Inter-server Computation Offloading and Resource Allocation in Multi-drone Aided Space-Air-Ground Integrated IoT Networks

Yongpeng Shi, Junjie Zhang, Ya Gao, and Yujie Xia

***Abstract*—Combining mobile edge computing (MEC), the multi-drone aided space-air-ground integrated Internet of things (SAG-IoT) networks can provide ground IoT devices (GIDs) high-quality wireless access and computing services. However, the diverse tasks, moving drones, and limited network resources reveal great challenges for the task offloading and resource allocation scheme exploitation. Especially, given the restricted computation resources, how to make full use of available applications deployed on MEC servers (MECSs) to compute various types of tasks, is even an important issue. To the best of our knowledge, it is an entirely new problem since most existing works in this line assume that all types of applications can be deployed on one MECS so as to process various offloaded tasks. Toward this end, we present this paper to investigate inter-server computation offloading, resource allocation, and drone deployment to minimize the overall computation overhead of all GIDs. An iteratively optimization algorithm is proposed which alternately utilizes heuristic greedy and successive convex approximation methods. Simulation results verify that, for different GID numbers, optimization schemes, and computing models, our devised schemes can not only significantly reduce the overall computation overhead but also achieve optimal decisions of computation offloading, spectrum allocation, and drone deployment.**

***Index Terms*—Bandwidth allocation, inter-server computation offloading, multi-drone, space-air-ground integrated IoT network.**

I. Introduction

DURING the past years, it is widely acknowledged that Internet of things (IoT) networks are experiencing an explosive growth in regard to both the number of participated equipment and the supported services [1]. By integrating smart sensing and wireless technologies, IoT can seamlessly combine various heterogeneous networks to construct powerful systems in diverse applied fields such as intelligent transportation, smart agriculture, wisdom health care, and smart logistics [2]. As is well known, most applications in IoT are computation intensive and require high-performance computations to accomplish. However, constrained by the physical size, the battery-powered IoT devices usually have limited computing capacity and cannot provide sustainable computation resource [3]. The contradiction between resource-restricted IoT devices and resource-hungry applications reveals an unparalleled challenge for the evolution and deployment of on-going and future IoT networks. How to introduce new computing architecture into IoT ecosystem to efficiently process applications and reduce energy consumption, so as to extend the battery life of IoT devices, is of great importance and deserves further exploring.

Having sufficient computation resources and storage capacity, cloud computing is able to dramatically reduce the computation latency and the energy consumption of IoT devices. Nevertheless, due to the long transmission latency caused by the long distance from terminal user to the cloud center, it cannot satisfy the requirements of time sensitive applications [4]. To address this challenge, mobile edge computing (MEC) [5] has been proposed as a key paradigm toward the fifth generation mobile communication (5G) [6], which deploys computation resource on MEC servers (MECSs) at the network edge, e.g., base stations (BSs), small cells, and wireless access points, to provide flexible and efficient computing services to the mobile users. Unlike traditional cloud computing, MEC can significantly reduce the accomplishing time and improve the communication reliability by offloading the tasks of mobile users to the adjacent MECSs. However, limited by the network coverage and capacity, only depending ground 5G systems cannot fulfill the increasing traffic and computation requirements of diverse IoT applications, especially in remote and rural areas, where IoT devices could be also widely deployed to conduct special services with high computing demands, such as high-definition sound or video information processing [7]. Due to the absence of ground network infrastructure, the typical cloud and edge computing are unable to be applied in such cases [8]. IoT must leverage new network architectures to widen the communication coverage and enhance the computation capacity for the large number of connected devices and provided services.

Interconnecting satellites, aerial platforms, and ground IoT devices (GIDs), the space-air-ground integrated IoT (SAG-IoT) network can provide seamless connectivity and enhanced capacity to diverse practical IoT applications [9]. In particular, the multiple drones, aka unmanned aerial vehicles (UAVs), aided SAG-IoT holds great promise for bringing lots of

Manuscript received November 11, 2021; revised March 14, 2022; approved for publication by Kyung-Joon Park, Division II Editor, April 14, 2022.

This work was supported in part by Key Scientific and Technological Projects under Grant 202102210120, 212102210553, and 222102210301, in pat by Key Scientific Research Program of Higher Education under Grant 21A510008, in part by Natural Science Foundation of Henan under Grant 202300410292, and in part by Henan Key Laboratory of E-commerce Big Data Processing & Analysis under Grant 2020-KF-6, Henan Province, China.

Y. Shi, J. Zhang, Y. Gao, and Y. Xia are with the Henan Key Laboratory of E-commerce Big Data Processing and Analysis, the School of Physics and Electronic Information, Luoyang Normal College, Luoyang 471934, China (e-mail: syp@lynu.edu.cn, junjiezhang@163.com, gaoya@lynu.edu.cn, xyj@lynu.edu.cn).

Y. Xia is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2022.000016

1229-2370/22/$10.00 © 2022 KICS

benefits in terms of high flexibility, high throughput, and high reliability [10]. On one hand, drones can not only act as aerial BSs to offer wireless access services to GIDs, but also play the role of transfers to forward the data traffic originated from GIDs to the satellites [11]. On the other hand, when installed with MECSs, drones are able to serve as the aerial edge servers to provide low latency and high efficiency computing functionality for GIDs.

In recent years, the multi-drone involved MEC in SAG-IoT systems have attracted lots of attentions, and a great deal of novel offloading schemes have been designed to optimize either processing latency [12], or energy consumption [13], or resource allocation for both communication and computation [14]. These research works, although presenting precious viewpoints for the offloading decision optimization and resource allocation in such networks, have one common limitation: all of them based on the assumption that the MECSs installed on drones can execute all types of applications to cope with various types of computation tasks offloaded by GIDs. As a matter of fact, impeded by the high cost of hardware deployment and maintenance, the computing and storage capacities of MECSs on drones cannot be as abundant as infinite. Accordingly, it is unpractical for one MECS to host all cartilaginous applications to compute unnumberable tasks. How to leverage the available limitation of application types deployed on MECSs to investigate effective cooperative computation offloading schemes among multiple MECSs on drones, deserves further exploration. Furthermore, offloading computation tasks to the MECSs in SAG-IoT can effectively alleviate the problem of resource constraints and reduce energy consumption of GIDs. Meanwhile, additional communication latency overhead will be inevitably required the tasks are transmitted through the wireless links. Therefore, task offloading in SAG-IoT must comprehensively consider the computation overhead in terms of both energy consumption and processing latency. To the best of our knowledge, we are the first to discuss the collaborative task offloading within multi-server for multi-task in the multi-drone enabled MEC. Motivated by this, we present this paper to study the problem of inter-server computation offloading and resource allocation in multi-drone aided SAG-IoT networks by introducing hybrid computing models of local, edge and remote cloud, to minimize the overall computation overhead of all GIDs. In particular, the main contributions of this paper are summarized as follows.

- Given the multi-drone aided space-air-ground integrated IoT network architecture, we mainly focus on the collaborative computation offloading of multi-server for multi-task by comprehensively leveraging local computing, edge computing, and remote cloud computing models.
- We present a novel joint optimization problem of inter-server task offloading and bandwidth allocation as well as drone deployment, and formulate it as a constrained optimization problem with the objective of minimizing the overall computation overhead of GIDs.
- By alternately adopting heuristic greedy method and successive convex approximation (SCA) technique, an iterative algorithm is exploited as our solution to the proposed problem while integrally considering the computation resource schedule within multiple servers and the communication resource allocation among different channels.
- Extensive simulation experiments have been conducted to verify the efficiencies of our proposed schemes. Numerical results show that for various number of GIDs and drones as well as application types, the proposed iterative scheme can not only implement the best computation offloading decision, but also obtain the optimal spectrum allocation and drone position deployment, so as to achieve the minimum overall computation overhead.

The rest of this paper is organized as follows. Section II reviews the related works in recent years. In Section III, we first introduces the system model, and then the joint optimization problem of inter-server task offloading and resource allocation is formulated. Section IV describes the detailed iterative algorithm for the problem. We present extensive numerical results in Section V, and finally conclude the whole paper in Section VI.

II. Related Work

Up to now, there have been many research works on the drone-enabled MEC systems in which the resource allocation schemes have been also proposed, including both single drone and multiple drones. For instance, with the constraints of energy-harvesting causal and the drone's velocity, Zhou *et al.* in [15] aimed to maximize the computation rate in a drone-enabled MEC wireless powered system through the joint optimization of processing unit frequency, user offloading latency and transmit power. Also in a drone-aided computing scenario, the authors in [16] proposed an alternative optimization scheme to minimize the total computation energy consumption by jointly optimizing offloading decisions, bit allocation and drone trajectory. While in [14], an optimization problem for minimizing the energy consumption of both communication and computation as well as drone's flight was proposed in the drone assisted MEC system by integrally considering the power allocation, drone trajectory design, and bits allocation. Considering the limited battery of IoT devices and the energy budget in a drone enhanced edge, Guo *et al.* in [17] proposed a coordinate descent based approach to reduce the overall energy consumption for task processing.

