#### Check for updates

#### OPEN ACCESS

EDITED BY Syed Agha Hassnain Mohsan, Zhejiang University, China

REVIEWED BY Gulista Khan, Teerthanker Mahaveer University, India Suni J, Annai Velankanni College, India Sajid Khan, Prince Sattam Bin Abdulaziz University, Saudi Arabia

\*CORRESPONDENCE Keyu Chen Mchenkeyu@xmu.edu.cn

RECEIVED 21 February 2025 ACCEPTED 21 April 2025 PUBLISHED 22 May 2025

#### CITATION

Cao W, Chen K and Cheng E (2025) Joint optimization of AoI and energy for AUVassisted data collection in underwater acoustic sensor networks. *Front. Mar. Sci.* 12:1580751. doi: 10.3389/fmars.2025.1580751

#### COPYRIGHT

© 2025 Cao, Chen and Cheng. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Joint optimization of AoI and energy for AUV-assisted data collection in underwater acoustic sensor networks

## Weinan Cao, Keyu Chen\* and En Cheng

Key Laboratory of Underwater Acoustic Communication and Marine Information Technology, Ministry of Education, Xiamen University, Xiamen, China

Underwater data collection is a key component of ocean observation systems, providing reliable data provenance through sensor networks deployed across diverse marine environments. Timely and energy-efficient data collection mechanisms are critical for ensuring the continuity and integrity of the collected data. For the sake of jointly optimizing the age of information (AoI) and energy consumption, we propose a multi-autonomous underwater vehicle (AUV)-assisted underwater data collection scheme in this paper. The optimization problem is formulated as a mixed integer linear programming (MILP) problem, which is non-deterministic polynomial (NP)-hard. In order to address the problem efficiently, we propose a two-stage method. First, considering the balance of energy consumption and the limitation of sensor communication ranges, we design a cluster-based AUV scheduling data transmission protocol to find a series of hovering points at which the AUVs hover to receive data and schedule data transmissions between the AUVs and sensors. Based on that, a combined heuristic algorithm is proposed to plan AUVs' trajectories so that the data collection task can be completed properly with a low cost. Simulation results show that the proposed algorithm outperforms three benchmark algorithms in terms of average AoI and collection utility. Specifically, the freshness of information and the collection utility improved by 10.23% and 33.36% on average, respectively. Additionally, the energy consumption of sensor nodes is more balanced, and the network lifetime is extended.

#### KEYWORDS

underwater acoustic sensor networks (UASNs), autonomous underwater vehicle (AUV), data collection, age of information (AoI), collection utility

# **1** Introduction

The ocean is regarded as the second living space for human beings, which covers approximately 70% of the Earth's surface (National Research Council, Division on Earth, Life Studies, Ocean Studies Board andCommittee on Exploration of the Seas, 2003). Ocean observation and exploration have emerged all over the world. Underwater acoustic sensor

networks (UASNs), being a powerful means of ocean observation and exploration, play an increasingly important role in smart oceans. With the development of the internet of underwater things (IoUT), the number of underwater equipment continues to grow rapidly. Massive heterogeneous data are generated underwater such that UASNs have entered the era of big data (Qiu et al., 2019).

Marine information collection is the basic function of UASNs, which supports the data requirements for marine science. However, the inherent characteristics of UASNs bring difficulties to underwater data collection, such as unreliable long-distance communication links, difficulties in replenishing the energy for underwater sensor nodes, and high bit error rates and delays in data transmission (Wang et al., 2019, 2020; Han et al., 2019). Owing to the convenience of power supply, mobility, and stronger computing and communication capabilities, autonomous underwater vehicles (AUVs) have been widely used in UASNs to improve performance (Yang et al., 2021). In general, existing underwater data collection schemes could be divided into two types, namely, multi-hop routing schemes and AUV-assisted schemes (de Souza et al., 2016; Gjanci et al., 2017). On the one hand, the multi-hop routing schemes relay sensed data to the sink node, causing the hot area problem and additional collection delay (Guan et al., 2019; Hwang and Kim, 2008; Li et al., 2016). Most of the existing routing protocols are applied in static networks, without considering the mobility of nodes (Khan et al., 2024, 2023). On the other hand, AUV-assisted underwater data collection schemes that focus on energy efficiency (Zhuo et al., 2020), network throughput (Luo et al., 2022), and multi-AUV collaboration (Han et al., 2021) have attracted increasing attention (Su et al., 2019).

