Collaborative Cloud-Edge-End Task Offloading in MEC-Based Small Cell Networks With Distributed Wireless Backhaul

Hui Xiao, Jiawei Huang, *Member, IEEE*, Zhigang Hu, Meiguang Zheng, and Keqin Li, *Fellow, IEEE*

***Abstract*—Collaborative cloud-edge-end computing is a promising solution to support computation-intensive and latency-sensitive tasks by utilizing rich computing resources of cloud datacenters and low access delay of mobile edge computing (MEC) servers. Compared with traditional cloud computing and MEC, the cloud-edge environment has a stronger heterogeneity of servers and networks, resulting in significant differences between servers in the computation speed and access delay. However, few studies on cloud-edge-end task offloading focused on the characteristic of 5G heterogeneous networks in the cloud-edge environment. In this paper, we study the task offloading problem for collaborative cloud-edge-end computing in MEC-enabled small cell networks with low-cost distributed wireless backhaul. We aim to minimize the energy consumption of all user devices (UDs) via jointly optimizing the offloading decision, UDs' transmission power, and the allocation of spectrum and computation resources. To solve the non-convex problem, we decouple the original problem into three subproblems, and design an efficient method with solving these three subproblems iteratively to obtain a high-quality solution. The simulation results indicate that our proposed method can lead to significant reduction in the energy consumption of all UDs compared with other conventional methods.**

***Index Terms*—Collaborative task offloading, mobile edge computing, distributed wireless backhaul.**

I. Introduction

With the explosive proliferation of intelligent user devices (UDs), various kinds of computation-intensive and latency-intensive applications (e.g., virtual reality, object detection) have emerged in our daily life [1]. However, due to the limited computation capacity and energy supply, it is a great challenge for UDs to execute complex applications within certain deadlines. One popular paradigm to overcome the computation bottleneck is cloud computing (CC) [2], [3], which enables UDs to transfer the tasks to the remote cloud datacenter (CDC) owning powerful computing power. Nevertheless, the transmission delay during the offloading process can be too high to meet the stringent latency constraints for some latency-intensive tasks because of the long distance between UDs and the CDC [4]. Recently, mobile edge computing (MEC) has emerged as a promising paradigm that pushes computation and storage resources to the network edge by deploying servers at the edge of the radio access network (RAN) (e.g., cells) [5]. With the idea of transferring resource-intensive tasks from the UDs to the edge servers (ESs), task offloading enables application tasks to be executed at a lower time and energy cost than local execution [6]. Due to resource limitation such as CPU, bandwidth, and storage, the optimization of resource utilization [7], service management [8], and quality of service (QoS) provisioning [9] are the main concerns in the research of MEC. Unfortunately, the limited resource capacity still leads to considerable performance bottlenecks for task offloading in MEC.

Inspired by the above characteristics, the collaboration of CC and MEC has been introduced to exploit the benefits of these two paradigms [10], [11], [12], i.e., using CC to guarantee adequate supply and high availability of resources and using MEC to support mobility and low latency requirement of UDs. Recently, some researchers have investigated the performance of collaborative cloud-edge-end computing architecture from the perspective of energy consumption and QoS. Many works aimed to minimize the energy consumption [13], [14] or task execution time [15], [16] by optimizing the offloading strategy and utilization of computation and communication resources. Nevertheless, as the network environment and task demands become increasingly complex and dynamic, traditional offloading approaches based on heuristic rules [17] and mathematical programming [15] may not be incapable of making real-time decisions in large-scale edge-cloud computing systems. Recently, there has been interesting research [18], [19] that adopts artificial intelligence methods (e.g., reinforcement learning, deep learning) to design real-time offloading schemes with prompt responses to incoming environment change. Since UDs directly communicate with their associated cells via RAN, all the communication between UDs and the CDC needs to be transferred by the cells. Therefore, the network configuration between cells and the CDC will exert great influence on the transmission delay and offloading performance [20]. However, the above studies

Manuscript received 4 September 2022; revised 15 February 2023; accepted 5 April 2023. Date of publication 11 April 2023; date of current version 12 December 2023. This work was supported in part by the Natural Science Foundation of China under Grant 62172442, and in part by the Youth Science Foundation of Natural Science Foundation of Hunan Province under Grant 2020JJ5775. The associate editor coordinating the review of this article and approving it for publication was K. Shiomoto. *(Corresponding author: Zhigang Hu.)*

Hui Xiao, Jiawei Huang, Zhigang Hu, and Meiguang Zheng are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: huixiao@csu.edu.cn; jiaweihuang@csu.edu.cn; zghu@csu.edu.cn; zhengmeiguang@csu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TNSM.2023.3266238

only considered a simple cross-tier communication mechanism between cells and the CDC, i.e., each cell has direct one-hop access to the CDC, and thus are not applicable for more complex 5G network situations.

In the current fifth-generation (5G) networks, a huge number of small cells (SCs) are densely deployed to meet the exponential growth of wireless data traffic [21], making the construction of traditional fiber-optic backhaul network costly and difficult. As the millimeter-wave (mm-wave) technologies develop rapidly, the distributed wireless backhaul (DWB) is emerging as a desiring solution for interconnecting SCs due to its high flexibility and cost-effectiveness [22]. By deploying the DWB, some SCs serve as gateways that gain access to the core network through fiber-optic cables. Other SCs transmit their data to their respective gateways via DWB links, and then the gateways forward the aggregated backhaul traffic to the core network. Note that the CDC is normally deployed in remote areas far from the city center [23]. Therefore, the gateway acts as a bridge between the densely-deployed small cells and the cloud and the deployment of DWB helps build communication links between a huge number of SCs and the CDC in a low-cost and high-efficiency manner. In such case, the allocation of wireless backhaul spectrum and wired bandwidth can exert a great influence on the transmission speed of the tasks offloaded to the CDC, and therefore deserves serious consideration during making offloading decision. Furthermore, the offloading decision can affect the allocation of computation and communication resources [12]. Therefore, the flexible offloading destinations (including the UD, ES, and CDC) and heterogeneous network environments in the cloud-edge computing system make it more complex and challenging to design a joint offloading decision and resource allocation scheme for achieving low energy consumption and high QoS level compared with the MEC system.

In this paper, we consider a heterogeneous cloud-edge-end cooperative network supported by densely deployed SCs with DWB, where each SC is attached with an ES for serving the UDs in its coverage area. Each task can be executed locally, or offloaded to the ES or the CDC, which typically corresponds to the scenario of serving non-partitionable tasks [24], [25], [26]. SCs are divided into two categories: the gateway connected to the CDC via fiber-optic links, and normal SCs connected to the gateway via DWB. Considering the different communication links between different categories of SCs and the CDC, we present a novel network communication model for calculating the task transmission delay under different offloading decisions. We focus on optimizing the energy consumption of UDs for the benefit of UDs due to the convenient and stable energy supply at the edge nodes and the limited battery capacity of UDs according to [7], [27], [28]. According to previous studies [7], [16], [29], the return of the result can be omitted since the downlink rate is normally much larger than the uplink rate and the result data size is much smaller than the input data size. Besides, it is also essential to meet the latency requirements of UDs and ensure a desired QoS level. Therefore, we formulate an energy-efficient task offloading problem where the energy consumption of all UDs incurred from local task execution and local task uploading is minimized under the requirements of latency deadline. The main contributions of this paper are summarized as follows:

- We present a DWB-interconnected small cell network architecture where each SC is associated with the gateway via DWB and the gateway with the CDC via fiber-optic links. Furthermore, each UD can either perform its task on the local device or offload its task to the ES or CDC. A sum UD energy consumption minimization problem with delay constraints is formulated which optimizes the offloading decision, transmission power, and joint spectrum and computation resource allocation.
- To solve the non-convex optimization problem, we decouple the original problem into three subproblems and design an efficient method to obtain the joint solution by solving these three subproblems iteratively. The offloading decision is obtained via sub-gradient based primal-dual method. The RAN spectrum allocation problem is solved in closed form based on Lagrangian multiplier method. The combination of interior-point method and Newton's method is used to reach the solution of the joint DWB spectrum and computation capacity allocation problem. We also analyze the complexity of the proposed method theoretically.
- We present some numerical results by comparing the proposed method with four other methods including the exhaustive search method, the random offloading method, an edge-end offloading method (JTORAA) and a cloud-edge-end offloading method (ISA-COO). Numerical results demonstrate that our proposed method can outperform other methods in reducing energy consumption of UDs and therefore evaluate the efficiency of the proposed method.

II. Related Work

Recently, the computation offloading mechanism has attracted significant attention in industry and academia and has gained extensive investigation. The related studies in the literature can be roughly reviewed from two aspects: edge-end offloading and cloud-edge-end offloading.

In the edge-end offloading case, each UD can choose to transfer its task to the ES deployed in its associated SC. Most related works considered minimizing latency or energy consumption as their optimization objective. For example, [9], [30], [31] studied the mobile edge computation offloading scheme in wireless cellular networks with the goal of minimizing the latency. Tang and Hu [9] developed a distributed successive convex approximation (SCA)-based algorithm to joint optimize the allocation of computation and communication resources under the limitation of battery capacity and inter-user interference. Wu et al. [30] integrated the nonorthogonal multiple access (NOMA) with MEC and designed a multi-UD task offloading scheme where the offloading ratios, the uploading duration and the downloading duration are jointly optimized. Fang et al. [31] focused on the partial computation offloading and proposed a bisection search-based iterative approach to optimize the tasks partition ratios and offloading transmit power.

References [7], [32], [33] have studied on computation offloading mechanisms for MEC to minimize the sum energy consumption of all UDs. El Haber et al. [32] developed a partial task offloading scheme in MEC-enabled heterogeneous networks based on successive convex approximation methods. Xu et al. [7] jointly optimized binary task offloading decision, UD transmission power, the allocation of local CPU frequency, ES computation capacity and subchannel resource considering a NOMA-based communication model. Bi et al. [33] developed a hybrid metaheuristic method which optimizes the offloading ratio, allocated bandwidth, CPU utilization, and UD transmission power. Tong et al. [34] designed a Lyapunov-based task offloading approach which optimizes the local calculation rate and the task offloading ratio to balance the incurred energy cost and virtual queue backlog. For some application scenarios that have strict requirements on both latency and energy consumption, it is necessary to comprehensively optimize the incurred latency and energy consumption during task offloading. For instance, Li et al. [35] developed a DRL-based task offloading algorithm to maximize the system reward considering the utility and energy consumption generated from task processing and the penalty caused by task dropping. To minimize the total system cost integrating energy consumption and delay, Lu et al. [36] designed a strategy of resource scheduling, task offloading, and selection of base stations and channels, while Liu et al. [37] developed a heuristic algorithm involving mobility prediction and CPU frequency control.

However, the above studies only focus on the task offloading between UDs and ESs in the MEC-enabled network architecture, where the finite computation capacities can be the main bottleneck. Therefore, a number of studies on collaborative offloading mechanisms in cloud-edge-end networks have been carried out to utilize the powerful computation capability of CDCs. For example, Ren et al. [15] presented a convex optimization-based task offloading method in a collaborative cloud-edge computing network for minimizing the weighted-sum communication latency of UDs without considering local computation. In [17], Li et al. designed a two-level task scheduling framework by using the artificial fish swarm optimization in an edge cloud environment to achieve load balancing. Gao et al. [19] modelled the offloading decision process as a Markov decision process (MDP) and developed a Q-learning-based approach for achieving the optimal offloading strategy.

In order to model the communication between ESs and the CDC more accurately, [13], [14], [38] considered a hybrid fiber-wireless network comprising of wireless access network and optic-fiber backhaul network. Guo and Liu [13] designed two collaborative computation offloading mechanisms based on greedy strategy and game theory respectively for minimizing the overall UDs' energy consumption. He et al. [14] designed an iterative searching method to optimize the offloading strategy, CPU frequency, and UDs' transmission power for various type of task requests. Chowdhury and Maier [38] presented a dynamical collaborative offloading approach taking the characteristics of tasks into account to minimize task completion time. Ebrahimzadeh and Maier [39] characterized the trade-off between the UDs' energy consumption and task execution latency by Pareto front analysis to study the offloading decision and computation capacity allocation. Different from the above works, Kai et al. [16] proposed a cloud-edge-end network architecture where UDs are connected to their associated edge nodes via wireless access links and the edge nodes communicate with the CDC via wireless fronthaul links. They expressed the task offloading problem as a sum latency minimization problem and developed a SCA-based method to reach the optimal solution of the problem.

