

# Computation-Efficient Aerial-Marine Integrated Networks for Search and Rescue via Cooperative HAPS, UAVs, and MASSs

Zhen Wang<sup>ID</sup>, Graduate Student Member, IEEE, Bin Lin<sup>ID</sup>, Senior Member, IEEE,  
and Qiang Ye<sup>ID</sup>, Senior Member, IEEE

**Abstract**—In this paper, we propose an aerial-marine integrated network architecture tailored for maritime search and rescue (MSAR) operations, where a high altitude platform station (HAPS) and maritime autonomous surface ships (MASSs) are equipped with edge servers, providing computing services for surveillance uncrewed aerial vehicles (UAVs). These UAVs generate substantial volumes of computation-intensive and delay sensitive tasks, some of which can be strategically offloaded to both MASSs and HAPSs for distributed processing. We formulate the joint task offloading and resource allocation as a mixed-integer nonlinear program (MINLP) to minimize the system computation overhead (CO), quantified by a weighted sum of task completion time and energy consumption. Given the stochastic nature of task arrivals from UAVs and the dynamic characteristics of marine channel conditions, we employ a two-stage optimization to decompose the problem into four tractable subproblems, i.e., 1) edge server selection, 2) transmission power allocation, 3) edge computing resource allocation, and 4) local computation resource allocation. In stage I, we optimize the edge server selection via many-to-one matching. In stage II, we determine the transmission power of each UAV with quasi-convex optimization and solve the edge computing and local computing resource allocation subproblems via a projected gradient descent method and convex optimization, respectively. Simulation results validate that the proposed scheme can achieve superior system performance over benchmark schemes.

**Index Terms**—Aerial-marine integrated networks, UAV, HAPS, maritime search and rescue, task offloading, MEC, server selection, computing resource allocation, MINLP.

## I. INTRODUCTION

MARITIME search and rescue (MSAR) operations are critical for ensuring safety and minimizing casualties in emergency situations at sea [1], [2]. It is paramount to perform timely search-and-rescue missions within the golden

Received 12 June 2025; revised 18 October 2025; accepted 30 November 2025. Date of publication 9 December 2025; date of current version 13 January 2026. This work was supported by the National Natural Science Foundation of China (No. 62371085 and No.51939001) and the Scientific research fund of Liaoning Provincial Department of Education (LJ212513631002). The associate editor coordinating the review of this article and approving it for publication was L. Wang. (*Corresponding author: Bin Lin.*)

Zhen Wang is with the Intelligent and Electronic Engineering College, Dalian Neusoft University of Information, Dalian 116023, China (e-mail: wangzhen\_jsj@neusoft.edu.cn).

Bin Lin is with the Information Science and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: binlin@dltmu.edu.cn).

Qiang Ye is with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: qiang.ye@ucalgary.ca).

Digital Object Identifier 10.1109/TCCN.2025.3642113

window of post-disaster rescue. In recent years, MSAR operations have faced increasing challenges due to the expanding scope of maritime activities, the complexity of rescue missions, and the vast and unpredictable nature of maritime environments [3], [4]. Traditional MSAR methods, which rely heavily on human resources and terrestrial networks, are often limited by geographical constraints, weather conditions, and availability of rescue assets [5]. In general, an effective MSAR operation necessitates an immediate response, including swift network deployment, real-time information collection, and low-latency data processing [6]. Nevertheless, the scarcity of conventional maritime communication and computing resources has posed potential obstacles to rescue operations [7], [8], [9]. To address these challenges, there is a growing interest in leveraging advanced technologies to enhance the efficiency and effectiveness of MSAR operations [10].

Multi-access edge computing (MEC) has emerged as a transformative technology that brings computational resources closer to the network edge, enabling low-latency and high-efficiency task processing [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. The integration of MEC into maritime networks can significantly enhance the network capability to handle computation-intensive and delay-sensitive tasks. Given the inherently limited maritime resources, most existing maritime MEC studies have established network models supported by shore-based, air-based, or space-based infrastructures as the communication foundation for optimizing maritime MEC network performance [5], [21], [22]. Qi et al. proposed an aerial-assisted maritime rescue system employing multiple surveillance UAVs and a relay UAV for disaster data acquisition in [5]. The study jointly optimizes task offloading decisions, relay UAV deployment, and surveillance UAV-rescue target associations to minimize total system latency. In [21], Zhang et al. proposed a shore-based edge computing network architecture, where an uncrewed surface vessel (USV) equipped with edge servers provides computing services for offshore users and can offload partial tasks to shore-based base stations (BSs) for processing. A three-tier edge computing network architecture for maritime devices is proposed in [22], where computational tasks generated by marine devices can be either processed locally, offloaded to satellites, or further relayed to shore-based BSs for further execution.

In the context of MSAR, the dynamic and resource-constrained nature of maritime environments necessitates efficient task offloading and resource allocation strategies to optimize the performance of maritime networks for MSAR [23]. The integration of aerial computing platforms, i.e., uncrewed aerial vehicles (UAVs) and high altitude platform station (HAPS) systems into MSAR operations has emerged as a promising solution, offering enhanced mobility, flexibility, and coverage [24], [25], [26]. UAVs are highly practical for deployment in disaster-affected areas due to their exceptional maneuverability, flexible deployment capabilities, reliable line-of-sight (LoS) communication, and cost efficiency [27], [28]. They play a critical role in supporting rescue missions, including disaster area monitoring, real-time data collection, and aerial search and rescue operations [29]. However, the limited onboard battery capacity and computing capabilities of UAVs can significantly restrict their endurance and efficiency when performing computation-intensive tasks required for rescue missions.

Recently, HAPS systems have emerged as key enablers for six generation (6G) networks [30], [31], [32], offering unique capabilities as stratospheric wireless communication stations operating at altitudes of around 20 km. These aerial platforms, such as airships or balloons, can provide a bird's-eye line-of-sight (LoS) view over extensive ground areas, enabling wide coverage with a radius of 50-500 km [33]. With their large payload capacity (typically  $\geq 100$  kg), HAPS can be equipped with advanced resources, i.e., large batteries, antennas, and computation and storage resources, making them powerful supplements for maritime MEC [34]. Considering the limitations of UAVs in terms of energy and computational prowess, a practical approach to enhance MEC efficiency entails allocating UAVs' tasks to both HAPS and sea surface stations for subsequent processing.

However, establishing UAV-HAPS-assisted maritime MEC for MSAR faces technical challenges. First, the dynamic nature of task generation from UAVs, coupled with the ever-fluctuating communication quality of maritime wireless channels, are susceptible to various marine environmental influences. Second, the aerial-marine integrated network comprises a multitude of computation options, making it particularly challenging to select suitable computation equipment and optimally utilize the available resources [35]. Third, there are multidimensional resources in aerial-marine integrated network, such as the computation resources of HAPS and sea surface stations, and the transmission power of UAVs. Efficiently allocating these resources and maximizing their utilization to enhance network performance poses a formidable challenge.

To overcome the aforementioned challenges, we propose a novel aerial-marine integrated network architecture for MSAR and introduce an approach for joint optimization of task offloading and resource allocation to minimize the system computation overhead (CO). The main contributions are summarized as follows:

- *System Architecture:* We delve into the exploration of an aerial-marine integrated network architecture for MSAR, wherein multiple surveillance UAVs installed

with lightweight computing resources are dispatched to a disaster area for an MSAR mission. A HAPS equipped with extensive computing resources is deployed to provide aerial edge capability to the UAVs. Concurrently, numerous maritime autonomous surface ships (MASSs) equipped with edge servers are deployed to the disaster areas to assist UAVs in processing tasks at closer distances. The computation-intensive tasks of UAVs can be offloaded to the HAPS or an MASS via multi-access approach.

- *Problem Formulation:* We propose a novel joint optimization framework for task offloading and resource allocation, formulated as a mixed-integer nonlinear program (MINLP). The objective is to minimize the system CO, defined theoretically as a weighted-sum of task completion time and energy consumption.
- *Algorithm Design:* Given the complexity of directly solving the formulated problem, we employ a two-stage optimization and decompose the primal problem into four subproblems: 1) sever selection; 2) transmission power allocation; 3) edge computing resource allocation; 4) local computing resource allocation. The first subproblem is addressed using a many-to-one matching approach, while the second subproblem is solved through quasi-convex optimization techniques. For the third subproblem, we apply a projected gradient descent method, and the fourth subproblem is resolved using convex optimization. On this basis, we introduce a joint computation offloading and resource allocation (JCORA) algorithm to optimize the total CO of the system.
- *Performance Evaluation:* We conduct extensive simulations to validate the effectiveness of the proposed solution. The results demonstrate that our proposed algorithms can minimize the CO of the system and improve the efficiency in comparison with the benchmark schemes.

The remainder of this paper is organized as follows. Section II summarizes the related work. The system model is presented in Section III. The problem formulation is given in Section IV. Section V presents the problem transformation and solution design. Section VI provides the performance evaluation results, and Section VII concludes the paper.

## II. RELATED WORK

In this section, we review existing research works related to aerial MEC and marine MEC approaches, respectively. The differences between the related works and our work is summarized in Table I.