However, few studies have considered optimizing information freshness, i.e., age of information (AoI) (Talak et al., 2018), which is defined as the time interval between when information is sampled and when it is sent to the data center (Kaul et al., 2012), in AUV-assisted underwater data collection schemes. Admittedly, the freshness of the sensed information is a crucial metric that cannot be ignored in delaysensitive applications. In terrestrial wireless sensor networks (TWSNs), a considerable amount of research about AoI has been studied. In Liu et al. (2020a), an unmanned aerial vehicle (UAV)-aided data collection framework is applied in TWSN to minimize the average AoI and max-AoI. Abd-Elmagid et al. noted that one UAV serves as a mobile relay to optimize the peak AoI from source to destination (Abd-Elmagid and Dhillon, 2018). Zhang et al. (2020) discussed the trade-off between sensing and communication through directly utilizing UAV to sense information in the target area. Unfortunately, the AoI-oriented optimization algorithms in TWSN data collection systems cannot be directly applied to UASNs due to the obvious differences between UASNs and TWSNs.

Motivated by the above-mentioned issues, this paper proposes a multi-AUV-assisted underwater data collection scheme, in which the joint optimization of AoI and energy efficiency is considered. The main contributions of this paper can be summarized as follows:

 We propose a multi-AUV-assisted data collection scheme (MADCS) for UASNs. Unlike existing solutions, MADCS jointly optimizes the average AoI and energy consumption via determining the AUV hovering positions, the association between the sensor nodes and AUVs, the data transmission protocol, and AUVs' trajectory.

- 2. The optimization problem is formulated as a mixed integer linear programming (MILP) problem, which is nondeterministic polynomial (NP)-hard and difficult to solve. To reduce the computational complexity, we solve this problem in two steps. First, we propose a cluster-based AUV scheduling data transmission protocol, in which a canopy-Kmeans (CK) clustering algorithm is applied to establish a cluster-based network topology and an AUV scheduling data transmission protocol is designed to reduce the energy consumption and AoI of sensors. Second, a path planning algorithm named PGS is used to plan traversal trajectories for AUVs, improving the information freshness and collection utility.
- 3. Sufficient simulations are conducted to evaluate the performance of our proposed scheme, MADCS, compared to the three benchmark algorithms. The simulation results demonstrate the advantages of MADCS, which outperforms the three compared algorithms in terms of average AoI and collection utility. In addition, MADCS balances the energy consumption of sensor nodes and thus extends the network lifetime of UASNs.

The rest of this paper is organized as follows. Section 2 presents related work on AUV-assisted data collection schemes in USANs with a focus on energy efficiency and AoI optimization. Section 3 provides the system model, including the underwater acoustic communication model, the proposed AUV-assisted data collection network, and the derivation of the problem. In Section 4, we introduce the details of the proposed MADCS. Simulation results and analysis are presented in Section 5. Finally, conclusions are drawn in Section 6.

# 2 Related studies and motivation

Collecting data and transmitting it to the sink node in a timely manner while minimizing the network energy consumption is a critical issue for UASNs. This section reviews previous works on data collection in UASNs.

Traditional underwater data collection schemes are mainly based on multi-hop routing. Low-energy adaptive clustering hierarchy (LEACH) (Heinzelman et al., 2002) combines the energy-efficient cluster-based routing and medium access control together with data aggregation to achieve good performance in terms of system lifetime, latency, and application-perceived quality. LEACH is extended to UASNs in Zhao et al. (2021), where mobile sink LEACH (MS-LEACH) and mobile sink LEACH based on energy and density (MS-LEACH-ED) are proposed to further save energy. In Guan et al. (2019), a distance vector-based opportunistic routing (DVOR) scheme is proposed to establish distance vectors for sensors and develop opportunity routes to forward packets. The authors designed an improved energy optimization clustering algorithm (EOCA) (Yu et al., 2020) for multi-hop acoustic cooperative sensor networks to prolong the network lifetime and maintain good communication performance. A novel chaotic search-and-rescue optimization-based multi-hop data transmission (CSRO-MHDT) protocol for UASNs is introduced in Anuradha et al. (2022). Residual energy, distance, and node degree are taken into account in the fitness function. The number of packets received is improved by the CSRO-MHDT protocol. The authors in ur Rahman et al. (2023) and Rahman et al. (2023) propose a new multi-hop energy-efficient and low-collision MAC protocol using the Q-learning technique for UASNs. By utilizing the Q-learning approach, the protocol is able to make informed decisions in selecting the optimal communication path while minimizing energy consumption and reducing the occurrence of collisions.