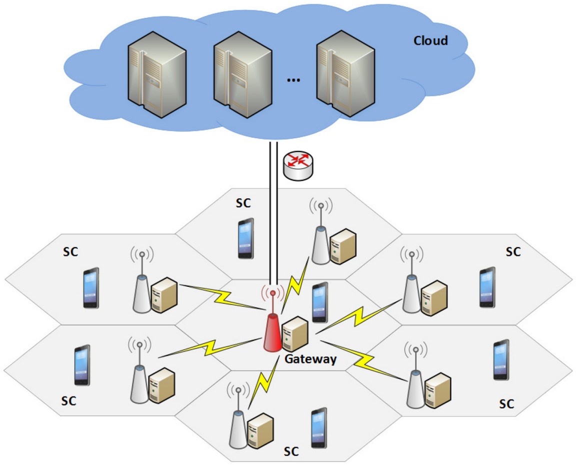

Fig. 1. The architecture of collaborative cloud-edge-end network.

Related works that considered collaborative cloud-edge-end computing paradigm assumed each ES is directly connected to the CDC via a one-hop backhaul/fronthaul link, which is much more costly and cumbersome in practice. As far as we are concerned, our work is the first to study the collaborative cloud-edge-end task offloading optimization problem in the small cell network with DWB. We build a heterogenous cloud-edge-end computing network where one designated gateway is connected to the CDC via optic-fiber links while other SCs communicate with the gateway via wireless backhaul links. Taking the heterogeneous nature of the cloud-edge network into account and for obtaining a high system energy efficiency, we aim to minimize the total energy consumption of UDs by optimizing the offloading decision, UDs' transmission power, wireless spectrum and computation capacity. Compared with the work [7] with the same comprehensive optimization parameters, including offloading decision and communication and computation resource allocation, this paper considers taking full advantage of the rich computation resources of the CDC. Although the work [14] integrated the centralized cloud and distributed ESs to enable collaborative task offloading, the impact of bandwidth resource allocation on the task transmission rate was neglected.

III. System Model

We consider a collaborative cloud-edge-end network where the coverage area is separated into many SCs as shown in Fig. 1. One of these SCs is designated as the gateway, which is connected to a CDC via high-capacity optic-fiber links. The remaining SCs are associated with the gateway via DWB links. We denote the gateway by index 0, and the set of remaining SCs by $\mathcal{N} = \{1, \ldots, N\}$. The set of all SCs is indexed by $\mathcal{M} = \{0 \cup \mathcal{N}\}$. Each SC $j \in \mathcal{M}$ is equipped with an ES to serve a set of UDs in its coverage area, denoted by $\mathcal{U}_j = \{1, \ldots, U_j\}$. The set of all UDs is denoted as $\mathcal{U} = \cup_{j \in \mathcal{M}} \mathcal{U}_j$. Here, the index of the SC is also used to denote its corresponding ES, and unless specified otherwise, $|\cdot|$ is used to denote the number of elements in the set.

Each UD $i \in \mathcal{N}$ has a computational task to be executed, which can be denoted by the tuple (D_i, F_i, T_i), where D_i represents the task input size, F_i describes the amount of CPU cycles needed to accomplish the task, and T_i represents the tolerable maximum delay. Here, the return of result is omitted as the size of result data is so small compared with the input data [29]. We define a binary offloading decision variable $x_{ijk} \in [0, 1] (k \in \mathcal{K} = \{0, 1, 2\})$ for UD i in the range of SC j, where $x_{ij0} = 1$ indicates the task is executed locally, $x_{ij1} = 1$ indicates the task is offloaded to the associated ES j, and $x_{ij2} = 1$ denotes the task is offloaded to the CDC.

A. Wireless Communication Model

In this subsection, we mainly concentrate on the uplink transmission and present the wireless communication models between UDs and their associated SCs, the normal SCs and the gateway, the gateway and the CDC. For saving space, the communication process via the optic-fiber network is introduced in (10) when calculating the total task execution latency in the CDC in Section III-B. To avoid the cross-tier interference, we consider a spectrum-splitted system [40] where the total radio spectrum is divided into two parts: $S^a \in (0, 1)$ for the wireless access communication between UDs and SCs (including the gateway), and $S^b = 1 - S^a$ for the backhaul transmission between the normal SCs and the gateway. Furthermore, we consider the wireless access communication with each SC and backhaul communication with the gateway to be based on the frequency division channel access (FDMA) method, and therefore the intra-cell interference can be ignored [41]. We also assume the inter-cell interference due to spectrum reuse between different SCs is negligibly small due to beamforming and the fact that wireless signals (e.g., mm-wave) are sensitive to blockage [42].

Whether the UD chooses to execute its task on the ES or in the CDC, the task needs to be uploaded to its associated SC first. We define s_{ijk}^a and $p_{ijk}, k \in \{1, 2\}$ as the fraction of radio spectrum and power assigned for the transmission from UD i to SC j when UD i offloads its task to ES j ($k = 1$) or the CDC ($k = 2$), respectively. The achievable uplink transmission rate of UD i associated to SC j can be calculated as

$$r_{ijk}^{a} = s_{ijk}^{a} B \log_2\left(1 + \frac{p_{ijk} g_{ij}}{s_{ijk}^{a} B N_0}\right), \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (1)$$

where B is the total bandwidth of radio spectrum, N_0 is the spectral density of noise, $g_{ij} = |g_0|^2 d_{ij}^{-\nu}$ represents the channel gain between UD i and SC j. g_0, d_{ij}, and ν here denote the Rayleigh fading channel coefficient with Gaussian nature, the distance between UD i and SC j and path loss exponent, respectively [43]. Here, we assume a constant value of g_{ij} due to the low mobility of UDs during the short offloading duration [32].

When UDs offload tasks to the CDC, the tasks uploaded to their associated SCs need to be first forwarded to the gateway via DWB and then transmitted to the CDC via optic-fiber links. Let s_{ij}^b indicate the fraction of spectrum assigned to SC j for forwarding the task of UD i to the gateway. The data forwarding rate from SC j to the gateway for UD i is given by

$$r_{ij}^{b} = s_{ij}^{b} B \log_2\left(1 + \frac{P_j \overline{g}_j}{s_{ij}^{b} B N_0}\right), \forall j \in \mathcal{N}, i \in \mathcal{U}_j \quad (2)$$

where P_j denotes the transmission power of SC j, and $\overline{g}_j$ is the channel gain from SC j to the gateway, similar to g_{ij}.

B. Computation Model

In the following, we present the computation models with respect to delay and energy consumption for tasks processed locally, on the ES, and in the CDC in this subsection.

In general, the CPU frequency is fixed at each UD and can vary over UDs. The resulting processing delay and energy consumption of local computation for UD i in SC j can be expressed as

$$T_{ij}^{l} = \frac{F_i}{f_i^l}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (3)$$

and

$$E_{ij}^{lc} = \kappa \left(f_i^l\right)^2 F_i, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (4)$$

respectively, where f_i^l is the computation capability in cycles/second of UD i, and κ denotes the effective switched capacitance related to the UD's chip architecture [44].

Based on the uplink data rate model described in (1), the transmission time and energy consumption for UD i to upload its task to SC j are given by

$$T_{ijk}^{at} = \frac{D_i}{r_{ijk}^{a}}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \{1, 2\}, \quad (5)$$

$$E_{ijk}^{at} = p_{ijk} T_{ijk}^{at}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \{1, 2\}, \quad (6)$$

respectively. Denote f_i^e as the computation capacity assigned to UD i by SC j. The resulting computation delay of the corresponding task on SC j can be computed as

$$T_{ij}^{ec} = \frac{F_i}{f_{ij}^{e}}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (7)$$

Accordingly, the total delay of UD i for remote task processing on the corresponding ES can be given as

$$T_{ij}^{e} = T_{ij1}^{at} + T_{ij}^{ec}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j. \quad (8)$$

All the normal SCs have to forward their tasks to the gateway via DBW to offload their tasks to the CDC. According to (2), the backhaul transmission delay of forwarding the task of UD i from SC j to the gateway can be calculated as

$$T_{ij}^{bt} = \begin{cases} \frac{D_i}{r_{ij}^{b}}, & \forall j \in \mathcal{N}, i \in \mathcal{U}_j, \\ 0, & \forall i \in \mathcal{U}_0. \end{cases} \quad (9)$$

Furthermore, due to the long transmission distance from the gateway to the CDC, we take into account the transmission delay and propagation delay for the uplink transmission from the gateway to the CDC [13]. Therefore, denoting f_{ij}^c as the computation capacity assigned to UD i by the CDC, the total delay of UD i for remote task processing in the CDC can be calculated as

$$T_{ij}^c = T_{ij2}^{at} + T_{ij}^{bt} + \frac{D_i}{C} + \chi + \frac{F_i}{f_{ij}^c}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (10)$$

where C is the bandwidth of the optic-fiber network, and χ denotes the propagation delay.

C. Problem Formulation

Our objective is to minimize the total energy consumption of all UDs by jointly considering the offloading decision $\boldsymbol{x} = \{x_{ijk}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \mathcal{K}\}$, power control $\boldsymbol{p} = \{p_{ijk}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \{1,2\}\}$, the resource allocation of radio spectrum $\boldsymbol{s^a} = \{s_{ijk}^a, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \{1,2\}\}$ and $\boldsymbol{s^b} = \{s_{ij}^b, \forall j \in \mathcal{N}, i \in \mathcal{U}_j\}$, computation capacity $\boldsymbol{f^e} = \{f_{ij}^e, \forall j \in \mathcal{M}, i \in \mathcal{U}_j\}$ and $\boldsymbol{f^c} = \{f_{ij}^c, \forall j \in \mathcal{M}, i \in \mathcal{U}_j\}$. The energy consumption resulting from accomplishing the task of UD i in the range of SC j can be given by

$$E_{ij} = x_{ij0} E_{ij}^{lc} + x_{ij1} E_{ij1}^{at} + x_{ij2} E_{ij2}^{at}, \quad (11)$$

and we can formulate the sum energy minimization problem as

$$\min_{\boldsymbol{x},\boldsymbol{p},\boldsymbol{s^a},\boldsymbol{s^b},\boldsymbol{f^e},\boldsymbol{f^c}} \sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{U}_j} E_{ij} \quad (12a)$$

$$\text{s.t.} \quad x_{ij0} T_i^l + x_{ij1} T_i^e + x_{ij2} T_i^c \le T_i, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (12b)$$

$$\sum_{i\in\mathcal{U}_j} \sum_{k\in\{1,2\}} x_{ijk} s_{ijk}^a \le S^a, \forall j \in \mathcal{M}, \quad (12c)$$

$$\sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{U}_j} x_{ij2} s_{ij}^b \le S^b, \quad (12d)$$

$$\sum_{k\in\{1,2\}} x_{ijk} p_{ijk} \le p_i^{\max}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (12e)$$

$$\sum_{i\in\mathcal{U}_j} x_{ij1} f_{ij}^e \le f_j^{\max}, \forall j \in \mathcal{M}, \quad (12f)$$

$$\sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{U}_j} x_{ij2} f_{ij}^c \le f_c^{\max}, \quad (12g)$$

$$\sum_{k\in\mathcal{K}} x_{ijk} = 1, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, \quad (12h)$$

$$x_{ijk} \in \{0,1\}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \mathcal{K}. \quad (12i)$$

Constraints (12b) represent the maximum delay requirements of the task completion time for all UDs. Constraints (12c) and (12d) ensure that radio spectrum allocation in each SC's RAN and in DWB is separate and non-overlapping. Constraints (12e) are the maximum power budget for each UD. Constraints (12f) and (12g) are to respect the computation capacity of each ES and the CDC. Constraints (12h) represent that each UD can only choose one place to perform its task. It can be observed that Problem (12) is a non-convex mixed-integer non-linear programming (MINLP) problem which is NP-hard and thus we cannot reach its optimal solution by using polynomial-time algorithms.

IV. Proposed Algorithm

In order to solve the proposed non-convex problem, we focus on the structural properties of Problem (12) to remove the strong coupling among different variables in the problem. We decouple the original problem into three subproblems, i.e., the offloading decision subproblem, the RAN spectrum allocation subproblem, and the joint DWB spectrum and computation capacity allocation subproblem, and present an efficient iterative method to obtain a high-quality sub-optimal solution. Specifically, we first solve the offloading decision problem considering the discrete nature of offloading decision variables. The optimal transmission power of offloading UDs can be achieved as a function of the remaining variables according to the fixed offloading decision. Based on the obtained optimal power, we give the solutions to the RAN spectrum allocation subproblem and the joint DWB spectrum and computation capacity allocation subproblem, respectively.