In the multi-drone aided MEC networks, Zhang *et al.* in [18] presented a Dinkelbach-based iterative optimization algorithm to maximize the computation efficiency by taking into account computation bits, energy consumption, user association, power and spectrum resources, and drone trajectory scheduling. In [19], Guo *et al.* proposed a coded distributed computing framework for task offloading from multi-drone to ground MECSs to save transmission and flying energy consumption of drones and reduce computing latency by designing cost optimal trajectory and code parameter schedule algorithms. To cope with the problem of offloading heavy tasks of drones and to achieve the optimal trade-off among energy consumption, latency, and computation overhead, the authors in [20] presented a non-cooperative game theory based strategy in

the drones assisted MEC system. To maximize the number of served IoT devices in the multi-drone enabled MEC network, Zhan *et al.* [21] presented a joint optimization scheme by adopting SCA method to alternatively optimizing computation offloading decisions, drone trajectory, and resource allocation. The authors in [22] presented a two-layer optimization strategy to minimize the total energy consumption in the multi-drone enabled MEC system consisting of large-scale IoT users by jointly optimizing the drone deployment and task scheduling. The authors designed a differential evolution algorithm and a greedy algorithm to solve the problem. In a multi-drone assisted SAG-IoT network, Cheng *et al.* [7] proposed a joint approach of computing resource allocation and task scheduling for MECs, and then devised a learning based method to optimize the offloading decisions with the objective of minimizing the computation cost.

Generally speaking, the existing works on drone-aided IoT-MEC systems have introduced many novel methods for computation offloading and resource allocation. However, most of them mainly leveraged the edge servers deployed on drones to provide computing functionality for GIDs while ignored the sufficient computation resources in the remote cloud center (RCC). More importantly, the available studies presumed that the MECSs on drones were deployed all kinds of applications so that they had the capability of processing various types of computation tasks requested by GIDs. Subjected to the power limitation of drones and the architecture constraint of MEC, it is infeasible for MECSs on drones to be deployed as huge computation resources as cloud center. Each MECS can host finite kinds of applications to compute the offloaded tasks. Accordingly, task offloading in multi-drone assisted IoT systems must utilize the available resource deployed on MECSs and consider the collaboration among multiple servers. Although Yao *et al.* in [23] considered that workload could be served across multiple geographically distributed data centers, and proposed a stochastic optimization based approach to minimize the power consumption without reducing the amount of workload served, Sun *et al.* [24] designed two double auction mechanisms to improve the system efficiency while jointly considering incentives and cross-server resource allocation in blockchain-driven MEC, they mainly focused on the inter-server computation within fiber-connected data centers or MEC servers, while taking no consideration of the variable wireless communication links. Toward this end, we present this paper to elaborate the inter-server computation offloading and resource allocation problem for the computation overhead minimization in the multi-drone aided IoT network while collaboratively adopting local, edge, and RCC computing models.

III. System Model and Problem Formulation

In this paper, we mainly consider a multi-task computation offloading scenario in the multi-drone aided SAG-IoT network, as shown in Fig. 1. M GIDs $\mathcal{M} \triangleq \{1, 2, ..., M\}$ are randomly deployed in a remote area covered by a low earth orbit (LEO) satellite to conduct certain tasks. In the air network, U drones $\mathcal{U} \triangleq \{1, 2, ..., U\}$ are hovering above this area within

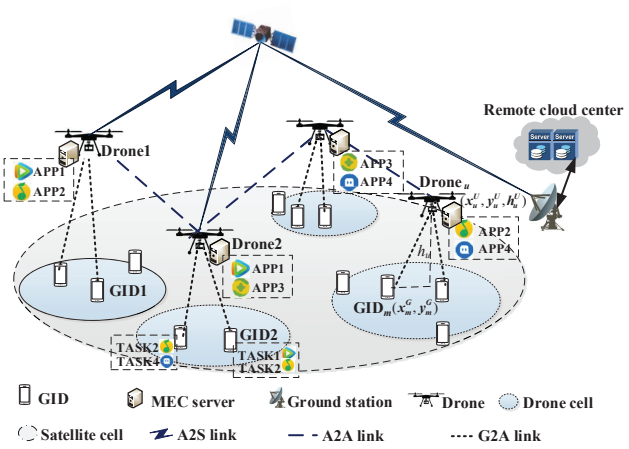


Fig. 1. A multi-drone aided space-air-ground integrated IoT network architecture. MEC servers deployed on the drones can process various types of tasks offloaded of MDs. MDs can offload their tasks to drones through ground to air links or RCC via the satellite backhaul links.

a height range of $[h_{min}, h_{max}]$, and each drone is installed an MECS to provide enhanced computing service to the GIDs. In the following, both drone and MECS are collectively called 'drone' for easy presentation. All drones have the capability to communicate with both GIDs and other drones as well as satellites by adopting decode-and-forward (DF) scheme [25] and full-duplex technology, while each GID must leverage drone as relay to connect with the satellite. There are total K types of computation tasks $\mathcal{K} \triangleq \{1, 2, ..., K\}$ requested by all GIDs, and each GID $m \in \mathcal{M}$ has K_m types of task to execute, denoted by $\mathcal{K}_m \triangleq \{k_{m,1}, k_{m,2}, ..., k_{m,K_m}\}$, where $\mathcal{K}_m \subset \mathcal{K}$, $K_m < K$. For GID m's type-n computation task, i.e., $k_{m,n}$, it can generally be defined as $k_{m,n} \triangleq (s_{m,n}, c_{m,n}, \tau_{m,n}^{max})$, where $s_{m,n}$ denotes the input data size of $k_{m,n}$, $c_{m,n}$ is the requisite CPU cycles to compute $k_{m,n}$, and $\tau_{m,n}^{max}$ is the maximum tolerated latency to accomplish $k_{m,n}$.

In order to process the computation tasks requested by GIDs, drones must be deployed corresponding applications. We call the application (APP) processing type-n computation task APP-n. Suppose that there are also total K types of APPs $\mathcal{J} \triangleq \{1, 2, ..., K\}$ deployed on all drones, and each drone $u \in \mathcal{U}$ can host at most K_u types of APPs $\mathcal{J}_u \triangleq \{j_{u,1}, j_{u,2}, ..., j_{u,K_u}\}$, $\mathcal{J}_u \subset \mathcal{J}$, $K_u < K$. That is, one drone is unable to process all types of computation tasks, if drone u does not host the corresponding APPs, the tasks must be transferred to other drones through the multi-hop wireless air-to-air links for processing, or be transmitted to the RCC via the satellite backhaul link, thus the inter-server computation offloading among multi-drone is caused.

A. Communication Model

As depicted in Fig. 1, there exist three kinds of communication links in the system: *1) G2A link* (GID-to-drone), *2) A2A link* (drone-to-drone), and *3) A2S link* (drone-to-satellite). For simplicity, we consider these communication links to be clear line of sight (LoS) links and neglect the impact of shadowing and small scale fading. Therefore, we can set the value of

path loss exponent, $\alpha = 2$. In addition, orthogonal frequency division multi-plexing (OFDM) transmission is adopted to avoid co-channel interference among different G2A, A2A, and A2S links caused by the strong LOS channels in the integrated system. Let $\mathcal{B}^{G2A} = \{\omega_{m,u}^{G2A} \geq 0, m \in \mathcal{M}, u \in \mathcal{U}\}$, $\mathcal{B}^{A2A} = \{\omega_{u,v}^{A2A} \geq 0, u, v \in \mathcal{U}, u \neq v\}$, and $\mathcal{B}^{A2S} = \{\omega_u^{A2S}, u \in \mathcal{U}\}$ denote the allocated bandwidth on the G2A links from GIDs to drones, the A2A links between two drones, and the A2S links from drones to the satellite, respectively, the bandwidth allocation matrix can be represented as

$$\mathcal{B} \triangleq \{(\mathcal{B}^{G2A})^T, \mathcal{B}^{A2A}, (\mathcal{B}^{A2S})^T\}. \tag{1}$$

The overall allocated bandwidth is

$$\Omega_{tot} = \sum_{i=1}^{U} \sum_{j=1}^{M+U+1} \omega_{i,j}, \forall \omega_{i,j} \in \mathcal{B}. \tag{2}$$

It is assume that both the GIDs and drones do not change their positions during the data delivery process. Let p_m^G and $\omega_{m,u}^{G2A}$ denote the transmit power of GID m and the allocated bandwidth to the G2A link from GID m to drone u, respectively, the uplink data rate can be expressed as

$$r_{m,u}^{G2A} = \omega_{m,u}^{G2A} \log 2 \left(1 + \frac{p_m^G g_{m,u}^{G2A}}{\sigma_A^2}\right), \tag{3}$$

where σ_A^2 is the additive white Gaussian noise (AWGN) at the drone receiver, $g_{m,u}^{G2A}$ is the channel gain of the G2A link from GID m to drone u,

$$g_{m,u}^{G2A} = \frac{g_0}{||\mathbf{q}_u^A - \mathbf{q}_m^G||^2}, \tag{4}$$

in which g_0 denotes the channel power gain at the reference distance $d_0 = 1$ m, $\mathbf{q}_m^G = (x_m^G, y_m^G, 0)$ and $\mathbf{q}_u^A = (x_u^A, y_u^A, z_u)$ are the positions of GID m and drone u in the 3-D Cartesian coordinate system, respectively.

