Environment-Aware AUV Trajectory Design and Resource Management for Multi-Tier Underwater Computing

Xiangwang Hou, *Student Member, IEEE,* Jingjing Wang, *Senior Member, IEEE*, Tong Bai, *Member, IEEE*, Yansha Deng, *Member, IEEE*, Yong Ren, *Senior Member, IEEE*, Lajos Hanzo, *Life Fellow, IEEE*

***Abstract*—The Internet of underwater things (IoUT) is envisioned to be an essential part of maritime activities. Given the IoUT devices' wide-area distribution and constrained transmit power, autonomous underwater vehicles (AUVs) have been widely adopted for collecting and forwarding the data sensed by IoUT devices to the surface-stations. In order to accommodate the diverse requirements of IoUT applications, it is imperative to conceive a multi-tier underwater computing (MTUC) framework by carefully harnessing both the computing and the communications as well as the storage resources of both the surface-station and of the AUVs as well as of the IoUT devices. Furthermore, to meet the stringent energy constraints of the IoUT devices and to reduce the operating cost of the MTUC framework, a joint environment-aware AUV trajectory design and resource management problem is formulated, which is a high-dimensional NP-hard problem. To tackle this challenge, we first transform the problem into a Markov decision process (MDP) and solve it with the aid of the asynchronous advantage actor-critic (A3C) algorithm. Our simulation results demonstrate the superiority of our scheme.**

***Index Terms*—Multi-tier computing, Internet of underwater things (IoUT), autonomous underwater vehicles (AUV), trajectory optimization, resource allocation, asynchronous advantage actor-critic (A3C).**

I. Introduction

As an extension of the Internet of things (IoT) in underwater environments, the Internet of underwater things (IoUT) is envisioned to be a crucial enabler for supporting diverse maritime activities [1]. More explicitly, the IoUT aims for constructing a "smart ocean" by connecting various underwater devices, e.g. sensors, robots, cameras, to monitor and reconstruct underwater objects and environments [2]. In contrast to the terrestrial IoT systems, radio frequency (RF)-based techniques are unsuitable for the IoUT, owing to the severe absorption of electromagnetic waves in underwater environments. As a remedy, underwater acoustic communications (UAC) [3], [4] are widely adopted, but it still remains unrealistic for energy-limited IoUT devices to directly transmit their collected data to a surface-station through long-distance propagation, because ten-times higher transmit power is required compared to RF-based communications. To cope with this issue, autonomous underwater vehicles (AUV) have been widely adopted for data collection in underwater environments [5], [6].

The seminal AUV-aided data collection techniques have routinely been based on a fixed AUV trajectory, such as an ellipse [7]. In this case, the IoUT devices distant from the AUV's trajectory have to aggregate their data at the IoUT devices in the close proximity of the AUV's trajectory for delivering it to AUVs. This inevitably leads to redundant communications and to potentially excessive energy requirements, especially at the data aggregation nodes. Hence, to overcome this impediment, recent studies opted for optimizing the AUV trajectory for actively collecting data from the IoUT devices [8]–[10]. However, only the specific locations of the IoUT devices are considered in these research contributions, while ignoring the impact of hostile environmental factors, such as dynamically fluctuating water velocity, vortex, etc., which may lead to excessive propulsion energy consumption and even disable the AUV.

Apart from the data collector node mentioned above, AUVs may also play the role of an intermediate node for data relaying. However, the requirement of ocean exploration activities is not limited to communications. Besides sensors, a large number of advanced devices have been harnessed, such as diverse underwater robots. Consequently, a large variety of computing and storage tasks has to be processed in a time-sensitive manner. For example, when considering robots, their

This work of Jingjing Wang was supported in part by the National Natural Science Foundation of China under grant No. 62071268 and grant No. 6222101, in part by the Young Elite Scientist Sponsorship Program by the China Association for Science and Technology under Grant No. 2020QNRC001, and in part by the Fundamental Research Funds for the Central Universities. T. Bai was supported in part by the National Natural Science Foundation of China under Grant 62101015. Y. Deng was partially supported by Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/W004348/1. Y. Ren was supported in part by the National Natural Science Foundation of China under grant No. 62127801, in part by the National Key R &D Program of China under Grant 2020YFD0901000, and in part by the project 'The Verification Platform of Multi-tier Coverage Communication Network for Oceans (LZC0020)' of Peng Cheng Laboratory. Moreover, L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). *(Corresponding author: Jingjing Wang.)*

X. Hou is with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China. (E-mail: xiangwanghou@163.com.)

J. Wang and T. Bai are with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China. (E-mail: drwangjj@buaa.edu.cn, tongbai@buaa.edu.cn.)

Y. Deng is with the Department of Engineering, King's College London, London WC2R 2LS, U.K. (E-mail: yansha.deng@kcl.ac.uk.)

Y. Ren is with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China, and also with the Network and Communication Research Center, Peng Cheng Laboratory, Shenzhen, 518055, China (E-mail: reny@tsinghua.edu.cn.)

L. Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. (E-mail: lh@ecs.soton.ac.uk.)

tasks have to be completed in time for adjusting the next mission. Although these devices are indeed equipped both with computing and storage capabilities, it is challenging to handle all the tasks locally, given their limited battery lives. Hence, it is beneficial to establish a multi-tier computing [11] framework by integrating both the computing and the communications as well as storage resources of surface-stations and of AUVs, as well as of the devices for providing on-demand computing services.

Both AUV-centric [12]–[15] and IoUT-centric [10], [16], [17] designs were considered in the open literature conceived either for latency-minimization or for energy-minimization. However, both types of designs have their limitations. As a remedy, we propose a system-level framework for maximizing the benefits of an intrinsically amalgamated hierarchical network comprised of IoUT devices, AUVs, and surface-stations. Note that it is not a simple conglomerate of its constituent components. For example, a rechargeable AUV and an IoUT device anchored underwater may consume the same energy but they have entirely different effects on the whole system, which deserves specific investigation.

Against this background, we design a multi-tier underwater computing (MTUC) framework intrinsically amalgamating both the computing and communications as well as storage resources of surface-stations and AUVs as well as IoUT devices for providing on-demand services for IoUT applications. Our new contributions are summarized as follows:

- To the best of our knowledge, this is the first attempt to integrate the surface-stations, AUVs, and IoUT devices to form an MTUC framework for providing on-demand underwater computing services instead of simply collecting the sensory data for satisfying the diverse requirements of advanced IoUT applications.
- Considering the limitations of both the AUV-centric and IoUT-centric designs, we conceive a system-level optimization model for maximizing the profits gleaned from the perspective of economics by integrating our environment-aware trajectory design, communication resource allocation, computation offloading and data caching.
- Since the problem formulated is NP-hard and high-dimensional, conventional methods cannot deal with it well. Hence, we transform it into a Markov decision process (MDP) and employ an asynchronous advantage actor-critic (A3C) algorithm [18] for solving it.
- Our simulation results show that the proposed scheme is capable of improving the system's profit by relying on environment-aware trajectory design and always exhibits better convergence speed and scalability in the face of an escalating problem dimension than other state-of-the-art schemes.

The remainder of the paper is organized as follows. Section II reviews the related state-of-the-art. In Section III, we describe the system model and formulate an optimization problem for maximizing the system's profit. Section IV introduces how we transform the optimization problem to an MDP and utilize A3C to solve it. In Section V, a range of experiments is carried out to show the efficiency of the proposed scheme. Section VI concludes the paper.

II. Related work

AUV-aided data collection has been extensively studied in recent years. Early efforts were focused on the collaborative transmissions of IoUT devices, while the trajectory of the AUV was usually assumed to be fixed [7], [19], [20]. As the first attempt to introduce AUV to relay the data of IoUT devices, Yoon *et al.* [7] proposed a new underwater routing scheme, where the sensing devices send their data to an aggregation device either directly or via a multi-hop transmission, and then the aggregation device transmits the data aggregated to the AUV when it passes by. For reducing the number of communication hops, the sensing devices intelligently select the next hop according to the aggregation device's preference. With the objective of minimizing the energy consumption of the IoUT devices, Chen *et al.* [19] conceived a novel routing protocol relying on the selective awake-sleep mechanism of IoUT devices and accurate estimation of the AUV's coverage range. Khan *et al.* [20] investigated an energy-efficient AUV-assisted clustering scheme, comprised of a fixed time-slot-based intra-cluster communication mechanism with a wake-up sleep cycle and a sectoring mechanism, which is capable of reducing the processing latency, while avoiding excessive energy consumption. However, the previous contributions relying on data aggregation among IoUT devices impose an excessive burden on the aggregation devices selected.

For enhancing the battery lives, AUVs may be intelligently configured to cruise and collect the data of all the IoUT devices. The trajectory design of AUVs will be revealed to have a significant impact on the performance of the IoUT system, including both IoUT devices and the AUV. Hence, a series of treatises were dedicated to the AUV's trajectory design, where some of them aim for reducing the operating cost of the AUV in terms of cruising distance or energy consumption [21]–[23]. Others focused on whether the IoUT devices' requirements are satisfied [10], [16], [17]. Specifically, considering the unreliable communications between the IoUT devices and AUVs, Hollinger *et al.* [21] formulated a communication-limited data collection problem as a special traveling salesperson problem (TSP) and presented both an AUV path planning method as well as a communication protocol to solve it. The efficiency of the proposed strategy was validated both under a deterministic access and a random access scenario. To reduce the cruising distance of the AUV, Ma *et al.* [22] designed a spanning tree covering algorithm for solving the path planning problem formulated. Faigl *et al.* [23] proposed employing a self-organizing map and an unsupervised learning technique to find a short path for the AUV considering the priority of the IoUT devices, which have a low computational complexity. This regime may also be readily extended to multi-AUV scenarios. With the emergence of advanced IoUT applications, such as mission-critical IoUTs [24], extremely stringent IoUT device requirements have to be considered. Hence, meeting these requirements of the IoUT applications with limited resources has drawn significant research attention. Bearing in mind that

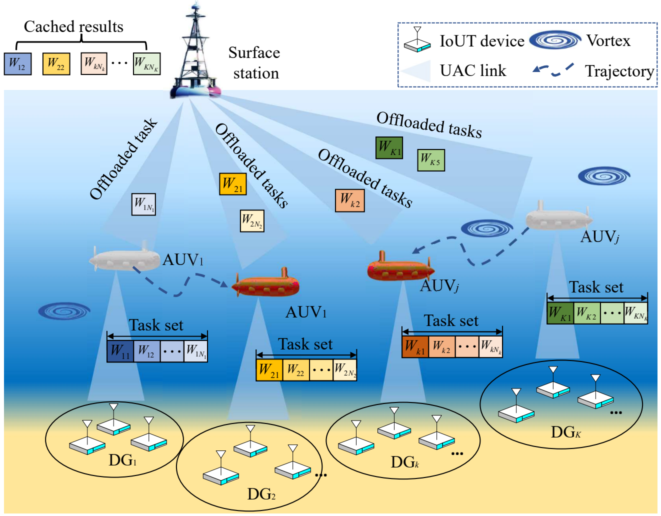


Fig. 1. The architecture of MTUC.

the value of the sensed data rapidly decays in time, the authors proposed a heuristic adaptive greedy AUV path-finding algorithm to find an optimal path having the maximal data value delivered to the aggregation devices [10]. Liu *et al.* [17] presented a hybrid data collection scheme, taking both the timeliness and energy efficiency requirements of IoUT devices into consideration. To guarantee the freshness of the collected data, Fang *et al.* [16] introduced the concept of age of information and designed a two-stage algorithm for the joint optimization of the resource allocation and trajectory planning of AUV-aided IoUTs. At the time of writing, however, there is no recommendation in the open literature for optimizing the amalgamated system's performance relying on integrating both the surface-station, AUVs, as well as the IoUT devices. It is beneficial to construct a system-level optimization framework for balancing the operating cost and meeting the IoUT devices' requirements. However there are some pioneering works on system-level optimization in terrestrial networks. Wang *et al.* [25] conceived a revenue-maximizing framework for cellular networks by jointly considering the computation offloading, resource allocation and content caching. Focusing on accuracy-aware machine learning (ML) tasks in the Internet of Industrial Things, Fan *et al.* [26] constructed a long-term average system cost optimization framework by jointly considering the resources of sensors, edge server and cloud server, as well as the inference accuracy of the ML tasks. However, when the application scenario changes from cellular networks to AUV-aided underwater networks, the research mentioned above is no longer applicable. Hence the system considered deserves further study. Therefore, in this paper, a max-profit problem, integrating environment-aware trajectory design, communication resource allocation, computation offloading and data caching, is conceived for filling this knowledge gap.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model

Fig. 1 shows our MTUC architecture, where multiple AUVs communicating with surface-stations perpetually cruising to provide computing service for a set of IoUT devices distributed in several device groups (DGs). Each AUV starts from the point of origin sight below the surface-station, and supports the assigned DGs in turn. We assume that there is a single surface-station, M AUVs, and K DGs. The M AUVs are denoted by the set $\boldsymbol{AUV} = \{AUV_1, AUV_2, \ldots, AUV_M\}$, while the K DGs are represented by the set $\boldsymbol{DG} = \{DG_1, DG_2, \ldots, DG_K\}$. Let us assume that there are a total of N_k IoUT devices located in DG_k, which are represented by a set $\boldsymbol{ND_k} = \{n_{k_1}, n_{k_2}, \ldots, n_{k_{N_k}}\}$. For brevity, let $\boldsymbol{M} = \{1, 2, \ldots, M\}$ represent the subscript of the AUVs, while $\boldsymbol{K} = \{1, 2, \ldots, K\}$ the subscript of the DGs, and $\boldsymbol{N_k} = \{1, 2, \ldots, N_k\}$ as the subscript of the IoUT devices located in DG$_k$. Let furthermore $\boldsymbol{P}_{\text{SS}} = (0, 0, H)$, $\boldsymbol{P}_j^{\text{A}} = \left(x_j^{\text{A}}, y_j^{\text{A}}, d_0\right)$, $\boldsymbol{P}_k^{\text{DG}} = \left(x_k^{\text{DG}}, y_k^{\text{DG}}, z_k^{\text{DG}}\right)$ and $\boldsymbol{P}_{ki}^{\text{S}} = \left(x_{ki}^{\text{S}}, y_{ki}^{\text{S}}, h_0\right)$ represent the three-dimensional (3D) Euclidean coordinates of the surface-station, AUV_k, DG_k, and IoUT device n_{k_i} located in DG_k, respectively.