A. Offloading Decision

With fixed $\{\boldsymbol{p}, \boldsymbol{s^a}, \boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$ and relaxing the integer constraints (12i), Problem (12) which optimizes $\boldsymbol{x}$ can be reformulated as

$$\min_{\boldsymbol{x}} \sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{U}_j} E_{ij} \quad (13a)$$

$$\text{s.t.} \quad 0 \le x_{ijk} \le 1, \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k \in \mathcal{K}, \quad (13b)$$

$$(12b) - (12h). \quad (13c)$$

Obviously, Problem (13) is convex and therefore we can effectively solved it via the dual method [45]. Define $\boldsymbol{\alpha} = \{\alpha_{ij} \ge 0, \forall j \in \mathcal{M}, i \in \mathcal{U}_j\}, \boldsymbol{\beta} = \{\beta_j \ge 0, \forall j \in \mathcal{M}\}$, $\gamma \ge 0, \boldsymbol{\epsilon} = \{\epsilon_{ij}, \forall j \in \mathcal{M}, i \in \mathcal{U}_j\}, \boldsymbol{\zeta} = \{\zeta_j \ge 0, \forall j \in \mathcal{M}\}$, and $\varphi \ge 0$ as the dual variables associated with Constraints (12b)-(12g), respectively. We have the following theorem to reach the optimal solution of Problem (13).

Theorem 1: The optimal offloading decision $\boldsymbol{x}^*$ of Problem (13) can be expressed as

$$x_{ijk}^* = \begin{cases} 1, & \text{if } k = \arg\min_{k\in\mathcal{K}} e_{ijk}, \\ 0, & \text{otherwise}, \end{cases} \quad (14)$$

where

$$e_{ijk} = \begin{cases} E_{ij}^{lc} + \alpha_{ij} T_i^l, \quad \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k = 0, \\ E_{ij1}^{at} + \alpha_{ij} T_i^e + \beta_j s_{ij1}^a + \epsilon_{ij} p_{ij1} + \zeta_j f_{ij}^e, \\ \qquad \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k = 1, \\ E_{ij2}^{at} + \alpha_{ij} T_i^c + \beta_j s_{ij2}^a + \gamma s_{ij}^b + \epsilon_{ij} p_{ij2} + \varphi f_{ij}^c, \\ \qquad \forall j \in \mathcal{M}, i \in \mathcal{U}_j, k = 2. \end{cases} \quad (15)$$

If there exists more than one value of k that generates the minimum value of e_{ijk}, we will select any of these values of k and make the corresponding offloading decision.

Proof: See Appendix A. ■

Theorem 1 provides the optimal solution to the primal variables $\boldsymbol{x}$ with respect to the dual variables. We accordingly develop a primal-dual algorithm where the primal and dual variables are iteratively updated until their values converge. The optimal primal solution depicted in Theorem 1 is utilized

Algorithm 1 Primal-Dual Offloading Decision Algorithm

```
Input: $\boldsymbol{p}, \boldsymbol{s^a}, \boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}$
Output: $\boldsymbol{x}^*$
repeat
    Initialize the dual variables $\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \boldsymbol{\epsilon}, \boldsymbol{\zeta}, \varphi$;
    Obtain the optimal offloading decision $\boldsymbol{x}$ according to (14);
    Update dual variables $\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \boldsymbol{\epsilon}, \boldsymbol{\zeta}, \varphi$ according to (16a)-(16f);
until the dual function converges with $\rho$;
return the optimal offloading decision $\boldsymbol{x}^*$.
```

to formulate the dual function (50). Problem (13) then is converted into maximizing its corresponding dual function (50) with respect to the dual variables. Here, the sub-gradient method [46] is used to update the value of $\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \boldsymbol{\epsilon}, \boldsymbol{\zeta}, \varphi$ based on their subgradients as follows, respectively.

$$\alpha_{ij} = \left[\alpha_{ij} + \delta\Big(x_{ij0} T^l_i + x_{ij1} T^e_i + x_{ij2} T^c_i - T_i\Big)\right]^+, \tag{16a}$$

$$\beta_j = \left[\beta_j + \delta\left(\sum_{i \in \mathcal{U}_j} \sum_{k \in \{1,2\}} x_{ijk} s^a_{ijk} - S^a\right)\right]^+, \tag{16b}$$

$$\gamma = \left[\gamma + \delta\left(\sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{U}_j} x_{ij2} s^b_{ij} - S^b\right)\right]^+, \tag{16c}$$

$$\epsilon_{ij} = \left[\epsilon_{ij} + \delta\left(\sum_{k \in \{1,2\}} x_{ijk} p_{ijk} - p^{\max}_i\right)\right]^+, \tag{16d}$$

$$\zeta_j = \left[\zeta_j + \delta\left(\sum_{i \in \mathcal{U}_j} x_{ij1} f^e_{ij} - f^{\max}_j\right)\right]^+, \tag{16e}$$

$$\varphi = \left[\varphi + \delta\left(\sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{U}_j} x_{ij2} f^c_{ij} - f^{\max}_c\right)\right]^+, \tag{16f}$$

where $[y]^+ = \max\{y, 0\}$, and δ is a sequence of step sizes which are dynamically chosen according to the self-adaptive scheme proposed in [46].

Accordingly, the primal-dual method for solving Problem (13) is introduced in Algorithm 1. For our implementation, at each iteration, the values of variables $\boldsymbol{x}$ and dual function (50) are updated based on the current value of dual variables. Then we calculate the values of the dual variables based on (16a)-(16f). The primal-dual iteration is continued until reaching the stopping criterion of desired accuracy level ρ. The convergence of Algorithm 1 is guaranteed since Problem (13) is a convex problem. With the convergence of the values of dual variables, the optimal solution of the primal variables can also be obtained by strong duality.

B. Power Control Optimization

The optimal power control with given offloading decision $\boldsymbol{x}$ can be obtained according to the following lemma.

Lemma 1: Given fixed offloading decision $\boldsymbol{x}$, the optimal power p^*_{ijk} can be expressed as

$$p^*_{ijk} = \frac{s^a_{ijk} B N_0}{g_{ij}} \left(2^{\frac{D_i}{s^a_{ijk} B T'_{ijk}}} - 1\right), \forall i \in \mathcal{U}_{jk}, k \in \{1, 2\}, \tag{17}$$

where $T'_{ij1} = T_i - T^{ec}_{ij}$, $T'_{ij2} = T_i - T^{bt}_{ij} - \frac{D_i}{C} - \chi - \frac{F_i}{f^c_{ij}}$, $\mathcal{U}_{j1}$ and $\mathcal{U}_{j2}$ represent the set of UDs in SC j that offload the task to the ES and the CDC, respectively.

Proof: See Appendix B. ■

According to Lemma 1, the optimal power p^*_{ijk} can be expressed a function of variables $\{\boldsymbol{s^a}, \boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$. Therefore, we replace p_{ijk} in Problem (12) by p^*_{ijk} in (17), and solve the Problem (12) with fixed offloading decision by optimizing $\{\boldsymbol{s^a}, \boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$.

C. Spectrum and Computation Resource Allocation

With fixed offloading decision $\boldsymbol{x}$, and substituting the optimal power p^*_{ijk} given in (17) into Problem (12), the spectrum and computation resource allocation problem can be formulated as

$$\min_{\boldsymbol{s^a}, \boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}} \sum_{j \in \mathcal{M}} \sum_{k \in \{1,2\}} \sum_{i \in \mathcal{U}_{jk}} \frac{s^a_{ijk} B N_0}{g_{ij}} \left(2^{\frac{D_i}{s^a_{ijk} B T'_{ijk}}} - 1\right) T'_{ijk} \tag{18a}$$

$$\text{s.t.} \sum_{k \in \{1,2\}} \sum_{i \in \mathcal{U}_{jk}} s^a_{ijk} \le S^a, \forall j \in \mathcal{M}, \tag{18b}$$

$$\sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{U}_{j2}} s^b_{ij} \le S^b, \tag{18c}$$

$$\frac{s^a_{ijk} B N_0}{g_{ij}} \left(2^{\frac{D_i}{s^a_{ijk} B T'_{ijk}}} - 1\right) \le p^{\max}_i,$$
$$\forall j \in \mathcal{M}, k \in \{1, 2\}, i \in \mathcal{U}_{jk}, \tag{18d}$$

$$\sum_{i \in \mathcal{U}_{j1}} f^e_{ij} \le f^{\max}_j, \forall j \in \mathcal{M}, \tag{18e}$$

$$\sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{U}_{j2}} f^c_{ij} \le f^{\max}_c. \tag{18f}$$

Note that the non-convexity of objective function (18a) results from the strong coupling of different variables in the term $s^a_{ijk} T'_{ijk}$. Thus, Problem (18) is a non-convex problem, which is complex and difficult to resolve. According to the coupling relationship between variables $\boldsymbol{s^a}$ and variables $\{\boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$, Problem (18) is decoupled into two subproblems, i.e., a RAN spectrum allocation subproblem optimizing $\boldsymbol{s^a}$ and a joint resource allocation subproblem optimizing $\{\boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$ with respect to T'_{ijk}. In the following subsections, we detail the solutions to these two subproblems.

D. RAN Spectrum Allocation

Given fixed $\{\boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$, the RAN spectrum allocation problem for the UDs associated with the same SC can be solved independently. The objective function (18a) is a monotonically decreasing and convex function of s^a_{ijk}. To show this, we define function $q_3(y) = y(e^{\frac{1}{y}} - 1), y > 0$, and we have

$$q_3'(y) = e^{\frac{1}{y}}\left(1 - \frac{1}{y} - \frac{1}{y^2}\right) - 1, \tag{19a}$$

$$q_3''(y) = e^{\frac{1}{y}}\left(\frac{1}{y^4} + \frac{3}{y^3}\right) > 0, \forall y > 0, \tag{19b}$$

the latter of which proves that $q_3(y)$ is a convex function. We can further observe that $\lim_{y\rightarrow+\infty} q_3'(y) = 0$, which indicates $q_3'(y) \le 0, \forall y > 0$, i.e., the function $q_3(y)$ monotonically decreases with y. Since the objective function (18a) monotonically decreases with s_{ijk}^a, the optimal $s_{ijk}^{a}{}^*$ should be achieved at the boundary of Constraints (18b) for energy saving purpose. Hence, for each SC $j \in \mathcal{M}$, we can reduce the constraint in (18b) to

$$\sum_{k\in\{1,2\}} \sum_{i\in\mathcal{U}_{jk}} s_{ijk}^a = S^a. \tag{20}$$

Furthermore, for each SC $j \in \mathcal{M}$, the constraint in (18d) can be equivalently transformed to

$$s_{ijk}^a \ge s_{ijk}^{a,\min}, \forall k \in \{1,2\}, i \in \mathcal{U}_{jk}, \tag{21}$$

where

$$s_{ijk}^{a,\min} = \left(-\frac{BT_{ijk}' W_0\left(O_{ijk}\right)}{D_i \ln 2} - \frac{BN_0}{g_{ij} p_i^{\max}}\right)^{-1}, \tag{22}$$

and

$$O_{ijk} = -\frac{D_i N_0 \ln 2}{T_{ijk}' g_{ij} p_i^{\max}} 2^{-\frac{D_i N_0}{T_{ijk}' g_{ij} p_i^{\max}}}. \tag{23}$$

The reason is that the inequalities in Constraints (18d) can be expressed as the following form

$$2^{a_{ijk} y_{ijk}} \le c_{ij} y_{ijk} + 1, \tag{24}$$

where $y_{ijk} = \frac{1}{s_{ijk}^a}$, $c_{ij} = \frac{g_{ij} p_i^{\max}}{BN_0}$, and $a_{ijk} = \frac{D_i}{BT_{ijk}'}$. For the function of the form $2^{ay} = cy + 1$, its inverse can be proved to be derived based on the principal branch of the Lambert W function W_0 [47] as

$$y = -\frac{W_0\left(-\frac{a \ln 2}{c} 2^{-\frac{a}{c}}\right)}{a \ln 2} - \frac{1}{c}. \tag{25}$$

Further, since the left term in (18d) decreases with the increase of s_{ijk}^a, we obtain (21).

Accordingly, for each SC $j \in \mathcal{M}$, the RAN spectrum allocation problem with given $\{\boldsymbol{s^b}, \boldsymbol{f^e}, \boldsymbol{f^c}\}$ can be rewritten as

$$\min_{\boldsymbol{s_j^a}} \sum_{k\in\{1,2\}} \sum_{i\in\mathcal{U}_{jk}} \frac{s_{ijk}^a BN_0}{g_{ij}} \left(2^{\frac{D_i}{s_{ijk}^a BT_{ijk}'}} - 1\right) T_{ijk}' \tag{26a}$$

$$\text{s.t.} \quad (20), (21). \tag{26b}$$

Due to the convexity of objective function (26a) and all constraints, Problem (26) is a convex problem, which can be effectively solved via the Lagrangian multiplier method [45]. Defining η_j as the Lagrange multiplier vector associated with Constraint (20), the following theorem is given to obtain the optimal RAN spectrum allocation.