### A. Aerial MEC

Aerial MEC mainly explores the integration of UAVs, HAPS, and other aerial platforms to enhance edge computing capabilities in challenging environments [1], [24], [36], [37]. Researchers have focused on leveraging the mobility, flexibility, and LoS advantages of aerial platforms to provide computational and communication support in areas with limited terrestrial infrastructure, such as remote regions,

TABLE I  
COMPARISON BETWEEN RELATED WORK AND OUR WORK

References	Search and rescue scenario	Offloading route (UAV→ HAPS)	Offloading route (UAV→ MASS)	Optimizing latency	Optimizing energy consumption	Task offloading	Resource allocation
[1]	✓	✗	✗	✓	✓	✓	✓
[5]	✓	✗	✗	✓	✗	✓	✗
[24]	✓	✗	✗	✓	✗	✓	✗
[36]	✗	✗	✗	✗	✗	✓	✓
[37]	✗	✗	✗	✓	✗	✓	✗
[38]	✓	✗	✗	✓	✓	✓	✓
[39]	✗	✗	✗	✗	✓	✓	✓
[40]	✗	✗	✗	✗	✗	✓	✓
[41]	✗	✗	✗	✗	✗	✓	✓
[42]	✗	✗	✗	✗	✓	✓	✓
[43]	✗	✗	✗	✗	✓	✓	✓
[44]	✗	✗	✗	✗	✓	✓	✓
[45]	✗	✗	✗	✗	✓	✓	✓
[46]	✗	✗	✗	✗	✗	✓	✓
[47]	✗	✓	✗	✗	✗	✓	✗
[48]	✗	✗	✓	✗	✓	✓	✓
[49]	✗	✗	✗	✗	✓	✓	✗
[50]	✗	✗	✗	✗	✓	✓	✗
[51]	✗	✗	✗	✗	✓	✓	✓
[52]	✓	✗	✗	✓	✓	✓	✓
[53]	✗	✗	✗	✓	✗	✓	✓
[54]	✗	✓	✗	✗	✓	✓	✗
[55]	✓	✗	✓	✓	✗	✓	✓
Our work	✓	✓	✓	✓	✓	✓	✓

disaster zones, and maritime environments. Some studies have demonstrated the potential of UAVs as flying edge servers, offering low-latency processing and data offloading for terminal devices [38], [39], [40], [41]. For instance, Liu et al. proposed a multiple input single output (MISO) UAV-assisted MEC network framework to tackle the challenge of degraded channel quality in traditional MEC networks caused by multipath propagation and blockages [39]. Zhao et al. investigated the deployment of UAVs to collaboratively provide edge computing services, jointly optimizing UAV trajectories and task management strategies to maximize the system throughput [40]. The HAPSs have also been explored as stable, long-endurance platforms for aerial MEC, capable of covering large areas and supporting heavy computational workloads [42], [43], [44], [45], [46], [47]. For instance, Ding et al. investigated a satellite-aerial integrated edge computing network to combine a low-earth-orbit(LEO) satellite and HAPSs to provide edge computing services for ground user equipment [42]. Lakew et al. proposed a heterogeneous aerial access Internet of Things (IoT) network comprising a HAPS and multiple UAVs with computation and communication capabilities to provide task offloading services for the resource-limited IoT Devices [43]. Jia et al. proposed a hierarchical model composed of an HAP and multi-UAVs in [54] to minimize the total energy cost with the chance constraint for remote ground users.

### B. Marine MEC

Marine MEC mainly focuses on extending edge computing capabilities to maritime environments, where traditional infrastructure is often unavailable or impractical. Researchers have explored the integration of uncrewed surface vehicles (USVs) [48], UAVs [49], HAPSs [44], autonomous underwater vehicles (AUVs) [50], LEO satellites [51] and other edge platforms to address the unique challenges of maritime operations, such as search and rescue, environmental monitoring, and vessel traffic management. In [48], Zeng et al. proposed an

energy-efficient USV fleets-assisted collaborative computation offloading scheme for smart maritime services, where the USVs were equipped with computing resources to provide edge services for the UAVs. In [52], Lyu et al. formulated a joint task offloading and resource allocation problem for a UAV emergency communication scenario, aiming to minimize the CO of the terminal device. In [44], Li et al. exploited the secure computation offloading with the assistance of a HAPS to perform computation offloading and provide cooperative jamming for USVs. Dai et al. proposed a hybrid offshore and aerial-based MEC network to enhance the offloading efficiency and minimize the maximum workloads latency of marine devices [53]. You et al. proposed a cooperative MEC framework composed of UAVs and vessels in [55] to minimize the total execution time, with consideration of uncertain MIoT tasks.

Although many existing studies focus on aerial MEC and marine MEC, the utilization of both HAPS and MASSs to create multiple layers of edge computing services for UAVs, especially for MSAR operations, need further investigation. In this paper, leveraging the high maneuverability and flexible deployment of UAVs, we utilize them as surveillance devices for MSAR and propose an innovative aerial-marine integrated network architecture, which enables the UAVs to offload computation-intensive tasks to either HAPS or MASSs for further processing.

## III. SYSTEM MODEL

This section first introduces an aerial-marine integrated network, where the UAVs can offload their tasks to an MASS or a HAPS edge for further processing. Then, the communication and computing models are presented.

### A. Network Model

We consider an aerial-marine integrated network with MEC functionality, consisting of one HAP and multiple MASSs,

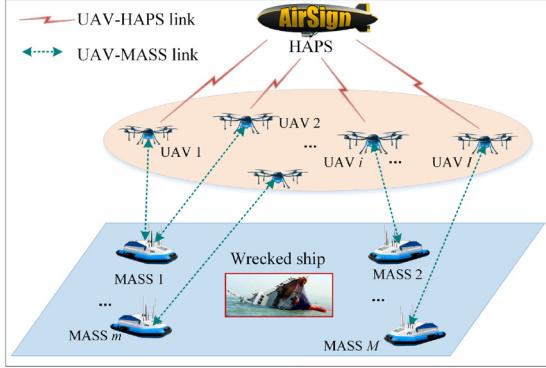


Fig. 1. Network model.

cooperatively supporting task offloading for connected UAVs, as shown in Fig. 1. In the considered scenario, a group of UAVs equipped with cameras are distributed to collect data from the disaster areas according to the pre-set trajectories. Meanwhile, an HAP and a certain number of MASSs equipped with edge servers are dispatched to receive the critical data collected by UAVs. Moreover, each MASS is assigned a dedicated service area that maintains non-overlapping boundaries with adjacent MASSs. Additionally, all UAVs maintain a constant altitude of  $H_0$  to minimize unnecessary energy expenditure associated with frequent altitude adjustments in response to potential obstacles [38]. We posit a one-to-one correspondence between HAPS and MASSs with edge servers. The set of UAVs and edge servers are denoted as  $\mathcal{I} = \{1, 2, \dots, i, \dots, I\}$  and  $\mathcal{M} = \{0, 1, 2, \dots, m, \dots, M\}$ , respectively. Therefore, there are  $M+1$  edge servers available to serve the UAVs. For convenience, we denote the  $m$ -th edge server as server  $m$ , where  $m = 0$  indicates the edge server connected with the HAP, and others mean edge servers connected with the MASSs. Each edge server is designed to provide communication and computing services for UAVs under its coverage, the number of which is denoted by  $\Lambda_m$  ( $0 \leq \Lambda_m \leq \Lambda_m^{\max}$ ).

Typically, UAVs generate a significant volume of computation-intensive tasks at the onset of each time slot, while being constrained by limited onboard computational resources and battery capacity. To address these limitations, UAVs can leverage the computational capabilities of either HAPS or MASS platforms through task offloading mechanisms. The server selection decision for UAV  $i$  is defined as  $s_i^m$ , where  $s_i^m = 1$  indicates that UAV  $i$  selects server  $m$  to handle tasks in the time slot  $t$ , or else  $s_i^m = 0$ . The set  $\mathbf{s} = \{s_i^m | i \in \mathcal{I}\}$  indicates the server selection decisions of all UAVs. In addition, the transmission power of UAV  $i$  during communication with an edge server is represented as  $p_i$ , and  $0 \leq p_i \leq p_i^{\max}$ . The set  $\mathbf{p} = \{p_i | i \in \mathcal{I}\}$  indicates the transmission power of all UAVs.

Let  $a_i$  denote the number of task arrivals at UAV  $i$ , which is randomly drawn from  $[0, a^{\max}]$  to capture the temporal variations in task arrivals.  $a^{\max}$  is the upper bound of the number of arriving tasks. The UAVs may request for different types of tasks which vary in input data size and required CPU cycles. To simplify the system model, we assume that each task has the same size of  $D$  (in bits) and the expected number

of CPU cycles required by one bit is  $\alpha$  [56]. Additionally, all the tasks generated by UAV  $i$  must be completed within its latency requirement  $T_i^{th}$ .

The three-dimensional Cartesian coordinate system is employed to represent the position coordinates of the HAPS, UAVs and MASSs. Generally, HAPSs can endure at a fixed position around the altitude of 20 km for several months, which can serve as stable BSs in the air due to large coverage and powerful payloads [57]. The position of an HAPS is expressed as  $\mathbf{q}_{HAP} = (x_H, y_H, z_H)$ . The position of UAV  $i$  is denoted as  $\mathbf{q}_i = (x_i, y_i, H_0)$ . Considering the safety distance between the UAVs, the position of each UAV should satisfy  $\|\mathbf{q}_i - \mathbf{q}_j\| \geq d^{\min}$ . Similarly, the position of MASS  $m$  is denoted as  $\mathbf{q}_m = (x_m, y_m, z_m)$ .

### B. Communication Model

The communication model consists of UAV-to-MASS communication and UAV-to-HAPS communication. The orthogonal frequency-division multiple access (OFDMA) protocol is employed for each MASS or the HAP channel access.

1) *UAV-to-MASS Communication Model:* Given the relatively open offshore environment, characterized by stronger direct signals, and considering that the primary factors influencing overseas radio channels are the multipath effect due to sea volatility and the impact of extreme weather, we regard the communication between UAVs and MASSs as air-to-sea channels, which exhibit Rician fading [58]. This fading can be viewed as a composite of large-scale and small-scale fading [59].

The large-scale path loss model is expressed as [60]

$$L_i^M |_{dB} = L_0 + 10\lambda \log_{10} \left( \frac{d_i^m}{d_0} \right) + X_\sigma + \varrho F, \quad (1)$$

where  $d_i^m = \sqrt{(x_m - x_i)^2 + (y_m - y_i)^2 + H_0^2}$  denotes the distance between UAV  $i$  and server  $m$  ( $m = 1, 2, \dots, M$ ),  $L_0$  is the path loss at the reference distance  $d_0$ ,  $\lambda$  denotes the path-loss exponent which could be reduced to less than 2 due to the ducting effect over the sea surface [60],  $X_\sigma$  describes the shadow fading caused by, e.g., sea waves under high sea state conditions,  $F$  is an adjustment parameter to incorporate the fast moving of UAVs, and the value of  $\varrho$  is set to 1 or  $-1$  depending on the moving direction of the UAVs (towards or away from the ground site) [61].

The small-scale Rician fading  $\tilde{u}_i^m$  is represented as

$$\tilde{u}_i^m = \sqrt{\frac{K_R}{1 + K_R}} + \sqrt{\frac{1}{1 + K_R}} g_i^m, \quad (2)$$

where  $g_i^m \in \mathcal{CN}(0, 1)$  and  $K_R$  is the Rician factor. On the ocean, MASSs typically navigate along established shipping routes, from which historical or pre-measured data can be obtained. Leveraging this derived data, we establish a correlation between specific locations and channel state information (CSI). With this correlation, we can determine the corresponding CSI for each location. Hence, we postulate that information on path loss and the Rician factor is accessible [59].