Recently, a variety of data collection schemes based on AUVs have been studied. AEEDCO and AEEDCOA are developed in Zhuo et al. (2020), and they solved the energy consumption utility maximization problem in four steps based on a single AUV. Luo et al. (2022) introduced an optical-acoustic hybrid communication network and then maximized network throughput by capturing channel variations and AUV mobility. In order to reduce the task load on a single AUV, a multi-AUV collaborative data collection algorithm is proposed in Han et al. (2021). The algorithm includes multi-AUV task allocation and Q-learning-based path planning that reduces the delay of data collection and leverages the energy consumption of a network. Considering the three-dimensional (3D) UASNs, a stratification-based data collection scheme was proposed in Han et al. (2018). In the upper layer, a forward set-based multihop forwarding algorithm is used while a neighbor density clustering-based AUV data-gathering algorithm is applied. Omeke et al. (2021) proposed a distance- and energy-constrained k-means clustering scheme for cluster head selection. The members in the cluster transmit their data to the cluster head, and then a single AUV visits all the clusters to collect data. Hao et al. (2023) proposed a novel AUV-based energy-efficient data collection scheme, in which a new cluster head selection method and a transmission strategy suitable for large-scale AUV-assisted UASNs are utilized to improve energy utility and efficiency. Considering the AUV's limited energy, the dynamic data upload demand of sensor nodes, and the diminishing value of information during data transmission, Xu et al. (2024) formulated the AUV path planning as a multi-objective optimization problem and employed the deep deterministic policy gradient algorithm to address it.

AoI optimization in underwater data collection is emerging. In Khan et al. (2019), end-to-end data freshness constraint is considered in the design of the AUV traversal path to improve the overall data freshness. With the purpose of minimizing the peak AoI, a multi-AUV-assisted heterogeneous underwater information collection scheme is proposed in Fang et al. (2021), and the limited service M/G/1 vacation queueing model is utilized to model the process of information exchange, where the optimal upper limit of the number of AUVs served in the queueing system and the steadystate distribution of the queue length is derived. An underwater linear network is considered in Al-Habob et al. (2021), where an AUV gathers data from a set of underwater devices. In order to reduce AoI, a deep reinforcement learning-based technique is developed to find the optimal locations of data gathering points as well as the hovering time. To improve the feasibility and practicality of AUV-assisted underwater data collection, the work in Li et al. (2023) used graph attention network to embed the information of ocean currents, time window, and sensor locations into the directed maneuver time-cost graph and then applied proximal policy optimization algorithm to select cluster head sensors and generate AUV routes. Jiang et al. (2024) proposed a target uncertainty map-assisted data collection scheme for AUV swarms based on the multiagent proximal policy optimization algorithm. By leveraging current and past search and collection results, the AUV swarms are guided to areas with higher probabilities of containing sensor nodes. Additionally, AoI is introduced to enable a comprehensive exploration of unknown environments.

As mentioned above, most previous studies have focused on the separate optimization of energy or AoI and rarely considered the joint optimization of them. In this paper, we propose a multi-AUVassisted underwater data collection scheme.