We assume that each IoUT device has a task that has to be solved. The task generated by IoUT device n_{k_i} can be represented by the twin tuple $W_{k_i} \triangleq \{Z_{k_i}, \alpha_{k_i}\}$, where Z_{k_i} represents the size of the input data (in bit), while α_{k_i} is the computational complexity (in cycles/bit) indicating how many CPU cycles are required to process 1 bit of the data [11]. Let $\boldsymbol{O} = \{o_{k_i}, k \in \boldsymbol{K}, i \in \boldsymbol{N_k}\}$ denote the offloading strategy vector. If task W_{k_i} is offloaded to the surface-station via an AUV, we have $o_{k_i} = 1$, and $o_{k_i} = 0$ otherwise. Let $\boldsymbol{r} = \{0 \leq r_{k_i} \leq 1, k \in \boldsymbol{K}, i \in \boldsymbol{N_k}\}$ denote the bandwidth allocation vector to represent the specific proportion of the bandwidth resources allocated to the device n_{k_i}. The caching strategy vector is denoted by $\boldsymbol{H} = \{h_{k_i}, k \in \boldsymbol{K}, i \in \boldsymbol{N_k}\}$. We have $h_{k_i} = 1$, if the surface-station has cached the data of the task W_{k_i} and $h_{k_i} = 0$ otherwise. For convenience, the notations are summarized in Table I.

B. Communication Model

UAC has complex propagation characteristics, where both the multi-path effects, Doppler effects and environmental noise influence the quality of the link. For simplicity, we consider a shallow-water acoustic propagation environment assumed to be both spatially and temporally homogenous.

1) Noise model: The environmental noise in the ocean may be caused by bubbles, shipping activity, surface wind fields, etc. According to [27], [28], the power spectral density (p.s.d) of the four main types of noise in dB per Hz at the communication frequency f can be characterized by

$$10 \log N_\vartheta(f) = 17 - 30 \log f, \tag{1}$$

$$10 \log N_s(f) = 40 + 20\left(s - \frac{1}{2}\right) + 26 \log f - 60 \log(f + 0.03), \tag{2}$$

$$10 \log N_w(f) = 50 + 7.5 w^{\frac{1}{2}} + 20 \log f - 40 \log(f + 0.4), \tag{3}$$

$$10 \log N_{th}(f) = -15 + 20 \log f, \tag{4}$$

where $N_\vartheta(f), N_s(f), N_w(f)$ and $N_{th}(f)$ represent the turbulence noise, the shipping noise, the waves noise, and the thermal noise, respectively. Furthermore, $s \in [0, 1]$ is the

TABLE I
NOTATIONS

Notation	Meaning
M	Number of AUV
K	Number of device group
f	Communication frequency
s	Shipping activity factor
w	Wind speed
H	Depth of water
$\mathcal{V}$	Viscosity of the fluid
h_0	Height of the IoUT device from the seabed
d_0	Height of the AUV from the seabed
r_0	Radius of the vortex
Ω_0	Strength of the vortex
C_d	Dragging coefficient
C_a	Cross-sectional area
k_s	Spreading factor
ρ_L	Density of seawater
Z_{ki}	Size of input data
f_{ki}	CPU cycles per second
B_{L}	Bandwidth between AUV and device
B_{H}	Bandwidth between AUV and surface-station
$P_{\mathrm{tr}}^{\mathrm{A}}$	Transmitted power of AUV
$P_{\mathrm{tr}}^{\mathrm{D}}$	Transmitted power of IoUT device
ζ	Conversion efficiency of electricity
η	Overall efficiency of electronic circuitry
Γ_b, Γ_s	Coefficient factors related to the channel gain
ω_{k_i}	Unit revenue of reducing time of IoUT device
λ_{k_i}	Unit revenue of saving energy consumption of IoUT device
ϱ	Unit cost to the surface-station
χ	Unit cost to the AUV

shipping activity factor, while w represents the wind velocity (m/s). Hence the combined noise $N(f)$ can be represented as

$$N(f) = N_\vartheta(f) + N_s(f) + N_w(f) + N_{th}(f). \tag{5}$$

There is a two-phase transmission protocol, if the IoUT devices offload their data to the surface-station, including the IoUT devices to AUV, and AUV to the surface-station phases, which can be modeled as follows:

2) The first phase transmission: IoUT device → AUV: The UAC channel is the superposition of the direct line-of-sight (LOS) path and a collection of non-line-of-sight (NLOS) paths, where the NLOS paths are typically reflected by underwater surfaces, the seabed and the water-air surface. Fig. 2(a) depicts the geometry of the UAC between IoUT devices and the AUV, where $\boldsymbol{m}_u = (x_u^m, y_u^m, H), u \in \{1, 2, \ldots, \alpha\}$ and $\boldsymbol{n}_u = (x_u^n, y_u^n, 0), u \in \{1, 2, \ldots, \beta\}$ are the reflection points at the sea surface and the seabed, respectively, while H is the depth of water. Hence, the Euclidean distance of the LOS path is calculated as

$$l_{\mathrm{L}} = \left\| \boldsymbol{P}_{ki}^{\mathrm{S}} - \boldsymbol{P}_j^{\mathrm{A}} \right\|_2, \tag{6}$$

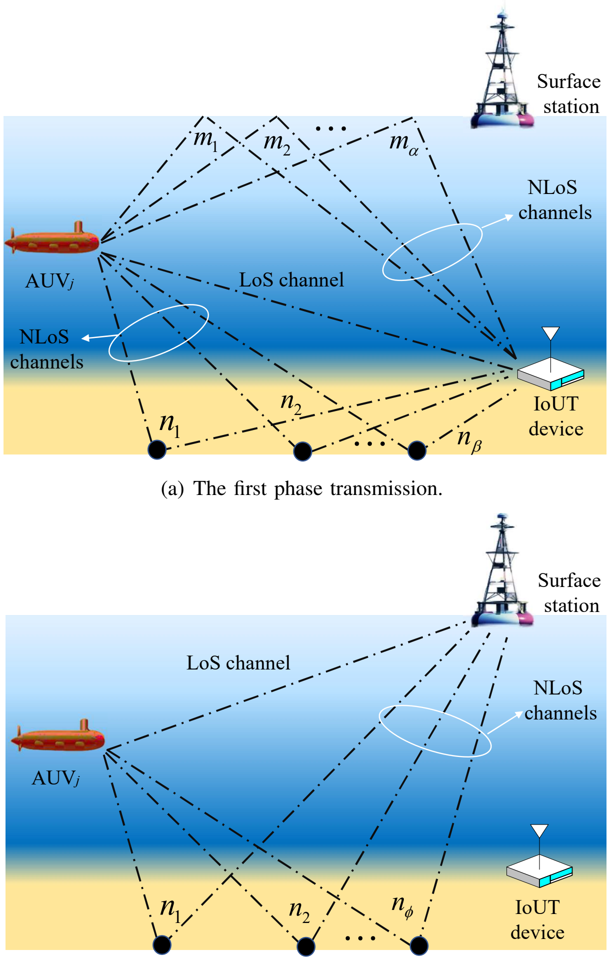
while the distance of the acoustic signal reflected from point $\boldsymbol{m}_u$ and point $\boldsymbol{n}_u$ of the NLOS propagation can be expressed as

$$l_m(\boldsymbol{m}_u) = \left\| \boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{m}_u \right\|_2 + \left\| \boldsymbol{P}_{ki}^{\mathrm{S}} - \boldsymbol{m}_u \right\|_2, \tag{7}$$

and

$$l_n(\boldsymbol{n}_u) = \left\| \boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{n}_u \right\|_2 + \left\| \boldsymbol{P}_{ki}^{\mathrm{S}} - \boldsymbol{n}_u \right\|_2, \tag{8}$$

respectively. Since NLOS paths have lost much of their energy after multiple reflections, we only have to pay attention to a finite number of significant paths [27]. Furthermore, obtaining the lower bound of the signal-to-noise ratio (SNR) is more beneficial by finding the minimum NLOS path lengths. We can easily to calculate the shortest NLOS path lengths $l_m(\boldsymbol{m}^\star)$ and $l_n(\boldsymbol{n}^\star)$ reflected from the top and bottom surfaces as



(a) The first phase transmission.

(b) The second phase transmission.

Fig. 2. Multi-path effect.

$$l_m(\boldsymbol{m}^\star) = \sqrt{\left(x_j^{\mathrm{A}} - x_{ki}^{\mathrm{S}}\right)^2 + \left(y_j^{\mathrm{A}} - y_{ki}^{\mathrm{S}}\right)^2 + (2H - h_0 - d_0)^2}, \tag{9}$$

and

$$l_n(\boldsymbol{n}^\star) = \sqrt{\left(x_j^{\mathrm{A}} - x_{ki}^{\mathrm{S}}\right)^2 + \left(y_j^{\mathrm{A}} - y_{ki}^{\mathrm{S}}\right)^2 + (h_0 + d_0)^2}, \tag{10}$$

respectively.

Let $A(l, f)$ be the attenuation at frequency f over the distance l, which is given by

$$A(l, f) = l^{k_s} a(f)^l, \tag{11}$$

where k_s represents the spreading factor, and $a(f)$ is the absorption coefficient, which can be expressed empirically in dB per km with f in KHz as follows [29]

$$10 \log a(f) = \frac{0.11 f^2}{1 + f^2} + \frac{44 f^2}{4100 + f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003. \tag{12}$$

Therefore, the normalized SNR of a signal with unity transmitted power and bandwidth can be represented as

$$\gamma(l,f)=\frac{1}{A(l,f)N(f)}. \tag{13}$$

The lower bound of the SNR considering the minimum NLOS path lengths derived from Eq. (9) and Eq. (10) can be expressed as [30]

$$\begin{aligned}\gamma\left(l_{\mathrm{L}},f\right)_{\min}=\frac{1}{N(f)}\cdot\Bigg\{&\frac{1}{\sqrt{A\left(l_{\mathrm{L}},f\right)}}-\frac{\alpha\Gamma_s}{\sqrt{A\left(l_m\left(\boldsymbol{m}^{\star}\right),f\right)}}\\&-\frac{\beta\Gamma_b}{\sqrt{A\left(l_n\left(\boldsymbol{n}^{\star}\right),f\right)}}\Bigg\}^2,\end{aligned} \tag{14}$$

where Γ_s and Γ_b characterize the channel gain of the shortest NLOS path reflected from the top and bottom surfaces, respectively.