Then, the channel coefficient is expressed as

$$h_i^m = (L_i^M)^{-1/2} \tilde{u}_i^m, \quad (3)$$

The channel gain between UAV  $i$  and server  $m$  ( $m = 1, 2, \dots, M$ ) is expressed as

$$H_i^m = G^U G^M |h_i^m|^2, m = 1, 2, \dots, M, \quad (4)$$

where  $G^U$  and  $G^M$  are the antenna gain of UAVs and MASSs, respectively.

*2) UAV-to-HAPS Communication Model:* In the UAV-HAPS communication model, each UAV communicates with one HAPS, which hovers at a fixed height in the stratosphere. According to [62], the path loss when UAV  $i$  communicates with the HAPS is calculated as

$$\begin{aligned} L_i^H &= 20 \log_{10} \left( \frac{4\pi f_i d_i^H}{c} \right) + \gamma_i^{LoS} \delta_i^{LoS} \\ &\quad + (1 - \gamma_i^{LoS}) \delta_i^{NLoS}, \end{aligned} \quad (5)$$

where  $f_i$  denotes the carrier frequency,  $c$  is the speed of light,  $d_i^H = \sqrt{(x_H - x_i)^2 + (y_H - y_i)^2 + (z_H - H_0)^2}$  means the distance between UAV  $i$  and the HAP,  $\delta_i^{LoS}$  and  $\delta_i^{NLoS}$  denote LoS and non-LoS (NLoS) link loss when UAV  $i$  communicates with the HAPS, respectively.  $\gamma_i^{LoS}$  denotes the LoS communication probability, calculated as

$$\gamma_i^{LoS} = \frac{1}{1 + \chi_1 \exp[-\chi_2 (\varphi_i - \chi_1)]} \quad (6)$$

where  $\varphi_i = \tan^{-1} \left[ (z_H - H_0) / \sqrt{(x_H - x_i)^2 + (y_H - y_i)^2} \right]$  means the elevation angle of UAV  $i$  to the HAPS, and the values of  $\chi_1$  and  $\chi_2$  are determined by the state of the environment. Without loss of generality, the channel gain when UAV  $i$  communicates with the HAPS is expressed as [57]

$$H_i^m = 10^{-L_i^H/10}, m = 0. \quad (7)$$

In summary, according to Shannon theorem, the transmission rate (link capacity) between UAV  $i$  and edge server  $m$  is calculated by

$$R_i^m = s_i^m W_i^m \log_2 \left( 1 + \frac{p_i H_i^m}{\sigma^2} \right), m = 0, 1, \dots, M \quad (8)$$

where  $W_i^m$  denotes the communication bandwidth allocated to UAV  $i$ .

### C. Computing Model

*1) Local Computing Model:* When the server selection decision  $s_i^m = 0$ , UAV  $i$  decides to process  $a_i$  tasks locally. We define  $\rho_i$  as the computation capacity of UAV  $i$ , which satisfies  $0 \leq \rho_i \leq \rho_i^{\max}$ . Thus, the computation completion time of  $a_i$  tasks is expressed as

$$t_i^{loc} = \frac{a_i D \alpha}{\rho_i}. \quad (9)$$

The energy consumed by local computation hinges on the integrated chip architecture of the UAVs [63]. Then, the energy consumed by local computation of UAV  $i$  is calculated as

$$E_i^{loc} = \varepsilon_i (\rho_i)^2 a_i D \alpha, \quad (10)$$

where  $\varepsilon_i$  means the effective switched capacitance [63].

*2) Edge Computing Model:* When the server selection decision  $s_i^m = 1$ , UAV  $i$  decides to offload  $a_i$  tasks to an edge server for further processing. The process of task offloading typically comprises three distinct phases: 1) UAV  $i$  conveys the tasks to an edge server via a OFDMA-based uplink; 2) the edge server acknowledges receipt of the computational tasks and allocates the necessary computational resources for task execution; 3) once the tasks have been successfully computed, the edge server transmits the result back to UAV  $i$ . Since the output of a computational task is generally significantly smaller in size compared to its input [52], [56], we omit the third phase.

The edge computing completion time  $t_i^{edge}$  of UAV  $i$  consists of two components, expressed as

$$t_i^{edge} = t_i^{off} + t_i^{com}, \quad (11)$$

where  $t_i^{off}$  and  $t_i^{com}$  denote the task offloading time and the task processing time, respectively.  $t_i^{off}$  mainly depends on the size of tasks and the transmission rate, calculated as

$$t_i^{off} = \frac{a_i D}{R_i^m}. \quad (12)$$

The corresponding energy dissipation required for offloading the computation tasks of UAV  $i$  is calculated as

$$E_i^{off} = p_i t_i^{off} = \frac{p_i a_i D}{R_i^m}. \quad (13)$$

$t_i^{com}$  mainly depends on the computation resource allocated to UAV  $i$ . We denote the computation capacity of edge server  $m$  (in the unit of CPU cycles per second) as  $F_m$ , and the ratio of computing resources allocated to UAV  $i$  as  $\beta_i^m \in [0, 1]$ , satisfying  $\sum_{i \in \mathcal{I}} \beta_i^m \leq 1$ . When  $a_i$  tasks of UAV  $i$  are offloaded to edge server  $m$ , the task execution time is calculated as

$$t_i^{com} = \frac{a_i D \alpha}{\beta_i^m F_m}. \quad (14)$$

The corresponding energy dissipation required for processing the computation offloading of UAV  $i$  at edge server  $m$  is given by

$$E_i^{com} = \varepsilon_m (\beta_i^m F_m)^2 a_i D \alpha, \quad (15)$$

where  $\varepsilon_m$  is the effective switched capacitance of edge server  $m$ . Then, Therefore, the total energy consumption of UAV  $i$  for edge computing is expressed as

$$E_i^{edge} = E_i^{off} + E_i^{com} = \frac{p_i a_i D}{R_i^m} + \varepsilon_m (\beta_i^m F_m)^2 a_i D \alpha. \quad (16)$$

## IV. PROBLEM FORMULATION

In an aerial-marine integrated MEC scenario, the CO of each UAV is mainly characterized by the task completion time  $T_i$  and energy consumption  $E_i$ , obtained as

$$T_i = s_i^m t_i^{edge} + (1 - s_i^m) t_i^{loc} \quad (17)$$

and

$$E_i = s_i^m E_i^{edge} + (1 - s_i^m) E_i^{loc}, \quad (18)$$

respectively. Our goal is to minimize the CO of all the UAVs. Considering the preferences of different UAVs, we define the CO function of UAV  $i$  as

$$\mathcal{G}_i = \mu_i^t T_i + \mu_i^e E_i, \quad (19)$$

where  $\mu_i^t, \mu_i^e \in [0, 1]$ , with  $\mu_i^t + \mu_i^e = 1, \forall i \in \mathcal{I}$ , indicate UAV  $i$ 's preference on task completion time and energy consumption, respectively, which are determined by the residual battery energy of the UAV itself and the requirement for task completion time. For instance, an UAV experiencing short battery life can opt to increase  $\mu_i^e$  while decreasing  $\mu_i^t$ , thereby conserving more energy at the cost of extending the task completion time.

Based on the system model described in Section III, we aim to minimize the CO of all the UAVs (denoted as  $\mathcal{G} = \sum_{i \in \mathcal{I}} \mathcal{G}_i$ ) by jointly optimizing the sever selection decision matrix of UAVs  $\mathbf{s} = \{s_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ , the power allocation matrix of UAVs  $\mathbf{p} = \{p_i\}_{i \in \mathcal{I}}$ , the computing resource allocation matrix of edge servers  $\boldsymbol{\beta} = \{\beta_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ , and the local computing resource allocation matrix of UAVs  $\boldsymbol{\rho} = \{\rho_i\}_{i \in \mathcal{I}}$ . Denote the decision set as  $Z = \{\mathbf{s}, \mathbf{p}, \boldsymbol{\beta}, \boldsymbol{\rho}\}$ . Then, the system CO minimization problem is formulated as

$$(P1): \min_Z \mathcal{G} \quad (20)$$

$$\text{s.t. } 0 \leq a_i \leq a_i^{\max}, \quad \forall i \in \mathcal{I}, \quad (20a)$$

$$s_i^m \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \quad (20b)$$

$$\sum_{m=0}^M s_i^m \leq 1, \quad \forall i \in \mathcal{I}, \quad (20c)$$

$$0 \leq p_i \leq p_i^{\max}, \quad \forall i \in \mathcal{I}, \quad (20d)$$

$$0 \leq \rho_i \leq \rho_i^{\max}, \quad \forall i \in \mathcal{I}, \quad (20e)$$

$$0 \leq \beta_i^m \leq 1, \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \quad (20f)$$

$$\sum_{i \in \Lambda_m} \beta_i^m \leq 1, \quad \forall m \in \mathcal{M}, \quad (20g)$$

$$t_i^{\text{loc}} \leq T_i^{\text{th}}, \quad \forall i \in \mathcal{I}, \quad (20h)$$

$$t_i^{\text{edge}} \leq T_i^{\text{th}}, \quad \forall i \in \mathcal{I}. \quad (20i)$$

In (P1), constraint (20a) indicates the task arrivals is bounded by  $a_i^{\max}$ , constraints (20b) and (20c) denote the server selection decision constraints, respectively, constraint (20d) guarantees that the transmission power of UAV  $i$  cannot exceed the maximum  $p_i^{\max}$ , constraint (20e) indicates the local CPU-cycle frequency of UAV  $i$  is bounded by  $\rho_i^{\max}$ , constraint (20f) denotes that the computing resource allocation ratio is between 0 and 1, constraint (20g) ensures that the total computation resource allocated to UAVs at edge server  $m$  is bounded by  $F_m$ , constraint (20h) and (20i) provide the maximum latency guarantee for local computing and edge computing, respectively.

