1}\delta_n^* > \Omega$. We can suitably reduce the positive δ_n^* such that the energy consumption is further reduced, which contradicts the optimality. Therefore, we can construct the following energy consumption minimization problem:

$$\min_{\boldsymbol{\delta}} \quad \sum_{n=1}^{m-1} E_n(F_n,\delta_n), \quad \text{(F.1)}$$

$$s.t. \quad \sum_{n=1}^{m-1}\delta_n = \Omega, \quad \text{(F.1a)}$$

$$0 \le \delta_n \le F_n/\Delta t_m, \quad \forall n = 1,\cdots,m-1, \quad \text{(F.1b)}$$

where $\boldsymbol{\delta} = [\delta_1,\cdots,\delta_{m-1}]^T$.

Based on (G.3), the second derivative of $E_n(F_n,\delta_n)$ with respect to δ_n is given by

$$\frac{\mathrm{d}^2 E_n(F_n,\delta_n)}{\mathrm{d}(\delta_n)^2} = \frac{6\kappa \Delta t_m \Delta t_{m+1}}{(\Delta t_m+\Delta t_{m+1})^2}\left[\delta_n(\Delta t_{m+1}-\Delta t_m)+F_n\right]. \quad \text{(F.2)}$$

We can infer that the second derivative of $E_n(F_n,\delta_n)$ is always positive in the considered region $-F_n/\Delta t_{m+1} \le \delta_n \le F_n/\Delta t_m$, no matter Δt_m is larger than or smaller than, or equal to Δt_{m+1}. Hence, $E_n(F_n,\delta_n)$ is convex with respect to δ_n. Thus, problem (F.1) is convex. The partial Lagrangian function of this problem is expressed as

$$\min_{\boldsymbol{\delta}} \quad \sum_{n=1}^{m-1} E_n(F_n,\delta_n) + \Upsilon\left(\sum_{n=1}^{m-1}\delta_n - \Omega\right), \quad \text{(F.3)}$$

$$s.t. \quad 0 \le \delta_n \le F_n/\Delta t_m, \quad \forall n = 1,\cdots,m-1, \quad \text{(F.3a)}$$

where Υ is the dual variable with respect to constraint (F.1a). Problem (F.3) can be decomposed into a series of $(m-1)$ parallel problems:

$$\min_{\delta_n} \quad E_n(F_n,\delta_n) + \Upsilon\delta_n, \quad \text{(F.4)}$$

$$s.t. \quad 0 \le \delta_n \le F_n/\Delta t_m, \quad \text{(F.4a)}$$

Denote the objective of (F.4) by J_n. Taking the derivative of J_n with respect to δ_n, we have

$$\begin{aligned}\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n} &= \frac{\mathrm{d}E_n(F_n,\delta_n)}{\mathrm{d}\delta_n} + \Upsilon, \\ &= \frac{3\kappa\Delta t_m \Delta t_{m+1}}{(\Delta t_m+\Delta t_{m+1})^2}\delta_n\left(\delta_n(\Delta t_{m+1}-\Delta t_m)+2F_n\right)+\Upsilon.\end{aligned} \quad \text{(F.5)}$$