Hence, the data rate between the AUV$_j$ and IoUT device n_{k_i} can be formulated as

$$R_{k_i}^{\mathrm{DA}}=r_{k_i}B_{\mathrm{L}}\log_2\left(1+\frac{\eta P_{\mathrm{tr}}^{\mathrm{D}}\gamma\left(l_{\mathrm{L}},f\right)_{\min}}{2\pi H_1(1\mu\mathrm{Pa})r_{k_i}B_{\mathrm{L}}}\right), \tag{15}$$

where B_{L} is the total bandwidth of AUV$_j$, while $P_{\mathrm{tr}}^{\mathrm{D}}$ denotes the transmitted power of the IoUT device, respectively. Furthermore, η is the overall efficiency of the electronic circuitry including both the power amplifier and transducer [9], while H_1 is the water depth of IoUT devices and r_{k_i} denotes the proportion of the bandwidth allocated to the IoUT device n_{k_i}, which satisfies

$$\begin{cases}r_{k_i}=0,\ if\ o_{k_i}=0,\forall k\in\boldsymbol{K},i\in\boldsymbol{N}_k,\\ r_{k_i}=0,\ if\ h_{k_i}=1,\forall k\in\boldsymbol{K},i\in\boldsymbol{N}_k,\\ \sum\limits_{i=1}^{N_k}o_{k_i}(1-h_{k_i})r_{k_i}\le 1,\forall k\in\boldsymbol{K}.\end{cases} \tag{16}$$

3) The second phase transmission: AUV → surface-station: As shown in Fig. 2(b), $\boldsymbol{w}_u=\left(x_u^w,y_u^w,0\right),u\in\{1,2,\ldots,\phi\}$ is the reflection point at the seabed of multi-path propagation. Hence, the Euclidean distance of the LOS path and the NLOS path is given by

$$l_{\mathrm{H}}=\left\|\boldsymbol{P}_j^{\mathrm{A}}-\boldsymbol{P}_{\mathrm{SS}}\right\|_2 \tag{17}$$

and

$$l_w^{\mathrm{H}}\left(\boldsymbol{w}_u\right)=\left\|\boldsymbol{P}_j^{\mathrm{A}}-\boldsymbol{w}_u\right\|_2+\left\|\boldsymbol{w}_u-\boldsymbol{P}_{\mathrm{SS}}\right\|_2, \tag{18}$$

respectively. Thus we can obtain the minimum NLOS path as

$$l_w^{\mathrm{H}}\left(\boldsymbol{w}^{\star}\right)=\sqrt{\left(x_j^{\mathrm{A}}\right)^2+\left(y_j^{\mathrm{A}}\right)^2+\left(h_0+d_0\right)^2}. \tag{19}$$

Similar to Eq. (11)-(14), the lower bound of SNR $\gamma\left(l_{\mathrm{H}}\right)_{\min}$ at the surface-station subjected to NLOS propagation is given by

$$\gamma\left(l_{\mathrm{H}},f\right)_{\min}=\frac{1}{N(f)}\cdot\left\{\frac{1}{\sqrt{A\left(l_{\mathrm{H}},f\right)}}-\frac{\beta\Gamma_b}{\sqrt{A\left(l_w^{\mathrm{H}}\left(\boldsymbol{w}^{\star}\right),f\right)}}\right\}^2, \tag{20}$$

The classic code division multiple access (CDMA) is adopted for the UAC links between the AUVs and surface-station, and the data rate between them can be calculated as

$$R_k^{\mathrm{AS}}=B_{\mathrm{H}}\log_2\left(1+\frac{\eta P_{\mathrm{tr}}^{\mathrm{A}}\gamma\left(l_{\mathrm{H}},f\right)_{\min}}{2\pi H_2(1\mu\mathrm{Pa})B_{\mathrm{H}}}\right), \tag{21}$$

where $\boldsymbol{P}_{\mathrm{tr}}^{\mathrm{A}}$ represents the transmitted power of the AUV, while B_{H} and H_2 represent the available bandwidth and the water depth of AUVs, respectively.

C. Caching Model

Because there are often repeated requests for tackling the same task, caching some data of the previous requested task is capable of reducing the backhaul latency and alleviate the pressure on the backhaul bandwidth [31], [32]. If task W_{k_i} is cached by the surface-station, $h_{k_i}=1$ and o_{k_i} should be 1, to avoid it being processed locally on the device for saving device's energy and reducing the processing latency. Hence the binary variable h_{k_i} of caching decision should satisfy

$$h_{k_i}\le o_{k_i}. \tag{22}$$

Moreover, since the storage capacity of surface-station is typically limited, the caching strategy should satisfy [33]

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}h_{k_i}Z_{k_i}\le C_e, \tag{23}$$

where C_e is the maximal storage capacity of the surface-station.

D. Computing Model

Next we discuss the processing time of local vs. offloaded surface-station based computation.

1) Local Computing: The computing capability of n_{k_i} is denoted by f_{k_i} and different IoUT devices have different computing capabilities. The duration of completing the task W_{k_i} locally is calculated as

$$T_{k_i}^{\mathrm{L}}=\frac{\alpha_{k_i}Z_{k_i}}{f_{k_i}}. \tag{24}$$

2) Computing at surface-station: If task W_{k_i} is offloaded to the surface-station for processing, two-stage transmission is needed. But if task W_{k_i} has already been cached at the surface-station, the data transmission procedure is eliminated. The transmission time of tackling the task at the surface-station can be represented as

$$T_{k_i}^{\mathrm{T}}=T_{k_i}^{\mathrm{DA}}+T_{k_i}^{\mathrm{AS}}, \tag{25}$$

where $T_{k_i}^{\mathrm{DA}}$ and $T_{k_i}^{\mathrm{AS}}$ is the duration of transmitting the data from the IoUT device to the AUV and that from the AUV to the surface-station, respectively. The duration of downloading the results from the surface-station is usually ignored, since the results are more compact than the input data size of W_{k_i} [34]. To elaborate, $T_{k_i}^{\mathrm{DA}}$ and $T_{k_i}^{\mathrm{AS}}$ are calculated as

$$T_{k_i}^{\mathrm{DA}}=(1-h_{k_i})\frac{Z_{k_i}}{R_{k_i}^{\mathrm{DA}}}, \tag{26}$$

and

$$T_{k_i}^{\mathrm{AS}}=(1-h_{k_i})\frac{Z_{k_i}}{R_k^{\mathrm{AS}}}, \tag{27}$$

respectively.

Furthermore, to improve efficiency and reduce the energy consumption, it is beneficial to sparingly activate the limited computing resources. Assume that the total computational resource allocated for each AUV by surface-station is denoted by F, and $\boldsymbol{F} = \{f_{k_i}^m, k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$ represents the computing resource allocation vector, where $f_{k_i}^m \in [0,1]$ is the specific proportion of F allocated to n_{k_i}. Hence the computing time of task W_{k_i} at the surface-station is given by

$$T_{k_i}^{\mathrm{M}} = \frac{\alpha_{k_i} Z_{k_i}}{f_{k_i}^m F}, \tag{28}$$

where the computing resource allocation vector should satisfy

$$\begin{cases} f_{k_i}^m = 0, \ if \ o_{k_i} = 0, \forall k \in \boldsymbol{K}, i \in \boldsymbol{N}_k, \\ \sum\limits_{i=1}^{N_k} o_{k_i} f_{k_i}^m \leq 1, \forall k \in \boldsymbol{K}. \end{cases} \tag{29}$$

Overall, the total duration of tackling task W_{k_i} is represented as

$$T_{k_i} = o_{k_i}(T_{k_i}^M + T_{k_i}^T) + (1 - o_{k_i})T_{k_i}^L, \tag{30}$$

while that of addressing all tasks in DG_k is given by

$$A_k = \max_{i=\{1,2,\ldots,N_k\}} T_{k_i}, \forall k \in \boldsymbol{K}. \tag{31}$$

E. Trajectory Model

We assume that AUV_j servers S_j DGs, and we define $\boldsymbol{P}_j^{\mathrm{A}}[\xi], \forall \xi \in \boldsymbol{S_j} = \{0, 1, 2, \ldots, S_j, S_j + 1\}$ as the trajectory of AUV_j, which starts from the surface-station, i.e., $\boldsymbol{P}_j^{\mathrm{A}}[0]$, passes through all the assigned DGs and finally returns to the surface-station, i.e., $\boldsymbol{P}_j^{\mathrm{A}}[S_j + 1]$ for recharging. Therefore, we have

$$\boldsymbol{P}_j^{\mathrm{A}}[S_j + 1] = \boldsymbol{P}_j^{\mathrm{A}}[0], \forall j \in \boldsymbol{M}. \tag{32}$$

We define $d_j[\xi]$ as the distance between the two points, which can be calculated as

$$d_j[\xi] = \left\| {\boldsymbol{P}_j}^{\mathrm{A}}[\xi + 1] - {\boldsymbol{P}_j}^{\mathrm{A}}[\xi] \right\|_2, \forall j \in \boldsymbol{M}, \forall \xi \in \{0, 1, 2, \ldots, S_j\}. \tag{33}$$

Let $Y_{j_k}[\xi] = 1$ represent that AUV_j selects the DG_k as its ξ-th hovering DG, otherwise $Y_{j_k}[\xi] = 0$. The AUV trajectory design strategy can be represented by $\boldsymbol{Y} = \{Y_{j_k}[\xi], j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}, k \in \boldsymbol{K}\}$. In order to guarantee that each DG can be covered and served only once, we have

$$\sum_{j=1}^{M} \sum_{\xi=1}^{S_j} Y_{j_k}[\xi] = 1, \forall j \in \boldsymbol{M}, \forall k \in \boldsymbol{K}, \tag{34}$$

$$\sum_{\xi=1}^{S_j} \sum_{k=1}^{K} Y_{j_k}[\xi] = S_j, \forall j \in \boldsymbol{M}, \tag{35}$$

and

$$\sum_{j=1}^{M} S_j = K, \forall j \in \boldsymbol{M}. \tag{36}$$

Therefore, the total hovering time of AUV_j can be expressed as

$$T_j^{\mathrm{H}} = \sum_{k=1}^{K} \sum_{\xi=1}^{S_j} Y_{j_k}[\xi] A_k, \tag{37}$$

where trajectory of AUV_j is composed of $S_j + 1$ sub-trajectories. We assume that each AUV moves along the segment at a constant velocity V_k. Therefore, the time of AUV_j in each sub-trajectory is given by

$$t_j^{\mathrm{F}}[\xi] = \frac{d_j[\xi]}{V_k}, \forall j \in \boldsymbol{M}, \forall \xi \in \{0, 1, 2, \ldots, S_j\}. \tag{38}$$

Furthermore, the total travelling distance and the travelling time of AUV_j are

$$I_j = \sum_{\xi=0}^{S_j} d_j[\xi], \forall j \in \boldsymbol{M}, \tag{39}$$

and

$$T_j^{\mathrm{F}} = \frac{I_j}{V_k}, \tag{40}$$

respectively. Consequently, the total cruising time of AUV_j in a cycle is given by

$$T_j^{\mathrm{AT}} = T_j^{\mathrm{F}} + T_j^{\mathrm{H}}, \forall j \in \boldsymbol{M}. \tag{41}$$

To strike a balance, we define ε as a constraint for limiting the difference of travelling time among different AUVs as

$$T_{\max}^{\mathrm{AT}} - T_{\min}^{\mathrm{AT}} \leq \varepsilon, \tag{42}$$

where $T_{\max}^{\mathrm{AT}} = \max\left\{T_j^{\mathrm{AT}}, j \in \boldsymbol{M}\right\}$ and $T_{\min}^{\mathrm{AT}} = \min\left\{T_j^{\mathrm{AT}}, j \in \boldsymbol{M}\right\}$.

F. Motion Model

The underwater oceanic environment is complex and hostile with dynamically fluctuating water velocity, vortex, etc., which may impose a significant impact on the AUV's movement. To quantify it, we construct a model for evaluating the effects of the turbulent oceanic environments on AUV's motion based on the Navier-Stokes equation [1]. Specifically, the oceanic current field can be represented as [36]

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\vec{V_{\mathrm{C}}} \nabla) \boldsymbol{\omega} = \mathcal{V} \Delta \boldsymbol{\omega}, \tag{43}$$

where $\vec{V_{\mathrm{C}}} = (V_x, V_y, V_z)$ represents the velocity field, while $\boldsymbol{\omega} = (\frac{\partial V_z}{\partial y} - \frac{\partial V_y}{\partial z})\vec{i} + (\frac{\partial V_x}{\partial z} - \frac{\partial V_z}{\partial x})\vec{j} + (\frac{\partial V_y}{\partial x} - \frac{\partial V_x}{\partial y})\vec{k}$ denotes the vorticity of the current. Furthermore, $\mathcal{V}$ is the viscosity of the fluid, while ∇ and Δ represent the gradient and Laplacian operator, respectively. To facilitate the analysis, we approximate the Navier-Stokes equation as

$$V_x(\boldsymbol{P}_j^{\mathrm{A}}) = -\frac{\Omega_0 \cdot \left(y_j^{\mathrm{A}} - y_0\right)}{2\pi \left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2} \left(1 - e^{-\frac{\left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2}{r_0^2}}\right), \tag{44}$$

$$V_y(\boldsymbol{P}_j^{\mathrm{A}}) = \frac{\Omega_0 \cdot \left(x_j^{\mathrm{A}} - x_0\right)}{2\pi \left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2} \left(1 - e^{-\frac{\left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2}{r_0^2}}\right), \tag{45}$$

[1] In practice, most commercial AUVs are equipped with the horizontal acoustic Doppler current profiler (H-ADCP) and Doppler velocity logger (DVL), which can measure ocean current velocity profiles up to hundreds of meters in front of the AUV with an accuracy of 1% of the measured magnitude ± 5 mm/s [35].

$$V_z(\boldsymbol{P}_j^{\mathrm{A}}) = \frac{\Omega_0 \cdot (d_0 - z_0)}{2\pi \left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2} \left(1 - e^{-\frac{\left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2}{r_0^2}}\right), \tag{46}$$

and

$$\boldsymbol{\omega}(\boldsymbol{P}_j^{\mathrm{A}}) = \frac{\Omega_0}{\pi r_0^2} e^{-\frac{\left\|\boldsymbol{P}_j^{\mathrm{A}} - \boldsymbol{P}_0\right\|_2^2}{r_0^2}}, \tag{47}$$

where $\boldsymbol{P}_j^{\mathrm{A}}$ and $\boldsymbol{P}_0 = (x_0, y_0, z_0)$ denote the coordinates of AUV_j and the center of the Lamb vortex [37], respectively. Furthermore Ω_0 and r_0 represent the strength and radius of the vortex, respectively. In fact, most of the energy consumption of the AUV is dissipated by overcoming the resistance of the water for maintaining the velocity V_k. To determine the propulsion force of AUV_j required for maintaining a given velocity V_k, the relative velocity between the AUV and the current should be derived, which can be expressed as

$$\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}) = V_k \cdot \vec{e}_k - \vec{V}_{\mathrm{C}}(\boldsymbol{P}_j^{\mathrm{A}}), \tag{48}$$

where $\vec{V}_{\mathrm{C}}(\boldsymbol{P}_j^{\mathrm{A}})$ denotes the water flow velocity, while $\vec{e}_k$ is the unit vector of the direction of the AUV_j. According to classic computational fluid dynamics (CFD) methods [38], the drag force required for floating and for moving can be expressed as

$$F_j^{\mathrm{H}} = \frac{1}{2}\rho_{\mathrm{L}} \left\|\vec{V}_{\mathrm{C}}(\boldsymbol{P}_j^{\mathrm{A}})\right\|_2^2 C_a C_d, \tag{49}$$

and

$$F_j^{\mathrm{F}} = \frac{1}{2}\rho_{\mathrm{L}} \left\|\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}})\right\|_2^2 C_a C_d, \tag{50}$$

respectively, where C_d denotes the dragging coefficient, while ρ_{L} and C_a represent the density of seawater and the cross-sectional area of the AUV moving along the current direction.