Similarly, let p_u^A and $g_{u,v}^{A2A}$ represent the transmit power of drone u and the channel gain of the A2A wireless link, respectively, the achievable data rate from drone u to drone v is

$$r_{u,v}^{A2A} = \omega_{u,v}^{A2A} \log_2 \left(1 + \frac{p_u^A g_{u,v}^{A2A}}{\sigma_A^2}\right), \tag{5}$$

where $g_{u,v}^{A2A} = g_0/||\mathbf{q}_u^A - \mathbf{q}_v^A||^2$. Furthermore, in order to avoid collision, the communication security distances among U drones must be guaranteed, i.e.,

$$||\mathbf{q}_u^A - \mathbf{q}_v^A|| \geq d_{min}, \tag{6}$$

where d_{min} is the minimum security distance.

Using H_S and σ_S^2 to separately denote the orbit height and the AWGN of the satellite, the uplink data rate of the A2S link can be given as

$$r_u^{A2S} = \omega_u^{A2S} \log_2 \left(1 + \frac{p_u^A g_u^{A2S} G_{tx} G_{rx}}{\sigma_S^2}\right), \tag{7}$$

where $g_u^{A2S} = g_0/(H_S - h_u)^2 \approx g_0/(H_S)^2$ is the channel gain of the A2S link, G_{tx} and G_{rx} are respectively the antenna gains of the drone and satellite. In general, we assume that each drone has an omni-directional antenna array element, as a result of which, $G_{tx} = 1$ is satisfied [26].

B. Computing Model

In general, there are three computing models for each computation task to select: *1) Local computing*, using GID's CPU for computing, *2) edge computing*, processing at the edge servers deployed on the drones, and *3) cloud computing*, offloading to the RRC for computing. Each GID m can select one of the three models to process its type-n task. In particular, we use $\lambda_{m,n} \in \{-1, 0, 1\}$ to denote task $k_{m,n}$'s computing model selection, where $\lambda_{m,n} = -1$ represents that GID m wants to process type-n task via its own CPU, $\lambda_{m,n} = 0$ means that task $k_{m,n}$ will be offloaded to the MECSs and be processed there, and $\lambda_{m,n} = 1$ states that GID m decides to compute its task $k_{m,n}$ on the RCC.

1) Local Computing: When $\lambda_{m,n} = -1$, i.e., GID m decides to process the type-n task by using its own CPU, the processing latency $t_{m,n}^l$ and the consumed energy $E_{m,n}^l$ can be separately calculated as

$$t_{m,n}^l = \frac{c_{m,n}}{f_m^l}, \tag{8}$$

$$E_{m,n}^l = \kappa_m c_{m,n} (f_m^l)^2, \tag{9}$$

where f_m^l, is the computational capability of GID m (in CPU cycles per second), and κ_m is a coefficient related to the GID m's CPU hardware architecture [27].

The computation overhead of $k_{m,n}$ is a function of the processing latency and energy consumption, in the local computing model, it can be defined as

$$\phi_{m,n}^l = \beta_{m,n}^t t_{m,n}^l + \beta_{m,n}^e E_{m,n}^l, \tag{10}$$

where $\beta_{m,n}^t \in [0, 1]$ and $\beta_{m,n}^e \in [0, 1]$ are respectively the weighted coefficients of the computing latency and energy consumption for task $k_{m,n}$, $\beta_{m,n}^t + \beta_{m,n}^e = 1$.

2) Edge Computing: When $\lambda_{m,n} = 0$, task $k_{m,n}$ will be offloaded to the MECSs for processing. In this model, GID m first upload $k_{m,n}$ to the connected drone u via the G2A wireless link. Then $k_{m,n}$ will be accomplished there if drone u hosts the corresponding APP, otherwise the task must be forwarded to other drones on which the required APP is deployed. Note that the time cost for sending back computed results from drones to GIDs is neglected in this paper since the data size of processed results is much smaller than that of input data for most IoT applications. On this condition, the accomplishing latency to accomplish task $k_{m,n}$ mainly includes two parts, i.e., the transmission latency for uploading and forwarding $k_{m,n}$ from GID m to the target drone which hosts the corresponding APP, and the computing time at the drone.

If task $k_{m,n}$ can be directly computed at drone u, GID m will transmit the task to this drone with the data rate of $r_{m,u}^{G2A}$, and the transmission latency can be easily calculate as $t_{m,n}^{e,tr} = s_{m,n}/r_{m,u}^{G2A}$. If drone u does not hosts the corresponding APP to process task $k_{m,n}$, it must forward $k_{m,n}$ to the other drone w on which the required APP is deployed via the multi-hop A2A links. Suppose that a drone can directly forward its data to at most one of other ones. Let $\mathcal{H}_{u,w} = \{h_i | 1 \leq i \leq |\mathcal{H}_{u,w}|\}$ denote the ordered set of drones on the routing path from drone u to w, obviously, $h_1 = u, h_{|\mathcal{H}_{u,w}|} = w$, where $|\mathcal{H}_{u,w}|$

is the cardinality of set $\mathcal{H}_{u,w}$. According to the DF protocol, the transmission rate for forwarding the task $k_{m,n}$ from drone u to w can be calculated as

$$r_{u,w}^{A2A} = \min\{r_{h_i,h_{i+1}}^{A2A}\}, 1 \le i \le |\mathcal{H}_{u,w}| - 1. \tag{11}$$

Thus, the data delivery rate for task $k_{m,n}$ from GID m to drone w is expressed as

$$r_{m,w}^{e,fw} = \min\{r_{m,u}^{G2A}, r_{u,w}^{A2A}\}, \tag{12}$$

and the delivery latency is

$$t_{m,n}^{e,fw} = \frac{s_{m,n}}{r_{m,w}^{e,fw}}. \tag{13}$$

The time cost for computing $k_{m,n}$ can be expressed as

$$t_{m,n}^{e,comp} = \frac{c_{m,n}}{f_u^e}\rho_{m,n}^u + \frac{c_{m,n}}{f_w^e}(1 - \rho_{m,n}^u), \tag{14}$$

where f_u^e and f_w^e denote the computing ability of the drones u and w, respectively, $\rho_{m,n}^u$ is a binary variable, $\rho_{m,n}^u = 1$ represents that task $k_{m,n}$ can be computed at drone u, $\rho_{m,n}^u = 0$ otherwise.

In this case, the accomplishing latency $t_{m,n}^e$ and the energy consumption of GID m, $E_{m,n}^e$, can be respectively given as

$$t_{m,n}^e = t_{m,n}^{e,tr}\rho_{m,n}^u + t_{m,n}^{e,fw}(1 - \rho_{m,n}^u) + t_{m,n}^{e,comp}, \tag{15}$$

and

$$E_{m,n}^e = p_m(t_{m,n}^{e,tr}\rho_{m,n}^u + t_{m,n}^{e,fw}(1 - \rho_{m,n}^u)). \tag{16}$$

The computation overhead of $k_{m,n}$ in the edge computing model is calculated as

$$\phi_{m,n}^e = \beta_{m,n}^t t_{m,n}^e + \beta_{m,n}^e E_{m,n}^e. \tag{17}$$

3) Cloud Computing: As for the case of offloading decision $\lambda_{m,n} = 1$, GID m will transmit its computation task $k_{m,n}$ to the RCC through the satellite links, and then the RCC server will process the task. It is noticeable that RCC servers are always deployed rich computation resource, thus, only the transmission latency and propagation latency from GID m to the RRC are taken into account and the computing latency of $k_{m,n}$ in this model can be neglected. Also based on DF protocol, the uplink data rate of $k_{m,n}$ from GID m to the satellite using drone u as relay can be expressed as

$$r_{m,n}^c = \min\{r_{m,u}^{G2A}, r_u^{A2S}\}. \tag{18}$$

Then the transmission latency can be obtained as $t_{m,n}^{c,tr} = s_{m,n}/r_{m,n}^c$, and the propagation latency is $t_{m,n}^{c,pr} = 2H_s/c$, $c = 3 \times 10^8$ m/s is the velocity of electromagnetic wave in vacuum. Then, when being computed in RRC, the processing latency $t_{m,n}^c$ of $k_{m,n}$ and the energy consumption of GID m, $E_{m,n}^c$, can be separately calculated as $t_{m,n}^c = t_{m,n}^{c,tr} + t_{m,n}^{c,pr}$ and $E_{m,n}^c = p_m t_{m,n}^{c,tr}$. The computation overhead of $k_{m,n}$ in the cloud computing model is given as

$$\phi_{m,n}^c = \beta_{m,n}^t t_{m,n}^c + \beta_{m,n}^e E_{m,n}^c. \tag{19}$$

C. Problem Formulation

According to the system models and assumptions discussed above, the main objective of this paper is to obtain the minimum computational overhead of all GIDs by jointly optimizing the task's computation offloading decision, spectrum allocation, and drone position deployment, while satisfying the maximum tolerated processing latency of all tasks and the given total available bandwidth Ω. For notational convenience, we define $\mathcal{S} = \{\lambda_{m,n}\}$, $\mathcal{Q} = \{\mathbf{q}_u^A\}$, and $\mathcal{P} = \{\mathcal{S}, \mathcal{B}, \mathcal{Q}\}$, the constrained computational overhead minimizing problem can be mathematically formulated as follows.