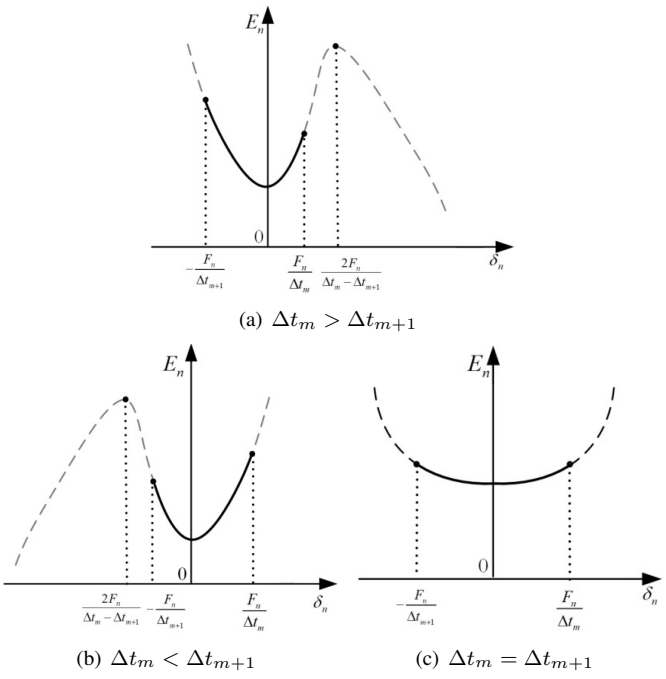
It can be deduced that $\frac{\mathrm{d}E_n(F_n,\delta_n)}{\mathrm{d}\delta_n}$ is non-negative when $0 \le \delta_n \le F_n/\Delta t_m$. If $\Upsilon \ge 0$, we have $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n} >= 0$. Thus, the optimal solution is achieved when $\delta_n = 0$ for all $n = 1,\cdots,m-1$, which contradicts (F.1a). Hence, we should have $\Upsilon < 0$. Due to $\frac{\mathrm{d}^2 J_n}{\mathrm{d}(\delta_n)^2} = \frac{d^2 E_n}{d(\delta_n)^2} > 0$ in the region of $[0, F_n/\Delta t_m]$, we can obtain that $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n}$ monotonously increases. Moreover, we have $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n}|_{\delta_n=0} = \Upsilon < 0$. Therefore, we consider the following two cases.

Case 1: $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n}|_{\delta_n=F_n/\Delta t_m} \le 0$, *i.e.,* $\Upsilon \le -\frac{3\kappa\Delta t_{m+1}F_n^2}{(\Delta t_m+\Delta t_{m+1})\Delta t_m}$. In this case, $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n} <= 0$ in the region of $0 \le \delta_n \le F_n/\Delta t_m$. Therefore, the optimal solution is $\delta_n^* = F_n/\Delta t_m$.

Case 2: $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n}|_{\delta_n=F_n/\Delta t_m} > 0$, *i.e.,* $\Upsilon > -\frac{3\kappa\Delta t_{m+1}F_n^2}{(\Delta t_m+\Delta t_{m+1})\Delta t_m}$. In this case, $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n}$ has a null point in the region of $0 \le \delta_n \le F_n/\Delta t_m$. Thus, J_n decreases first and then increases. Through solving $\frac{\mathrm{d}J_n}{\mathrm{d}\delta_n} = 0$, we obtain that

$$\delta_n^* = \begin{cases} \dfrac{-F_n + \sqrt{F_n^2 - \Upsilon\Xi(\Delta t_{m+1}-\Delta t_m)}}{\Delta t_{m+1}-\Delta t_m}, & \text{if } \Delta t_m < \Delta t_{m+1} \\ & \text{or } \Delta t_m > \Delta t_{m+1}, \\ -\dfrac{\Upsilon\Xi}{2F_n}, & \text{if } \Delta t_m = \Delta t_{m+1}, \end{cases} \quad \text{(F.6)}$$

where $\Xi = \frac{(\Delta t_m+\Delta t_{m+1})^2}{3\kappa\Delta t_m\Delta t_{m+1}}$. Meanwhile, the optimal Υ^* should satisfy constraint (F.1a). Obviously, both the above two cases satisfy $\delta_n^* > 0$ $(n = 1,\cdots,m-1)$, completing the proof. ■



(a) $\Delta t_m > \Delta t_{m+1}$

(b) $\Delta t_m < \Delta t_{m+1}$ (c) $\Delta t_m = \Delta t_{m+1}$

Fig. 9: Illustrations of three cases for the analysis of energy consumption in Appendix G.

Appendix G
Proof of Lemma 7

Denote $\delta_n = f_{n,m} - f_{n,m+1}$. Since $f_{n,m}$ and $f_{n,m+1}$ should be larger than or equal to zero, we can deduce that $-F_n/\Delta t_{m+1} \le \delta_n \le F_n/\Delta t_m$. Hence, we can obtain that

$$f_{n,m} = \frac{F_n + \delta_n \Delta t_{m+1}}{\Delta t_m + \Delta t_{m+1}},\ f_{n,m+1} = \frac{F_n - \delta_n \Delta t_m}{\Delta t_m + \Delta t_{m+1}}. \quad \text{(G.1)}$$