Theorem 2: The optimal RAN spectrum allocation of Problem (26) is

$$s_{ijk}^a = h_{ijk}^{-1}(-\eta_j)|_{s_{ijk}^{a,\min}}, \forall k \in \{1,2\}, i \in \mathcal{U}_{jk} \tag{27}$$

where $y_1|_{y_2} = \max\{y_1, y_2\}$, $h_{ijk}^{-1}(s_{ijk}^a)$ is the inverse function of $h_{ijk}(s_{ijk}^a)$,

$$h_{ijk}\left(s_{ijk}^a\right) = \frac{BN_0 T_{ijk}'}{g_{ij}} \left(2^{\frac{D_i}{BT_{ijk}' s_{ijk}^a}} \left(1 - \frac{D_i \ln 2}{BT_{ijk}' s_{ijk}^a}\right) - 1\right), \tag{28}$$

and η_j is the solution of

$$\sum_{k\in\{1,2\}} \sum_{i\in\mathcal{U}_{jk}} h_{ijk}^{-1}(-\eta_j)|_{s_{ijk}^{a,\min}} = S^a. \tag{29}$$

Proof: See Appendix C. ■

In fact, $h_{ij}(s_{ij}^a)$ is a monotonically increasing function of s_{ij}^a since

$$h_{ij}'\left(s_{ij}^a\right) = \frac{N_0 D_i \ln^2 2}{g_{ij}\left(s_{ijk}^a\right)^3} 2^{\frac{D_i}{BT_{ijk}' s_{ijk}^a}} > 0, \forall s_{ij}^a > 0. \tag{30}$$

Accordingly, its inverse function $h_{ij}^{-1}(s_{ij}^a)$, and further the left term of function (29) are monotonically increasing with s_{ij}^a. Hence, we can effectively reach the unique solution to η_j for equation (29) based on the bisection method [48].

E. Joint DWB Spectrum and Computation Capacity Allocation

In this subsection, we focus on the optimization of DWB spectrum and computation capacity for Problem (18) with fixed RAN spectrum. In such case, there exists a high degree of decoupling in the objective function and constraints. Specifically, Problem (18) can be decoupled into $|\mathcal{M}|+1$ subproblems, i.e., $|\mathcal{M}|$ subproblems each of which optimizes the computation capacity $\boldsymbol{f_j^e} = \{f_{ij}^e, i \in \mathcal{U}_{j1}\}$ of a SC $j \in \mathcal{M}$, and one subproblem jointly optimizing $\{\boldsymbol{s^b}, \boldsymbol{f^c}\}$ for UDs that offload the task to the CDC.

For each SC $j \in \mathcal{M}$, the ES computation capacity allocation problem is expressed as

$$\min_{\boldsymbol{f_j^e}} \sum_{i\in\mathcal{U}_{j1}} \frac{s_{ij1}^a BN_0}{g_{ij}} \left(2^{\frac{D_i}{s_{ij1}^a BT_{ij1}'}} - 1\right) T_{ij1}' \tag{31a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{U}_{j1}} f_{ij}^e \le f_j^{\max}, \tag{31b}$$

$$f_{ij}^e \ge f_{ij}^{e,\min}, \forall i \in \mathcal{U}_{j1}, \tag{31c}$$

where $f_{ij}^{e,\min} = \frac{F_i}{T_i - T_{ij1}^{\min}}$ and we have the subproblem jointly optimizing $\boldsymbol{\sigma} = \{\boldsymbol{s^b}, \boldsymbol{f^c}\}$ as

$$\min_{\boldsymbol{\sigma}} \sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{U}_{j2}} \frac{s_{ij2}^a BN_0}{g_{ij}} \left(2^{\frac{D_i}{s_{ij2}^a BT_{ij2}'}} - 1\right) T_{ij2}' \tag{32a}$$

$$\text{s.t.} \quad T_{ij2}' \ge T_{ij2}^{\min}, \forall j \in \mathcal{M}, i \in \mathcal{U}_{j2}, \tag{32b}$$

$$(18c), (18f). \tag{32c}$$

Constraints (31c) and (32b) are derived from (18d), where

$$T_{ijk}^{\min} = \frac{D_i}{s_{ijk}^a B \log_2\left(1 + \left(g_{ij} p_i^{\max}\right) \Big/ \left(s_{ijk}^a B N_0\right)\right)}. \tag{33}$$

Theorem 3: Problem (31) and (32) are convex optimization problems.

Proof: See Appendix D. ■

To solve these problems, we employ the interior method [45] and define the logarithmic barrier functions of Problems (31) and (32) as

$$\begin{aligned} z_j\left(\boldsymbol{f_j^e}\right) = & \sum_{i \in \mathcal{U}_{j1}} \frac{s_{ij1}^a B N_0}{g_{ij}} \left(2^{\frac{D_i}{s_{ij1}^a T'_{ij1}}} - 1\right) T'_{ij1} \\ & - \mu \log\left(f_j^{\max} - \sum_{i \in \mathcal{U}_{j1}} f_{ij}^e\right) \\ & - \mu \sum_{i \in \mathcal{U}_{j1}} \log\left(f_{ij}^e - \frac{F_i}{T_i - T_{ij1}^{\min}}\right), \end{aligned} \tag{34}$$

$$\begin{aligned} c(\boldsymbol{\sigma}) = & \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{U}_{j2}} \frac{s_{ij2}^a B N_0}{g_{ij}} \left(2^{\frac{D_i}{s_{ijk}^a B T'_{ij2}}} - 1\right) T'_{ij2} \\ & - \mu \log\left(S^b - \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{U}_{j2}} s_{ij}^b\right) \\ & - \mu \log\left(f_c^{\max} - \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{U}_{j2}} f_{ij}^c\right) \\ & - \mu \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{U}_j} \log\left(T'_{ij2} - T_{ij2}^{\min}\right), \end{aligned} \tag{35}$$

respectively, where μ is a barrier parameter. Then, Problem (31) and (32) can be transformed into the following unconstrained convex optimization problems

$$\min_{\boldsymbol{f_j^e}} z_j, \tag{36}$$

$$\min_{\boldsymbol{\sigma}} c, \tag{37}$$

respectively. According to Newton's method, the following iterative functions are provided to obtain the optimal solutions to $\boldsymbol{f_j^e}$ and $\boldsymbol{\sigma}$, respectively:

$$\boldsymbol{f_j^{e(k+1)}} = \boldsymbol{f_j^{e(k)}} - \frac{\nabla z\left(\boldsymbol{f_j^{e(k)}}\right)}{H\left(\boldsymbol{f_j^{e(k)}}\right)}, \tag{38}$$

$$\boldsymbol{\sigma}^{(k+1)} = \boldsymbol{\sigma}^{(k)} - \frac{\nabla z\left(\boldsymbol{\sigma}^{(k)}\right)}{H\left(\boldsymbol{\sigma}^{(k)}\right)}, \tag{39}$$

where $\nabla(\cdot)$ denotes the gradient matrix, $H(\cdot)$ denotes the Hessian matrix, and k is the loop index in Newton's method. The joint DWB spectrum and computation capacity allocation approach is detailed in Algorithm 2.

Algorithm 2 Joint DWB Spectrum and Computation Capacity Allocation

```
Input: $\boldsymbol{x}, \boldsymbol{s^a}$
Output: $\boldsymbol{f^{e*}}, \boldsymbol{f^{c*}}, \boldsymbol{s^{b*}}$
Set accuracy of the interior point method $\rho > 0$;
for $j \in \mathcal{M}$ do
    Initialize barrier parameter $\mu^{(1)} > 0$, iteration index $t = 0$, and generate an arbitrary feasible solution $\boldsymbol{f_j^{e(0)}}$;
    repeat
        $t \leftarrow t + 1$;
        Generate barrier function $z(\boldsymbol{f_j^e})$ with given $\mu^{(t)}$;
        Obtain the extreme point $\boldsymbol{f_j^{e*}}\left(\mu^{(t)}\right)$ with initial point $\boldsymbol{f_j^{e(t-1)}}$ according to (38);
        $\boldsymbol{f_j^{e(t)}} \leftarrow \boldsymbol{f_j^{e*}}\left(\mu^{(t)}\right), \mu^{(t+1)} \leftarrow Q\mu^{(t)}$;
    until $\left\|\boldsymbol{f_j^{e(t)}} - \boldsymbol{f_j^{e(t-1)}}\right\| \le \rho$;
end for
Initialize barrier parameter $\mu^{(1)} > 0$, iteration index $t = 0$, and generate an arbitrary feasible solution $\boldsymbol{\sigma}^{(0)}$;
repeat
    $t \leftarrow t + 1$;
    Generate barrier function $c(\boldsymbol{\sigma})$ with given $\mu^{(t)}$;
    Obtain the extreme point $\boldsymbol{\sigma}^*\left(\mu^{(t)}\right)$ with initial point $\boldsymbol{\sigma}^{(t-1)}$ according to (39);
    $\boldsymbol{\sigma}^{(t)} \leftarrow \boldsymbol{\sigma}^*\left(\mu^{(t)}\right), \mu^{(t+1)} \leftarrow Q\mu^{(t)}$;
until $\left\|\boldsymbol{\sigma}^{(t)} - \boldsymbol{\sigma}^{(t-1)}\right\| \le \rho$;
return the optimal allocation of DWB spectrum and computation capacity $\boldsymbol{f^{e*}} = \{\boldsymbol{f_j^{e*}}\}_{j \in \mathcal{M}}, \boldsymbol{\sigma}^* = \{\boldsymbol{f^{c*}}, \boldsymbol{s^{b*}}\}$.
```

F. Algorithm Implementation and Analysis

The Iterative Offloading decision and Resource allocation (IOR) method for solving Problem (12) is presented in Algorithm 3. Fig. 2 illustrates the layered structure of the proposed IOR algorithm. Starting with an initialized feasible solution, we iteratively optimize the offloading decision, RAN spectrum allocation, and joint DWB spectrum and computation capacity allocation. The obtained values of the above variables can uniquely determine the transmission power of UDs. In detail, at each τ-th iteration, we first obtain the offloading decision via sub-gradient based primal-dual method as shown in Algorithm 1. Second, the RAN spectrum allocation is optimized based on Lagrangian multiplier method and KKT conditions. Then, we employ the combination of interior-point method and Newton's method to solve the joint DWB spectrum and computation capacity allocation problem. Finally, the transmission power of UDs is uniquely obtained based on the offloading decision, RAN and DWB spectrum, and computation capacity.