$$\begin{aligned}
\mathbf{P1}: &\Phi = \min_{\{\mathcal{P}\}} \sum_{m=1}^{M}\sum_{n=1}^{K_m} \phi_{m,n} \\
\text{s.t.} &C1: t_{m,n} \le \tau_{m,n}^{max}, 1 \le n \le K_m, \forall m \in \mathcal{M}, \\
&C2: \Omega_{tot} \le \Omega, \\
&C3: \omega_{i,j} \ge 0, 1 \le i \le U, 1 \le j \le (M + U + 1), \\
&C4: \sum_{m=1}^{M}\sum_{n=1}^{K_m} r_{m,u}^{G2A} I_{\{\lambda_{m,n} \ne -1\}} + \sum_{v=1, v\ne u}^{U} r_{v,u}^{A2A} \ge \\
&\quad \sum_{v=1, v\ne u}^{U} r_{u,v}^{A2A} + \sum_{m=1}^{M}\sum_{n=1}^{K_m} r_u^{A2S} I_{\{\lambda_{m,n}=1\}}, \forall u \in \mathcal{U}, \\
&C5: \sum_{u=1}^{U}\sum_{m=1}^{M}\sum_{n=1}^{K_m} \rho_{m,n}^u \le 1, \rho_{m,n}^u \in \{0,1\}, \\
&C6: ||\mathbf{q}_u^A - \mathbf{q}_v^A|| \ge d_{min}, \forall u, v \in \mathcal{U}, u \ne v, \\
&C7: h_{min} \le h_u \le h_{max}, \forall u \in \mathcal{U}, \\
&C8: \lambda_{m,n} \in \{-1, 0, 1\}, 1 \le n \le K_m, \forall m \in \mathcal{M},
\end{aligned} \tag{20}$$

where

$$\begin{aligned}
\phi_{m,n} = &\phi_{m,n}^l I_{\{\lambda_{m,n}=-1\}} + \phi_{m,n}^e I_{\{\lambda_{m,n}=0\}} \\
&+ \phi_{m,n}^c I_{\{\lambda_{m,n}=1\}},
\end{aligned} \tag{21}$$

and

$$t_{m,n} = t_{m,n}^l I_{\{\lambda_{m,n}=-1\}} + t_{m,n}^e I_{\{\lambda_{m,n}=0\}} + t_{m,n}^c I_{\{\lambda_{m,n}=1\}}, \tag{22}$$

$I_{\{\cdot\}}$ is an indicator function, $I_{\{\cdot\}} = 1$ if $\{\cdot\}$ holds, otherwise $I_{\{\cdot\}} = 0$.

In problem **P1**, the constraint $C1$ ensures that the accomplishing latency of each task must meet the requirement of its maximum tolerated processing time. $C2$ denotes that the totally allocated bandwidth in the system cannot exceed the available spectrum, and $C3$ means the allocated bandwidth cannot be negative. $C4$ is the flow conservation constraint [28] in the multi-hop relay wireless networks, which declares the total outgoing flows of each drone should be less than or equal to its sum of incoming ones. While the constraint $C5$ states that each task can be computed at most on one drone. $C6$ and $C7$ guarantee the communication security distance among drones. $C8$ represents that each task $k_{m,n}$ can select any but only one computing model from the three offloading choices. It is easy to prove that problem **P1** is NP-hard [29] and it cannot be solved in polynomial time.

IV. Joint optimization of Inter-Server Task Offloading and Rource Allocation

The objective function of problem **P1** is multivariate with respect to variables $\mathcal{S}$, $\mathcal{B}$, and $\mathcal{Q}$. Constraints $C1$, $C4$, $C6$ and the objective function are all non-convex. What's more, **P1** is also a mixed integer programming problem due to the variables $\lambda_{m,n}$ and $\rho^u_{m,n}$. Therefore, **P1** is a non-convex mixed integer programming optimization problem, which is difficult to solve. To efficiently handle problem **P1**, we decompose it into three subproblems by alternately optimizing the offloading decision $\mathcal{S}$, bandwidth allocation $\mathcal{B}$, and drone deployment $\mathcal{Q}$. And then the minimum computational overhead and the optimal solution to **P1** are obtained through an iterative method.

A. Computation Offloading Optimization

Given bandwidth allocation $\mathcal{B}$ and drone deployment $\mathcal{Q}$, problem **P1** can be formulated as

$$\mathbf{P1.1}: \min_{\{\mathcal{S}\}} \sum_{m=1}^{M}\sum_{n=1}^{K_m} \phi_{m,n}. \quad \text{s.t.} C1, C4, C5, C8. \tag{23}$$

Obviously, **P1.1** is an integer programming problem and can be solved by adopting the enumeration method, which has high computational complexity of $O(3^{MK_m})$ and is not applicable to large scale problems. Notice that for the fixed bandwidth allocation and drone position deployment strategies, which computing model will be selected for each task depends on whether it can achieve the minimum overhead with the constraint of maximum processing latency. Based on this, we propose a heuristic greedy offloading optimization algorithm (HGOA) to solve problem **P1.1**. In HGOA, we first select the tasks of which the accomplishing latency meets their maximum tolerated processing time and record them into set $\mathcal{K}^{sat}$, then assign the offloading decisions of the tasks in $\mathcal{K}^{sat}$ according to their minimum computational overhead by comparing three computing models, respectively. The details of the proposed HGOA are elaborated in Algorithm 1.

In Algorithm 1, calculating $t^e_{m,n}$ and $\phi^e_{m,n}$ means that task $k_{m,n}$ is chosen to be computed at the drone. As discussed in Section III-B, this task cannot be processed until the specific drone hosting the corresponding APP is searched. To obtained the minimum forwarding latency from drone u to w, we proposed an open shortest path first (OSPF) routing strategy based APP searching method to obtain the maximum forwarding data rate $r^{A2A}_{u,w}$, which is summarized in Procedure 1.

B. Bandwidth Allocation Optimization

According to (21), when the computation offloading scheme is known, $\phi^l_{m,n}$ is a constant, and $\phi_{m,n}$ varies only with $\phi^e_{m,n}$ and $\phi^c_{m,n}$. Therefore, for any given $\mathcal{S}$, problem **P1** can be redefined as

$$\mathbf{P1.2}: \min_{\{\mathcal{B}\}} \sum_{m=1}^{M}\sum_{n=1}^{K_m} (\phi^e_{m,n} I_{\{\lambda_{m,n}=0\}} + \phi^c_{m,n} I_{\{\lambda_{m,n}=1\}}) \quad \text{s.t.} \quad C1 - C4. \tag{24}$$

Algorithm 1 A heuristic greedy offloading optimization algorithm (HGOA).

Input: $\mathcal{M}$, $\mathcal{U}$, the fixed bandwidth allocation $\mathcal{B}$ and drone position deployment $\mathcal{Q}$.
Output: $\mathcal{S}^{opt}$-an approximately optimal offloading decision.

```
Initialize K^sat = ∅;
for m = 1 to M do
    for n = 1 to K_m do
        compute t^l_{m,n}, t^e_{m,n}, t^c_{m,n}, φ^l_{m,n}, φ^e_{m,n}, φ^c_{m,n};
        if min{t^l_{m,n}, t^e_{m,n}, t^c_{m,n}} ≤ τ^max_{m,n} then
            K^sat = K^sat ∪ {k_{m,n}};
        end if
    end for
end for
for each task k_{m,n} ∈ K^sat do
    if min{φ^e_{m,n}, φ^c_{m,n}} ≤ φ^l_{m,n} OR t^l_{m,n} ≥ τ^max_{m,n} then
        λ_{m,n} = 0 if φ^e_{m,n} ≤ φ^c_{m,n}, otherwise λ_{m,n} = 1;
    else
        λ_{m,n} = −1;
    end if
end for
return S^opt = {λ_{m,n}}.
```

Procedure 1 An inter-server APP search procedure

Input: $\mathcal{U}$, $\mathcal{B}$, $\mathcal{Q}$, $\mathcal{S}^t$, and the source drone u, .
Output: the target drone w which has the required APP and the maximum forwarding data rate $r^{A2A}_{u,w}$.

```
for each o ∈ U and o ≠ u do
    search the required APP on drone o;
    if drone o hosting the required APP then
        obtain H_{u,o}, the routing path from drone v to o
        using OSPF routing strategy;
        calculate r^{A2A}_{u,o} using (11);
    end if
end for
w = arg max_{o∈U,o≠u} r^{A2A}_{u,o};
r^{A2A}_{u,w} = max_{o∈U,o≠u} r^{A2A}_{u,o};
return w and r^{A2A}_{u,w}.
```