Therefore, the energy consumption in time slot t_m and t_{m+1} can be given by

$$\begin{aligned} E_n &= \kappa f_{n,m}^3 \Delta t_m + \kappa f_{n,m+1}^3 \Delta t_{m+1}, \\ &= \kappa \frac{(F_n + \delta_n \Delta t_{m+1})^3}{(\Delta t_m + \Delta t_{m+1})^3} \Delta t_m + \kappa \frac{(F_n - \delta_n \Delta t_m)^3}{(\Delta t_m + \Delta t_{m+1})^3} \Delta t_{m+1}. \end{aligned} \quad \text{(G.2)}$$

Taking the first derivative of E_n with respect to δ_n, we have

$$\frac{\mathrm{d}E_n}{\mathrm{d}\delta_n} = \frac{3\kappa \Delta t_m \Delta t_{m+1}}{(\Delta t_m + \Delta t_{m+1})^2} \delta_n \left(\delta_n (\Delta t_{m+1} - \Delta t_m) + 2F_n\right). \quad \text{(G.3)}$$

Equation (G.3) has two null points: 0 and $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}}$. We consider the following three cases.

Case 1: $\Delta t_m > \Delta t_{m+1}$. In this case, we have $0 < \frac{2F_n}{\Delta t_m - \Delta t_{m+1}}$. The energy consumption decreases when $\delta_n < 0$ and $\delta_n > \frac{2F_n}{\Delta t_m - \Delta t_{m+1}}$ while increases when $0 \le \delta_n \le \frac{2F_n}{\Delta t_m - \Delta t_{m+1}}$, as shown in Fig. 9(a). Since we can easily prove that $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}} > F_n/\Delta t_m$, the minimum energy consumption is obtained when $\delta_n = 0$, i.e., $f_{n,m}^* = f_{n,m+1}^*$.

Case 2: $\Delta t_m < \Delta t_{m+1}$. In this case, we have $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}} < 0$. The energy consumption increases when $\delta_n < \frac{2F_n}{\Delta t_m - \Delta t_{m+1}}$ and $\delta_n > 0$ while decreases when $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}} \le \delta_n \le 0$, as shown in Fig. 9(b). Similarly, since we can prove that $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}} < -F_n/\Delta t_{m+1}$, the minimum energy consumption is obtained when $\delta_n = 0$.

Case 3: $\Delta t_m = \Delta t_{m+1}$. In this case, two null points coincide, i.e., $\frac{2F_n}{\Delta t_m - \Delta t_{m+1}} = 0$. Therefore, the energy consumption decreases when $\delta_n < 0$ while increases when $\delta_n \ge 0$, as shown in Fig. 9(c). $\delta_n = 0$ is the solution that minimizes energy consumption.

In summary, the energy consumption when $f_{n,m}^* = f_{n,m+1}^* = \frac{F_n}{\Delta t_m + \Delta t_{m+1}}$ is the most energy efficient solution. ■

Appendix H
Proof of Proposition 4

According to Theorem 3, we have $x_{i-1,i} > 0$ ($\forall i \in \{2, \cdots, K+1\}$). Hence, according to (11), we have $\xi^* - \sum_{n=i+1}^{K} \rho_n^* h_{\pi_n} \eta P_0 - \omega_i^* F_{\max} = \sum_{n=1}^{i-1} 2\kappa \frac{(x_{n,i})^3}{(\Delta t_i^*)^3} + \rho_i^* \frac{2\lambda (A_{\pi_i})^3}{h_{\pi_i} (\Delta t_i^*)^3} > 0$. Therefore, it can be derived that $\Delta t_i^* = \sqrt[3]{\frac{2\kappa h_{\pi_i} \sum_{n=1}^{i-1} (x_{n,i})^3 + 2\rho_i^* \lambda (A_{\pi_i})^3}{\xi^* h_{\pi_i} - \sum_{n=i+1}^{K} \rho_n^* (h_{\pi_n})^2 \eta P_0 - \omega_i^* h_{\pi_i} F_{\max}}}$ ($\forall i \in \{2, \cdots, K-1\}$). Similarly, based on (12) and (13), we have $\Delta t_K^* = \sqrt[3]{\frac{2\kappa h_{\pi_K} \sum_{n=1}^{K-1} (x_{n,K})^3 + 2\rho_K^* \lambda (A_{\pi_K})^3}{\xi^* h_{\pi_K} - \omega_K^* h_{\pi_K} F_{\max}}}$ and $\Delta t_{K+1}^* = \sqrt[3]{\frac{2\kappa \sum_{n=1}^{K} (x_{n,K+1})^3}{\xi^* - \omega_{K+1}^* F_{\max}}}$.