1) Optimality Analysis: The optimal offloading decision can be obtained via the primal-dual method as shown in Algorithm 1. The optimal RAN spectrum allocation problem can be achieved according to (27). The combination of interior-point method and Newton's method as shown in Algorithm 2 can reach the optimal solution of the joint DWB spectrum and computation capacity allocation problem. The optimal transmission power can be calculated according to (17). Hence, Algorithm 3 achieves a sub-optimal solution to the original problem (12). Besides,

Algorithm 3 Iterative Offloading Decision and Resource Allocation (IOR)

```
Input: The set of UDs' tasks
Output: The optimal offloading decision and resource allocation $\left\{\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{s}^{a*}, \boldsymbol{s}^{b*}, \boldsymbol{f}^{e*}, \boldsymbol{f}^{c*}\right\}$
Set the iteration number $\tau = 0$, the maximal iteration number $\tau_{\max}$, the tolerance $\rho$
Obtain the initial solution as $\left(\boldsymbol{x}^{(0)}, \boldsymbol{p}^{(0)}, \boldsymbol{s}^{a(0)}, \boldsymbol{s}^{b(0)}, \boldsymbol{f}^{e(0)}, \boldsymbol{f}^{c(0)}\right)$;
Calculate the objective function value $V^{(0)}$ based on $\left(\boldsymbol{x}^{(0)}, \boldsymbol{p}^{(0)}, \boldsymbol{s}^{a(0)}, \boldsymbol{s}^{b(0)}, \boldsymbol{f}^{e(0)}, \boldsymbol{f}^{c(0)}\right)$;
repeat
    $\tau \leftarrow \tau + 1$;
    Obtain the optimal $\boldsymbol{x}^{(\tau)}$ of Problem (13) given $\left(\boldsymbol{s}^{a(\tau-1)}, \boldsymbol{s}^{b(\tau-1)}, \boldsymbol{f}^{e(\tau-1)}, \boldsymbol{f}^{c(\tau-1)}\right)$;
    Obtain the optimal $\boldsymbol{s}^{a(\tau)}$ of Problem (26) given $\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{b(\tau-1)}, \boldsymbol{f}^{e(\tau-1)}, \boldsymbol{f}^{c(\tau-1)}\right)$;
    Obtain the optimal $\boldsymbol{f}^{e(\tau)}$ of Problem (31) and $\left(\boldsymbol{s}^{b(\tau)}, \boldsymbol{f}^{c(\tau)}\right)$ of Problem (32) given $\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{a(\tau)}\right)$;
    Obtain the optimal $\boldsymbol{p}^{(\tau)}$ according to (17) given $\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{a(\tau)}, \boldsymbol{s}^{b(\tau)}, \boldsymbol{f}^{e(\tau)}, \boldsymbol{f}^{c(\tau)}\right)$;
    Calculate the objective function value $V^{(\tau)}$ based on $\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau)}, \boldsymbol{s}^{a(\tau)}, \boldsymbol{s}^{b(\tau)}, \boldsymbol{f}^{e(\tau)}, \boldsymbol{f}^{c(\tau)}\right)$;
until $|V^{(\tau)} - V^{(\tau-1)}| \leq \rho$ or $\tau > \tau_{\max}$.
```

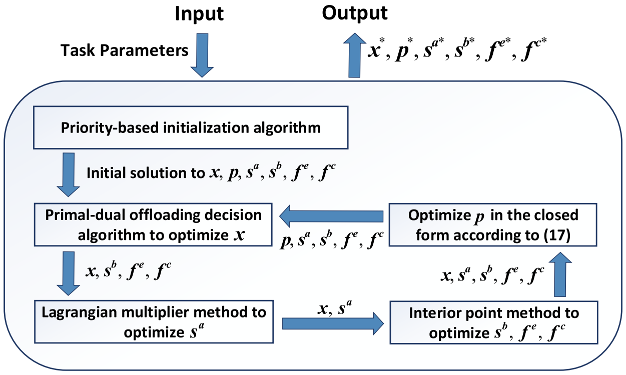
Input
Task Parameters
Output
$x^*, p^*, s^{a*}, s^{b*}, f^{e*}, f^{c*}$
Priority-based initialization algorithm
Initial solution to x, p, s^a, s^b, f^e, f^c
Primal-dual offloading decision algorithm to optimize x
p, s^a, s^b, f^e, f^c
Optimize p in the closed form according to (17)
x, s^b, f^e, f^c
x, s^a, s^b, f^e, f^c
Lagrangian multiplier method to optimize s^a
x, s^a
Interior point method to optimize s^b, f^e, f^c

Fig. 2. Layered structure of the proposed algorithm IOR.

simulations are conducted to demonstrate the near-optimal performance of the proposed method.

2) Convergence Analysis: To demonstrate the convergence of the proposed method, a theoretical proof is presented in Theorem 4.

Theorem 4: Algorithm 3 can converge to a solution within a finite number of steps.

Proof: See Appendix E. ■

3) Complexity Analysis: The complexity of Algorithm 3 is primarily related to the complexity of solving the three subproblems of $\boldsymbol{x}$, $\boldsymbol{s}^a$, and $\{\boldsymbol{s}^b, \boldsymbol{f}^e, \boldsymbol{f}^c\}$ respectively.

The process of solving offloading decision $\boldsymbol{x}$ for Problem (13) is presented in Algorithm 1. The complexity of solving offloading decision $\boldsymbol{x}$ is $\mathcal{O}(3|\mathcal{U}|)$ according to (14). The complexity of updating the corresponding dual variables $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \boldsymbol{\epsilon}, \boldsymbol{\zeta}, \varphi\}$ is $\mathcal{O}(2|\mathcal{U}| + 2|\mathcal{M}| + 2)$ according to (16a)-(16f). Supposing that Algorithm 1 needs I_1 iterations to converge, the total complexity of solving $\boldsymbol{x}$ is $\mathcal{O}(C_1) = \mathcal{O}(I_1(5|\mathcal{U}| + 2|\mathcal{M}| + 2))$.

The complexity of solving $\boldsymbol{s}_j^a$ for SC j in Problem (26) is $\mathcal{O}((|\mathcal{U}_{j1}| + |\mathcal{U}_{j2}|)\log(1/\theta_1)\log(1/\theta_2))$, where $\mathcal{O}(log(1/\theta_1))$ is the complexity of deriving the inverse function $h_{ijk}^{-1}(\cdot)$, and $\mathcal{O}(log(1/\theta_2))$ is the complexity of using the bisection method to solve (29). Thus, the complexity of obtaining the solution to $\boldsymbol{s}^a$ is $\mathcal{O}(C_2) = \mathcal{O}(\sum_{j\in\mathcal{M}}((|\mathcal{U}_{j1}| + |\mathcal{U}_{j2}|)\log(1/\theta_1)\log(1/\theta_2)))$.

Algorithm 2 optimizes $\boldsymbol{f}^e$ and $\{\boldsymbol{s}^b, \boldsymbol{f}^c\}$ separately. By using the interior method and Newton's method, the complexity of optimizing $\boldsymbol{f}^e$ and $\{\boldsymbol{s}^b, \boldsymbol{f}^c\}$ is $\mathcal{O}(\sum_{j\in\mathcal{M}}|\mathcal{U}_{j1}|^{3.5})$ and $\mathcal{O}((2\sum_{j\in\mathcal{M}}|\mathcal{U}_{j2}|)^{3.5})$ respectively. Consequently, the complexity of obtaining the solution to $\{\boldsymbol{s}^b, \boldsymbol{f}^e, \boldsymbol{f}^c\}$ is $\mathcal{O}(C_3) = \mathcal{O}(\sum_{j\in\mathcal{M}}|\mathcal{U}_{j1}|^{3.5} + (2\sum_{j\in\mathcal{M}}|\mathcal{U}_{j2}|)^{3.5})$.

Supposing that Algorithm 3 needs I_2 iterations to converge, the total complexity of Algorithm 3 is $\mathcal{O}(I_2(C_1 + C_2 + C_3))$.

G. Priority-Based Initialization Algorithm

It is challenging to use standard methods to reach an initial feasible solution of Problem (12) due to the non-convex feasible set caused by the strong coupling and mixed-integer nature of the offloading decision variables and resource allocation variables. In this subsection, we propose a priority-based algorithm to approach an initial solution of Problem (12), as shown in Algorithm 4.

In detail, for each SC $j \in \mathcal{M}$, we first filter the set of its associated UDs that can locally accomplish the task under the maximal latency constraint, i.e., $\mathcal{U}_j^l = \{i \in \mathcal{U}_j | \frac{F_i}{f_i^l} \leq T_i\}$. Then, the set of UDs that need the help of the ES or CDC to meet the task latency requirements can be denoted by $\mathcal{U}_j^o = \mathcal{U}_j \setminus \mathcal{U}_j^l$. To effectively reach a feasible solution, it is preferred to consider the different requirements of these UDs' tasks and fully utilize the communication and computation resources for task execution to reduce transmission and computation latency. Due to the different magnitude orders of the task requirements in terms of (D_i, F_i, T_i), we normalize the task requirements to remove their related units as follows:

$$\tilde{D}_i = \frac{D_i}{D^{\max}},\ \tilde{F}_i = \frac{F_i}{F^{\max}},\ \tilde{T}_i = \frac{T_i}{T^{\max}} \tag{40}$$

where $D^{\max}$, $F^{\max}$ and $T^{\max}$ indicate the maximum value of input size, computation cycles, and deadline among all tasks, respectively.

Since each UD $i \in \mathcal{U}_j^o$ has to first upload its task to SC j whether for task execution in the ES or the CDC, we first define the weight for the spectrum allocation as

$$W_i^s = \frac{\tilde{D}_i}{\tilde{T}_i}, \tag{41}$$

Algorithm 4 Priority-Based Initialization Algorithm

```
Initialize $CR \leftarrow 0, ER_j \leftarrow 0, \mathcal{U}_{j1}, \mathcal{U}_{j2}, \mathcal{U}^c_j \leftarrow \phi, \forall j \in \mathcal{M}$
for $j \in \mathcal{M}$ do
    Resort set $\mathcal{U}^o_j$ in descending order of the priority $PR_{ij}$ as $\bar{\mathcal{U}}^o_j$;
    for $i \in \bar{\mathcal{U}}^o_j$ do
        Calculate the value of $s^a_{ij}$ according to (42);
        Set the value of $p_{ij}$ according to (43);
        Calculate the value of $f^e_{ij}$ according to (44);
        if $f^e_{ij} + ER_j \leq f^{\max}_j$ then
            $x_{ij1} \leftarrow 1, s^a_{ij1} \leftarrow s^a_{ij}, p_{ij1} \leftarrow p_{ij}, ER_j \leftarrow ER_j + f^e_{ij}, U_{j1} \leftarrow U_{j1} \cup \{i\}$;
        else
            $\mathcal{U}^c_j \leftarrow \mathcal{U}^c_j \cup i$
        end if
    end for
end for
for $j \in \mathcal{M}$ do
    for $i \in \mathcal{U}^c_j$ do
        Calculate the value of $s^b_{ij}$ according to (46);
        Calculate the value of $f^c_{ij}$ according to (47);
    end for
    if $f^c_{ij} + CR \leq f^{\max}_c$ then
        $x_{ij2} \leftarrow 1, s^a_{ij2} \leftarrow s^a_{ij}, p_{ij2} \leftarrow p_{ij}, CR \leftarrow CR + f^c_{ij}, U_{j2} \leftarrow U_{j2} \cup \{i\}$;
    end if
end for
```

and assign the RAN spectrum to UD $i \in \mathcal{U}^o_j$ according to

$$s^a_{ij} = \frac{W^s_i}{\sum_{k\in\mathcal{U}^o_j} W^s_k} S^a, \forall i \in \mathcal{U}^o_j, \tag{42}$$

which implies the task with lager input size and tighter deadline is assigned more RAN spectrum resources. Each UD $i \in \mathcal{U}^o_j$ uses its maximum transmission power to upload its task as

$$p_{ij} = p^{\max}_i. \tag{43}$$

Note that we do not distinguish between s^a_{ij1} and s^a_{ij2}, p_{ij1} and p_{ij2} here as the task offloading destination remains undetermined.

Then, a available ES computation capacity allocation for UD $i \in \mathcal{U}^o_j$ is given by

$$f^e_{ij} = f^{e,\min}_{ij}, \forall j \in \mathcal{M}, k \in \{1, 2\}, i \in \mathcal{U}_{jk}, \tag{44}$$

based on (31c). However, the UD can only offload its task to its associated ES if the maximum computation capacity constraint of this ES can be guaranteed. Therefore, we define a priority for selecting the UD to offload the task to the corresponding ES among the UDs served by SC j:

$$PR_{ij} = \frac{\tilde{D}_i}{\tilde{T}_i \tilde{F}_i}, \forall i \in \mathcal{U}^o_j. \tag{45}$$

Considering the ES's proximity to UDs and limited computation capacity, the task with lager input size, lower computation workload, and tighter deadline is assigned a higher priority to be offloaded to the ES. When the total amount of required computation capacity ER_j exceeds the maximum computation capacity $f^{\max}_j$ of the ES, each remaining UD $i \in \mathcal{U}^c_j = \mathcal{U}^o_j \setminus \mathcal{U}_{j1}$ have to offload its task to the CDC. For these UDs, we assign the DWB spectrum according to

$$s^b_{ij} = \frac{W^s_i}{\sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{U}^c_j} W^s_k} S^b, \forall j \in \mathcal{N}, i \in \mathcal{U}^c_j. \tag{46}$$

which is similar to the RAN spectrum allocation.

Then, a available CDC computation capacity allocation for UD $i \in \mathcal{U}^c_j$ is given by

$$f^c_{ij} = f^{c,\min}_{ij}, \forall j \in \mathcal{M}, i \in \mathcal{U}^c_j, \tag{47}$$

where $f^{c,\min}_{ij}$ can be obtained by solving $T'_{ij2} = T^{\min}_{ij2}$ derived from (32b).

Due to the necessity of satisfying the latency constraints (12b), we have the feasibility condition for Problem (12) as

$$\sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{U}^c_j} f^{c,\min}_{ij} \leq f^{\max}_c. \tag{48}$$

According to (48), each UD $i \in \mathcal{U}^c_j, \forall j \in \mathcal{M}$ can only offload its task to the CDC on the premise of guaranteeing the computation capacity constraint of the CDC, i.e., the total amount of required computation capacity *CR* is no bigger than $f^{\max}_c$. When the feasibility condition in (48) can not be satisfied, it is infeasible to find a solution that satisfies each UD's task latency requirements. In this paper, we assume that with the assistance of the CDC, the computation capacity of each SC can satisfy the requirements of its serving UDs [7], [16]. Therefore, the decision rules in Algorithm 4 guarantee the obtained initial solution is a feasible solution to Problem (12).