Define $\mathcal{X} = \{x_{m,u} > 0, \forall m \in \mathcal{M}, \forall u \in \mathcal{U}\}$, $\mathcal{Y} = \{y_{u,v} > 0, \forall u, v \in \mathcal{U}, u \neq v\}$, and $\mathcal{Z} = \{z_u > 0, \forall u \in \mathcal{U}\}$, $\phi^c_{m,n}$ and $\phi^e_{m,n}$ can be separately expended as

$$\begin{aligned}\phi^c_{m,n} &= \beta^t_{m,n} t^{c,pr}_{m,n} + (\alpha^t_{m,n} + \beta^e_{m,n} p_m)\frac{s_{m,n}}{\min\{r^{G2A}_{m,u}, r^{A2S}_u\}} \\ &\leq \beta^t_{m,n} t^{c,pr}_{m,n} + (\beta^t_{m,n} + \beta^e_{m,n} p_m) s_{m,n} \max\{x_{m,u}, z_u\} \\ &\triangleq (\phi^c_{m,n})',\end{aligned} \tag{25}$$

and

$$\begin{aligned}\phi^e_{m,n} &= \beta^t_{m,n} t^{e,sev}_{m,n} + (\alpha^t_{m,n} + \alpha^e_{m,n} p_m)\Big(\frac{s_{m,n}}{r^{G2A}_{m,u}}\rho^u_{m,n} + \\ &\quad \frac{s_{m,n}}{\min\{r^{G2A}_{m,u}, r^{A2A}_{u,w}\}}(1-\rho^u_{m,n})\Big) \\ &\le \alpha^t_{m,n} t^{e,sev}_{m,n} + (\alpha^t_{m,n} + \alpha^e_{m,n} p_m) s_{m,n}\big(x_{m,u}\rho^u_{m,n} + \\ &\quad \max\{x_{m,u}, y_{u,w}\}(1-\rho^u_{m,n})\big) \triangleq (\phi^e_{m,n})'. \end{aligned} \tag{26}$$

Similarly, the expansion of constraint $C1$ can be expressed as

$$\begin{aligned} t_{m,n} &= t^l_{m,n} I_{\{\lambda_{m,n}=-1\}} + t^{e,sev}_{m,n} I_{\{\lambda_{m,n}=0\}} + t^{c,pr}_{m,n} I_{\{\lambda_{m,n}=1\}} + \\ &\quad \Big(\frac{s_{m,n}}{r^{G2A}_{m,u}}\rho^u_{m,n} + \frac{s_{m,n}}{\min\{r^{G2A}_{m,u}, r^{A2A}_{u,w}\}}(1-\rho^u_{m,n})\Big) I_{\{\lambda_{m,n}=0\}} \\ &\quad + \frac{s_{m,n}}{\min\{r^{G2A}_{m,u}, r^{A2S}_{u}\}} I_{\{\lambda_{m,n}=1\}} \\ &\le t^l_{m,n} I_{\{\lambda_{m,n}=-1\}} + t^{e,sev}_{m,n} I_{\{\lambda_{m,n}=0\}} + t^{c,pr}_{m,n} I_{\{\lambda_{m,n}=1\}} + \\ &\quad s_{m,n}\left(x_{m,u}\rho^u_{m,n} + \max\{x_{m,u}, y_{u,w}\}(1-\rho^u_{m,n})\right) I_{\{\lambda_{m,n}=0\}} \\ &\quad + s_{m,n}\max\{x_{m,u}, z_u\} I_{\{\lambda_{m,n}=1\}} \triangleq (t_{m,n})'. \end{aligned} \tag{27}$$

Thus, problem **P1.2** is transformed into

$$\begin{aligned} \mathbf{P1.2}': \min_{\{\mathcal{B}\}} & \sum_{m=1}^{M}\sum_{n=1}^{K_m}\left((\phi^e_{m,n})' I_{\{\lambda_{m,n}=0\}} + (\phi^c_{m,n})' I_{\{\lambda_{m,n}=1\}}\right) \\ \text{s.t.} \quad & C1': (t_{m,n})' \le \tau^{max}_{m,n}, 1 \le n \le K_m, \forall m \in \mathcal{M}, \\ & C2 - C4, \\ & C9: x_{m,u} r^{G2A}_{m,u} \ge 1, \forall m \in \mathcal{M}, \forall u \in \mathcal{U}, \\ & C10: y_{u,v} r^{A2A}_{u,v} \ge 1, \forall u, v \in \mathcal{U}, u \ne v, \\ & C11: z_u r^{A2S}_{u} \ge 1, \forall u \in \mathcal{U}. \end{aligned} \tag{28}$$

From (3), (5), and (7), we can see that $r^{G2A}_{m,n}$, $r^{A2A}_{u,v}$, and r^{A2S}_{u} are linear functions with respect to $\omega^{G2A}_{m,u}$, $\omega^{A2A}_{u,v}$, and ω^{A2S}_{u}, respectively. Therefore, both the objective function and the constraints of problem **P1.2′** are convex, and it can be solved in optimization solvers such as YALMIP toolbox [30].

C. Drone Deployment Optimization

Under given computation offloading scheme $\mathcal{S}$ and bandwidth allocation $\mathcal{B}$, the problem of drone deployment optimization can be transformed into

$$\begin{aligned} \mathbf{P1.3}: \min_{\{Q\}} & \sum_{m=1}^{M}\sum_{n=1}^{K_m}(\phi^e_{m,n} I_{\{\lambda_{m,n}=0\}} + \phi^c_{m,n} I_{\{\lambda_{m,n}=1\}}) \\ \text{s.t.} \quad & C1, C4, C6, C7. \end{aligned} \tag{29}$$

In problem **P1.3**, constraints $C1$, $C4$, and $C6$ as well as the objective function are non-convex due to that $r^{G2A}_{m,u}$, $r^{A2A}_{u,v}$, and $||\mathbf{q}^A_u - \mathbf{q}^A_v||^2$ are non-convex with respect to $\mathbf{q}^A_u$ and $\mathbf{q}^A_v$. However, $r^{G2A}_{m,u}$ and $r^{A2A}_{u,v}$ are respectively convex with respect to $||\mathbf{q}^A_u - \mathbf{q}^G_m||^2$ and $||\mathbf{q}^A_u - \mathbf{q}^A_v||^2$, while $||\mathbf{q}^A_u - \mathbf{q}^A_v||^2$ is convex with regard to $||\mathbf{q}^A_u - \mathbf{q}^A_v||$. It is known the first-order Taylor expansion of any convex function is its global lower bound at any point [31]. Therefore, we can adopt the SCA technique [32] to cope with the non-convexity of **P1.3**. Let $(\mathbf{q}^A_u)^i$ and $(\mathbf{q}^A_v)^i$ denote the positions of drone u and v in the i-th iteration, respectively, by applying the first-order Taylor expansion of $r^{G2A}_{m,u}$ and $r^{A2A}_{u,v}$, we have

$$\begin{aligned} r^{G2A}_{m,u} &= \omega^{G2A}_{m,u}\log_2\left(1 + \frac{p^G_m g_0}{||\mathbf{q}^A_u - \mathbf{q}^G_m||^2\sigma^2_A}\right) \\ &\ge \omega^{G2A}_{m,u}\log_2\left(1 + \frac{p^G_m g_0}{||(\mathbf{q}^A_u)^i - \mathbf{q}^G_m||^2\sigma^2_A}\right) \\ &\quad - \frac{\omega^{G2A}_{m,u} p^G_m g_0\left(||\mathbf{q}^A_u - \mathbf{q}^G_m||^2 - ||(\mathbf{q}^A_u)^i - \mathbf{q}^G_m||^2\right)}{\ln 2||(\mathbf{q}^A_u)^i - \mathbf{q}^G_m||^2(||(\mathbf{q}^A_u)^i - \mathbf{q}^G_m||^2\sigma^2_A + p^G_m g_0)} \\ &\triangleq (r^{G2A}_{m,u})', \end{aligned} \tag{30}$$

$$\begin{aligned} r^{A2A}_{u,v} &= \omega^{A2A}_{u,v}\log_2\left(1 + \frac{p^A_u g_0}{||\mathbf{q}^A_u - \mathbf{q}^A_v||^2\sigma^2_A}\right) \\ &\ge \omega^{A2A}_{u,v}\log_2\left(1 + \frac{p^A_u g_0}{||(\mathbf{q}^A_u)^i - (\mathbf{q}^A_v)^i||^2\sigma^2_A}\right) \\ &\quad - \frac{\omega^{A2A}_{u,v} p^A_u g_0\left(||\mathbf{q}^A_u - \mathbf{q}^A_u||^2 - ||(\mathbf{q}^A_u)^i - (\mathbf{q}^A_u)^i||^2\right)}{\ln 2||\left(\mathbf{q}^A_u\right)^i - (\mathbf{q}^A_v)^i||^2(||(\mathbf{q}^A_u)^i - (\mathbf{q}^A_v)^i||^2\sigma^2_A + p^A_u g_0)} \\ &\triangleq (r^{A2A}_{u,v})', \end{aligned} \tag{31}$$

and

$$\begin{aligned} C4': & \sum_{m=1}^{M}\sum_{n=1}^{K_m}(r^{G2A}_{m,u})' I_{\{\lambda_{m,n}\ne -1\}} + \sum_{v=1, v\ne u}^{U}(r^{A2A}_{v,u})' \\ & \ge \sum_{v=1, v\ne u}^{U}(r^{A2A}_{u,v})' + \sum_{m=1}^{M}\sum_{n=1}^{K_m} r^{A2S}_{u} I_{\{\lambda_{m,n}=1\}}, \forall u \in \mathcal{U}. \end{aligned} \tag{32}$$