## V. PROBLEM TRANSFORMATION AND SOLUTION DESIGN

In (P1), the variable  $s_i^m$  is a binary integer variable that governs server selection decisions. Notably, the objective function incorporates terms involving the product of this binary variable and continuous variables, a formulation that introduces nonlinearity. Given the presence of both integer

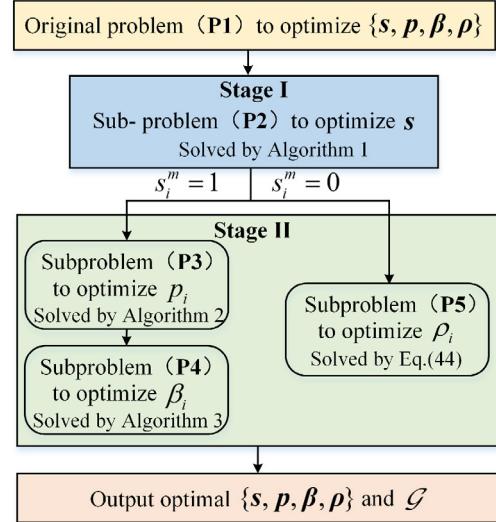


Fig. 2. The proposed solution framework for solving the formulated problems.

and continuous variables and nonlinear components, (P1) is identified as an MINLP. It is well-established that finding the optimal solution to an MINLP typically requires exponential time complexity, which reflects the inherent computational challenge of this type of optimization problems. Considering the large number of decision variables that scale exponentially with the number of UAVs and edge servers, we aim to devise a sub-optimal solution that boasts low complexity. We transform the original problem into four subproblems and employ a two-stage optimization that separately handles integer and continuous variables. In Stage I, we determine UAV-server selection decisions  $\mathbf{s}$  through a matching game-based mechanism, detailed in Section V-A. Subsequently, in Stage II, we derive the resource allocation parameters  $\mathbf{p}$ ,  $\boldsymbol{\beta}$ , and  $\boldsymbol{\rho}$ , given the Stage I solution. Specifically, after each UAV determines which edge server to offload the tasks to, we introduce a transmission power allocation algorithm in Section IV-B. Regarding the computing resource allocation of the edge servers, a projected gradient descent based algorithm is devised in Section IV-C. Furthermore, the local computing resource allocation is optimized in Section IV-D. Finally, we propose a JCORA algorithm to optimize the total CO of all the UAVs. The entire solution framework for solving the formulated problems is illustrated in Fig. 2.

### A. Server Selections of the UAVs: A Matching-Based Scheme

We first assume that each UAV is assigned a default transmission power  $p_i^{\text{def}}$ . Then, the server selection problem is formulated as

$$(P2): \begin{aligned} & \min_{\mathbf{s}} \mathcal{G} \\ & \text{s.t. } (20b), (20c). \end{aligned} \quad (21)$$

Since  $s_i^m$  is a binary variable, it is difficult to solve the problem with traditional optimization methods. In the following, we opt to utilize the matching game method to tackle the problem. We formulate (P2) as a two-sided matching problem, involving

two distinct sets of agents: one set consisting of the UAVs, and the other set comprising the edge servers. Since each server can serve multiple UAVs, yet one UAV can connect to only one edge server, we model **(P2)** as a many-to-one matching game.

a) *Fundamentals*: We describe the matching as a triplet of  $(\Theta, \Psi, \Phi)$ :

- $\Theta = \{\Theta_1, \Theta_2\}$  denotes two disjoint sets of agents where  $\Theta_1 \in \mathcal{I}$  is the set of UAVs that have not decided where to process tasks in time slot  $t$ , and  $\Theta_2 \in \mathcal{M}$  is the set of the available edge servers.
- $\Psi = \{\Psi_i^m, \Psi_m^i\}$  denotes the preference lists of the UAVs and edge servers. Each  $i \in \Theta_1$  has a descending ordered preferences on the servers, i.e.,  $\Psi_i^m = \{m | m \in \Theta_2, m \succ_i m'\}$ , where  $\succ_i$  indicates the preference of UAV  $i$  towards the servers. Similarly, each server has a descending ordered preference list over the UAVs, i.e.,  $\Psi_m^i = \{i | i \in \Theta_1, i \succ_m i'\}$ .
- $\Phi \subseteq \{i | i \in \mathcal{I}\} \times \{m | m \in \mathcal{M}\}$  denotes the many-to-one matching between the UAVs and the edge servers. Specifically, each UAV is matched with at most one edge server, i.e.,  $|\Phi(i)| \leq 1, \forall i \in \mathcal{I}$ , while each edge server is matched with multiple UAVs, i.e.,  $\Phi(m) \in \{i | i \in \mathcal{I}\}$ .

b) *Preference List*: The preference lists of each agent are constructed as follows.

- For UAV  $i$ , it prefers edge servers with larger available channel capacity  $R_i^m$  and larger residual computing capacity  $F_m^{res}$ . Therefore, the preference value for each UAV is expressed as

$$PL_i(m) = \xi_i^1 R_i^m + \xi_i^2 F_m^{res}, \quad (22)$$

where  $\xi_i^1$  and  $\xi_i^2$  indicate the weighted parameters. Then, we can construct the preference list for UAV  $i$  by ranking the preference values as a descending order, yielding

$$PL_i(m) > PL_i(m') \Leftrightarrow m \succ_i m', \Psi_i^m = \{m, m'\}. \quad (23)$$

Furthermore, each UAV has a threshold of the preference value, i.e.,  $PL_i^{th}$ , and UAV  $i$  will process the tasks locally when  $PL_i(m) < PL_i^{th}, \forall m \in \mathcal{M}$ .

- For edge server  $m$ , it prefers UAVs with more tasks and larger tolerated delay [47]. Therefore, the preference list for each server is expressed as

$$PL_m(i) = \zeta_m^1 a_i + \zeta_m^2 T_i^{th}, \quad (24)$$

where  $\zeta_m^1$  and  $\zeta_m^2$  indicate the weighted parameters. Then, we can construct the preference list for edge server  $m$  by ranking the preference values as a descending order, yielding

$$PL_m(i) > PL_m(i') \Leftrightarrow i \succ_m i', \Psi_m^i = \{i, i'\}. \quad (25)$$

Based on the aforementioned analysis, we propose Algorithm 1 to illustrate the many-to-one matching between UAVs and edge servers.

In particular, the matching should be stable, the blocking pair in many-to-one matching is defined as follows.

---

**Algorithm 1** Many-to-One Matching Between UAVs and Edge Servers

---

**Input:** The set of UAVs  $\Theta_1 \in \mathcal{I}$ , the set of the available edge servers  $\Theta_2 \in \mathcal{M}$ , the set of the default transmission power  $p^{def}$ ;

**Output:** The optimal matching list  $\Phi$ , the optimal server selection strategy  $s$ ;

```

1 Initialization:  $\Psi_i^m = \emptyset, \Psi_m^i = \emptyset, \Phi = \emptyset$ , the rejected set  $\Upsilon = \Theta_1$ ;
2 Calculate the preference value for UAVs and servers with Eq. (22) and Eq. (24), respectively;
3 Establish the preference list of UAVs and servers with Eq. (23) and Eq. (25), respectively;
4 while  $\Upsilon \neq \emptyset$  do
5   for  $i \in \Upsilon$  do
6     if  $PL_i(m) < PL_i^{th}, \forall m \in \mathcal{M}$  then
7       UAV  $i$  processes the tasks locally;
8     else
9       UAV  $i$  selects the most preferred server  $m'$  as its partner;
10      Update the matching list of UAV  $i$  as  $\Phi(i) = \Phi(i) \cup m'$  ;
11      Update the matching list of server  $m'$  as  $\Phi(m') = \Phi(m') \cup i$  ;
12    end
13  end
14  for  $m \in \Theta_2$  do
15    if  $\sum_{i \in \mathcal{I}} s_i^m \leq \Lambda_m$  and server  $m$  has enough computation resources then
16      The server  $m$  keeps all the matched UAVs and remove the matched UAVs from  $\Upsilon$ ;
17    else
18      Update the matching list of server  $m$  as  $\Phi(m) = \Phi(m) \setminus \Upsilon^{rej}$ , where  $\Upsilon^{rej}$  is the set of less preferred UAVs ;
19      Update  $\Upsilon$  as  $\Upsilon = \Upsilon \cup \Upsilon^{rej}$  ;
20      for  $i \in \Upsilon^{rej}$  do
21        Update the preference list of UAV  $i$  as  $\Psi_i^m = \Psi_i^m \setminus m$  ;
22        Update the matching list of UAV  $i$  as  $\Phi(i) = \Phi(i) \setminus m$  ;
23      end
24    end
25  end
26 end
```

---

*Definition 1: Blocking Pair in Many-to-One Matching*: A pair  $(i, m)$  is deemed as a blocking pair, if: 1) UAV  $i$  is unserved or UAV  $i$  prefers server  $m$  to its current matching  $\Phi(i)$  and 2) server  $m$  is underutilized or prefers UAV  $i$  to at least one UAV in existing matching  $\Phi(m)$ .

*Definition 2*: A matching is stable if and only if there are no blocking pairs.

*Theorem 1*: The optimal matching  $\Phi$  in Algorithm 1 is stable for all  $i \in \mathcal{I}$  and  $m \in \mathcal{M}$ .

*Proof:* We assume that matching  $\Phi$  is unstable, i.e., there is one unmatched pair  $i \in \mathcal{I}$  and  $m \in \mathcal{M}$  that prefer each other than their current matching partners, i.e.,  $i \notin \Phi(m)$  and  $m \notin \Phi(i)$ . Then, the following conditions are satisfied:

$$m \succ_i \Phi(i), \quad (26a)$$

$$i \succ_m i', i' \in \Phi(m) \quad (26b)$$

$$i \notin \Phi(m), \quad (26c)$$

$$m \notin \Phi(i). \quad (26d)$$

We prove the theorem by showing that the necessary conditions in (26) cannot be satisfied simultaneously [64]. If (26a) holds true, we deduce that UAV  $i$  prefers server  $m$  to its current matching  $\Phi(i)$ . However, condition (26c) indicates that UAV  $i$  is less preferred by server  $m$  than any  $i' \in \Phi(m)$ , i.e.,  $i' \succ_m i, i' \in \Phi(m)$ , which contradicts the condition in (26b). In this case, Although UAV  $i$  prefers server  $m$ , server  $m$  does not prefer UAV  $i$ . Therefore, the blocking pair cannot be formed between  $i \in \mathcal{I}$  and  $m \in \mathcal{M}$ . Then, the matching obtained from Algorithm 1 is stable.