Besides, according to (13), we have $\xi^* = \sum_{n=1}^{K} 2\kappa \frac{(x_{n,K+1})^3}{(\Delta t_{K+1}^*)^3} + \omega_{K+1}^* F_{\max} > 0$ since $\sum_{n=1}^{K} (x_{n,K+1})^3 > (x_{K,K+1})^3 = (A_{\pi_K} I_{\pi_K})^3 > 0$. Therefore, we can obtain that $\sum_{i=0}^{K+1} \Delta t_i^* = T$. Furthermore, based on (10), we have $\xi^* = \sum_{n=1}^{K} \rho_n^* h_{\pi_n} \eta P_0$. Due to that ξ^* is positive, there exists at least an $n \in \mathcal{K}$ such that $\rho_n^* > 0$. This indicates that for energy causality constraints (6a), at least a device is run out of energy after offloading, i.e., this device uses all the harvested energy for transmission. Substituting (9) into (10), we have $\rho_1^* \frac{2\lambda (A_{\pi_1})^3}{h_{\pi_1} (\Delta t_1^*)^3} = \rho_1^* h_{\pi_1} \eta P_0$. If $\rho_1^* > 0$, we can deduce that $\Delta t_1^* = \sqrt[3]{\frac{2\lambda}{(h_{\pi_1})^2 \eta P_0}} A_{\pi_1}$. Moreover, since $\frac{\lambda (A_{\pi_1})^3}{h_{\pi_1} (\Delta t_1)^2} = \Delta t_0 h_{\pi_1} \eta P_0$, we have $\Delta t_0^* = \sqrt[3]{\frac{\lambda}{4(h_{\pi_1})^2 \eta P_0}} A_{\pi_1}$. If $\rho_1^* = 0$, $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\Delta t_1^*} = 0$ can be guaranteed for arbitrary Δt_1 satisfying $\frac{\lambda (A_{\pi_1})^3}{h_{\pi_1} (\Delta t_1)^2} \le \Delta t_0 h_{\pi_1} \eta P_0$. That means any pairs of Δt_0 and Δt_1 satisfying $\frac{\lambda (A_{\pi_1})^3}{h_{\pi_1} (\Delta t_1)^2} \le \Delta t_0 h_{\pi_1} \eta P_0$ and $\Delta t_0 + \Delta t_1 = T - \sum_{i=2}^{K+1} \Delta t_i^*$ are the optimal solutions. Hence, according to $\Delta t_0 + \Delta t_1 = T - \sum_{i=2}^{K+1} \Delta t_i$, we should have $\frac{\lambda (A_{\pi_1})^3}{h_{\pi_1} (\Delta t_1)^2} \le \left(T - \sum_{i=2}^{K+1} \Delta t_i - \Delta t_1\right) h_{\pi_1} \eta P_0$, i.e., $\Psi(\Delta t_1) \triangleq \left(T - \sum_{i=2}^{K+1} \Delta t_i - \Delta t_1\right) (\Delta t_1)^2 (h_{\pi_1})^2 \eta P_0 - \lambda (A_{\pi_1})^3 \ge 0$. Taking the derivative of $\Psi(\Delta t_1)$, we have $\Psi'(\Delta t_1) = \Delta t_1 \left(2T - 2\sum_{i=2}^{K+1} \Delta t_i - 3\Delta t_1\right) (h_{\pi_1})^2 \eta P_0$, which has two null points $\Delta t_1 = 0$ and $\Delta t_1 = \frac{2}{3}\left(T - \sum_{i=2}^{K+1} \Delta t_i\right)$. Thus, $\Psi(\Delta t_1)$ increases in the region of $\left(0, \frac{2}{3}\left(T - \sum_{i=2}^{K+1} \Delta t_i\right)\right]$ and decreases in $\left(\frac{2}{3}\left(T - \sum_{i=2}^{K+1} \Delta t_i\right), \left(T - \sum_{i=2}^{K+1} \Delta t_i\right)\right)$. Additionally, we can obtain that $\Psi(0) = \Psi\left(T - \sum_{i=2}^{K+1} \Delta t_i\right)$. Therefore,

we can choose Δt_1^* as the unique null point of $\Psi(\Delta t_1)$ in the range of $\Delta t_1 \in \left(0, \frac{2}{3}\left(T - \sum_{i=2}^{K+1} \Delta t_i\right)\right]$ without loss of generality. ■