Similarly, $||\mathbf{q}^A_u - \mathbf{q}^A_v||^2$ can be substituted with its convex lower bound at a given local point in each iteration, i.e.,

$$\begin{aligned} ||\mathbf{q}^A_u - \mathbf{q}^A_v||^2 &\ge -||(\mathbf{q}^A_u)^i - (\mathbf{q}^A_v)^i||^2 + \\ &\quad 2((\mathbf{q}^A_u)^i - (\mathbf{q}^A_v)^i)^T(\mathbf{q}^A_u - \mathbf{q}^A_v) \triangleq ((d^{A2A}_{u,v})')^2. \end{aligned} \tag{33}$$

Also by introducing the auxiliary variables $\mathcal{X}$ and $\mathcal{Y}$, we replace $r^{G2A}_{m,u}$ and $r^{A2A}_{u,v}$ with $(r^{G2A}_{m,u})'$ and $(r^{A2A}_{u,v})'$ in (26), (25) and (27) to obtain $(\phi^e_{m,n})''$, $(\phi^c_{m,n}))''$, and $(t_{m,n})''$, respectively, problem **P1.3** can be converted into

$$\begin{aligned} \mathbf{P1.3}': \min_{\{Q\}} & \sum_{m=1}^{M}\sum_{n=1}^{K_m}(\phi_{m,n})'' \\ \text{s.t.} \quad & C1'': (t_{m,n})'' \le \tau^{max}_{m,n}, 1 \le n \le K_m, \forall m \in \mathcal{M}, \\ & C4', C7, \\ & C6': ((d^{A2A}_{u,v})')^2 \ge (d_{min})^2, \forall u, v \in \mathcal{U}, u \ne v, \\ & C9': x_{m,u}(r^{G2A}_{m,u})' \ge 1, \forall m \in \mathcal{M}, \forall u \in \mathcal{U}, \\ & C10': y_{u,v}(r^{A2A}_{u,v})' \ge 1, \forall u, v \in \mathcal{U}, u \ne v, \end{aligned} \tag{34}$$

where

$$(\phi_{m,n})'' = \left((\phi^e_{m,n})'' I_{\{\lambda_{m,n}=0\}} + (\phi^c_{m,n})'' I_{\{\lambda_{m,n}=1\}}\right). \tag{35}$$

Notably, problem **P1.3′** is convex and can be solved by using YALMIP solver.

Algorithm 2 The joint optimization algorithm (JOA) for problem **P1**.

Input: $\mathcal{M}$, $\mathcal{U}$, Ω, L_{max}-the maximum iteration number, and ζ-an infinitesimal positive number.

Output: $\mathcal{P}^*$-the optimal solutions to **P1**, Φ^*-the total minimum computational overhead.

Initialize $l = 0$, generate an initial scheme of computation offloading decision $\mathcal{S}0$, bandwidth allocation $\mathcal{B}^0$ and drone position deployment $\mathcal{Q}^0$;
while $l \leq L_{max}$ **do**
  $l = l + 1$;
  calculate $\Phi^{l-1}(\mathcal{S}^{l-1}, \mathcal{B}^{l-1}, \mathcal{Q}^{l-1})$;
  solve **P1.1** to obtain $\mathcal{S}^l$ for given $\mathcal{B}^{l-1}$, and $\mathcal{Q}^{l-1}$;
  solve **P1.2′** to obtain $\mathcal{B}^l$ for given $\mathcal{S}^l$ and $\mathcal{Q}^{l-1}$;
  solve **P1.3′** to obtain $\mathcal{Q}^l$ for given $\mathcal{S}^l$ and $\mathcal{B}^l$;
  calculate $\Phi^l(\mathcal{S}^l, \mathcal{B}^l, \mathcal{Q}^l)$;
  if $|\Phi^l - \Phi^{l-1}| \leq \zeta$ **then return** $\mathcal{P}^* = \{\mathcal{S}^l, \mathcal{B}^l, \mathcal{Q}^l\}$ and $\Phi^* = \Phi^l$.
  end if
end while

D. Overall Algorithm

Based on the block coordinate descent method [33], a joint optimization algorithm (JOA) is devised by alternately solving the three sub-problems **P1.1**, **P1.2′**, and **P1.3′** in an iterative manner, which is described in Algorithm 2.

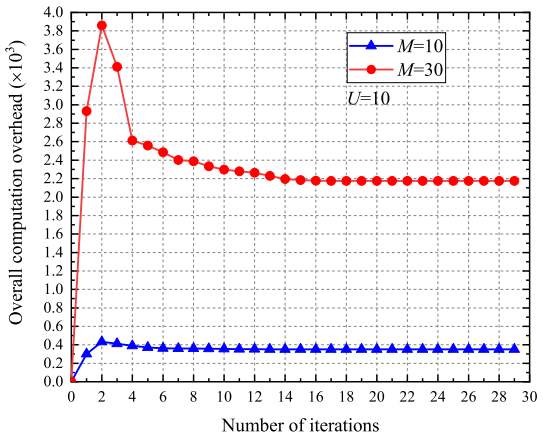
E. Computational Complexity Analysis

From the whole solving process of problem **P1** described above, one can see that the proposed joint optimization scheme uses an iterative method to obtain the minimum computation offloading overhead of all GIDs. The computational complexity of the overall scheme is mainly determined by the resolution progress of the three sub-problems via the while-loop in Algorithm 2. The heuristic greedy computation offloading optimization scheme in Algorithm 1, which is used to solve **P1.1**, has the computational complexity of $O((MK_m)^2)$, MK_m is the total number of the computation tasks. Problem **P1.2′** and **P1.3′** are solved by adopting a primal-dual interior point algorithm in YALMIP toolbox with complexity of $O((MK_m)^3 log(\zeta^{-1}))$ [18], where ζ is the accepted duality gap, i.e., the allowance error. Accordingly, if the maximum iteration numbers of Algorithm 2 is l_{max}, the overall computation complexity for the proposed joint optimization algorithm can be calculated as $O(l_{max}((MK_m)^2 + (MK_m)^3 log(\zeta^{-1})))$.

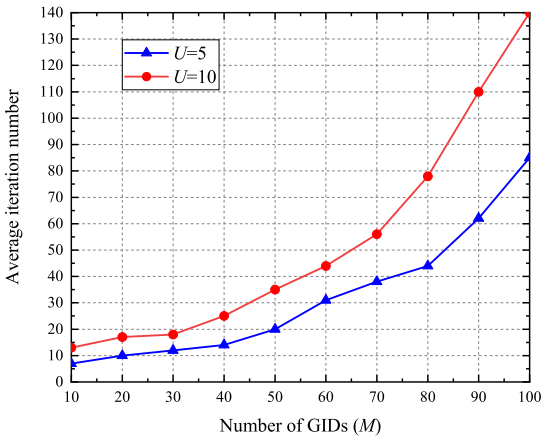
V. Numerical Results

A. Parameter Settings

Without loss generality, we consider a remote 3 km × 3 km square area in which 50 GIDs are randomly distributed. 10 drones are hovering at the height of 100 m ~ 150 m above this area, the minimum distance between UAVs is set as 100 m. Each drone has the same computation capacity of 6 GHz, and the CPU frequency of each GID is 0.8 ~ 1 GHz. There are total 10 types of computation tasks requested by all GIDs and each GID has two types of tasks to process. There are also 10 types of APPs deployed on all drones and each drone can host two types of APPs. The input data size of a task is set as 0.5 ~ 1 MB, and the CPU cycles for computing it is assumed to be 0.1 ~ 1 Gigacycles. The maximum tolerated accomplishing latency of each task is 0.1 ~ 2 s, and the weighted coefficients β^t and β^e are all set as 0.5. Furthermore, the transmit power of each GID and UAV are set as 0.1 W and 3 W, respectively. The LEO satellite's orbit height is set as 500 km, and its receiving antenna gain is set to be 45 dBi. The G2A, A2A, and the A2S channels are all working at C-band and the totally allocatable bandwidth is 200 MHz, and the AWGN is assumed to be −110 dBm.



(a)

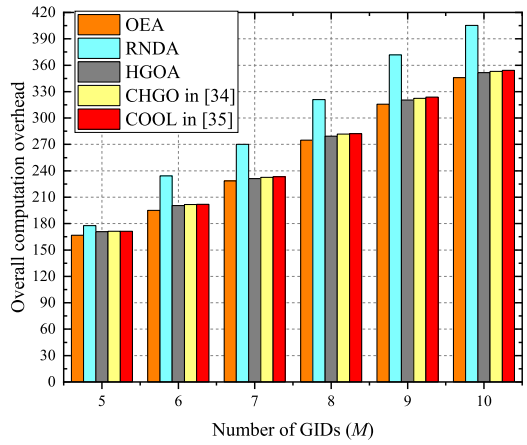


(b)

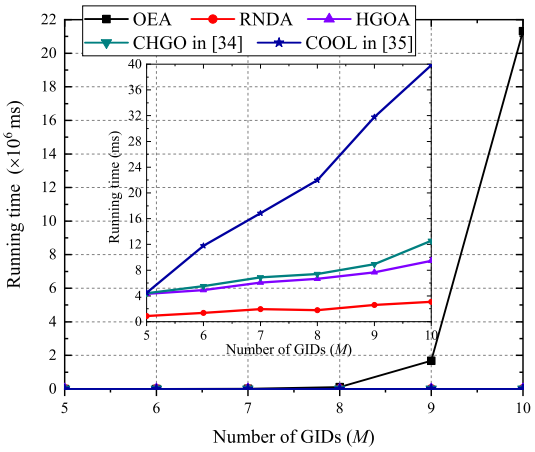
Fig. 2. Convergence illustration of the proposed joint optimization scheme, i.e., JOA: a) convergence behavior of JOA with regard to the overall computation overhead; b) average iterations for the convergence of JOA with different number of GIDs.