G. Energy Consumption Model

In the following, we analyze the energy consumption of the MTUC from the perspective of the user (i.e., IoUT devices) and the service provider (i.e., surface-station and AUVs), respectively.

1) The energy consumption of users: For IoUT devices, the energy is mainly consumed either by local computations or by transmissions, when tasks are offloaded to the surface-station. If task W_{k_i} solved locally, the energy consumption of computing is formulated by

$$E_{k_i}^{\mathrm{L}} = \mu (f_{k_i})^{\sigma} T_{k_i}^{\mathrm{L}}, \tag{51}$$

where f_{k_i} is the CPU frequency of the IoUT device n_{k_i}. According to [39], [40], μ is a constant that depends on the average switched capacitance and the average activity factor, while σ is a constant close to 3. By contrast, if task W_{k_i} is transmitted to the AUV for further processing at the surface-station, the corresponding transmit energy consumption consumed of IoUT device n_{k_i} is given by [13]

$$E_{k_i}^{\mathrm{DA}} = (1 - h_{k_i}) \frac{2\pi (1\mu\mathrm{Pa}) r_{k_i} B_{\mathrm{L}}}{\eta\gamma\left(l_L, f\right)_{\min}} \left[2^{\frac{R_{k_i}^{\mathrm{DA}}}{r_{k_i} B_{\mathrm{L}} T_{k_i}^{\mathrm{DA}}}} - 1\right] T_{k_i}^{\mathrm{DA}}. \tag{52}$$

2) The energy consumption of service provider: The energy consumption of the service provider is composed of two parts, including that of the surface-station solving the tasks and that of the AUVs for cruising and forwarding the tasks.

Specifically, for the surface-station, similar to (51), when task W_{k_i} is offloaded to the surface-station, the energy consumption of this is given by

$$E_{k_i}^{\mathrm{M}} = \mu (f_{k_i}^{m} F)^{\sigma} T_{k_i}^{\mathrm{M}}. \tag{53}$$

Furthermore, as for the AUVs, similar to Eq. (52), upon forwarding task W_{k_i} from n_{k_i} to a surface-station, the transmit energy consumption consumed by a AUV is formulated by

$$E_{k_i}^{\mathrm{AS}} = (1 - h_{k_i}) \frac{2\pi (1\mu\mathrm{Pa}) B_{\mathrm{H}}}{\eta\gamma\left(l_{\mathrm{H}}, f\right)_{\min}} \left[2^{\frac{R_k^{\mathrm{AS}}}{B_{\mathrm{H}} \cdot T_{k_i}^{\mathrm{AS}}}} - 1\right] T_{k_i}^{\mathrm{AS}}. \tag{54}$$

As for the energy consumption of the AUV's movement, it should be discussed in two scenarios, namely for hovering above the DGs and for moving between two destinations. According to Eq. (49), the drag force required to stay afloat above the ξ-th DG is formulated by

$$F_j^{\mathrm{H}}[\xi] = \frac{1}{2}\rho_{\mathrm{L}} \left\|\vec{V}_C(\boldsymbol{P}_j^{\mathrm{A}}[\xi])\right\|_2^2 C_a C_d. \tag{55}$$

Consequently, the electric power generating the required force is calculated as

$$P_j^{\mathrm{H}}[\xi] = \frac{1}{\zeta} F_j^{\mathrm{H}}[\xi] \left\|\vec{V}_C(\boldsymbol{P}_j^{\mathrm{A}}[\xi])\right\|_2, \tag{56}$$

where ζ is the electricity conversion efficiency. Since the water flow velocity is different at each point during the movement of the AUV, this will impose significant challenges on our further analysis. Therefore, we approximate the average relative flow velocity in a sub-trajectory by the average of the relative flow velocity at the starting point, the midpoint and the end of this sub-trajectory. The more DGs are deployed in the same area, the closer the approximation to reality. The average relative flow velocity of AUV_j moving from the ξ-th DG to the $(\xi+1)$-st DG is given by

$$\begin{aligned} &\overline{\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}[\xi])} = \\ &\frac{1}{3}(\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}[\xi]) + \vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}[\xi+1]) + \vec{V}_{R_k}(\hat{\boldsymbol{P}}_j^{\mathrm{A}}[\xi_{mid}])) \\ &\forall \xi \in \{0, 1, 2, \ldots, S_j\}, \end{aligned} \tag{57}$$

where $\boldsymbol{P}_j^{\mathrm{A}}[\xi]$ and $\boldsymbol{P}_j^{\mathrm{A}}[\xi+1]$ are the coordinates of the ξ-th DG and the $(\xi+1)$-st DG, while $\hat{\boldsymbol{P}}_j^{\mathrm{A}}[\xi_{mid}]$ represents the coordinates of the middle point between the ξ-th DG and the $(\xi+1)$-st DG. Therefore, according to Eq. (50), the drag force required for supporting AUV_j movement from the ξ-th DG to the $(\xi+1)$-st DG is calculated as

$$F_j^{\mathrm{F}}[\xi] = \frac{1}{2}\rho_L \left\|\overline{\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}[\xi])}\right\|_2^2 C_a C_d. \tag{58}$$

Consequently, the corresponding electric power is represented by

$$P_j^{\mathrm{F}}[\xi] = \frac{1}{\zeta} \cdot F_j^{\mathrm{F}}[\xi] \cdot \left\|\overline{\vec{V}_{R_k}(\boldsymbol{P}_j^{\mathrm{A}}[\xi])}\right\|_2. \tag{59}$$

As a result, the energy consumption of AUV_j cruising through a specific cycle is given by

$$E_j = \sum_{k=1}^{K}\sum_{\xi=1}^{S_j} Y_{j_k}[\xi] A_k P_j^{\mathrm{H}}[\xi] + \sum_{\xi=0}^{S_j} t_j^{\mathrm{F}}[\xi] P_j^{\mathrm{F}}[\xi]. \tag{60}$$

H. Utility Function

Our proposed MTUC framework aims for maximizing the profit of the whole system. Specifically, the latency and energy consumption improvement of the users, i.e., IoUT devices, are deemed to be the revenue, while the cost is the energy consumption imposed on the service provider, namely the AUV and the surface-station. The profit is calculated by the revenue minus cost.

To elaborate, with the assistance of our MTUC framework, the computation latency and energy consumption of the task W_{k_i} can be reduced to

$$T_{k_i}^{\mathrm{S}} = o_{k_i}\left[T_{k_i}^{\mathrm{L}} - (T_{k_i}^{\mathrm{M}} + T_{k_i}^{\mathrm{T}})\right], \tag{61}$$

and

$$E_{k_i}^{\mathrm{S}} = o_{k_i}(E_{k_i}^{\mathrm{L}} - E_{k_i}^{\mathrm{DA}}), \tag{62}$$

respectively. Accordingly, the revenue that the MTUC framework can obtain is given by [25], [41]

$$Re = \sum_{k=1}^{K}\sum_{i=1}^{N_k} \omega_{k_i} T_{k_i}^{\mathrm{S}} + \lambda_{k_i} E_{k_i}^{\mathrm{S}}, \tag{63}$$

where the ω_{k_i} and λ_{k_i} are the unit revenue attained by reducing the time and by saving energy for the IoUT device n_{k_i}, respectively.

The cost that the MTUC framework has to bear is composed of the cost of solving the computing task and supporting the AUVs' movements. Specifically, the cost of the surface-station and of the AUV for solving task W_{k_i} is given by

$$CT_{k_i}^{\mathrm{M}} = o_{k_i}\varrho E_{k_i}^{\mathrm{M}}, \tag{64}$$

and

$$CT_{k_i}^{\mathrm{AS}} = o_{k_i}\chi E_{k_i}^{\mathrm{AS}}, \tag{65}$$

respectively, where ϱ denotes the unit cost to the surface-station, while χ is the unit cost to the AUV. It is noted that the values of ϱ and χ are different because of the difference in the difficulty of replenishing the energy of the surface-station and of the AUV. Therefore, the cost of solving all the tasks for the MTUC framework is calculated as

$$CT = \sum_{k=1}^{K}\sum_{i=1}^{N_k}\left(CT_{k_i}^{\mathrm{M}} + CT_{k_i}^{\mathrm{AS}}\right), \tag{66}$$

In fact, most of the energy consumption is dissipated by the AUV's movement[2]. Hence the cost caused by AUV's movement is given by

$$CF = \sum_{j=1}^{M}\chi E_j, \tag{67}$$

Therefore, the profit of the MTUC can be obtained by

$$\begin{aligned} Pr &= Re - CT - CF \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N_k}(\omega_{k_i} T_{k_i}^{\mathrm{S}} + \lambda_{k_i} E_{k_i}^{\mathrm{S}} - CT_{k_i}^{\mathrm{M}} - CT_{k_i}^{\mathrm{AS}}) - \sum_{j=1}^{M}\chi E_j. \end{aligned} \tag{68}$$

For maximizing the profit defined by Eq. (68), we jointly optimize the computation offloading strategy $\boldsymbol{O}$, caching strategy $\boldsymbol{H}$, bandwidth allocation $\boldsymbol{R}$, computing resource allocation $\boldsymbol{F}$ and trajectory design strategy $\boldsymbol{Y}$. This optimization problem is formulated as

$$\mathcal{P}1: \max_{\boldsymbol{O},\boldsymbol{H},\boldsymbol{R},\boldsymbol{F},\boldsymbol{Y}} Pr \tag{69a}$$

$$s.t. \quad r_{k_i} = 0,\ if\ o_{k_i} = 0, \forall k \in \boldsymbol{K}, i \in \boldsymbol{N}_k, \tag{69b}$$

$$r_{k_i} = 0,\ if\ h_{k_i} = 1, \forall k \in \boldsymbol{K}, i \in \boldsymbol{N}_k, \tag{69c}$$

$$\sum_{i=1}^{N_k} o_{k_i}(1 - h_{k_i}) r_{k_i} \le 1, \forall k \in \boldsymbol{K}, \tag{69d}$$

$$f_{k_i}^{m} = 0,\ if\ o_{k_i} = 0, \forall k \in \boldsymbol{K}, i \in \boldsymbol{N}_k, \tag{69e}$$

$$\sum_{j=1}^{M}\sum_{\xi=1}^{S_j} Y_{j_k}[\xi] = 1, \forall j \in \boldsymbol{M}, \forall k \in \boldsymbol{K}, \tag{69f}$$

$$\sum_{i=1}^{N_k} o_{k_i} f_{k_i}^{m} \le 1, \forall k \in \boldsymbol{K}, \tag{69g}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k} h_{k_i} Z_{k_i} \le C_e, \tag{69h}$$

$$\boldsymbol{P}_j^{\mathrm{A}}[S_j + 1] = \boldsymbol{P}_j^{\mathrm{A}}[0], \forall j \in \boldsymbol{M}, \tag{69i}$$

$$\sum_{\xi=1}^{S_j}\sum_{k=1}^{K} Y_{j_k}[\xi] = S_j, \forall j \in \boldsymbol{M}, \tag{69j}$$

$$\sum_{j=1}^{M} S_j = K, \forall j \in \boldsymbol{M}, \tag{69k}$$

$$T_{\max}^{\mathrm{AT}} - T_{\min}^{\mathrm{AT}} \le \varepsilon, \tag{69l}$$

$$h_{k_i} \le o_{k_i}. \tag{69m}$$

As for $\mathcal{P}1$, we have following proposition:

Proposition 1: $\mathcal{P}1$ is NP-hard, and we cannot find an optimal solution in polynomial time.