### B. Optimization of Transmission Power Allocation

Once UAV  $i$  identifies the appropriate MASS or HAP for task offloading, we can proceed to optimize the transmission power in accordance with their specific requirements, thereby minimizing the CO. The transmission power allocation problem is modeled as

$$\begin{aligned} (\mathbf{P3}): \quad & \min_{\mathbf{p}} \sum_{i \in \mathcal{I}} \left( \mu_i^t \cdot \frac{a_i D}{R_i^m} + \mu_i^e \cdot \frac{p_i a_i D}{R_i^m} \right) \\ & \text{s.t.} \quad (20d). \end{aligned} \quad (27)$$

Substituting (5) into (27), (P3) is reformulated as

$$\begin{aligned} (\mathbf{P3.1}): \quad & \min_{\mathbf{p}} \sum_{i \in \mathcal{I}} \frac{a_i D}{W_i^m} \cdot \frac{\mu_i^t + \mu_i^e p_i}{\log_2 \left( 1 + \frac{p_i H_i^m}{\sigma^2} \right)} \\ & \text{s.t.} \quad (20d). \end{aligned} \quad (28)$$

For convenience, let  $B_i = \frac{a_i D}{W_i^m}$ ,  $C_i = H_i^m / \sigma^2$ . We introduce  $\Xi(p_i)$  to solve (P3.1) and define it as

$$\Xi(p_i) = B_i \cdot \frac{\mu_i^t + \mu_i^e p_i}{\log_2 (1 + C_i p_i)}. \quad (29)$$

(P3.1) is non-convex as the second-order derivative of  $\Xi(p_i)$  is not always positive. In view of this, we employ the quasi-convex optimization technique to address problem (28) based on the following lemma.

*Lemma 1:*  $\Xi(p_i)$  is quasi-convex in the domain, and it satisfies the quasi-convex second-order condition and there is a point  $p_i^0$  satisfying  $\Xi'(p_i) = 0$  and  $\Xi''(p_i) > 0$ .

*Proof:* Obviously,  $\Xi(p_i)$  is twice differentiable on  $\mathbb{R}$ , The first-order derivative of  $\Xi(p_i)$  is given by

$$\frac{\partial \Xi(p_i)}{\partial p_i} = B_i \frac{\mu_i^e \log_2 (1 + C_i p_i) - \frac{C_i(\mu_i^t + \mu_i^e p_i)}{\ln 2 (1 + C_i p_i)}}{[\log_2 (1 + C_i p_i)]^2}. \quad (30)$$

When  $\Xi'(p_i^0) = 0$ , we obtain

$$\Omega(p_i^0) = \mu_i^e \log_2 (1 + C_i p_i^0) - \frac{C_i (\mu_i^t + \mu_i^e p_i^0)}{\ln 2 (1 + C_i p_i^0)} = 0. \quad (31)$$

The second-order derivative of  $\Xi(p_i)$  is given in (32), shown at the bottom of the page.

Substituting  $p_i^0$  into the second-order derivative of  $\Xi(p_i)$ , we obtain

$$\frac{\partial^2 \Xi(p_i^0)}{\partial p_i^0} = B_i \frac{C_i^3 (\mu_i^t + \mu_i^e p_i^0)^2}{(\ln 2)^2 \mu_i^e (1 + C_i p_i^0)^3 \log_2^3 (1 + C_i p_i^0)} \geq 0. \quad (33)$$

Thus,  $\Xi(p_i)$  is quasi-convex in the domain  $[0, p_i^{\max}]$ .

Note that a quasi-convex function reaches a local optimum at the point of decreasing first-order derivatives to 0. Any local optimum of a quasi-convex function is a global optimum [52]. Since  $\Omega(0) = -\frac{C_i \mu_i^t}{\ln 2} < 0$ , we obtain  $\Xi'(0) < 0$  and  $\Xi'(p_i)$  is a monotonically increasing function and is negative at the starting point  $p_i = 0$ . Therefore, based on Lemma 1, we confirm that the optimal solution  $p_i^*$  either lies at the constraint border, i.e.,  $p_i^* = p_i^{\max}$ , or satisfies  $p_i^* = p_i^0$ , where  $\Xi'(p_i^0) = 0$ . Then, the optimal  $p_i^*$  is expressed as

$$p_i^* = \min \{p_i^0, p_i^{\max}\}. \quad (34)$$

Based on the above analysis, we design a transmission power allocation (TPA) algorithm based on the low-complexity bisection method to evaluate  $\Xi'(p_i)$  in each iteration, so as to obtain the optimal  $p_i^*$ , as shown in Algorithm 2.

### C. Optimization of Edge Computing Resource Allocation

After UAV  $i$  successfully offloads tasks to the designated edge server, we will optimize the allocation of edge computing resources to minimize the CO, formulated as

$$\begin{aligned} (\mathbf{P4}): \quad & \min_{\boldsymbol{\beta}} \sum_{i \in \mathcal{I}} \left\{ \mu_i^t \cdot \frac{a_i D \alpha}{\beta_i^m F_m} + \mu_i^e \cdot \varepsilon_m (\beta_i^m F_m)^2 a_i D \alpha \right\} \\ & \text{s.t.} \quad (20f), (20g). \end{aligned} \quad (35)$$

Notice that the constraints in (20f) and (20g) are convex and the second-order derivative of the objective function is greater than zero, (P4) is a convex problem. We employ the projected Gradient descent method to find the optimal  $\beta_i^{m*}$ . For convenience, let

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{I}} \left\{ \mu_i^t \cdot \frac{a_i D \alpha}{\beta_i^m F_m} + \mu_i^e \cdot \varepsilon_m (\beta_i^m F_m)^2 a_i D \alpha \right\}. \quad (36)$$

$$\frac{\partial^2 \Xi(p_i)}{\partial p_i} = B_i \frac{\ln 2 C_i \log_2 (1 + C_i p_i) [C_i (\mu_i^t + \mu_i^e p_i) - 2\mu_i^e (1 + C_i p_i)] + 2C_i^2 (\mu_i^t + \mu_i^e p_i)}{[\ln 2 (1 + C_i p_i)]^2 \log_2^3 (1 + C_i p_i)}. \quad (32)$$

**Algorithm 2** TPA Algorithm

**Input:** The set of UAVs  $\mathcal{I}$ , the set of edge servers  $\mathcal{M}$ ;  
**Output:** The optimal transmission power allocation strategy  $p$ ;

---

```

1 Initialization: The optimality tolerance parameter  $\delta$ ;
2 Calculate  $\Omega(p_i^{\max})$  with Eq. (31);
3 if  $\Omega(p_i^{\max}) \leq 0$  then
4   | Set  $p_i^* = p_i^{\max}$  ;
5 else
6   | Set the lower bound as  $p_i^L = 0$  and the upper
    | bound as  $p_i^U = p_i^{\max}$  ;
7   while  $p_i^U - p_i^L > \delta$  do
8     | Calculate the current value of
      |  $p_i^{cur} = (p_i^U + p_i^L) / 2$  ;
9     | if  $\Omega(p_i^{cur}) \leq 0$  then
10    |   | Set  $p_i^* = p_i^{cur}$  ;
11    | else
12    |   | Set  $p_i^U = p_i^{cur}$  ;
13    | end
14  end
15  Set  $p_i^* = (p_i^U + p_i^L) / 2$  ;
16 end
```

---

The gradient of  $\mathcal{L}(\beta)$  is denoted as  $\nabla \mathcal{L} = \left\{ \frac{\partial \mathcal{L}}{\partial \beta_i^m} \right\}$ , where

$$\frac{\partial \mathcal{L}}{\partial \beta_i^m} = 2\mu_i^e \varepsilon_m F_m^2 a_i D \alpha \beta_i^m - \frac{\mu_i^t a_i D \alpha}{F_m (\beta_i^m)^2}. \quad (37)$$

The update rule of gradient descent method is denoted as

$$(\beta_i^m)^{k+1} = (\beta_i^m)^k - \eta \frac{\partial \mathcal{L}}{\partial \beta_i^m} \Big|_{\beta_i^m=(\beta_i^m)^k}, \quad (38)$$

where  $(\beta_i^m)^k$  is the value of  $\beta_i^m$  at the  $k$ -th iteration,  $\eta$  is the learning rate (step size), which needs to be manually set.

Subsequently, we will employ a projection operation to ensure that the value of  $\beta_i^m$  remains within the bounds defined by the constraint conditions.

- Projection to the  $[0, 1]$  interval: We first need to perform constraint projection on each  $\beta_i^m$  to ensure that  $0 \leq \beta_i^m \leq 1$ , yielding:

$$(\beta_i^m)^{k+1} = \max \left\{ 0, \min \left[ 1, (\beta_i^m)^k \right] \right\}. \quad (39)$$

- Projection to  $\sum_{i \in \Lambda_m} \beta_i^m \leq 1$ : After each update, we check the value of  $\sum_{i \in \Lambda_m} \beta_i^m$ , if  $\sum_{i \in \Lambda_m} \beta_i^m \geq 1$ , we need to scale each  $\beta_i^m$  so that the sum  $\sum_{i \in \Lambda_m} \beta_i^m$  does not exceed 1. Let  $scale = \frac{1}{\sum_{i \in \Lambda_m} (\beta_i^m)^{k+1}}$ , then, each  $(\beta_i^m)^{k+1}$  is scaled by

$$(\beta_i^m)^{k+1} = (\beta_i^m)^k \times scale. \quad (40)$$

Driving from the above analysis, we propose a computation resource allocation (CRA) algorithm base on the projected Gradient descent method to obtain the optimal  $\beta_i^{m*}$  for each UAV, as shown in Algorithm 3, where  $\text{clip}(x, x_{\min}, x_{\max}) = \max[x_{\min}, \min(x, x_{\max})]$  ensures the solution remains within the feasible domain by constraining values between a lower bound and an upper bound.