After analyzing Algorithm 4, we can obtain that the computational complexity of achieving a initial solution of joint offloading decision and resource allocation is $\mathcal{O}(\sum_{j\in\mathcal{M}}(|\mathcal{U}_j| + |\mathcal{U}^o_j| \log |\mathcal{U}^o_j| + |\mathcal{U}^o_j| + |\mathcal{U}^c_j|))$, where the term $|\mathcal{U}^o_j| \log |\mathcal{U}^o_j|$ is the complexity of reordering set $\mathcal{U}^o_j$ based on the rapid sorting method, and the other three terms $|\mathcal{U}_j|$, $|\mathcal{U}^o_j|$ and $|\mathcal{U}^c_j|$ denote the complexity of determining the tasks to be processed by the UD, the ES and the CDC, respectively.

V. Numerical Results

In this section, we present numerical results obtained from various simulations implemented in MATLAB software on a server with Intel i7 processor. The network topology consists of 5 SCs, one of which is served as the gateway which is connected to a CDC, and 20 UDs randomly distributed among these SCs. The channel gain follows an exponential distribution with mean 1 and the default values of all remaining system parameters are summarized in Table I, and they are set based on the works in [7], [16].

Depending on different task characteristics, we consider two types of tasks, i.e., latency-sensitive and latency-tolerant tasks. The parameters of these two type of tasks are listed in Table II. As shown in Table II, compared with latency-sensitive tasks, latency-tolerant tasks have heavier computation

TABLE I
SIMULATION PARAMETERS

Parameter	Value
The number of UDs ($\lvert\mathcal{U}\rvert$)	20
The number of SCs ($\lvert\mathcal{M}\rvert$)	5
Maximum computation capacity of UDs f_i^l	5×10^9 cycles/s
Maximum computation capacity of ESs $f_j^{\max}$	9×10^{10} cycles/s
Maximum computation capacity of the CDC $f_{\max}^c$	9×10^{12} cycles/s
Maximum transmission power of UDs	0.2 W
Maximum transmission power of ESs	4 W
Total bandwidth of the RAN spectrum	20 MHz
Total bandwidth of the DWB spectrum	20 MHz
Total bandwidth of optical fiber	1 Gbps
Uplink Propagation delay χ	50 ms
Noise power N_0	-176 dbm/Hz
Pathloss exponent ν	3.7
Coefficient κ	10^{-29}
Desired accuracy level as convergence criterion (ρ)	10^{-3}

TABLE II
TASK SETTINGS

Type	Data size (Mbits)	Computation resource requirements (cycles)	Maximum latency (s)
Latency-sensitive	1	6×10^8	0.25
Latency-tolerant	1.5	6×10^9	1

loads and looser latency requirements. Unless specified otherwise, the ratio of latency-sensitive to latency-tolerant tasks is always 5:5.

In our simulations, the performance of the proposed method IOR is compared with the other four offloading methods as follows:

EXH: Due to the high complexity of obtaining the optimal offloading decision and resource allocation via exhaustive search, this method employs the exhaustive search to reach the optimal offloading decision and uses the proposed method to optimize the joint communication and computation resource allocation.

Random: Each UD chooses to execute its task on the local device, the ES or in the CDC randomly.

JTORAA [7]: A joint task offloading and resource allocation method where each task is processed on the UD or the ES.

ISA-CCO [14]: A collaborative cloud-edge-end task offloading scheme with fixed spectrum resources.

Fig. 3 illustrates the convergence behavior of the proposed method IOR with different number of UDs $\lvert\mathcal{U}\rvert$. Here, the horizontal axis denotes the number of iterations τ given in Algorithm 3. The UD energy consumption at the beginning is relatively high since the initial solution forces each UD to conduct the task locally if the task deadline can be met and incurs high local computation energy consumption. After iterative optimization of offloading decision, spectrum, and computation resources, the number of tasks conducted locally can be reduced and thus the UD energy can be saved. It can be seen that the convergence speed of IOR method gets slightly slower with the increase of $\lvert\mathcal{U}\rvert$ as two iterations are required to converge when $\lvert\mathcal{U}\rvert = 10$ and four iterations when $\lvert\mathcal{U}\rvert = 30$. During the offloading decision process, each UD is only provided three choices for executing the task, i.e., the local device, the associated ES, and the CDC, which can significantly restrict the solution space and improve the efficiency of achieving a high-quality solution. In addition, the proposed IOR method can reach the optimal solution for each subproblem in terms of optimizing the allocation of transmission power, RAN Spectrum, DWB spectrum, and computation resources, and thus can improve the convergence speed. Therefore, the proposed method converges rapidly in general as shown in Fig. 3, which implies the effectiveness of the proposed method.

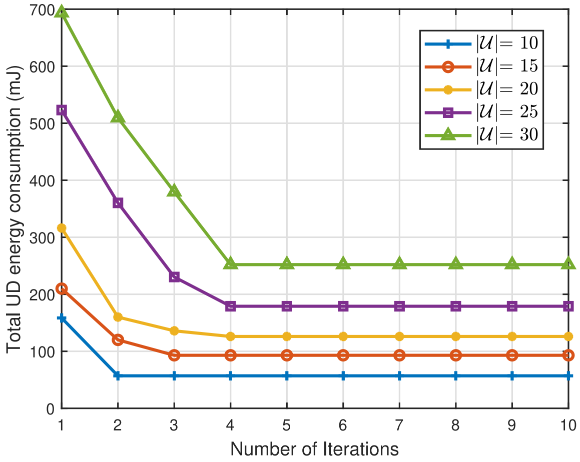

Fig. 3. Convergence behavior of the proposed method with different number of UDs.

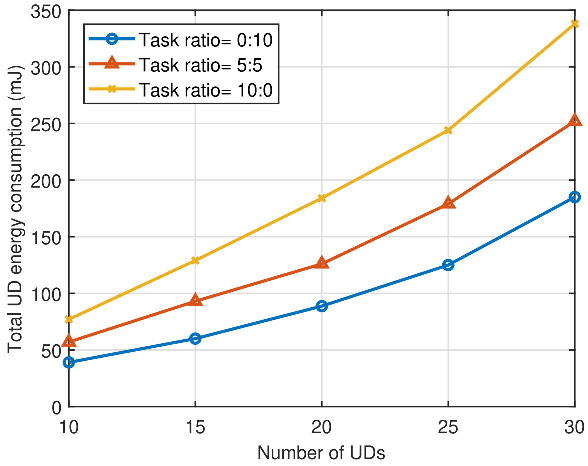

Fig. 4. Total UD energy consumption versus the number of UDs considering different task ratios.

In Fig. 4, we depict the total UD energy consumption versus the number of UDs with different ratios of latency-sensitive to latency-tolerant tasks. It is seen that increasing the number of UDs leads to a higher UDs' energy consumption. That is expected due to growing UDs generates a larger number of computation tasks that need to be processed simultaneously. Furthermore, the UDs' energy consumption rises as the ratio of latency-sensitive to latency-tolerant tasks increases. Specifically, the energy consumption with task ratio = 0:10 is the lowest and with task ratio = 10:0 is the highest. Latency-sensitive tasks with stringent low-latency requirements impose strict limits on the task transmission time. Therefore, due to

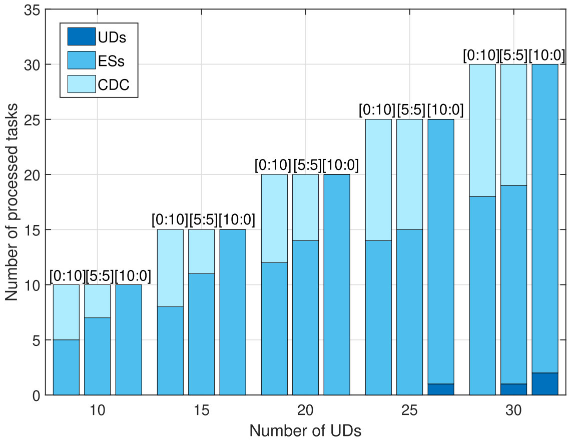

Fig. 5. Number of tasks processed at each layer versus the number of UDs considering different task ratios.

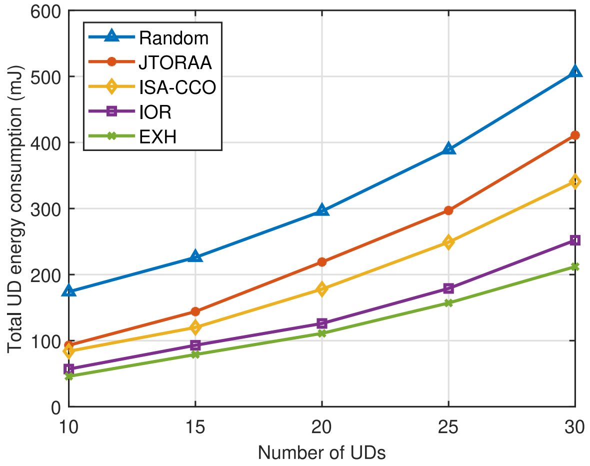

Fig. 6. Total UD energy consumption versus the number of UDs considering different methods.

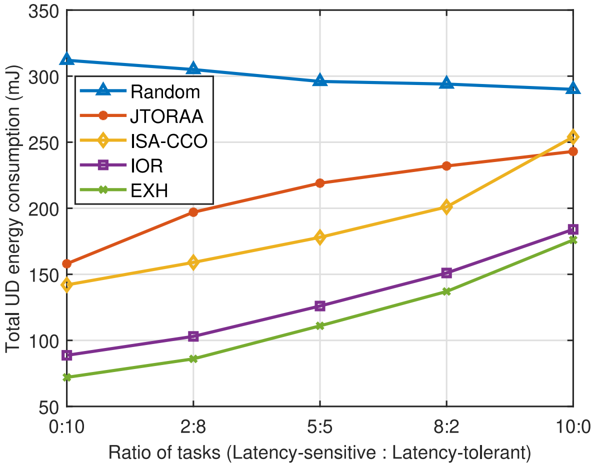

Fig. 7. Total UD energy consumption versus the ratio of tasks considering different methods.

the long transmission distance to the CDC, latency-sensitive tasks are preferentially executed locally or on ESs rather than in the CDC, which can be shown in Fig. 5. However, local task execution can incur high computational energy and offloading latency-sensitive tasks to ESs consumes high upload energy since UDs have to increase the transmission power to satisfy the latency bound.

Fig. 5 plots the detailed number of tasks processed on each layer including UDs, ESs, and the CDC. We can observe that increasing the ratio of latency-sensitive to latency-tolerant tasks reduces the proportion of tasks to be offloaded to the CDC (e.g., from 5/10 with ratio=0:10 to 0/10 with ratio=10:0 when there are 10 UDs). This is because the long transmission latency to the CDC can violate the low-latency bound of latency-sensitive tasks. Moreover, as the number of UDs increases, the rising resource demands intensify the competition for the limited spectrum and computation resources of ESs, causing some UDs have to execute their task locally (e.g., 2 tasks are processed locally with task ratio = 10:0 when there are 30 UDs). The proportion of tasks processed in the CDC can be noticeable with the increasing number of latency-tolerant tasks, which indicates the advantage of utilizing the CDC with powerful computing power.

To further verify the necessity and the performance of our proposed method, we compare the total energy consumption achieved by our proposed method with the other four methods as outlined above. The total UD energy consumption of different methods versus the number of UDs is depicted in Fig. 6. From this figure, one can easily observe that IOR can achieve lower energy consumption than the others except the EXH method. Both IOR and ISA-CCO outperform the JTORAA method, since the CDC can help handle the increasing computational requests and thus relieve computation stress for UDs. Moreover, we can notice a wider gap of energy consumption between IOR and ISA-CCO as the number of UDs rises, which indicates the importance of optimizing the spectrum resources, especially for heavy task workload. The energy consumption of IOR keeps close to that of EXH especially for small number of UDs, which demonstrates that the proposed IOR method can reach the near-optimal solution.