Proof: The detail proof is provided in Appendix. A. ■

[2] A DG typically has a dozen to dozens of IoUT devices [9], [13], [29]. Assume that each device has an image processing task that has to be solved with the data size of 300 Kb and computational complexity of 2000 cycles/bit. If all the IoUT devices select to offload their task to surface-station, the transmit energy consumption of the AUV is about 5000 J, while the computing energy consumption of the surface-station is about 1000 J. Furthermore, the energy consumption of the AUV for floating above a DG and for moving from the DG to the next DG is about 6000 J. Hence it is feasible to consider the energy consumption of computation and transmission together with the energy consumption of motion of the AUV.

IV. Deep Reinforcement Learning Solution

Since the problem formulated is non-convex and NP-hard, which is generally intractable for conventional optimization methods, therefore, we introduce A3C [42], an efficient distributed deep reinforcement learning approach, for solving $\mathcal{P}1$. Briefly, A3C combines the advantages of both value-based and policy-based reinforcement learning algorithms, which can deal with both continuous and discrete valued problems and implement an asynchronous update for improving learning efficiency.

A. Modeling of Deep Reinforcement Learning Environment

Specifically, we need to transform $\mathcal{P}1$ to an MDP firstly, which consists of state space, action space, policy, state transition matrix function, and reward function.

State Space: At each episode ϑ, the state $s(\vartheta) \in \mathcal{S}$ includes the following parts:

- The coordinates of AUVs at episode ϑ: $\left\{\boldsymbol{P}_j^{\mathrm{A}}(\xi, \vartheta), j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}\right\}$;
- The offloading strategy at episode $\vartheta - 1$: $\{o_{k_i}(\vartheta-1), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The caching strategy at episode $\vartheta - 1$: $\{h_{k_i}(\vartheta-1), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The bandwidth allocation at episode $\vartheta - 1$: $\{r_{k_i}(\vartheta-1), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The computing resource allocation at episode $\vartheta - 1$: $\left\{f_{k_i}^{m}(\vartheta-1), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\right\}$;
- The trajectory design strategy at episode $\vartheta - 1$: $\{Y_{j_k}(\xi, \vartheta-1), j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}, k \in \boldsymbol{K}\}$;

Hence, the state at episode ϑ can be summarized as

$$s(\vartheta) = \left\{\boldsymbol{P}_j^{\mathrm{A}}(\xi, \vartheta), o_{k_i}(\vartheta-1), h_{k_i}(\vartheta-1), r_{k_i}(\vartheta-1),\right.$$
$$\left. f_{k_i}^{m}(\vartheta-1), Y_{j_k}(\xi, \vartheta-1), j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}, k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\right\}. \tag{70}$$

Action Space: At each episode ϑ, the agent selects an action $a(\vartheta) \in \mathcal{A}$ according to the observed state $s(\vartheta)$, where $a(\vartheta)$ consists of the following parts:

- The offloading strategy at episode ϑ: $\{o_{k_i}(\vartheta), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The caching strategy of task at episode ϑ: $\{h_{k_i}(\vartheta), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The bandwidth allocation at episode ϑ: $\{r_{k_i}(\vartheta), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\}$;
- The computing resource allocation at episode ϑ: $\left\{f_{k_i}^{m}(\vartheta), k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\right\}$;
- The trajectory design strategy at episode ϑ: $\{Y_{j_k}(\xi, \vartheta), j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}, k \in \boldsymbol{K}\}$;

Hence, the action at episode ϑ can be formulated as

$$\begin{aligned} a(\vartheta) = &\left\{o_{k_i}(\vartheta), h_{k_i}(\vartheta), r_{k_i}(\vartheta), f_{k_i}^{m}(\vartheta), Y_{j_k}(\xi, \vartheta)\right. \\ &\left. f_{k_i}^{m}(\vartheta-1), Y_{j_k}(\xi, \vartheta-1), j \in \boldsymbol{M}, \xi \in \boldsymbol{S_j}, k \in \boldsymbol{K}, i \in \boldsymbol{N}_k\right\}. \end{aligned} \tag{71}$$

Policy: Let $\pi(a \mid s) = \mathcal{P}(a \mid s)$ denote the policy function, which is a probability distribution based on the observed state to make a decision to select an action.

State Transition Function: Let $\mathcal{P}[s(\vartheta+1) \mid s(\vartheta), a(\vartheta)]$ be the transition probability at each episode, which is the probability of entering into the state $s(\vartheta+1)$ after executing action $a(\vartheta)$ at the observed state $s(\vartheta)$.

Reward Function: The reward function is the objective of Eq. (69a) for the sake of maximizing the profit of the MTUC framework, which is represented as

$$\begin{aligned} &r\left(s(\vartheta), a(\vartheta)\right) = \\ &\sum_{k=1}^{K}\sum_{i=1}^{N_k}(\omega_{k_i} T_{k_i}^{\mathrm{S}} + \lambda_{k_i} E_{k_i}^{\mathrm{S}} - CT_{k_i}^{\mathrm{M}} - CT_{k_i}^{\mathrm{AS}}) - \sum_{j=1}^{M} \chi E_j. \end{aligned} \tag{72}$$

B. A3C-Based Joint Optimization Algorithm

Here, A3C is adopted to deal with the large-scale optimization problem formulated. The architecture of the A3C-based joint optimization algorithm is shown in Fig. 3. In contrast to the traditional deep reinforcement learning method, A3C can realize efficient distributed asynchronous learning. In the A3C-based joint optimization algorithm, the agent consists of a global network and multiple workers. Both the global network and the workers have the same network architecture, which is composed of two neural networks, namely the policy network (actor) with parameter θ_A and the value network (critic) with parameter θ_C. The workers learn in parallel by interacting with their environments separately to compute their new gradients and send them to the global networks, when reaching the terminal state or the maximum number of iterations. Instead of interacting with the environment directly, the global network is only responsible for updating the global network parameters with the gradient fetched from the workers and distributing the global network parameters to each worker at regular intervals.

Specifically, in each episode, the estimated state value predicted by the value network is denoted by $V[s(\vartheta); \theta_C]$. The agent executes an action $a(\vartheta)$ according to the policy $\pi[a(\vartheta) \mid s(\vartheta)]$ at the current state $s(\vartheta)$, and then the environment will change to the next state $s(\vartheta+1)$ and generate a reward $r(\vartheta)$. The state value function of A3C is represented as [43]

$$V\left(s(\vartheta); \theta_C\right) = E\left[\sum_{c=0}^{\infty} \Psi^{c} r(\vartheta + c)\right], \tag{73}$$

where Ψ is the discount factor, which denotes how future rewards affect the current state value. A3C employs a $\mathbb{K}$-step reward for updating the parameters, which can be represented as

$$R(\vartheta) = \sum_{l=0}^{\mathbb{K}-1} \Psi^{l} r(\vartheta + l) + \Psi^{\mathbb{K}} V\left(s_{\vartheta+\mathbb{K}}; \theta_C\right), \tag{74}$$

where $\mathbb{K}$ is the number of time steps required for calculating $\mathbb{K}$-step returns. Aiming for reducing the estimation variance and improving the decision-making capability of the agent, we define the advantage function $\hat{A}(\vartheta)$ as follows:

$$\hat{A}\left[s(\vartheta), a(\vartheta); \theta_A, \theta_C\right] = R(\vartheta) - V\left[s(\vartheta); \theta_C\right]. \tag{75}$$

Furthermore, the loss function of the actor is represented as

$$J_{\pi}(\theta_A) = \log \pi\left[a(\vartheta) \mid s(\vartheta); \theta_A\right] \hat{A}(\vartheta) + \Theta H\left(\pi\left[s(\vartheta); \theta_A\right]\right), \tag{76}$$

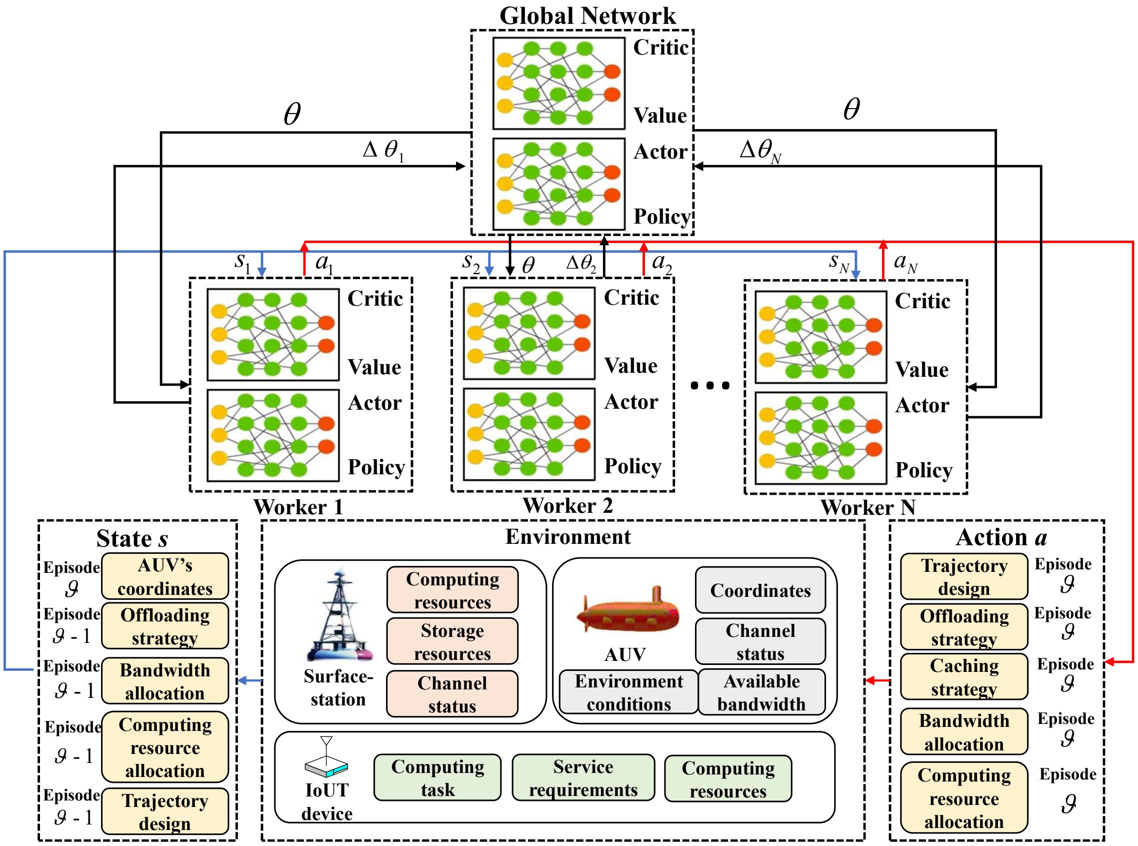


Fig. 3. The architecture of A3C-based joint optimization algorithm.

where $H\left(\pi\left[s(\vartheta);\theta_A\right]\right)$ is an entropy item introduced for encouraging exploration and for avoiding to fall into a local optimum, while Θ manages the strength of the entropy regularization. By contrast, the loss function of the critic network is denoted by

$$J_C(\theta_A)=\hat{A}(\vartheta)^2. \tag{77}$$

As the updating process, the accumulated gradient of the policy network is calculated as

$$\begin{aligned}d\theta_A \leftarrow d\theta_A &+ \nabla_{\theta_A}\log\pi\left[a(\vartheta)\mid s(\vartheta);\theta_A\right]\hat{A}(\vartheta)\\ &+\delta\nabla_{\theta_A}H\left(\pi\left[s(\vartheta);\theta_A\right]\right),\end{aligned} \tag{78}$$

while the accumulated gradient of the value network is calculated as

$$d\theta_C \leftarrow d\theta_C+\frac{\partial\hat{A}(\vartheta)^2}{\partial\theta_C}. \tag{79}$$

To train the A3C framework effectively, the RMSProp algorithm [43] is adopted, which can significantly improve the speed of gradient descent. The estimated gradient relying on the RMSProp algorithm can be formulated as

$$\Upsilon=\Lambda\Upsilon+(1-\Lambda)(\Delta\theta)^2, \tag{80}$$

where $\Delta\theta$ represents the accumulated gradients of the loss function of the policy or value networks, while Λ is the momentum. Relying on Eq. (80), we update the parameters of the policy and value networks by

$$\theta_A \leftarrow \theta_A-\Xi\frac{\Delta\theta_A}{\sqrt{\Upsilon+\epsilon}} \tag{81}$$

and

$$\theta_C \leftarrow \theta_C-\Xi\frac{\Delta\theta_C}{\sqrt{\Upsilon+\epsilon}}, \tag{82}$$