REFERENCES

[1] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art," *IEEE Access*, vol. 8, pp. 116 974–117 017, June 2020.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[3] H. Zhou, Z. Zhang, D. Li, and Z. Su, "Joint Optimization of Computing Offloading and Service Caching in Edge Computing-Based Smart Grid," *IEEE Trans. Cloud Comp.*, vol. 11, no. 2, pp. 1122–1132, Apr. 2023.

[4] X. Li, H. Zhang, H. Zhou, N. Wang, K. Long, S. Al-Rubaye, and G. K. Karagiannidis, "Multi-Agent DRL for Resource Allocation and Cache Design in Terrestrial-Satellite Networks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 8, pp. 5031–5042, Aug. 2023.

[5] H. Zhou, T. Wu, X. Chen, S. He, D. Guo, and J. Wu, "Reverse Auction-Based Computation Offloading and Resource Allocation in Mobile Cloud-Edge Computing," *IEEE Trans. Mobile Comp.*, vol. 22, no. 10, pp. 6144–6159, Oct. 2023.

[6] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing," *IEEE Int. Things J.*, vol. 6, no. 3, pp. 4188–4200, June 2019.

[7] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, "Energy-Efficient NOMA-Based Mobile Edge Computing Offloading," *IEEE Commun. Letters*, vol. 23, no. 2, pp. 310–313, Feb. 2019.

[8] Z. Yu, Y. Tang, L. Zhang, and H. Zeng, "Deep Reinforcement Learning Based Computing Offloading Decision and Task Scheduling in Internet of Vehicles," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Xiamen, China, July 2021, pp. 1166–1171.

[9] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 2, pp. 757–789, 2014.

[10] Z. Zhang, H. Pang, A. Georgiadis, and C. Cecati, "Wireless Power Transfer—An Overview," *IEEE Trans. Industrial Electron.*, vol. 66, no. 2, pp. 1044–1058, Feb. 2019.

[11] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[12] S. Bi and Y. J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 6, pp. 4177–4190, June 2018.

[13] P. Chen, B. Lyu, Y. Liu, H. Guo, and Z. Yang, "Multi-IRS Assisted Wireless-Powered Mobile Edge Computing for Internet of Things," *IEEE Trans. Green Commun. Netw.*, pp. 1–1, Sep. 2022.

[14] X. Hu, K.-K. Wong, and K. Yang, "Wireless Powered Cooperation-Assisted Mobile Edge Computing," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.

[15] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, July 2017, pp. 1–6.

[16] Z. Zhu, J. Peng, X. Gu, H. Li, K. Liu, Z. Zhou, and W. Liu, "Fair Resource Allocation for System Throughput Maximization in Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 5332–5340, Jan. 2018.

[17] F. Zhou and R. Q. Hu, "Computation Efficiency Maximization in Wireless-Powered Mobile Edge Computing Networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3170–3184, Feb. 2020.

[18] Y. Xu, P. Cheng, Z. Chen, M. Ding, Y. Li, and B. Vucetic, "Task offloading for large-scale asynchronous mobile edge computing: An index policy approach," *IEEE Trans. Signal Proc.*, vol. 69, pp. 401–416, Dec. 2020.

[19] Y. Dai, M. Sheng, J. Liu, N. Cheng, and X. Shen, "Delay-efficient offloading for NOMA-MEC with asynchronous uploading completion awareness," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. Waikoloa, HI, USA: IEEE, Feb. 2019, pp. 1–6.

[20] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous Mobile-Edge Computation Offloading: Energy-Efficient Resource Management," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 11, pp. 7590–7605, Nov. 2018.

[21] Y. Hu, M. Chen, Y. Wang, Z. Li, M. Pei, and Y. Cang, "Discrete-Time Joint Scheduling of Uploading and Computation for Deterministic MEC Systems Allowing for Task Interruptions and Insertions," *IEEE Wirel. Commun. Letters*, vol. 12, no. 1, pp. 21–25, Jan. 2023.