B. Simulation Results

Considering that our proposed joint optimization algorithm is an alternately iterative method, its convergence should be firstly analyzed. Fig. 2 presents the convergence behavior of the joint optimization scheme, i.e., JOA, with the growing



(a)



(b)

Fig. 3. Performance comparison of different task offloading algorithms, i.e., optimal enumeration algorithm, random algorithm, our proposed algorithm, and algorithms designed in [34] and [35], respectively: (a) comparison results of the overall computation overhead, (b) comparison results of the running time.

number of GIDs. Especially, Fig. 2(a) illustrates the convergence analysis of JOA with regard to the overall computation overhead, in which the numbers of GIDs M are set as 10 and 30, respectively. Fig. 2(b) depicts the average iterations for the convergence of JOA with different number of GIDs and drones. As can be seen from these figures, given different number of GIDs and drones, JOA can always obtain the minimum computation overhead after a certain number of iteration, which verifies that our proposed joint optimization scheme has a fast and stable convergence.

Next, we evaluate the effectiveness of our proposed task offloading algorithm (HGOA) by comparing the computation overhead with other offloading schemes, i.e., an optimal enumeration algorithm (OEA) and a random algorithm (RNDA). Further, other two task offloading methods, CHGO and COOL, which were designed in [34] and [35], respectively, and collaboratively utilized three computation models, are also adopted in the simulation. In particular, CHGO is a centralized offloading method in which each task first selects the computing model that can achieve the minimum computation overhead, and then the GID reports its offloading decision to the RCC. One of the tasks makes its current optimal offloading decision after the RCC broadcasts all the wireless connections to the drones and GIDs. While COOl is a decentralized game theory based offloading mechanism, which regards each task as a player and it can probably choose an appropriate decision given the offloading decision of other tasks. Considering the high computational complexity of OEA, the number of GIDs varies from 5 to 10. Fig. 3 demonstrates the performance comparison among these five algorithms. Fig. 3(a) depicts the comparison results of the overall computation overhead. As can intuitively be seen from this figure, given the same number of GIDs, HGOA is able to obtain the minimum computation overhead very close to OEA, and it has better performance than both CHGO and COOL. Since CHGO must report each GID's offloading decision to the RCC, which inevitably increases the processing latency, leading to higher computation overhead. Fig. 3(b) shows that our proposed offloading algorithm has a much shorter running time than that of OEA. Even compared to CHGO and COOL, HGOA still has a higher computational efficiency. It is verified from these two figures that HGOA can achieve a near-optimal computation overhead within a very short running time. Furthermore, although the random algorithm is the fastest offloading scheme, it obtains the worst performance of computation overhead.

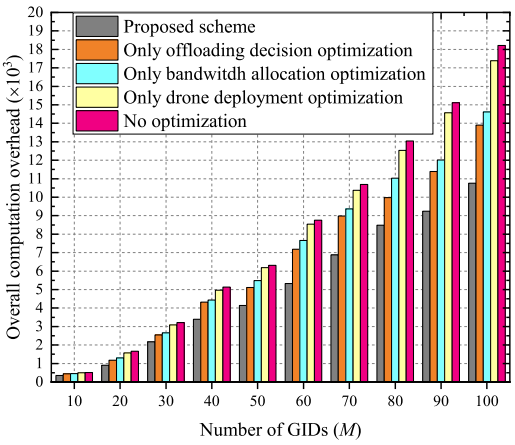


Fig. 4. Comparison results of overall computation overhead under different optimization schemes, i.e., only optimizing offloading decision, only optimizing bandwidth allocation, only optimizing drone deployment, no optimization, and our proposed one.

In order to evaluate our proposed joint optimization scheme, we provide some comparisons on the overall computation overhead of GIDs under diverse optimization schemes, including scheme only optimizing offloading decision, scheme only optimizing bandwidth allocation, scheme only optimizing drone deployment, scheme with no optimization, and our proposed one. Specially, in the non-optimized scheme, the random offloading decision, average bandwidth allocation, and random drone deployment are adopted. Fig. 4 demonstrates the

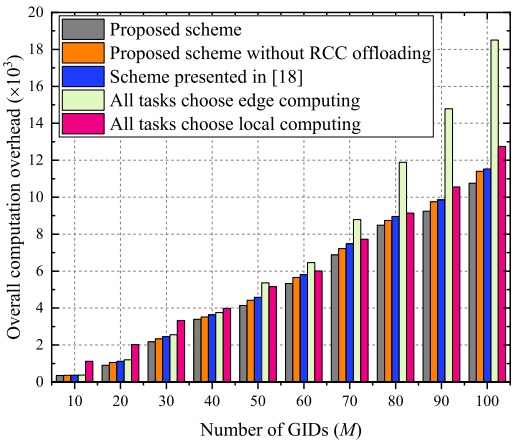


Fig. 5. Comparison results of overall computation overhead under different computing offloading schemes, i.e., computing all tasks locally, offloading all tasks to the MECSs, the proposed scheme, the proposed scheme without RCC computing, and the scheme in [18].

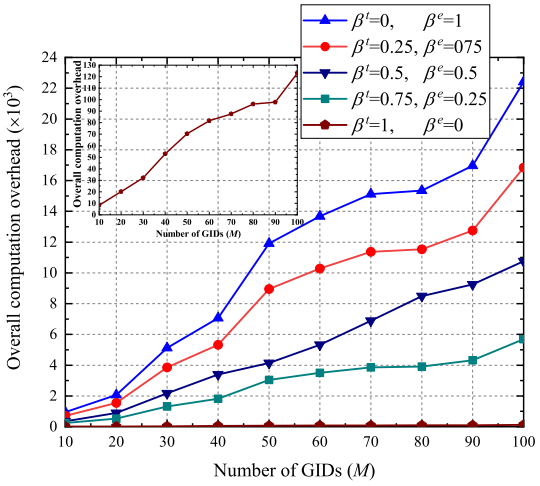


Fig. 6. Comparison results of overall computation overhead under different weighted coefficients.

compared results as the number of GIDs varying from 10 to 100. It can be seen from the comparison shown in this figure, when the number of GIDs is small, the overall computation overhead obtained by the joint optimization scheme is near to the that by other schemes, while as the GID number increasing, the gap between them becomes larger. Fig. 4 also shows that the scheme with only drone deployment optimization is slightly better than the one with no optimization, which has the worst performance. This is due the fact that, optimizing drone position deployment can improve the channel gains of G2A and A2A links so as to enhance the data delivery rate and reduce the transmission latency, but such improvement is inferior than that of the offloading decision and bandwidth allocation optimization. Finally, our proposed joint optimization scheme is much more preferred to the other four schemes as the scale of GIDs becomes larger.

To further validate the performance of our proposed task offloading scheme, we compare the achieved overall computation overhead by adopting different offloading schemes, i.e., computing all tasks locally, offloading all tasks to the MECSs, and our proposed scheme. In addition, to show the advantage of introducing RCC computing, the scheme of joint optimization without RCC model, is added into the comparison. What's more, we also introduce the scheme presented in [18] into the comparison, which jointly optimized offloading decision, resource allocation, and drone trajectory, but took no consideration for RCC model. Fig. 5 provides the comparison results among these five schemes. As seen from the figure, one can find that our proposed offloading scheme outperforms the other four ones. Moreover, when the number of GIDs is small, all tasks computing at the edge can produce less computation overhead than local computing thanks to the abundant computation resource on the MECSs. However, with the growth of GID number, the overhead of edge computing gradually exceeds that of local computing and increases extremely. Evidently, given the limited spectrum resource, a large number of GIDs choosing edge computing will greatly decrease the data rate, leading to longer transmission latency and larger computation overhead. Furthermore, our proposed scheme can obtain better performance than the one without RCC computing, which corroborates the necessity of introducing RCC computing model in the SAG-IoT MEC networks. Fig. 5 also verifies that, even without RCC model, our proposed joint optimization scheme is slightly superior to the one in [18].

Notice that the computation overhead consists of processing latency and energy consumption for the accomplishment of tasks, and the weighted coefficients β^t and β^e decide which component will dominate the final overhead. In the following, we set β^t and β^e as different values to verify how these coefficients effect on the overall computation overhead. Fig. 6 demonstrates the simulation results when the number of GIDs increasing from 10 to 100. One can easily observed from this figure that, given the same number of GIDs, with β^t increasing and β^e decreasing, the overall computation overhead monotonously declines. Especially when $\beta^t = 1$ and $\beta^e = 0$, the computation overhead will contain only processing latency. Therefore, the value settings of weighted coefficients should take into account the actual application scenarios of computation tasks. For instance, if the task is time sensitive, we should give priority to the processing latency and set β^t as a larger value.