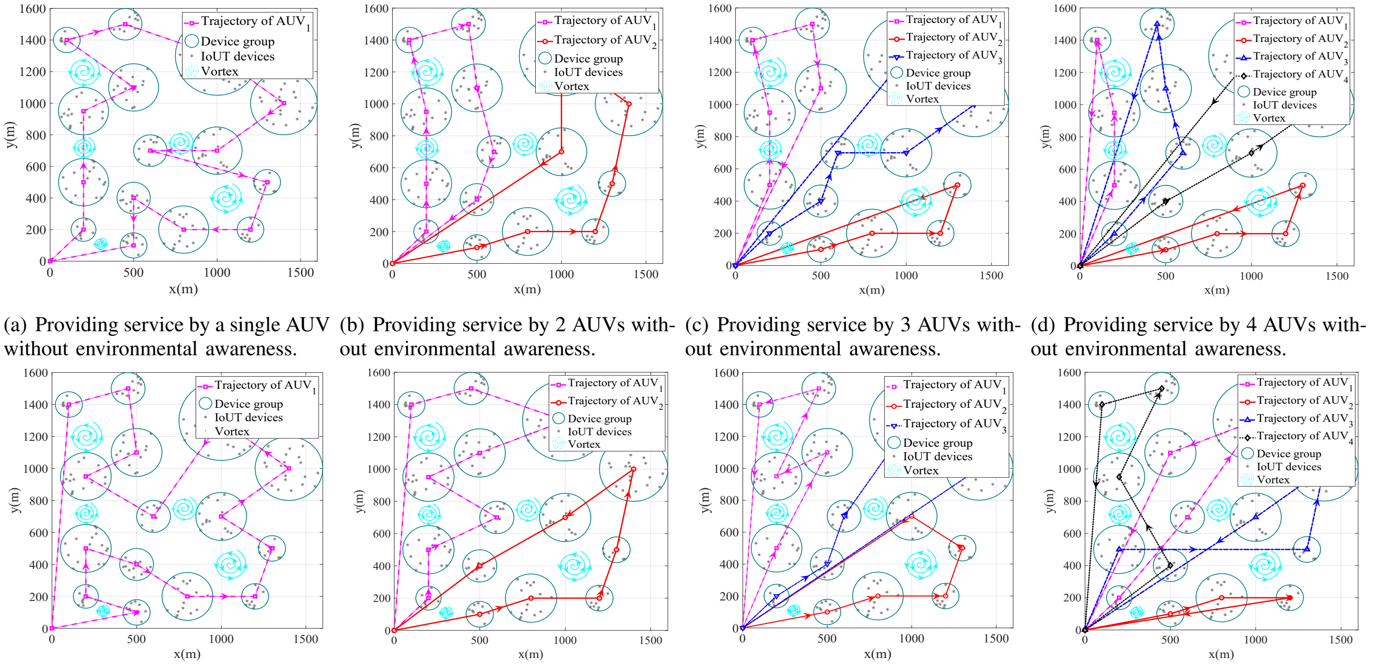
respectively, where ϵ is a tiny positive step, while Ξ is the learning rate. The procedure designed is summarized in Algorithm 1.

V. Simulation Results

In this section, we provide the experimental results for validating the superiority of our proposed scheme. Unless specified, otherwise, the number of the AUVs is set to 4, while the numbers of the DGs and IoUT devices are set to 15 and 190, respectively. The main parameters are summarized in Table II.

A. Impact of the Hostile Underwater Environment on the System

Fig. 4 shows the difference between the trajectory design with and without environmental awareness. Observe that each AUV starts from the origin, providing services to the DGs assigned, and then returns to the starting point for recharging after completing one cycle. Furthermore, as we can observe, compared to the AUVs in Fig. 4(a)-4(d), the AUVs in Fig. 4(e)-4(h) relying on environmental awareness can select the optimal trajectories without vortex, which can avoid the extra energy consumption of the vortex and yield a high profit for the MTUC framework. Although sometimes the AUV



(a) Providing service by a single AUV without environmental awareness.

(b) Providing service by 2 AUVs without environmental awareness.

(c) Providing service by 3 AUVs without environmental awareness.

(d) Providing service by 4 AUVs without environmental awareness.

(e) Providing service by a single AUV with environmental awareness.

(f) Providing service by 2 AUVs with environmental awareness.

(g) Providing service by 3 AUVs with environmental awareness.

(h) Providing service by 4 AUVs with environmental awareness.

Fig. 4. Comparison between environment-agnostic and environment-aware trajectory design.

TABLE II
VALUES OF MAIN PARAMETERS

Parameter	Value	Parameter	Value
f	30 kHz	$P_{\mathrm{tr}}^{\mathrm{D}}$	30 mW
s	0.5	$P_{\mathrm{tr}}^{\mathrm{A}}$	36 mW
w	0	B_{H}	10 kHz
H	200 m	B_{L}	10 kHz
h_0	10 m	$\mathcal{V}$	1
d_0	20 m	V_k	5 knot
Ω_0	8	C_a	0.0314
ρ_L	$1020\ \mathrm{kg/m^3}$	α	100
Z_{ki}	$\mathcal{U}[10^5, 3*10^5]$ bit	β	100
f_{ki}	$\mathcal{U}[1,4]$ GHz	ϕ	100
α_{ki}	$\mathcal{U}[1500, 2000]$ cycles/bit	ε	2 s
C_e	100 Mb	r_0	100 m
C_d	0.117	σ	3
k_s	1.5	μ	1.25×10^{-26}
Γ_s	1	ζ	0.8
Γ_b	0.0139	η	0.2
H_1	180 m	H_2	190 m
ω_{k_i}	$\mathcal{U}[10,20]$	λ_{k_i}	$\mathcal{U}[1,2]$
ϱ	1	χ	2

relying on environmental awareness selects a longer path than that without environmental awareness, the profit of the whole system still settles on the global optimum.

In Fig. 5, we show the profit comparison between environment-aware and environment-agnostic trajectory design versus the number of AUVs, corresponding to the results shown in Fig. 4. Observe that the environment-aware trajectory design outperforms its agnostic counterpart. Furthermore, as the number of the AUVs increases, the profit of the whole system increases, because the collaboration of multiple AUVs exhibits more flexibility than a single AUV. However, we will conjecture that having a higher number of AUVs does not necessarily result in a higher profit for the system.

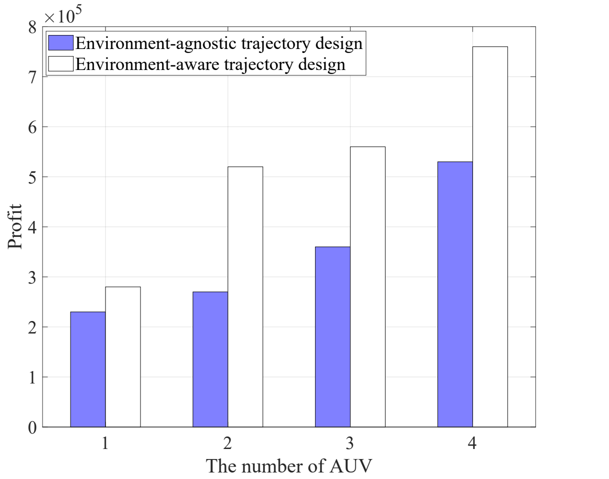


Fig. 5. Comparison of the profit between environment-aware and environment-agnostic trajectory design versus the number of AUVs.

Fig. 6 portrays the profit versus the number of AUVs serving different numbers of IoUT devices. Observe that there is always an optimal solution for the number of AUVs for serving a given number of IoUT devices. For example, for 300 IoUT devices, employing 5 AUVs to provide services achieves the highest profit. This phenomenon can provide us with a tangible philosophy for guiding the AUV deployment. Furthermore, we can observe in Fig. 6 that if all other conditions remain the same, then increasing the number of devices increases the benefit of the system. The reason for this is that when the number of devices increases, assigning the same energy consumption to the AUV's movement can support more IoUT devices, thereby obtaining higher revenue

Algorithm 1 Asynchronous advantage actor-critic Algorithm

```
Initialize the maximum counters $\mathcal{T}_{max}$, $\vartheta_{max}$, and all the parameters as shown in Table II, respectively.
Initialize the global policy network and global value network with parameters $\theta_A$ and $\theta_C$.
Initialize global shared counter as $\mathcal{T} = 0$ and thread-specific counter as $\vartheta = 1$.
Initialize the thread-specific policy network parameters $\theta'_A$ and value network parameters $\theta'_C$.
for $\mathcal{T} < \mathcal{T}_{max}$ do
    for each worker do
        Initialize the gradients of agent as $d\theta_A = 0$ and $d\theta_C = 0$.
        Synchronous parameters of each worker with global parameters $\theta'_A = \theta_A$ and $\theta'_C = \theta_C$.
        for $\vartheta \leq \vartheta_{max}$ do
            Obtain the state $s(\vartheta)$.
            Perform $a(\vartheta)$ relying on the policy $\pi(a(\vartheta) \mid s(\vartheta); \theta'_A)$.
            Obtain reward $r(\vartheta)$ and new state $s(\vartheta + 1)$.
            $\vartheta = \vartheta + 1$.
        end for
        $\hat{V} = \begin{cases} 0, & \text{for terminal state} \\ V\left(s(\vartheta), \theta'_v\right), & \text{for non-terminal state} \end{cases}$
        for $\vartheta = \vartheta_{max}$ do
            $\hat{V} = r(\vartheta) + \Psi \hat{V}$
            Obtain the accumulate gradient with respect to $\theta'_A$ by Eq. (78);
            Obtain the accumulate gradient with respect to $\theta'_C$ by Eq. (79);
        end for
        Update $\theta_A$ and $\theta_C$ according to Eq. (81) and Eq. (82).
        $\mathcal{T} = \mathcal{T} + 1$
    end for
end for
```

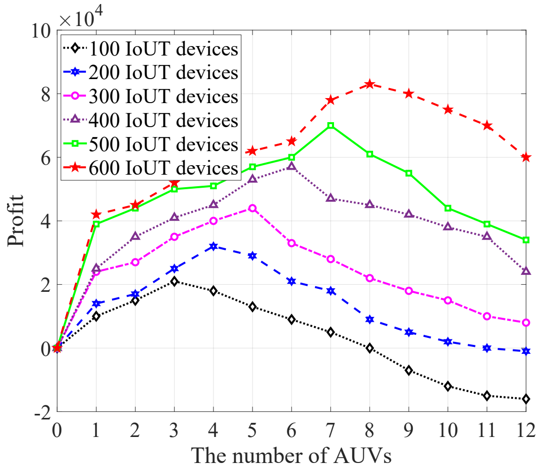


Fig. 6. The profit versus the number of AUVs serving different numbers of IoUT devices.

and further improving the profit.

B. Impact of Different Resource Allocation Schemes on the System's Profit

To characterize the impact of the offloading scheme on the system's profit, in Fig. 7, we show the profit of different task offloading schemes. The full offloading scheme represents that all IoUT devices select to offload their tasks to the surface-station for processing, while the non-offloading scheme means that all IoUT devices address their tasks locally. Moreover, the random offloading scheme represents that each device randomly chooses whether to offload their computing tasks to the surface-station, while the partial offloading scheme means that we designate a proportion of tasks to offload to the surface-station and leave some tasks to be processed locally. Although the non-offloading scheme can satisfy the requirements of the devices, the cost that it has to pay is substantially higher than that of offloading the tasks to the MTUC framework for processing due to the energy dissipation of IoUT devices that are difficult to recharge. The IoUT devices are also harder to recharge than the AUVs that can be continuously recharged. Similarly, the surface-stations may be more readily recharged. Furthermore, when we choose to offload some tasks to the MTUC, the profit gleaned increases significantly. Explicitly, the proposed scheme consistently outperforms the other offloading schemes because it can search for an optimal offloading strategy to maximize the profit with limited resources.

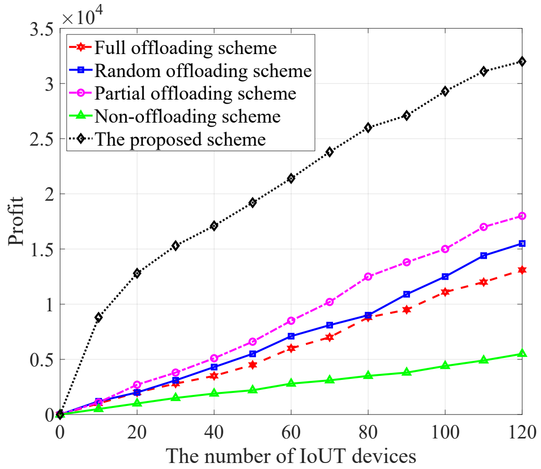


Fig. 7. The profit of different task offloading schemes versus the number of IoUT devices.