Fig. 7 provides the energy consumption of different methods versus the ratio of latency-sensitive to latency-tolerant tasks. Random method randomly chooses tasks to be processed locally, making the local computation the leading cause of energy consumption. Note that compared with latency-tolerant tasks, latency-sensitive tasks require less computation and quicker transmission, where the computation energy is always of a higher magnitude. Therefore, the energy consumption of Random approach decreases slightly with the ratio of latency-sensitive to latency-tolerant tasks as shown in Fig. 7. Different from Random method, the other four methods offload as many tasks as possible to external servers to save local computation energy, which means their energy consumption comes mainly from task data uploading. Hence, the energy consumption of these four methods increases with the ratio of latency-sensitive to latency-tolerant tasks in Fig. 7. Despite the different trend among these methods, IOR always has advantageously lower energy consumption than the others except EXH, especially when the proportion of latency-tolerant tasks is high.

Fig. 8 plots the total UD energy consumption of different methods versus the number of SCs. As depicted in Fig. 8, the total energy consumption decreases gradually as the number of SCs increases. This is because the increase of SCs means

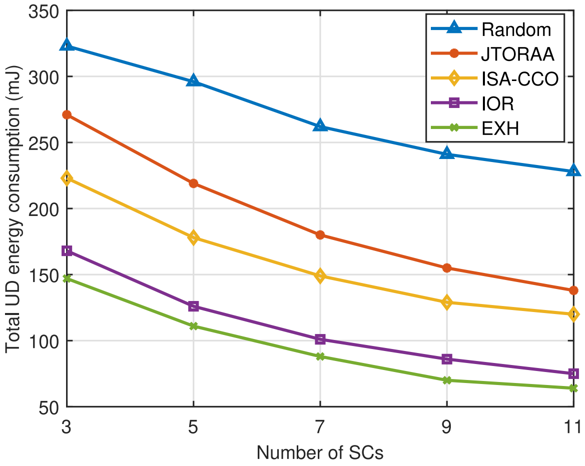

Fig. 8. Total UD energy consumption versus the number of SCs considering different methods.

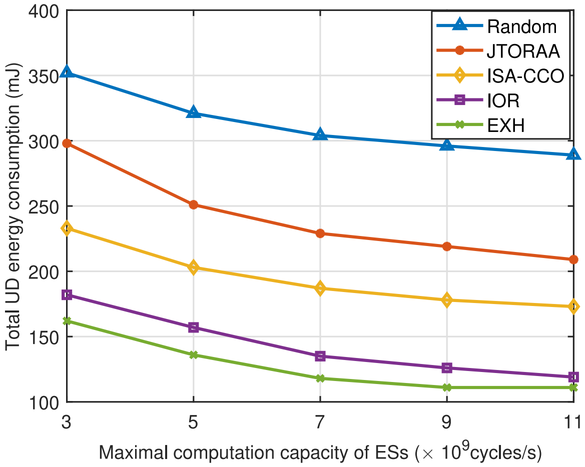

Fig. 9. Total UD energy consumption versus the maximal computation capacity of ESs of different methods.

more available ESs and weaker competition for RAN spectrum and computation resources among UDs, and thus allows more tasks to be offloaded with a lower transmission power. We can also conclude that the energy advantage of leveraging the CDC becomes less obvious with the increase of the number of SCs as the gap between JTORAA method and ISA-COO and IOR methods is narrowing in Fig. 8. The reason is that the computation resources at the edge tend to be saturated and sufficient to serve the computation requests from UDs.

Fig. 9 reveals the energy consumption of different methods versus the maximal computation capacity of SCs. We can observe that increasing the computation capacity of SCs leads to lower UDs' energy consumption, which can be explained by the reason similar to Fig. 8. In addition, from the comparison of Fig. 8 and Fig. 9, it can be shown that the total energy consumption when $|\mathcal{M}| = 3, f_j^{\max} = 9 \times 10^{10}$ is higher than when $|\mathcal{M}| = 5, f_j^{\max} = 5 \times 10^{10}$, even the total amount of computation capacity is larger in the former case. Compared with increasing the ESs' computation capacity, deploying more SCs can weaken the competition of RAN spectrum resources between UDs due to the spectrum reuse of each SC, thereby reducing the energy consumption during task uploading.

VI. Conclusion

In this paper, we investigated the energy-aware collaborative computation offloading combining MEC and CC, where tasks can be processed locally, on the ES, or in the CDC. We designed a cloud-edge-end architecture for collaborative computation offloading over small cell networks with DWB. To minimize the total energy consumption of the delay-constrained UDs, we jointly optimized the offloading decision, transmission power, spectrum and computation resources. The formulated optimization problem is decoupled into three subproblems to remove the tight coupling between different variables. Depending on the structure characteristics of these subproblems, we applied a few different convex optimization techniques to obtain the optimal solution to each subproblem and proposed an iterative method to reach the high-quality solution to the original problem. Numerical results demonstrated that the proposed method achieves better performance than existing methods in terms of the total UDs' energy consumption. For future work, the energy consumption optimization for the whole network architecture, which considers the energy consumed by the UDs, ESs and the CDC, is a practical and urgent problem.

Appendix A
Proof of Theorem 1

We can derive the partial Lagrangian function of Problem (13) as

$$\begin{aligned}
\mathcal{L}_1(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi) &= \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{U}_j} E_{ij} \\
&+ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{U}_j} \alpha_{ij}\left(x_{ij0}T_i^l + x_{ij1}T_i^e + x_{ij2}T_i^c - T_i\right) \\
&+ \sum_{j\in\mathcal{M}} \beta_j\left(\sum_{i\in\mathcal{U}_j}\sum_{k\in\{1,2\}} x_{ijk}s_{ijk}^a - S^a\right) \\
&+ \gamma\left(\sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{U}_j} x_{ij2}s_{ij}^b - S^b\right) \\
&+ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{U}_j} \epsilon_{ij}\left(\sum_{k\in\{1,2\}} x_{ijk}p_{ijk} - p_i^{\max}\right) \\
&+ \sum_{j\in\mathcal{M}} \zeta_j\left(\sum_{i\in\mathcal{U}_j} x_{ij1}f_{ij}^e - f_j^{\max}\right) \\
&+ \varphi\left(\sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{U}_j} x_{ij2}f_{ij}^c - f_c^{\max}\right).
\end{aligned} \tag{49}$$

Then, the dual function of Problem (13) can be given as

$$\begin{aligned}
&g(\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi) \\
&= \begin{cases} \min\limits_{\boldsymbol{x}} \mathcal{L}_1(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi) \\ \text{s.t.}\ \ \sum_{k\in\mathcal{K}} x_{ijk} = 1, \forall j\in\mathcal{M}, i\in\mathcal{U}_j, \\ \quad 0 \le x_{ijk} \le 1, \forall j\in\mathcal{M}, i\in\mathcal{U}_j, k\in\mathcal{K}, \end{cases}
\end{aligned} \tag{50}$$

and the dual problem of Problem (13) can be derived as

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi} \quad & g(\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi) \\ \text{s.t.} \quad & \alpha_{ij},\beta_j,\gamma,\epsilon_{ij},\zeta_j,\varphi \geq 0, \forall j \in \mathcal{M}, i \in \mathcal{U}_j \end{aligned} \tag{51}$$

Given any fixed $\{\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi\}$, we can obtain $g(\boldsymbol{\alpha},\boldsymbol{\beta},\gamma,\delta,\boldsymbol{\epsilon},\boldsymbol{\zeta},\varphi)$ by solving Problem (50). In fact, objective function (50) is a linear combination of x_{ijk}. Therefore, the optimal task assignment decision $\boldsymbol{x}^*$ can be expressed as (14), i.e., the offloading decision variable corresponding to the smallest e_{ijk} is set to 1 for each UD.

Appendix B
Proof of Lemma 1

According to Constraints (12b), we have

$$p_{ijk} \geq \frac{s^a_{ijk} B N_0}{g_{ij}} \left(2^{\frac{D_i}{s^a_{ijk} B T'_{ijk}}} - 1\right), \forall i \in \mathcal{U}_{jk}, k \in \{1,2\}. \tag{52}$$

Notice that the term with respect to p_{ijk} in objective function (12a) is function E^{at}_{ijk}. Recall

$$E^{at}_{ijk} = \frac{p_{ijk} D_i}{s^a_{ijk} B \log_2\left(1 + \left(p_{ijk} g_{ij}\right) \Big/ \left(s^a_{ijk} B N_0\right)\right)}, \tag{53}$$

which is an increasing function of p_{ijk}. To prove this, we define function $q_1(y) = \frac{y}{\ln(1+y)}, y > 0$, and we have

$$q_1'(y) = \frac{1}{\ln(1+y)} - \frac{y}{(1+y)\ln^2(1+y)}, \tag{54a}$$

$$q_1''(y) = \frac{2y - (y+2)\ln(1+y)}{(1+y)^2 \ln^3(1+y)}. \tag{54b}$$

Define $q_2(y) = 2y - (y+2)\ln(1+y)$, and we have

$$q_2'(y) = 1 - \ln(1+y) - \frac{1}{1+y}, \tag{55a}$$

$$q_2''(y) = -\frac{y}{(y+1)^2} < 0, \forall y > 0. \tag{55b}$$

Since $q_2'(0) = 0$, we can obtain $q_2'(y) < 0, \forall y > 0$. Then based on $q_2(0) = 0$, we can deduce $q_2(y) < 0, \forall y > 0$. Hence, we have $q_1''(y) < 0, \forall y > 0$. Since $\lim_{y \to +\infty} q_1'(y) = 0$, we have $q_1'(y) > 0, \forall y > 0$, which indicates $q_1(y)$ is monotonically increasing when $y > 0$. Analogically, objective function (12a) monotonically increases with p_{ijk}. Thus, the equality of the power constraint holds for (52) to save the UD energy, i.e., the optimal p^*_{ijk} can be calculated as (17).

Appendix C
Proof of Theorem 2

The Lagrangian dual of Problem (26) can be formulated as

$$\begin{aligned} \mathcal{L}_2 = & \sum_{k \in \{1,2\}} \sum_{i \in \mathcal{U}_{jk}} \frac{s^a_{ijk} B N_0}{g_{ij}} \left(2^{\frac{D_i}{s^a_{ijk} B T'_{ijk}}} - 1\right) T'_{ijk} \\ & + \eta_j \left(\sum_{k \in \{1,2\}} \sum_{i \in \mathcal{U}_{jk}} s^a_{ijk} - S^a\right). \end{aligned} \tag{56}$$

The Karush-Kuhn-Tucker (KKT) conditions of Problem (26) are as follows:

$$\nabla_{s^a_{ijk}} \mathcal{L}_2 = h_{ijk}\left(s^a_{ijk}\right) + \eta_j, \forall k \in \{1,2\}, i \in \mathcal{U}_{jk}, \tag{57a}$$

$$\sum_{k \in \{1,2\}} \sum_{i \in \mathcal{U}_{jk}} s^a_{ijk} = S^a, \tag{57b}$$

$$s^a_{ijk} \geq s^{a,\min}_{ijk}, \forall k \in \{1,2\}, i \in \mathcal{U}_{jk}, \tag{57c}$$

where $h_{ijk}(s^a_{ijk})$ is given in (28). Setting $\nabla_{s^a_{ijk}} \mathcal{L}_2 = 0$ yields

$$s^a_{ijk} = h^{-1}_{ijk}(-\eta_j), \forall k \in \{1,2\}, i \in \mathcal{U}_{jk}. \tag{58}$$

Due to the existence of (57c), we derive (27). Then, we derive (29) by combining (57b) and (27).

Appendix D
Proof of Theorem 3

The objective functions (31a) and (32a) can be expressed as a sum of composite functions of the following form:

$$l_{ijk}\left(T'_{ijk}\right), k \in \{1,2\}, \tag{59}$$

where

$$l_{ijk}(y) = s^a_{ijk} N_0 \left(2^{\frac{D_i}{s^a_{ijk} y}} - 1\right) y, \tag{60a}$$

$$T'_{ij1}\left(f^e_{ij}\right) = T_i - \frac{F_i}{f^e_{ij}}, \tag{60b}$$

$$\begin{aligned} & T'_{ij2}\left(f^c_{ij}, s^b_{ij}\right) \\ & = \begin{cases} T_i - \frac{D_i}{C} - \chi - \frac{F_i}{f^c_{ij}}, \text{if } j = 0, \\ T_i - \frac{D_i}{s^b_{ij} \log_2\left(1 + \frac{P_j}{s^b_{ij} N_0}\right)} - \frac{D_i}{C} - \chi - \frac{F_i}{f^c_{ij}}, \text{if } j \neq 0. \end{cases} \end{aligned} \tag{60c}$$

As proved in the beginning of Section IV-D, $l_{ijk}(y)$ is a convex and monotonically decreasing function of y when $y > 0$. Since T'_{ij1} is a concave function of f^e_{ij} and T'_{ij2} is also a concave function of $\{f^c_{ij}, s^b_{ij}\}$, the composite function $l_{ijk}(T'_{ijk}), k \in \{1,2\}$ is a convex function according to [45]. Besides, Constraints (31b)-(31c) and (32b)-(32c) are convex. Therefore, Problem (31) and (32) are convex.