[22] S. Eom, H. Lee, J. Park, and I. Lee, "Asynchronous Protocol Designs for Energy Efficient Mobile Edge Computing Systems," *IEEE Trans. Veh. Techn.*, vol. 70, no. 1, pp. 1013–1018, Jan. 2021.

[23] K. Guo and T. Q. S. Quek, "On the Asynchrony of Computation Offloading in Multi-User MEC Systems," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7746–7761, Dec. 2020.

[24] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial Offloading Scheduling and Power Allocation for Mobile Edge Computing Systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[25] P. Cai, F. Yang, J. Wang, X. Wu, Y. Yang, and X. Luo, "JOTE: Joint Offloading of Tasks and Energy in Fog-Enabled IoT Networks," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3067–3082, April 2020.

[26] Z. Liang, H. Chen, Y. Liu, and F. Chen, "Data Sensing and Offloading in Edge Computing Networks: TDMA or NOMA?" *IEEE Trans. Wirel. Commun.*, vol. 21, no. 6, pp. 4497–4508, June 2022.

[27] S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, "Lyapunov-Guided Deep Reinforcement Learning for Stable Online Computation Offloading in Mobile-Edge Computing Networks," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 11, pp. 7519–7537, Nov. 2021.

[28] F. Wang, J. Xu, and S. Cui, "Optimal Energy Allocation and Task Offloading Policy for Wireless Powered Mobile Edge Computing Systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2443–2459, Apr. 2020.

[29] J. Liu, K. Xiong, D. W. K. Ng, P. Fan, Z. Zhong, and K. B. Letaief, "Max-Min Energy Balance in Wireless-Powered Hierarchical Fog-Cloud Computing Networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 11, pp. 7064–7080, Nov. 2020.

[30] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-Efficient UAV-Assisted Mobile Edge Computing: Resource Allocation and Trajectory Optimization," *IEEE Trans. Veh. Techn.*, vol. 69, no. 3, pp. 3424–3438, Mar. 2020.

[31] D. P. Bertsekas, *Convex optimization Theory*. Athena Scientific Belmont, 2009.

[32] Y. Yu, X. Bu, K. Yang, H. Yang, X. Gao, and Z. Han, "UAV-Aided Low Latency Multi-Access Edge Computing," *IEEE Trans. Veh. Techn.*, vol. 70, no. 5, pp. 4955–4967, May 2021.

[33] A. Ibrahim, O. A. Dobre, T. M. N. Ngatched, and A. G. Armada, "Bender's Decomposition for Optimization Design Problems in Communication Networks," *IEEE Netw.*, vol. 34, no. 3, pp. 232–239, May 2020.

[34] D. W. K. Ng, Y. Wu, and R. Schober, "Power Efficient Resource Allocation for Full-Duplex Radio Distributed Antenna Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 4, pp. 2896–2911, Apr. 2016.

[35] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[36] D. W. K. Ng and R. Schober, "Secure and Green SWIPT in Distributed Antenna Networks With Limited Backhaul Capacity," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 9, pp. 5082–5097, Sep. 2015.

[37] "GUROBI Optimization, State-of-the-Art Mathematical Programming Solver, v5.6". Apr. 2014. [Online]. Available: http://www.gurobi.com/

[38] "MOSEK ApS: Software for Large-Scale Mathematical Optimization Problems, Version 7.0.0.111". Apr. 2014. [Online]. Available: http://www.mosek.com/

[39] Y. Mezentsev, "Binary cut-and-branch method for solving mixed integer programming problems," in *Proc. Construct. Nonsmooth Analy. Related Topics (CNSA)*, St. Petersburg, Russia, May 2017, pp. 1–3.

Yihan Cang (Student Member, IEEE) received the B.S. degree in communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2019. He is currently working towards his Ph.D. degree in information and communications engineering with National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. His research interests include mobile edge computing, edge AI, and resource allocation in communications.