**Algorithm 3** CRA Algorithm

**Input:** Learning rate  $\eta$ , convergence tolerance  $\epsilon$ , maximum iterations  $K$ ;  
**Output:** The optimal solution  $\beta^*$ ;

---

```

1 Initialization: 0  $\leftarrow k$ , set each  $\beta_i^m$  with an initial
  | value  $\beta_i^{m0}$ ;
2 while  $k \leq K$  do
3   | Calculate gradient  $\nabla \mathcal{L}$  with Eq. (37);
4   | for  $i \in \mathcal{I}$  do
5     |   | Update  $(\beta_i^m)^k$  with Eq. (38);
6   | end
7   | Project onto feasible region:
      |    $(\beta_i^m)^k \leftarrow \text{clip} \left[ (\beta_i^m)^k, 0, 1 \right]$ ;
8   | if  $\sum_{i \in \Lambda_m} \beta_i^m \geq 1$  then
9     |   | Update  $(\beta_i^m)^k$  with Eq. (40);
10    | end
11   | if  $\| (\beta_i^m)^{k+1} - (\beta_i^m)^k \| < \epsilon$  then
12     |   | break ;
13   | end
14   | Update  $\beta_i^m = (\beta_i^m)^{k+1}$ ;
15 end
```

---

*D. Optimization of Local Computation Resource Allocation*

When  $s_i^m = 0$ , the tasks of UAV  $i$  will be executed locally. By selecting optimal processing parameters, the CO can be effectively minimized. Following this, we will address the optimization of local computing resource allocation, formulated as:

$$\begin{aligned} \text{(P5): } \min_{\rho} \sum_{i \in \mathcal{I}} & \left( \mu_i^t \cdot \frac{a_i D \alpha}{\rho_i} + \mu_i^e \cdot \varepsilon_i \rho_i^2 a_i D \alpha \right) \\ \text{s.t. } & (20e), (20h). \end{aligned} \quad (41)$$

**(P5)** is a convex optimization problem regarding  $\rho$  and decoupled with respect to each UAV. Its solution is obtained at the poles or at the boundary. We first obtain its pole value  $\rho_i^0$  by assigning its first-order derivative function to zero. Then, we get

$$\rho_i^0 = \sqrt[3]{\frac{\mu_i^t}{2\mu_i^e \varepsilon_i}}. \quad (42)$$

According to constraint (20h), we have

$$\rho_i \geq \frac{a_i D \alpha}{T_i^{th}} = \rho_i^{\min}. \quad (43)$$

With Eq. (42) and Eq. (43), we obtain the optimal  $\rho_i^*$  as

$$\rho^* = \begin{cases} \rho_i^{\min}, \rho_i^0 < \rho_i^{\min} \\ \rho_i^0, \rho_i^{\min} \leq \rho_i^0 \leq \rho_i^{\max} \\ \rho_i^{\max}, \rho_i^0 > \rho_i^{\max}. \end{cases} \quad (44)$$

Based on the aforementioned analysis, we propose the JCORA algorithm to jointly optimize server selection, transmission power allocation, edge computing resource allocation, and local computing resource allocation, as outlined in Algorithm 4.

**Algorithm 4** JCORA Algorithm

---

**Input:** The set of UAVs  $\mathcal{I}$ , the set of edge servers  $\mathcal{M}$ , the default transmission power  $p^{def}$ , the optimality tolerance  $\delta$ , learning rate  $\eta$ , convergence tolerance  $\epsilon$ , maximum iterations  $K$ ;

**Output:** the optimal  $s^*$ ,  $p^*$ ,  $\beta^*$ ,  $\rho^*$  and the optimal CO  $\mathcal{G}^*$ ;

- 1 Obtain  $s^*$  with Algorithm 1;
- 2 **for**  $i \in \mathcal{I}$  **do**
- 3   **if**  $s_i^m = 0$  **then**
- 4     | Calculate  $\rho_i^*$  with Eq. (44) ;
- 5   **else**
- 6     | Obtain  $p_i^*$  with Algorithm 2 ;
- 7     | Obtain  $\beta_i^{m*}$  with Algorithm 3 ;
- 8   **end**
- 9   Calculate  $\mathcal{G}_i^*$  for each UAV ;
- 10 **end**

---

Finally, we analyze the computational complexity of the proposed algorithms. For Algorithm 1, in the worst case, any UAV could be rejected at most  $\mathcal{M}$  times, i.e., the UAV is rejected by all edge servers. Whenever a UAV is rejected by its preferred server, in the subsequent iteration, there are at most  $\min\{\mathcal{I}, \mathcal{M}\}$  servers that need to update their preference lists. Therefore, the worst-case complexity of Algorithm 1 is  $\mathcal{O}(\mathcal{M} \cdot (2\mathcal{I} + \min\{\mathcal{I}, \mathcal{M}\}))$ , where  $\mathcal{I}$  and  $\mathcal{M}$  are the number of UAVs and edge servers, respectively. We denote the number of iterations of Algorithm 2 and 3 as  $Q$  and  $K$ , respectively. Then, we obtain the computation complexity of Algorithm 2 and 3 as  $\mathcal{O}(\log_2 Q)$  and  $\mathcal{O}(K)$  for each UAV, respectively. Therefore, the total computational complexity of the proposed algorithms is  $\mathcal{O}(\mathcal{M} \cdot (2\mathcal{I} + \min\{\mathcal{I}, \mathcal{M}\}) + \mathcal{I}(\log_2 Q + K))$ . We see that the proposed algorithms have a low complexity compared to conventional solutions, e.g., the exhaustive search policy with a complexity of  $\mathcal{O}(\mathcal{M}^{\mathcal{I}})$ .

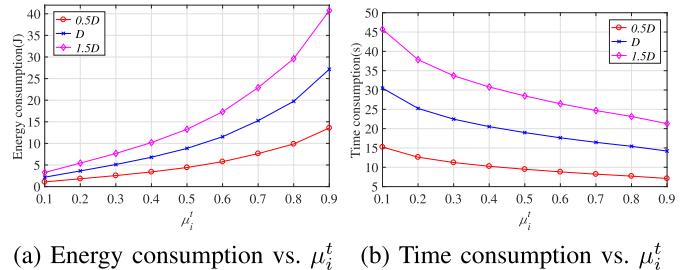
## VI. PERFORMANCE EVALUATION

In this section, we conduct simulations to verify the effectiveness of the proposed algorithms. More specifically, we assess the impact of key parameters and compare our proposed scheme with the following benchmark schemes.

- (a) Local Computing (LC): All UAVs choose to perform the tasks locally, i.e.,  $s_i^m = 0, \forall i \in \mathcal{I}$ .
- (b) Only Match Game (OMG) [65]: This scheme determines task offloading through a match game without incorporating joint resource allocation. Each UAV utilizes a constant transmission power, whereas the edge servers adopt an equitable sharing strategy for computing resource allocation.
- (c) Distributed edge user allocation (DEUA) scheme: This scheme is extended from [66] to be applied to our model. When the UAVs compete for the matched edge servers, the winner is selected randomly.

TABLE II  
SIMULATION PARAMETER SETTINGS

Parameters	Values
Carrier frequency ( $f_i$ )	0.2 GHz
Speed of light ( $c$ )	$3 \times 10^8$ m/s
Transmission bandwidth of MASS	10 MHz
Transmission bandwidth of HAPS	10 MHz
Background noise power ( $\sigma^2$ )	$1 \times 10^{-13}$ W
Path-loss exponent ( $\lambda$ )	1.5
Height of HAPS ( $Z_H$ )	20km
Height of UAV $i$ ( $Z_i$ )	200m
Link loss ( $\delta_i^{LOS}$ and $\delta_i^{NLOS}$ ) [57]	0.1, 21
Environmental parameters ( $\chi_1$ and $\chi_2$ ) [57]	4.88, 0.43s
Learning rate ( $\eta$ )	0.01
Number of CPU cycles for processing one bit of data ( $\alpha$ )	$1 \times 10^3$
Maximum computing capability of UAVs ( $\rho_i^{\max}$ )	$5 \times 10^9$ cycles/s



(a) Energy consumption vs.  $\mu_i^t$       (b) Time consumption vs.  $\mu_i^t$

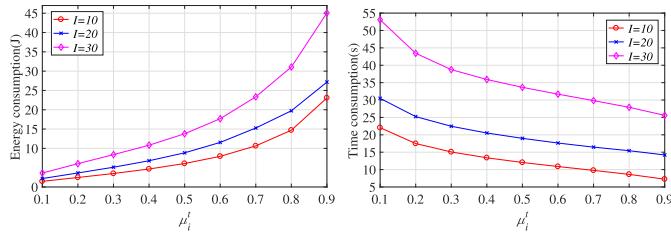
Fig. 3. The total energy consumption and overall latency associated with different task sizes under different UAV preferences for time.

### A. System Setup

We conduct all the simulations using MATLAB on a PC configured with a Core i7-10510U 1.80 GHz CPU and 8 GB of RAM. We consider an aerial-marine integrated network comprised of one HAPS and four MASSs which navigate autonomously on the ocean. We assume that all UAVs are randomly distributed within the service range of the HAPS. The maximum number of task arrivals of each UAV is 10, and the size of each task is 0.5 Mbits. Each UAV communicates with the corresponding MASS via OFDMA, utilizing a channel bandwidth of 10 MHz, and each UAV communicates with the HAPS through OFDMA, employing a channel bandwidth of 10 MHz. The main parameters are shown in Table II.

### B. Performance Evaluation

Fig. 3 depicts the total energy consumption and the overall latency for completing all UAVs' tasks associated with different task sizes, respectively, as we adjust the UAVs' preference for time,  $\mu_i^t$ , from 0.1 to 0.9 while changing the UAVs' preference for energy accordingly as  $\mu_i^e = 1 - \mu_i^t, \forall i \in \mathcal{I}$ . We observe that an increase in  $\mu_i^t$  leads to a reduction in the overall latency, albeit at the expense of elevated energy expenditure. Moreover, with the expansion of task size, a corresponding increase was observed in both energy consumption and the overall latency. The observed trend stems from inherent system limitations, as the increased computational demands from the escalating task size of UAVs necessitates greater temporal and energy resources for task execution.



(a) Energy consumption vs.  $\mu_i^t$  (b) Time consumption vs.  $\mu_i^t$

Fig. 4. The total energy consumption and overall latency associated with different number of UAVs under different UAV preferences for time.

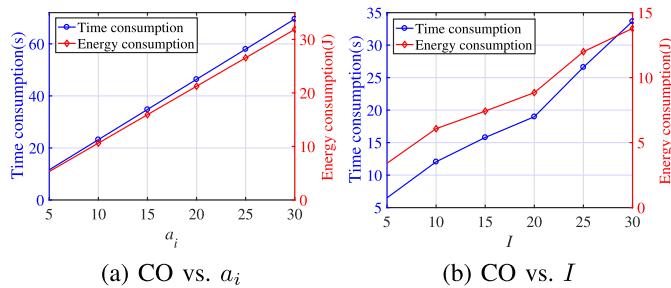


Fig. 5. The CO associated with different number of tasks and number of UAVs, respectively.