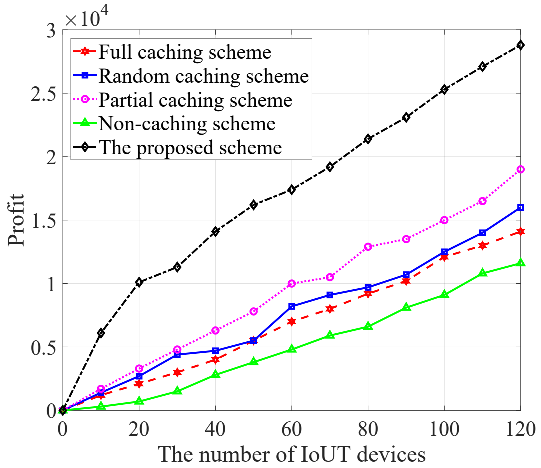


Fig. 8. The profit of different caching schemes versus the number of IoUT devices.

Fig. 8 shows the benefit of task caching. The full caching scheme represents that all tasks are cached on the surface-station, while the non-caching scheme represents that none of the tasks is cached. Moreover, the random caching scheme

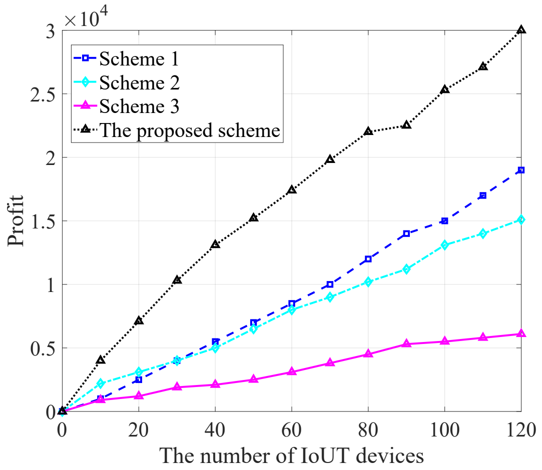


Fig. 9. The profit of different computing and communication allocation schemes versus the number of IoUT devices (Scheme 1 is with optimal bandwidth resource allocation and average computing resource allocation, scheme 2 is with average bandwidth resource allocation and optimal computing resource allocation, and scheme 3 is with average bandwidth resource allocation, and average computing resource allocation.).

means that we randomly choose some of the tasks to cache by the surface-station, while the partial caching scheme represents that we designate a certain proportion of tasks to cache by the surface-station. Firstly, we can observe that task caching significantly improves the system's profit, when there are repeated task computing requests. This is because task caching avoids repeated communication and computation, consequently reducing the processing latency and the energy consumption. Furthermore, upon increasing the number of IoUT devices, the profit increases dramatically. The reason for this is that the more IoUT devices we have, the higher the probability of repeated computing requests. Moreover, we can see that the proposed scheme outperforms other schemes without optimization. This is because the scheme advocated comprehensively considers both the popularity of the tasks and the storage capacity of the surface-station for formulating an optimal caching strategy so as to attain the highest system profit.

To investigate the impact of computing and communication resource configuration on the system's profit, we compare the profit of different schemes in Fig. 9. Observe that the scheme relying on the average bandwidth resource allocation and average computing resource allocation is the worst, because it ignores the differences in tasks and the resource states between different IoUT devices. By contrast, optimizing both the bandwidth resource allocation and computing resource allocation dramatically increases the system's profit. Furthermore, we can observe that the proposed scheme is much better than all other schemes that optimize a single resource individually, which indicates that the configuration of both types of resources significantly improves the system's profit.

C. The Performance Analysis of The A3C Algorithm

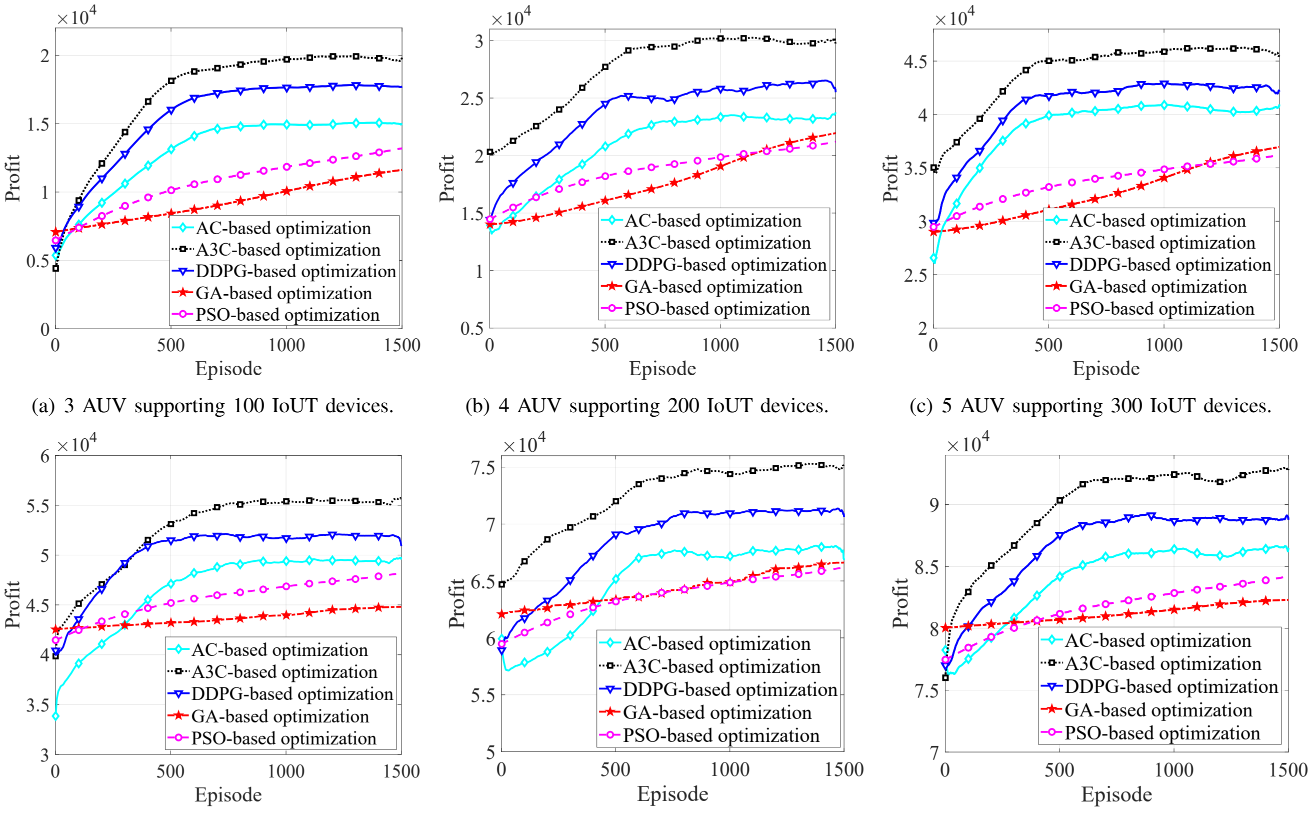
Conventional methods falter in tackling $\mathcal{P}1$, because it is typically NP-hard and has a high dimensionality. In Fig. 10, we compare the performance of state-of-the-art algorithms in tackling this problem in multiple scenarios, including the popular genetic algorithm (GA) [44], particle swarm optimization (PSO) algorithm [45], actor-critic (AC) algorithm [46], deep deterministic policy gradient (DDPG) algorithm [47], and our A3C algorithm. Observe that the heuristic algorithms, i.e., GA and PSO-based optimization strategies have poor convergence performance. By contrast, the deep reinforcement learning algorithms, i.e., AC, DDPG, and A3C-based optimization strategies, perform better. The reason is that the deep reinforcement learning algorithms are more suitable for solving high-dimensional problems as a direct of the neural networks' powerful function fitting capability. Furthermore, the A3C algorithm is better than DDPG and AC-based optimization, because it can find better solutions within the same number of iterations as a benefit of its distributed parallel operating paradigm.

The setting of the hyperparameters in deep reinforcement learning is of pivotal importance, since it may seriously affect the performance of the algorithms. As a significant hyperparameter in A3C, the learning rate dramatically affects the convergence rate, but fails to obtain a theoretical optimal value. If the learning rate is set too low, it will slow down the convergence of the algorithm and increase the training time. By the contrast, if the learning rate is excessive, the parameters may swing back and forth on both sides of the optimal value, failing to converge. In Fig. 11, we investigate the impact of the learning rate on the convergence performance of A3C. As we can observe, the algorithm having an adaptive learning rate is superior to others, which will gradually adjust the learning rate according to the training process.

VI. Conclusions

To satisfy the stringent requirements of IoUT applications, we proposed an MTUC framework by judiciously allocating the computing, communication, and storage resources of both the surface-station, as well as of the AUVs, and of the IoUT devices. Furthermore, under this framework, we conceived a system-level optimization problem for the sake of maximizing the profit of the MTUC framework relying on jointly optimizing the environment-aware trajectory design of the AUVs, computation offloading, data caching, communication, and computing resource allocation. Since the problem formulated is NP-hard and of high dimensionality, we transformed it into an MDP and further employed the A3C algorithm to solve it. Finally, we conducted a range of experiments to validate the efficiency of the proposed scheme.

In the near future, we plan to study the impact of underwater environments on IoUT applications. For instance, hostile underwater environments may cause a high probability of device failure, reducing the success probability of IoUT applications. Hence it is beneficial to explore how to guarantee the success probability of IoUT applications. Moreover, the conception of having low communication overhead in the face of limited UAC communication resources is also worth pursuing, for example by using federated learning.



(a) 3 AUV supporting 100 IoUT devices.

(b) 4 AUV supporting 200 IoUT devices.

(c) 5 AUV supporting 300 IoUT devices.

(d) 6 AUV supporting 400 IoUT devices.

(e) 7 AUV supporting 500 IoUT devices.

(f) 8 AUV supporting 600 IoUT devices.

Fig. 10. The profit of different algorithms.

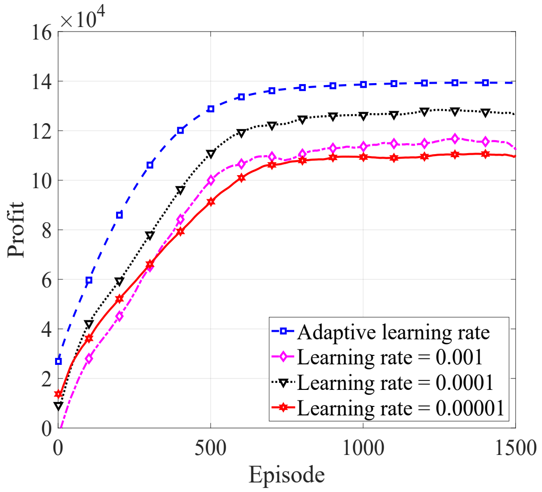


Fig. 11. Impact of the learning rate on the convergence performance of A3C.

Appendix A
Proof of Proposition 1

Let us consider a particular case, where the values of computation offloading strategy $\boldsymbol{O}$, caching strategy $\boldsymbol{H}$, bandwidth allocation $\boldsymbol{R}$ and computing resource allocation $\boldsymbol{F}$ are given, which satisfy the constraints demonstrated in problem $\mathcal{P}1$. Consequently, we can obtain a sub-problem of $\mathcal{P}1$ as

$$\mathcal{P}2: \quad \max_{\boldsymbol{Y}} \qquad C - \sum_{j=1}^{M} \chi \hat{E}_j \tag{A.1a}$$

$$s.t. \quad \sum_{j=1}^{M} \sum_{\xi=1}^{S_j} Y_{j_k}[\xi] = 1, \forall j \in \boldsymbol{M}, \forall k \in \boldsymbol{K}, \tag{A.1b}$$

$$\boldsymbol{P}_j^{\mathrm{A}}[S_j+1] = \boldsymbol{P}_j^{\mathrm{A}}[0], \forall j \in \boldsymbol{M}, \tag{A.1c}$$

$$\sum_{\xi=1}^{S_j} \sum_{k=1}^{K} Y_{j_k}[\xi] = S_j, \forall j \in \boldsymbol{M}, \tag{A.1d}$$

$$\sum_{j=1}^{M} S_j = K, \forall j \in \boldsymbol{M}, \tag{A.1e}$$

$$T_{\max}^{\mathrm{AT}} - T_{\min}^{\mathrm{AT}} \leq \varepsilon. \tag{A.1f}$$

where C is a constant associated with the first term of Eq. (69a), while $\hat{E}_j$ is represented as

$$\hat{E}_j = \sum_{k=1}^{K} \sum_{\xi=1}^{S_j} Y_{j_k}[\xi] \hat{A}_k P_j^{\mathrm{H}}[\xi] + \sum_{\xi=0}^{S_j} \frac{d_j[\xi]}{V_k} P_j^{\mathrm{F}}[\xi], \tag{A.2}$$

where $\hat{A}_k$ is a constant as a result of Eq. (31) after the computation offloading strategy $\boldsymbol{O}$, caching strategy $\boldsymbol{H}$, bandwidth allocation $\boldsymbol{R}$ and computing resource allocation $\boldsymbol{F}$ are given. In fact, problem $\mathcal{P}2$ can be equivalent to

$$\mathcal{P}3: \quad \min_{\boldsymbol{Y}} \sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{\xi=1}^{S_j} Y_{j_k}[\xi] \hat{A}_k P_j^{\mathrm{H}}[\xi] + \sum_{\xi=0}^{S_j} \frac{d_j[\xi]}{V_k} P_j^{\mathrm{F}}[\xi] \tag{A.3a}$$

$$s.t. \quad \text{Eq. (A.1b)} \sim \text{(A.1f)}. \tag{A.3b}$$

$\mathcal{P}3$ can be seen as a variant of the multiple traveling salesman problem (MTSP), which is essentially a generalization of the well-known traveling salesman problem (TSP). Furthermore, since TSP has already been proven to be NP-hard and can be reduced to the MTSP, MTSP is an NP-hard problem [48]. Consequently, $\mathcal{P}3$ is NP-hard. Furthermore, due to $\mathcal{P}3$ is a sub-problem of $\mathcal{P}1$, we can determine that $\mathcal{P}1$ is also an NP-hard problem. Therefore, if $P \neq NP$, there is no algorithm can solve $\mathcal{P}1$ in polynomial time. Thus the proof of *Proposition 1* is completed.