Appendix E
Proof of Theorem 4

Denote the objective value of Problem (12) under the solution $(\boldsymbol{x},\boldsymbol{p},\boldsymbol{s^a},\boldsymbol{s^b},\boldsymbol{f^e},\boldsymbol{f^c})$ by $E(\boldsymbol{x},\boldsymbol{p},\boldsymbol{s^a},\boldsymbol{s^b},\boldsymbol{f^e},\boldsymbol{f^c})$. At the t-th iteration, we first obtain $\boldsymbol{x}^{(\tau)}$ by solving Problem (13) with fixed $(\boldsymbol{p}^{(\tau-1)},\boldsymbol{s^a}^{(\tau-1)},\boldsymbol{s^b}^{(\tau-1)},\boldsymbol{f^e}^{(\tau-1)},\boldsymbol{f^c}^{(\tau-1)})$, and thus we have

$$\begin{aligned} & E\left(\boldsymbol{x}^{(\tau-1)},\boldsymbol{p}^{(\tau-1)},\boldsymbol{s^a}^{(\tau-1)},\boldsymbol{s^b}^{(\tau-1)},\boldsymbol{f^e}^{(\tau-1)},\boldsymbol{f^c}^{(\tau-1)}\right) \\ & \geq E\left(\boldsymbol{x}^{(\tau)},\boldsymbol{p}^{(\tau-1)},\boldsymbol{s^a}^{(\tau-1)},\boldsymbol{s^b}^{(\tau-1)},\boldsymbol{f^e}^{(\tau-1)},\boldsymbol{f^c}^{(\tau-1)}\right). \end{aligned}$$

Secondly, given $(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{e}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{c}(\tau-1)})$, the $\boldsymbol{s}^{\boldsymbol{a}(\tau)}$ can be obtained by solving Problem (26), which guarantees the following equation

$$E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau-1)}, \boldsymbol{s}^{\boldsymbol{a}(\tau-1)}, \boldsymbol{s}^{\boldsymbol{b}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{e}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{c}(\tau-1)}\right) \geq E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau-1)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{e}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{c}(\tau-1)}\right).$$

Then, with fixed $(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)})$, we approach the solutions of Problem (31) and Problem (32), i.e., $\boldsymbol{f}^{\boldsymbol{e}(\tau)}$ and $\{\boldsymbol{s}^{\boldsymbol{b}(\tau)}, \boldsymbol{f}^{\boldsymbol{c}(\tau)}\}$. Therefore, we have

$$E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau-1)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{e}(\tau-1)}, \boldsymbol{f}^{\boldsymbol{c}(\tau-1)}\right) \geq E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau-1)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau)}, \boldsymbol{f}^{\boldsymbol{e}(\tau)}, \boldsymbol{f}^{\boldsymbol{c}(\tau)}\right).$$

Finally, $\boldsymbol{p}^{(\tau)}$ can be uniquely determined based on $(\boldsymbol{x}^{(\tau)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau)}, \boldsymbol{f}^{\boldsymbol{e}(\tau)}, \boldsymbol{f}^{\boldsymbol{c}(\tau)})$, which does not change the objective value, i.e.,

$$E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau-1)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau)}, \boldsymbol{f}^{\boldsymbol{e}(\tau)}, \boldsymbol{f}^{\boldsymbol{c}(\tau)}\right) = E\left(\boldsymbol{x}^{(\tau)}, \boldsymbol{p}^{(\tau)}, \boldsymbol{s}^{\boldsymbol{a}(\tau)}, \boldsymbol{s}^{\boldsymbol{b}(\tau)}, \boldsymbol{f}^{\boldsymbol{e}(\tau)}, \boldsymbol{f}^{\boldsymbol{c}(\tau)}\right).$$

The above equations demonstrate that the value of objective function (12a) is non-increasing when sequence $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{s}^{\boldsymbol{a}}, \boldsymbol{s}^{\boldsymbol{b}}, \boldsymbol{f}^{\boldsymbol{e}}, \boldsymbol{f}^{\boldsymbol{c}})$ is updated after each iteration. Furthermore, the value of objective function in (12a) always keeps positive. Hence, Algorithm 3 can converge in a finite number of iterations.

References

[1] S. T. Arzo, C. Naiga, F. Granelli, R. Bassoli, M. Devetsikiotis, and F. H. P. Fitzek, "A theoretical discussion and survey of network automation for IoT: Challenges and opportunity," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12021–12045, Aug. 2021.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[3] J. Hu, K. Li, C. Li, J. Chen, and K. Li, "Coalition formation for deadline-constrained resource procurement in cloud computing," *J. Parallel Distrib. Comput.*, vol. 149, pp. 1–12, Mar. 2021.

[4] C. R. Panigrahi, J. L. Sarkar, B. Pati, R. Buyya, P. Mohapatra, and A. Majumder, "Mobile cloud computing and wireless sensor networks: A review integration architecture and future directions," *IET Netw.*, vol. 10, no. 4, pp. 141–161, 2021.

[5] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020.

[6] S. Yue et al., "TODG: Distributed task offloading with delay guarantees for edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 7, pp. 1650–1665, Jul. 2022.

[7] C. Xu, G. Zheng, and X. Zhao, "Energy-minimization task offloading and resource allocation for mobile edge computing in NOMA heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16001–16016, Dec. 2020.

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

[9] L. Tang and H. Hu, "Computation offloading and resource allocation for the Internet of Things in energy-constrained MEC-enabled HetNets," *IEEE Access*, vol. 8, pp. 47509–47521, 2020.

[10] H. Yuan and M. Zhou, "Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1277–1287, Jul. 2021.

[11] Z. Hong, W. Chen, H. Huang, S. Guo, and Z. Zheng, "Multi-hop cooperative computation offloading for industrial IoT-edge-cloud computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 12, pp. 2759–2774, Dec. 2019.

[12] Y. Ding, K. Li, C. Liu, and K. Li, "A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 6, pp. 1503–1519, Jun. 2022.

[13] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[14] C. He, R. Wang, and Z. Tan, "Energy-aware collaborative computation offloading over mobile edge computation empowered fiber-wireless access networks," *IEEE Access*, vol. 8, pp. 24662–24674, 2020.

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

[16] C. Kai, H. Zhou, Y. Yi, and W. Huang, "Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 624–634, Jun. 2021.

[17] C. Li, C. Wang, and Y. Luo, "An efficient scheduling optimization strategy for improving consistency maintenance in edge cloud environment," *J. Supercomput.*, vol. 76, pp. 6941–6968, Jan. 2020.

[18] X. Chen, J. Zhang, B. Lin, Z. Chen, K. Wolter, and G. Min, "Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 683–697, Mar. 2022.

[19] Z. Gao, W. Hao, Z. Han, and S. Yang, "Q-learning-based task offloading and resources optimization for a collaborative computing system," *IEEE Access*, vol. 8, pp. 149011–149024, 2020.

[20] G. Kalfas et al., "Next generation fiber-wireless Fronthaul for 5G mmWave networks," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 138–144, Mar. 2019.

[21] S. Li, L. D. Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[22] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: Challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, Nov./Dec. 2014.

[23] M. Raithatha, A. U. Chaudhry, R. H. M. Hafez, and J. W. Chinneck, "Locating gateways for maximizing backhaul network capacity of 5G ultra-dense networks," in *Proc. Wireless Telecommun. Symp. (WTS)*, 2020, pp. 1–6.

[24] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[25] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[26] P. Wang, K. Li, B. Xiao, and K. Li, "Multiobjective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11737–11748, Jul. 2022.

[27] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[28] Y. Guo, Z. Zhao, K. He, S. Lai, J. Xia, and L. Fan, "Efficient and flexible management for Industrial Internet of Things: A federated learning approach," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108122.

[29] Q. Li, S. Wang, A. Zhou, X. Ma, F. Yang, and A. X. Liu, "QoS driven task offloading with statistical guarantee in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 278–290, Jan. 2022.

[30] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE J. Sel. Topics Signal Process*, vol. 13, no. 3, pp. 392–407, Jun. 2019.

[31] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7867–7881, Dec. 2020.

[32] E. El Haber, T. M. Nguyen, C. Assi, and W. Ajib, “Macro-cell assisted task offloading in MEC-based heterogeneous networks with wireless backhaul,” *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1754–1767, Dec. 2019.

[33] J. Bi, H. Yuan, S. Duanmu, M. Zhou, and A. Abusorrah, “Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization,” *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3774–3785, Mar. 2021.

[34] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, “Dynamic energy-saving offloading strategy guided by Lyapunov optimization for IoT devices,” *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19903–19915, Oct. 2022.

[35] Z. Li, V. Chang, J. Ge, L. Pan, H. Hu, and B. Huang, “Energy-aware task offloading with deadline constraint in mobile edge computing,” *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–24, 2021.

[36] Y. Lu, X. Chen, Y. Zhang, and Y. Chen, “Cost-efficient resources scheduling for mobile edge computing in ultra-dense networks,” *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 3163–3173, Sep. 2022.

[37] Y. Liu, C. Liu, J. Liu, Y. Hu, K. Li, and K. Li, “Mobility-aware and code-oriented partitioning computation offloading in multi-access edge computing,” *J. Grid Comput.*, vol. 20, p. 11, Mar. 2022.

[38] M. Chowdhury and M. Maier, “Toward dynamic HART-centric task offloading over FiWi infrastructures in the tactile Internet Era,” *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 123–128, Nov. 2019.

[39] A. Ebrahimzadeh and M. Maier, “Cooperative computation offloading in FiWi enhanced 4G HetNets using self-organizing MEC,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4480–4493, Jul. 2020.

[40] Q.-V. Pham, N. Iradukunda, N. H. Tran, W.-J. Hwang, and S.-H. Chung, “Joint placement, power control, and spectrum allocation for UAV wireless backhaul networks,” *IEEE Netw. Lett.*, vol. 3, no. 2, pp. 56–60, Jun. 2021.

[41] W. Wu, Q. Yang, R. Liu, T. Q. S. Quek, and K. S. Kwak, “Online spectrum partitioning for LTE-U and WLAN coexistence in unlicensed spectrum,” *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 506–520, Jan. 2020.

[42] Y. Shi, E. Alsusa, and M. W. Baidas, “Energy-efficient decoupled access scheme for cellular-enabled UAV communication systems,” *IEEE Syst. J.*, vol. 16, no. 1, pp. 701–712, Mar. 2022.

[43] C. Kai, H. Li, L. Xu, Y. Li, and T. Jiang, “Energy-efficient device-to-device communications for green smart cities,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1542–1551, Apr. 2018.

[44] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, “Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

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

[46] D. P. Bertsekas, *Convex Optimization Theory*. Belmont, MA, USA: Athena Sci., 2009.

[47] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the LambertW function,” *Adv. Comput. Math.*, vol. 5, pp. 329–359, Dec. 1996.

[48] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in UAV-enabled mobile edge computing Networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

Hui Xiao received the B.E. degree from Shandong University in 2017, and the M.E. degree from Central South U niversity in 2020, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. Her main research interests are in the area of mobile edge computing and cloud computing.

Jiawei Huang (Member, IEEE) received the bachelor’s degree from the School of Computer Science, Hunan University in 1999, and the master’s and Ph.D. degrees from the School of Computer Science and Engineering, Central South University, China, in 2004 and 2008, respectively, where he is currently a Professor with the School of Computer Science and Engineering. His research interests include cloud computing, data center networks, Internet video, Web performance, and programmable switching architectures.

Zhigang Hu received the M.S. and Ph.D. degrees from Central South University in 1988 and 2002, respectively, where he is currently a Professor with the School of Computer Science and Engineering. His research interests include high performance computing, cloud computing, edge computing, and machine learning.

Meiguang Zheng received the B.S. and Ph.D. degrees from Central South University in 2005 and 2011, respectively, where she is currently an Associate Professor with the School of Computer Science and Engineering. Her research interests include cloud computing, edge computing, and big data.

Keqin Li (Fellow, IEEE) is a SUNY Distinguished Professor of Computer Science with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. He has authored or coauthored over 900 journal articles, book chapters, and refereed conference papers. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world’s top five most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has received several best paper awards and chaired many international conferences. He is currently an Associate Editor of the *ACM Computing Surveys* and the *CCF Transactions on High Performance Computing*. He has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing. He is an AAAS Fellow and an AAIA Fellow and also a Member of Academia Europaea (Academician of the Academy of Europe).