Fig. 4 illustrates the total energy consumption and the overall latency for completing all UAVs' tasks associated with different number of UAVs, respectively, when we adjust the UAVs' preference for time,  $\mu_i^t$ , from 0.1 to 0.9 while changing the UAVs' preference to energy accordingly as  $\mu_i^e = 1 - \mu_i^t, \forall i \in \mathcal{I}$ . Consistent with the trends observed in Fig. 3, we observe that increasing  $\mu_i^t$  effectively reduces system latency, though this performance enhancement comes at the cost of increased energy consumption. Furthermore, an increase in the number of UAVs results in a corresponding surge in both energy consumption and overall latency. This is attributable to the heightened competition among UAVs for limited resources, wherein the increasing number of UAVs reduces the offloading efficiency and creates computational bottlenecks.

Fig. 5 illustrates the impact of task quantity  $a_i$  and the number of UAVs on the CO, i.e., the overall latency and the energy consumption, respectively. Specifically, Fig. 5(a) illustrates that system performance degrades with increasing values of  $a_i$ , manifesting in elevated latency and amplified energy consumption. This phenomenon originates from the cumulative resource requirements of the system, where the growing workload necessitates additional processing capabilities from MASSs, UAVs, and HAPS, thereby intensifying both temporal and energy expenditures. Fig. 5(b) indicates that both the latency and energy consumption exhibit progressive deterioration with increasing UAV numbers, due to the inherent computational demands, where expanding UAV numbers generate greater processing workloads that demand additional time and energy resources.

Fig. 6 illustrates the performance comparison of the proposed scheme with other three benchmarks under different parameters (i.e., task quantity, the number of UAVs, the UAV preferences for time and the computational capability

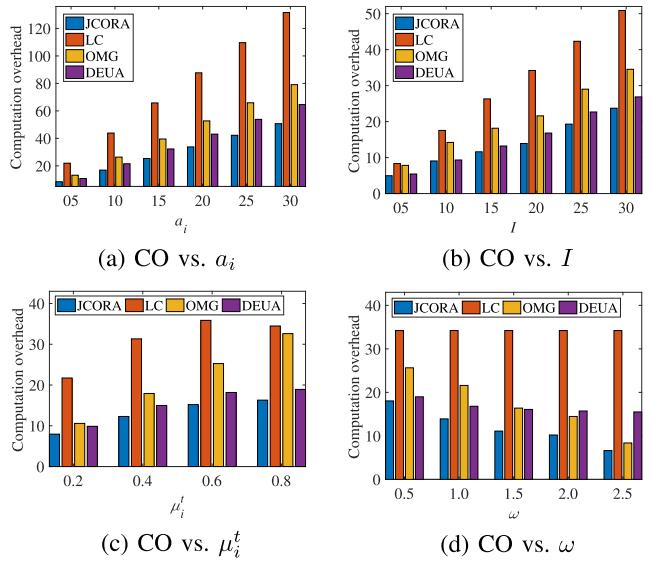


Fig. 6. Performance comparison of our proposed scheme with other benchmarks under different parameters (i.e.,  $a_i$ ,  $I$ ,  $\mu_i^t$  and  $\omega$ ).

of MASSs). Specifically, in Fig. 6(d),  $\omega$  indicates the multiple of MASSs' computing resources  $F_m$ , namely  $0.5F_m$ ,  $1.0F_m$ ,  $1.5F_m$ ,  $2.0F_m$  and  $2.5F_m$ . Figs. 6(a), 6(b), 6(c) and 6(d) clearly demonstrate the superiority of our proposed scheme, which effectively diminishes the overall CO. This is attributed to our method of determining optimal task offloading decisions and resource allocation strategies by minimizing the total CO.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have considered an MSAR scenario and proposed a joint task offloading and resource allocation scheme for surveillance UAVs, by optimizing the server selection, transmission power allocation, edge computing resource allocation, and local computing resource allocation. The objective is to minimize the system total CO, quantified by a weighted-sum of task completion time and energy consumption of each UAV. We define a novel scenario where both MASSs and the HAPS are equipped with edge-servers to provide computing services for surveillance UAVs and the computation workloads of surveillance UAVs can be offloaded to MASSs and the HAPS via a multi-access approach. To address the stochastic UAV task arrivals and dynamic marine channel conditions, we decompose the original problem into four subproblems via a two-stage optimization: (1) edge server selection through many-to-one matching, (2) UAV transmission power allocation via quasi-convex optimization, (3) edge computing resource distribution using projected gradient descent, and (4) local computation resource assignment through convex optimization. Finally, we propose a JCORA algorithm aimed at minimizing the overall CO of the system. Simulation results demonstrate that our scheme outperforms benchmark schemes, highlighting its potential to significantly improve the efficiency of MSAR operations. For future work, we will consider the integration of advanced artificial intelligence (AI)-driven techniques to further enhance the decision-making process in task offloading and resource allocation for MSAR.

## REFERENCES

- [1] C. Lei, S. Wu, Y. Yang, J. Xue, and Q. Zhang, "Joint trajectory and communication optimization for heterogeneous vehicles in maritime SAR: Multi-agent reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12328–12344, Sep. 2024.
- [2] Z. Jiang, T. Yang, L. Zhou, Y. Yuan, and H. Feng, "Maritime search and rescue networking based on multi-agent cooperative communication," *J. Commun. Inf. Netw.*, vol. 4, no. 1, pp. 42–53, Mar. 2019.
- [3] T. Yang, Z. Jiang, R. Sun, N. Cheng, and H. Feng, "Maritime search and rescue based on group mobile computing for unmanned aerial vehicles and unmanned surface vehicles," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7700–7708, Dec. 2020.
- [4] G. Zhang, S. Liu, X. Zhang, and W. Zhang, "Event-triggered cooperative formation control for autonomous surface vehicles under the maritime search operation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21392–21404, Nov. 2022.
- [5] S. Qi, B. Lin, Y. Deng, X. Chen, and Y. Fang, "Minimizing maximum latency of task offloading for multi-UAV-assisted maritime search and rescue," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13625–13638, Sep. 2024.
- [6] D. Jia and Q. J. Ye, "Mobility-adaptive digital twin modeling for post-disaster network traffic prediction," in *Proc. IEEE 100th Veh. Technol. Conf. (VTC-Fall)*, Oct. 2024, pp. 1–7.
- [7] Z. Wang, B. Lin, Q. Ye, and H. Peng, "Two-tier task offloading for satellite-assisted marine networks: A hybrid Stackelberg-Bargaining game approach," *IEEE Internet Things J.*, vol. 12, no. 9, pp. 13047–13060, May 2025.
- [8] J. Zhang, H. Liu, Y. He, W. Gao, N. Xu, and C. Liu, "MCLORA: Maritime ad-hoc communication system based on LoRa," *High-Confidence Comput.*, vol. 5, no. 3, Sep. 2025, Art. no. 100275.
- [9] C. Zhang, B. Lin, X. Hu, S. Qi, L. Qian, and Y. Wu, "Resource management and trajectory optimization for UAV-IRS assisted maritime edge computing networks," *Tsinghua Sci. Technol.*, vol. 30, no. 4, pp. 1600–1616, Aug. 2025.
- [10] Z. Chen, H. Liu, Y. Tian, R. Wang, P. Xiong, and G. Wu, "A particle swarm optimization algorithm based on time-space weight for helicopter maritime search and rescue decision-making," *IEEE Access*, vol. 8, pp. 81526–81541, 2020.
- [11] Q. Ye, W. Zhuang, S. Zhang, A.-L. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9896–9910, Oct. 2018.
- [12] V. Niazmand and Q. Ye, "Joint task offloading, DNN pruning, and computing resource allocation for fault detection with dynamic constraints in industrial IoT," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 5, pp. 3486–3501, Oct. 2025.
- [13] D. Han et al., "Two-timescale learning-based task offloading for remote IoT in integrated satellite–terrestrial networks," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10131–10145, Jun. 2023.
- [14] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, Jun. 2018.
- [15] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, "Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 272–288, 2021.
- [16] Y. Dai et al., "A survey of graph-based resource management in wireless networks—Part II: Learning approaches," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 4, pp. 2101–2122, Aug. 2025.
- [17] Y. Dai et al., "A survey of graph-based resource management in wireless networks—Part I: Optimization approaches," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 4, pp. 2078–2100, Aug. 2025.
- [18] Z. A. Khan and I. A. Aziz, "Dynamic OBL-driven whale optimization algorithm for independent tasks offloading in fog computing," *High-Confidence Comput.*, vol. 5, no. 4, Dec. 2025, Art. no. 100317.
- [19] Q. Wang, C. Yu, S. Chen, W. Fang, and N. Xiong, "Joint adaptive resolution selection and conditional early exiting for efficient video recognition on edge devices," *Big Data Mining Analytics*, vol. 8, no. 3, pp. 661–677, Jun. 2025.
- [20] S. Zhang, Y. Zhang, and H. Zheng, "SAGVN task offloading and resource allocation with spectrum resource constraints," *Tsinghua Sci. Technol.*, Sep. 2025, doi: [10.26599/TST.2024.9010190](https://doi.org/10.26599/TST.2024.9010190).
- [21] C. Zhang, B. Lin, Z. Chen, L. X. Cai, and J. Duan, "Mobile edge deployment and resource management for maritime wireless networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 5, pp. 7928–7939, May 2025.
- [22] H. Zhang, S. Xi, B. Shang, P. Zhang, S. Wu, and C. Jiang, "Energy oriented three-tier computation offloading scheme in maritime edge computing network," *IEEE Trans. Veh. Technol.*, vol. 74, no. 5, pp. 8126–8140, May 2025.
- [23] W. Wu et al., "Split learning over wireless networks: Parallel design and resource management," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1051–1066, Apr. 2023.
- [24] C. Zhou, J. Liu, K. Qu, M. Sheng, J. Li, and W. Zhuang, "Delay-aware UAV computation offloading and communication assistance for post-disaster rescue," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 19110–19125, Dec. 2024.
- [25] Q. Ren, O. Abbasi, G. K. Kurt, H. Yanikomeroglu, and J. Chen, "Caching and computation offloading in high altitude platform station (HAPS) assisted intelligent transportation systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9010–9024, Nov. 2022.
- [26] M. Li, L. P. Qian, Q. Wang, Y. Wu, B. Lin, and X. Yang, "High altitude platforms-assisted hierarchical computing offloading in marine-IoT networks: A delay minimization approach," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 3783–3788.
- [27] H. Guo, X. Zhou, J. Wang, J. Liu, and A. Benslimane, "Intelligent task offloading and resource allocation in digital twin based aerial computing networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3095–3110, Oct. 2023.
- [28] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 53–87, 1st Quart., 2022.
- [29] H. Sun, X. Zhang, B. Zhang, K. Sha, and W. Shi, "Optimal task offloading and trajectory planning algorithms for collaborative video analytics with UAV-assisted edge in disaster rescue," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6811–6828, May 2024.
- [30] R. Men, X. Fan, K.-L.-A. Yau, A. Shan, and G. Yuan, "Hierarchical aerial computing for task offloading and resource allocation in 6G-enabled vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 4, pp. 3891–3904, Jul. 2024.
- [31] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2022.
- [32] W. Wu et al., "AI-native network slicing for 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.
- [33] Q. Ren, O. Abbasi, G. K. Kurt, H. Yanikomeroglu, and J. Chen, "Handoff-aware distributed computing in high altitude platform station (HAPS)—Assisted vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8814–8827, Dec. 2023.
- [34] W. Jaafar and H. Yanikomeroglu, "HAPS-ITS: Enabling future ITS services in trans-continental highways," *IEEE Commun. Mag.*, vol. 60, no. 10, pp. 80–86, Oct. 2022.
- [35] X. Huang, R. Gu, and Y. Huang, "Towards efficient serverless MapReduce computing on cloud-native platforms," *Big Data Mining Analytics*, vol. 8, no. 3, pp. 575–591, 2025.
- [36] X. Zhou, L. Huang, T. Ye, and W. Sun, "Computation bits maximization in UAV-assisted MEC networks with fairness constraint," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20997–21009, Nov. 2022.
- [37] Y. Wang, J. Zhu, H. Huang, and F. Xiao, "Bi-objective ant colony optimization for trajectory planning and task offloading in UAV-assisted MEC systems," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12360–12377, Dec. 2024.
- [38] G. Sun et al., "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8582–8600, Sep. 2024.
- [39] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, May 2022.
- [40] M. Zhao, R. Zhang, Z. He, and K. Li, "Joint optimization of trajectory, offloading, caching, and migration for UAV-assisted MEC," *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 1981–1998, Mar. 2025.
- [41] Y. Ding et al., "Online edge learning offloading and resource management for UAV-assisted MEC secure communications," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 54–65, Jan. 2023.
- [42] C. Ding, J.-B. Wang, H. Zhang, M. Lin, and G. Y. Li, "Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1362–1377, Feb. 2022.