REFERENCES

[1] M. Jahanbakht, W. Xiang, L. Hanzo, and M. Rahimi Azghadi, "Internet of underwater things and big marine data analytics—A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 904–956, 2021.

[2] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, "Underwater Internet of things in smart ocean: System architecture and open issues," *IEEE Trans. Industr. Inform.*, vol. 16, no. 7, pp. 4297–4307, 2020.

[3] R. Zhang, X. Ma, D. Wang, F. Yuan, and E. Cheng, "Adaptive coding and bit-power loading algorithms for underwater acoustic transmissions," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5798–5811, 2021.

[4] H. Ramezani and G. Leus, "Localization packet scheduling for underwater acoustic sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1345–1356, 2015.

[5] Y. Yang, Y. Xiao, and T. Li, "A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 815–841, 2021.

[6] X. Wei, H. Guo, X. Wang, X. Wang, and M. Qiu, "Reliable data collection techniques in underwater wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 404–431, 2022.

[7] S. Yoon, A. K. Azad, H. Oh, and S. Kim, "AURP: An AUV-aided underwater routing protocol for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 2, pp. 1827–1845, 2012.

[8] G. Han, X. Long, C. Zhu, M. Guizani, Y. Bi, and W. Zhang, "An AUV location prediction-based data collection scheme for underwater wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6037–6049, 2019.

[9] R. Duan, J. Du, C. Jiang, and Y. Ren, "Value-based hierarchical information collection for AUV-enabled Internet of underwater things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9870–9883, 2020.

[10] P. Gjanci, C. Petrioli, S. Basagni, C. A. Phillips, L. Bölöni, and D. Turgut, "Path finding for maximum value of information in multimodal underwater wireless sensor networks," *IEEE Trans. Mob. Comput.*, vol. 17, no. 2, pp. 404–418, 2017.

[11] K. Wang, W. Chen, J. Li, Y. Yang, and L. Hanzo, "Joint task offloading and caching for massive MIMO-aided multi-tier computing networks," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1820–1833, 2022.

[12] G. Han, S. Shen, H. Song, T. Yang, and W. Zhang, "A stratification-based data collection scheme in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 671–10 682, 2018.

[13] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in Internet of underwater things networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1775–1789, 2022.

[14] M. Huang, K. Zhang, Z. Zeng, T. Wang, and Y. Liu, "An AUV-assisted data gathering scheme based on clustering and matrix completion for smart ocean," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9904–9918, 2020.

[15] J. Yan, X. Yang, X. Luo, and C. Chen, "Energy-efficient data collection over AUV-assisted underwater acoustic sensor network," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3519–3530, 2018.

[16] Z. Fang, J. Wang, C. Jiang, Q. Zhang, and Y. Ren, "AoI-inspired collaborative information collection for AUV-assisted Internet of underwater things," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14 559–14 571, 2021.

[17] Z. Liu, X. Meng, Y. Liu, Y. Yang, and Y. Wang, "AUV-aided hybrid data collection scheme based on value of information for Internet of underwater things," *IEEE Internet Things J.*, pp. 1–1, 2021.

[18] A. B. Labao, M. A. M. Martija, and P. C. Naval, "A3C-GS: Adaptive moment gradient sharing with locks for asynchronous actor–critic agents," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1162–1176, 2021.

[19] Y.-S. Chen and Y.-W. Lin, "Mobicast routing protocol for underwater sensor networks," *IEEE Sens. J.*, vol. 13, no. 2, pp. 737–749, 2012.

[20] M. T. R. Khan, S. H. Ahmed, and D. Kim, "AUV-aided energy-efficient clustering in the Internet of underwater things," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 4, pp. 1132–1141, 2019.

[21] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 899–911, 2012.

[22] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1472–1483, 2012.

[23] J. Faigl and G. A. Hollinger, "Autonomous data collection using a self-organizing map," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1703–1715, 2018.

[24] X. Hou, J. Wang, Z. Fang, X. Zhang, S. Song, X. Zhang, and Y. Ren, "Machine-learning-aided mission-critical Internet of underwater things," *IEEE Netw.*, vol. 35, no. 4, pp. 160–166, 2021.

[25] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing," *IEEE Trans. Wireless. Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug 2017.

[26] W. Fan, S. Li, J. Liu, Y. Su, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for accuracy-aware machine-learning-based IIoT applications," *IEEE Internet Things J.*, vol. Early access, pp. 1–1, 2022.

[27] P. Abichandani, S. Torabi, S. Basu, and H. Benson, "Mixed integer nonlinear programming framework for fixed path coordination of multiple underwater vehicles under acoustic communication constraints," *IEEE J. Ocean. Eng.*, vol. 40, no. 4, pp. 864–873, 2015.

[28] F. B. Jensen, W. A. Kuperman, M. B. Porter, H. Schmidt, and A. Tolstoy, *Computational ocean acoustics*. Springer, 2011, vol. 2011.

[29] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun, "AUV-aided energy-efficient data collection in underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10 010–10 022, 2020.

[30] P. Abichandani, S. Torabi, S. Basu, and H. Benson, "Mixed integer nonlinear programming framework for fixed path coordination of multiple underwater vehicles under acoustic communication constraints," *IEEE J. Ocean. Eng.*, vol. 40, no. 4, pp. 864–873, 2015.

[31] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11 365–11 373, 2018.

[32] X. Yang, Z. Fei, J. Zheng, N. Zhang, and A. Anpalagan, "Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 018–11 030, 2019.

[33] W. Wen, Y. Cui, T. Q. S. Quek, F.-C. Zheng, and S. Jin, "Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7879–7894, 2020.

[34] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel. Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, 2015.

[35] Z. Zeng, K. Sammut, A. Lammas, F. He, and Y. Tang, "Efficient path re-planning for AUVs operating in spatiotemporal currents," *J. Intell. Robot. Syst.*, vol. 79, no. 1, pp. 135–153, 2015.

[36] L. Shi, R. Zheng, S. Zhang, and M. Liu, "Cooperative estimation to reconstruct the parametric flow field using multiple AUVs," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[37] S. Shuai and M. H. Kasbaoui, "Accelerated decay of a Lamb–Oseen vortex tube laden with inertial particles in Eulerian–Lagrangian simulations," *J. Fluid Mech.*, vol. 936, 2022.

[38] M. M. Bhatti, M. Marin, A. Zeeshan, and S. I. Abdelsalam, "Recent trends in computational fluid dynamics," *Front. Phys.*, vol. 8, p. 593111, 2020.

[39] K. Wang, Y. Zhou, J. Li, L. Shi, W. Chen, and L. Hanzo, "Energy-efficient task offloading in massive MIMO-aided multi-pair fog-computing networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2123–2137, 2021.

[40] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling," *IEEE Trans. Commun.*, vol. 65, no. 18, pp. 3571–3584, 2017.

[41] S. Zheng, Z. Ren, X. Hou, and H. Zhang, "Optimal communication-computing-caching for maximizing revenue in UAV-aided mobile edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Decemeber, 2020.

[42] J. Du, W. Cheng, G. Lu, H. Cao, X. Chu, Z. Zhang, and J. Wang, “Resource pricing and allocation in MEC enabled blockchain systems: An A3C deep reinforcement learning approach,” *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 33–44, 2022.

[43] J. Wang, L. Kaiyang, and J. Pan, “Online UAV-mounted edge server dispatching for mobile-to-mobile edge computing,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1375–1386, Feb 2020.

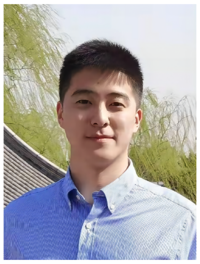
[44] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, “Automatically designing CNN architectures using the genetic algorithm for image classification,” *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3840–3854, 2020.

[45] X. Ji, Y. Zhang, D. Gong, and X. Sun, “Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 794–808, 2021.

[46] X. Wang, Q. Wang, and C. Sun, “Prescribed performance fault-tolerant control for uncertain nonlinear MIMO system using actor-critic learning structure,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2021.

[47] Z. Gu, C. She, W. Hardjawana, S. Lumb, D. McKechnie, T. Essery, and B. Vucetic, “Knowledge-assisted deep reinforcement learning in 5G scheduler design: From theoretical framework to implementation,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2014–2028, 2021.

[48] O. Cheikhrouhou and I. Khoufi, “A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy,” *Comput. Sci. Rev.*, vol. 40, p. 100369, 2021.



Xiangwang Hou (Student Member, IEEE) is currently pursuing his Ph.D. degree in Electronics and Communication Engineering at Tsinghua University, Beijing, China. And he received the B.E. degree in Electronic Information Engineering from Shandong University of Technology, Shandong, China in 2017 and the M.E. degree in Information and Communication Engineering from Xidian University, Xi’an, China in 2020. His research interests include UAV/AUV networks, federated learning and wireless AI.

Jingjing Wang (Senior Member, IEEE) received his B.S. degree in Electronic Information Engineering from Dalian University of Technology, Liaoning, China in 2014 and the Ph.D. degree in Information and Communication Engineering from Tsinghua University, Beijing, China in 2019, both with the highest honors. From 2017 to 2018, he visited the Next Generation Wireless Group chaired by Prof. Lajos Hanzo, University of Southampton, UK. Dr. Wang is currently an associate professor at School of Cyber Science and Technology, Beihang University. His research interests include AI enhanced next-generation wireless networks, UAV swarm intelligence and confrontation. He has published over 100 IEEE Journal/Conference papers. Dr. Wang was a recipient of the Best Journal Paper Award of IEEE ComSoc Technical Committee on Green Communications & Computing in 2018, the Best Paper Award of IEEE ICC and IWCMC in 2019.

Tong Bai (Member, IEEE) received the B.Sc. degree in telecommunications from Northwestern Polytechnical University, Xi’an, China, in 2013, and the M.Sc. and Ph.D. degrees in communications and signal processing from the University of Southampton, Southampton, U.K., in 2014 and 2019, respectively. From 2019 to 2020, he was a Postdoctoral Researcher with Queen Mary University of London, London, U.K. Since 2020, he has been with Beihang University (BUAA) as an Assistant Professor. His research interests include edge intelligence and wireless communications.

Yansha Deng (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Queen Mary University of London, U.K., in 2015. From 2015 to 2017, she was a Post-Doctoral Research Fellow with King’s College London, U.K, where she is currently a Senior Lecturer (an Associate Professor) with the Department of Engineering. Her research interests include molecular communication and machine learning for 5G/6G wireless networks. She was a recipient of the Best Paper Awards from ICC 2016 and GLOBECOM 2017 as the first author and IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region 2021. She also received the Exemplary Reviewers of the IEEE Transactions on communications in 2016 and 2017 and IEEE Transactions on wireless communications in 2018. She has served as a TPC Member for many IEEE conferences, such as IEEE GLOBECOM and ICC. She is currently an Associate Editor of the IEEE Transactions on communications and IEEE Transactions on molecular, biological and multi-scale communications, a Senior Editor of the IEEE communication letters, and the Vertical Area Editor of IEEE Internet of things magazine.

Yong Ren (Senior Member, IEEE) received his B.S, M.S and Ph.D. degrees in electronic engineering from Harbin Institute of Technology, China, in 1984, 1987, and 1994, respectively. He worked as a post doctor at Department of Electrical Engineering, Tsinghua University, China from 1995 to 1997. Now he is a full professor of Department of Electronic Engineering and serves as the director of the Complexity Engineered Systems Lab in Tsinghua University. Moreover, he is also a guest professor of the Network and Communication Research Center in Peng Cheng Laboratory. He has authored or co-authored more than 400 technical papers in the area of computer network and mobile telecommunication networks. He has served as a reviewer of more than 40 international journals or conferences. His current research interests include marine information network, swarm intelligence and wireless AI.

Lajos Hanzo (Life Fellow, IEEE) (http://www-mobile.ecs.soton.ac.uk, https://en.wikipedia.org/wiki/Lajos_Hanzo) received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 2000+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. Over 40 of them are Professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is also a Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP. He is the recipient of the 2022 Eric Sumner Field Award.