To identify how the computing resource of drones affects the overall computation overhead, we change the number of APPs deployed on each drone and compare the obtained results. In this group of simulations, the number of GIDs is set as 50, the types of tasks requested by each GID and the APPs hosted on each drone are all vary from 2 to 10 with the increment of 2. Fig. 7 presents the comparisons of the calculated overall computation overhead under different numbers of task and APP types. It can be easily observed that, when the number of APP types hosted on each drone, i.e., K_u, is fixed, the overall computation overhead increases following the growth of task

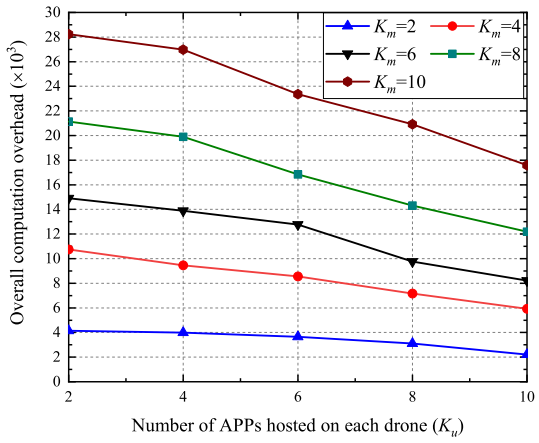


Fig. 7. Comparison results of overall computation overhead under different numbers of task types requested by each GID and the numbers of APP types hosted on each drone.

types requested by each GID, i.e., K_m. While for the same K_m, the overall overhead presents a monotone decreasing tendency with K_u growing. The reason is that if one drone hosts all types of APPs, it will has the capability of process all types of tasks and the inter-server data forwarding is unnecessary, which can significantly reduce the task's total accomplishing latency and the overall computation overhead.

VI. Conclusions

This paper mainly investigated the problem of inter-server task offloading and bandwidth allocation for computation overhead minimization in the multi-drone aided SAG-IoT network. The issue was formally defined as a constrained optimization problem according to communication and computing models. To efficiently tackle this problem, we first decomposed it into three sub-problems, and then leveraged heuristic greedy algorithm and successive convex approximation method to solve them. Finally, an iteratively joint optimization scheme was proposed by alternately optimizing offloading decision, bandwidth allocation, and drone deployment. Various numbers of GIDs and different optimization schemes as well as diverse offloading models have been adopted to evaluate the performance of our proposed algorithms. Numerical results demonstrated that, to the joint optimization problem of inter-server offloading and resource allocation, our presented solutions could achieve the minimum overall computation overhead of all GIDs in the multi-drone assisted SAG-IoT systems.

References

[1] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted Internet of things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4021, Mar. 2021.

[2] Y. Song, F. R. Yu, L. Zhou, X. Yang, and Z. He, "Applications of the Internet of things (IoT) in smart logistics: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4250–4274, Mar. 2021.

[3] 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.

[4] M. Ijaz, G. L. L. Lin, O. Cheikhrouhou, H. Hamam, and A. Noor, "Integration and applications of fog computing and cloud computing based on the Internet of things for provision of healthcare services at home," *Electronics*, vol. 10, no. 9, pp. 1–12, May 2021.

[5] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, Mar. 2017.

[6] M. Shafi *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Apr. 2017.

[7] N. Cheng, F. Lyu, W. Quan *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[8] Y. Chen, B. Ai, Y. Niu, H. Zhang, and Z. Han, "Energy-constrained computation offloading in space-air-ground integrated networks using distributionally robust optimization," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12113–12125, Nov. 2021.

[9] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tut.*, vol. 20, no. 4, pp. 2714–2741, 4th Quart. 2018.

[10] Y. H. Kim, I. A. Chowdhury, and I. Song, "Design and analysis of UAV-assisted relaying with simultaneous wireless information and power transfer," *IEEE Access*, vol. 8, pp. 27 874–27 886, Feb. 2020.

[11] Y. Shi, J. Liu, Z. M. Fadlullah, and N. Kato, "Cross-layer data delivery in satellite-aerial-terrestrial communication," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 138–143, Jul. 2018.

[12] Q. Zhang, J. Chen, L. Ji, Z. Feng, Z. Han, and Z. Chen, "Response delay optimization in mobile edge computing enabled UAV swarm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3280–3295, Jan. 2020.

[13] Z. Lv, J. Hao, and Y. Guo, "Energy minimization for mec-enabled cellular-connected UAV: Trajectory optimization and resource scheduling," in *Proc. IEEE INFOCOM WKSHPS*, Jul. 2020, pp. 1–9.

[14] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Inf.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020.

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

[16] J. Xiong, H. Guo, and J. Liu, "Task offloading in UAV-aided edge computing: Bit allocation and trajectory optimization," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 538–541, Mar. 2019.

[17] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of things at the edge," *IEEE Trans. Ind. Inf.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[18] J. Zhang, L. Zhou, F. Zhou, B.-C. Seet, H. Zhang, Z. Cai, and J. Wei, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[19] Y. Guo, S. Gu, Q. Zhang, N. Zhang, and W. Xiang, "A coded distributed computing framework for task offloading from multi-UAV to edge servers," in *Proc. IEEE WCNC*, 2021, pp. 1–6.

[20] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an UAV network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[21] C. Zhan, H. H. Z. Liu, Z. Wang, and S. Mao, "Multi-UAV-enabled mobile edge computing for time-constrained IoT applications," *IEEE Internet Things J.*, pp. 1–15, 2021, in press, DOI 10.1109/JIOT.2021.3073208.

[22] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3977, Sep. 2020.

[23] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM, Orlando, FL, USA*, 25-30, Mar. 2012, pp. 1431–1439.

[24] W. Sun, J. Liu, Y. Yue, and P. Wang, "Joint resource allocation and incentive design for blockchain-based mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 6050–6064, Sep. 2020.

[25] R. Khan and D. N. K. Jayakody, "Full duplex component-forward cooperative communication for a secure wireless communication system," *Electronics*, vol. 9, no. 2, pp. 1–17, Dec. 2020.

[26] Y. Shi, Y. Xia, and Y. Gao, "Joint gateway selection and resource allocation for cross-tier communication in space-air-ground integrated IoT networks," *IEEE Access*, vol. 9, pp. 4303–4314, Feb. 2021.

[27] 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.
[28] P. Li and J. Xu, "UAV-enabled cellular networks with multi-hop backhauls: Placement optimization and wireless resource allocation," in *Proc. IEEE ICCS*, 2018, pp. 110–114.
[29] C. Li and X. Tang, "On fault-tolerant bin packing for online resource allocation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 817–829, OCt. 2020.
[30] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *In Proc. IEEE CACSD*, 2004.
[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[32] Y. Yang and M. Pesavento, "A unified successive pseudoconvex approximation framework," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.
[33] Z. Kang, C. You, and R. Zhang, "3d placement for multi-UAV relaying: An iterative gibbs-sampling and block coordinate descent optimization approach," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 2047–2062, Mar. 2021.
[34] H. Guo, J. Zhang, and J. Liu, "Fiwi-enhanced vehicular edge computing networks: Collaborative task offloading," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 45–53, Mar. 2019.
[35] B. Chen, H. Zhou, J. Yao, and H. Guan, "RESERVE: An energy-efficient edge cloud architecture for intelligent multi-UAV," *IEEE Trans. Services Comput.*, pp. 1–14, 2019, in press, DOI: 10.1109/TSC.2019.2962469.

Yongpeng Shi (Member, IEEE) received his B.S. degree in electronic information science from Shaanxi Normal University in 2001, and M.S. and Ph.D. degree in computer science from Xidian University in 2008 and 2018, respectively. He is currently an Associate Professor in the School of Physics and Electronic Information, Luoyang Normal University. His research interests cover space-air-ground integrated network, edge computing, SDN and NFV.

Junjie Zhang received his B.S. degree in electronic information science from Shaanxi Normal University in 2001, and M.S. in microelectronics from Xiangtan University in 2008, respectively. He is currently a Lecturer in the School of Physics and Electronic Information, Luoyang Normal University. His main research interest is intelligent information processing

Ya Gao (Member, IEEE) received M.S. degree in Information and Communications Engineering from Central South University, Changsha, China, in 2010, and Ph.D. degree in Information and Communications Engineering from Xidian University, Xi'an, China, in 2018, respectively. She worked as a visiting Postgraduate Student at Institute of Computing Technology Chinese Academy of Sciences, Beijing, China, from 2008 to 2010. She worked with the Luoyang Normal University since 2010. Her research interests focus on B5G/6G wireless networks with emphasis on statistical QoS provisioning, energy efficient wireless networks, wireless powered communications networks, the Internet of things networks, and ultra dense networks.

Yujie Xia received his B.S. degree in electronic engineering from Henan Normal University, Xinxiang, China, in 2001 and his M.S. degree in communication and information systems from Harbin Engineering University, Harbin, China, in 2004. He received his Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2014. Since 2004, He has been with the Luoyang Normal University, Luoyang, China. His research interests are in the area of the next generation wireless communications, communication signal processing and multiple user access technology.