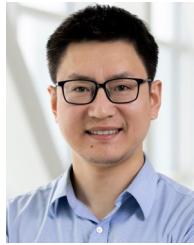
- [43] D. S. Lakew, A.-T. Tran, N.-N. Dao, and S. Cho, "Intelligent offloading and resource allocation in heterogeneous aerial access IoT networks," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5704–5718, Apr. 2023.
- [44] M. Li, L. P. Qian, X. Dong, B. Lin, Y. Wu, and X. Yang, "Secure computation offloading for marine IoT: An energy-efficient design via cooperative jamming," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6518–6531, May 2023.
- [45] X. Yu, X. Zhang, Y. Rui, K. Wang, X. Dang, and M. Guizani, "Joint resource allocations for energy consumption optimization in HAPS-aided MEC-NOMA systems," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3632–3646, Dec. 2024.
- [46] X. Qin, Z. Na, X. Wu, M. Xiong, and B. Lin, "Secure mobile edge computing for integrated HAP and UAV networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, China, Aug. 2023, pp. 1–6.
- [47] Z. Jia, Q. Wu, C. Dong, C. Yuen, and Z. Han, "Hierarchical aerial computing for Internet of Things via cooperation of HAPs and UAVs," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5676–5688, Apr. 2023.
- [48] H. Zeng et al., "USV fleet-assisted collaborative computation offloading for smart maritime services: An energy-efficient design," *IEEE Trans. Veh. Technol.*, vol. 73, no. 10, pp. 14718–14733, Oct. 2024.
- [49] Y. Liao, X. Chen, J. Liu, Y. Han, N. Xu, and Z. Yuan, "Cooperative UAV-USV MEC platform for wireless inland waterway communications," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3064–3076, Feb. 2024.
- [50] H. Chen, W. Cai, and M. Zhang, "AUV-aided computing offloading for multi-tier underwater computing: A Stackelberg game learning approach," *Ocean Eng.*, vol. 297, Apr. 2024, Art. no. 117109.
- [51] Z. Lin, J. Yang, Y. Chen, C. Xu, and X. Zhang, "Maritime distributed computation offloading in space-air-ground-sea integrated networks," *IEEE Commun. Lett.*, vol. 28, no. 7, pp. 1614–1618, Jul. 2024.
- [52] T. Lyu, H. Xu, F. Liu, M. Li, L. Li, and Z. Han, "Computing offloading and resource allocation of NOMA-based UAV emergency communication in marine Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15571–15586, May 2024.
- [53] M. Dai et al., "Latency minimization oriented hybrid offshore and aerial-based multi-access computation offloading for marine communication networks," *IEEE Trans. Commun.*, vol. 71, no. 11, pp. 6482–6498, Nov. 2023.
- [54] Z. Jia et al., "Distributionally robust optimization for aerial multi-access edge computing via cooperation of UAVs and HAPs," *IEEE Trans. Mobile Comput.*, vol. 24, no. 10, pp. 10853–10867, Oct. 2025.
- [55] J. You, Z. Jia, C. Dong, Q. Wu, and Z. Han, "Joint computation offloading and resource allocation for uncertain maritime MEC via cooperation of AAVs and vessels," *IEEE Trans. Veh. Technol.*, vol. 74, no. 11, pp. 18081–18095, Nov. 2025.
- [56] Z. Wang, B. Lin, Q. Ye, Y. Fang, and X. Han, "Joint computation offloading and resource allocation for maritime MEC with energy harvesting," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19898–19913, Jun. 2024.
- [57] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: A distributed online approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8129–8142, Aug. 2024.
- [58] X. Li, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "Maritime coverage enhancement using UAVs coordinated with hybrid satellite-terrestrial networks," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2355–2369, Apr. 2020.
- [59] X. Li, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "UAV-enabled accompanying coverage for hybrid satellite-UAV-terrestrial maritime communications," in *Proc. 28th Wireless Opt. Commun. Conf. (WOCC)*, May 2019, pp. 1–5.
- [60] J. Wang et al., "Wireless channel models for maritime communications," *IEEE Access*, vol. 6, pp. 68070–68088, 2018.
- [61] Z. Wang, B. Lin, and Q. Ye, "Double-edge-assisted computation offloading and resource allocation for space-air-marine integrated networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 9, pp. 14501–14514, Sep. 2025.
- [62] H. Liao, Z. Zhou, X. Zhao, and Y. Wang, "Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5250–5263, Apr. 2021.
- [63] F. Pervez, A. Sultana, C. Yang, and L. Zhao, "Energy and latency efficient joint communication and computation optimization in a multi-UAV-assisted MEC network," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 1728–1741, Mar. 2024.
- [64] Z. Sun, G. Sun, Y. Liu, J. Wang, and D. Cao, "BARGAIN-MATCH: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1655–1673, Feb. 2024.
- [65] Y. Wang et al., "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1525–1539, Aug. 2022.
- [66] Q. He et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.



**Zhen Wang** (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Tianjin University, China, in 2010, the M.S. degree in communication and information systems from Beijing University of Posts and Telecommunications, China, in 2015, and the Ph.D. degree in information and communication engineering from Dalian Maritime University, Dalian, China, in 2025. She is currently an Associate Professor with Dalian Neusoft University of Information. Her research interests include maritime communication, edge/fog computing, resource allocation, and artificial intelligence.



**Bin Lin** (Senior Member, IEEE) received the B.S. and M.S. degrees from Dalian Maritime University, Dalian, China, in 1999 and 2003, respectively, and the Ph.D. degree from the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2009. She is currently a Full Professor and the Dean of Communication Engineering Department, School of Information Science and Technology, Dalian Maritime University. She was a Visiting Scholar with George Washington University, Washington, DC, USA, from 2015 to 2016. Her current research interests include wireless communications, network dimensioning and optimization, resource allocation, artificial intelligence, maritime communication networks, edge/cloud computing, wireless sensor networks, and the Internet of Things. She is an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and IET Communications.



**Qiang (John) Ye** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2016. Since September 2023, he has been an Assistant Professor with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary (UCalgary), AB, Canada. Before joining UCalgary, he was an Assistant Professor with the Department of Computer Science, Memorial University of Newfoundland, NL, Canada, from September 2021 to August 2023; and the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, USA, from September 2019 to August 2021. He was with the Department of Electrical and Computer Engineering, University of Waterloo, as a Post-Doctoral Fellow and then a Research Associate, from December 2016 to September 2019. He has published over 90 research papers in top-ranked journals and conference proceedings. He received the Best Paper Award in IEEE/CIC International Conference on Communications in China (ICCC) in 2024 and the IEEE Transactions on Cognitive Communications and Networking (TCCN) Exemplary Editor Award in 2023. He is/was the General Co-Chair, the Publication Co-Chair, the Publicity Co-Chair, the TPC Co-Chair, and the Symposium Co-Chair of different reputable international conferences and workshops, such as INFOCOM, GLOBECOM, VTC, ICCC, and ICCT. He also serves/served as the IEEE Vehicular Technology Society (VTS) Region 7 Chapter Coordinator in 2024, the IEEE Communications Society (ComSoc) Southern Alberta Chapter Vice Chair in 2024, and the VTS Regions 1–7 Chapters Coordinator from 2022 to 2023. He is also the Leading Chair of the Special Interest Group (SIG) in IEEE ComSoc-Internet of Things, Ad Hoc, and Sensor Networks (IoT-AHSN) Technical Committee. He serves as an Associate Editor for prestigious IEEE journals, such as IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. He has been selected as an IEEE ComSoc Distinguished Lecturer for the class of 2025–2026.