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3358237

Ming Chen (Member, IEEE) received his Ph.D. degrees from mathematics department of Nanjing University, Nanjing, China, in 1996. In July of 1996, he came to National Mobile Communications Research Laboratory of Southeast University in Nanjing to be a Lecturer. From 1998 to 2003 he has been an Associate Professor and from 2003 to now a professor at the laboratory. His research interests include baseband signal processing, radio and computing resource allocation and network planning for all generation mobile communication systems, visible light communication systems, mobile edge computing systems and satellite mobile communications. He has completed more than 40 research projects in the role of director by now, among which there are more than 15 projects are issued by national ministries, supervised more than 30 Ph. D. and 100 M. Sc. Students and published more than 300 journal papers as coauthor. He won twice the first level provincial progress awards in science and technology.

Yijin Pan (Member, IEEE) received the B.S. and M.S. degree in communication engineering from Chongqing University, Chongqing, China, in 2011 and 2014, respectively, and the PhD degree from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2018. She held the 2019 2021 Royal Society Newton Fellowship. She currently holds an associated professor with the School of Information Science and Engineering, Southeast University, Nanjing, China. She has been a TPC Member of IEEE ICC since 2015 and GLOBECOM since 2017. Her current research interests are in wireless mobile communications, covering topics including wireless signal processing, network resource and performance optimization, embedded intelligent network terminal equipments, etc.

Zhaohui Yang (Member, IEEE) (S'14-M'18) received the Ph.D. degree from Southeast University, Nanjing, China, in 2018. From 2018 to 2020, he was a Post-Doctoral Research Associate at the Center for Telecommunications Research, Department of Informatics, King's College London, U.K. From 2020 to 2022, he was a Research Fellow at the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a ZJU Young Professor with the Zhejiang Key Laboratory of Information Processing Communication and Networking, College of Information Science and Electronic Engineering, Zhejiang University, and also a Research Scientist with Zhejiang Laboratory. His research interests include joint communication, sensing, and computation, federated learning, and semantic communication. He received the 2023 IEEE Marconi Prize Paper Award, 2023 IEEE Katherine Johnson Young Author Paper Award, 2023 IEEE ICCCN best paper award, and the first prize in Invention and Entrepreneurship Award of the China Association of Inventions. He was the Co-Chair for international workshops with more than ten times including IEEE ICC, IEEE GLOBECOM, IEEE WCNC, IEEE PIMRC, and IEEE INFOCOM. He is an Associate Editor for the IEEE Communications Letters, IET Communications, and EURASIP Journal on Wireless Communications and Networking. He has served as a Guest Editor for several journals including IEEE Journal on Selected Areas in Communications.

Ye Hu (Member, IEEE) (S'17) is an assistant professor in the Industrial and Systems Engineering Department at University of Miami. She received her Ph.D. degree from Virginia Tech, VA, USA, in 2021. After graduation, she has served as a postdoctoral research scientist at the Columbia University, and the North Carolina State University. Her research interests span from unmanned aerial vehicle networks, low earth orbit satellite, cyber physical human system, network security to distributed machine learning. She is also the recipient of the best paper award at IEEE GLOBECOM 2020 for her work on meta-learning for drone-based communications.

Haijian Sun (Member, IEEE) is an Assistant Professor in the School of Electrical and Computer Engineering at The University of Georgia, where he joined in 2022 as an Assistant Professor. He obtained her Ph.D. degree in the Department of Electrical and Computer Engineering from Utah State University, USA, in 2019. His current research interests include vehicular communication, wireless communication for 5G and beyond, machine learning at the edge, cyber security, IoT communications, wireless systems, and optimization analysis. Dr. Sun directs ESI Wireless Lab at The University of Georgia and has published extensively in the field of wireless communication.

Mingzhe Chen (Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering and Institute of Data Science and Computing at University of Miami. His research interests include federated learning, reinforcement learning, virtual reality, unmanned aerial vehicles, and Internet of Things. He has received four IEEE Communication Society journal paper awards including the IEEE Marconi Prize Paper Award in Wireless Communications in 2023, the Young Author Best Paper Award in 2021 and 2023, and the Fred W. Ellersick Prize Award in 2022, and four conference best paper awards at ICCCN in 2023, IEEE WCNC in 2021, IEEE ICC in 2020, and IEEE GLOBECOM in 2020. He currently serves as an Associate Editor of IEEE Transactions on Mobile Computing, IEEE Wireless Communications Letters, IEEE Transactions on Green Communications and Networking, and IEEE Transactions on Machine Learning in Communications and Networking.