# 3 System model

In this article, we consider a multi-AUV-assisted data collection scenario in 3D heterogeneous UASNs, which consists of a data center  $s_0$  located on the sea surface, N sensor nodes  $S = \{s_1, s_2, \dots, s_N\}$  deployed in the area of interest, and K AUVs denoted by  $\mathcal{V} = \{v_1, v_2, \dots, v_K\}$ , as shown in Figure 1. The location of the sensor  $s_n$  can be represented by  $\mathbf{q}_n^s = (x_n^s, y_n^s, z_n^s)$ . We assume that the locations of the sensor nodes can be obtained during the initialization phase of the network. In the network, the sensor nodes are responsible for collecting environmental data. In order to avoid the drawbacks caused by multi-hop and longdistance transmission, such as unbalanced energy consumption and unreliable communication links, the AUVs are dispatched to move closely to visit sensor nodes for data collection. AUVs depart from the data center and then return back to offload the collected data after traversing all the sensor nodes.

To reduce the path length of the AUV, sensor nodes are clustered into groups. Traditionally, one sensor node is selected as the cluster head for each group to receive data from cluster members and forward the data to the AUV. However, this approach leads to the hotspot problem, i.e., the cluster head consumes energy at a faster rate. Though cluster head rotation methods can be applied to balance the energy consumption of sensor nodes, the energy consumed by secondary forwarding from cluster head to the AUV is unnecessary. In this paper, we choose a virtual cluster head for each cluster, which is a hovering point (HP) for the AUV reaches the HP, it acts as the cluster head and establishes acoustic communication links with sensor nodes in the cluster. In this way, the sensor nodes directly send data to the AUV, avoiding secondary forwarding. Suppose *N* sensor nodes are divided



into *M* clusters. Meanwhile, there are *M* HPs  $\mathcal{H} = {\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M}$ . It should be noted that the value of *M* and the location of the HPs are not pre-known and are optimized by the proposed solution later. To guarantee that each AUV can visit at least one cluster, we have  $1 \le K \le M$ . Table 1 and Table 2 present the notions and abbreviations used in this paper.

## 3.1 Underwater acoustic channel model

The underwater acoustic communication model applied in this work is described in Chaitanya et al. (2011) specifically. The path loss depends on communication distance d in km and carrier frequency  $f_c$  in kHz is expressed as Equation 1.

$$A(d, f_c) = A_0 d^{\zeta} a(f_c)^d \tag{1}$$

where  $A_0$  is a normalized constant and  $\zeta$  is the propagation loss coefficient. The value of  $\zeta$  takes either 2 or 1 in spherical spreading and cylindrical spreading, respectively; generally,  $\zeta = 1.5$  in practical spreading.  $a(f_c)$  is the absorption coefficient, which can be expressed in dB/km as the function of  $f_c$  according to Thorp's formula (Brekhovskikh et al., 2003), i.e., Equation 2:

$$10 \log a(f_c) = \frac{0.11f_c^2}{1+f_c^2} + \frac{44f_c^2}{4100+f_c^2} + 2.75 \times 10^{-4}f_c^2 + 0.003$$
(2)

In addition, the path loss model of the acoustic channel can be expressed in dB as Equation 3:

$$10 \log A(d, f_c) = 10 \log A_0 + \zeta 10 \log d + d10 \log a(f_c)$$
(3)

Ambient noise in underwater acoustic communication is determined by several factors, such as turbulence, the shipping activity in the surrounding region, the surface motion caused by wind-driven waves, and finally thermal noise. The constant surface motion due to wind-driven waves is a significant factor contributing to the noise at the operating frequencies of interest for underwater systems (100 Hz to 100 kHz). The noise can be modeled (Stojanovic, 2007):

$$10\log N_t(f_c) = 17 - 30\log f_c \tag{4a}$$

 $10 \log N_s(f_c) = 40 + 20(s - 0.5) + 26 \log f_c - 60 \log (f_c + 0.03)$ (4b)

$$10 \log N_w(f_c) = 50 + 7.5w^{1/2} + 20 \log f_c - 40 \log (f_c + 0.4)$$
(4c)

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

where  $N_t(f_c)$ ,  $N_s(f_c)$ ,  $N_w(f_c)$ , and  $N_{th}(f_c)$  represent turbulence, shipping, waves, and thermal noise, and can be calculated by Equations 4a-4d, respectively.  $f_c$  is in kHz, *s* models the surface shipping activity ranging from 0 to 1, and *w* denotes the wind speed in m/s. The total acoustic channel noise consists of the four components, i.e., Equation 5:

$$N(f_c) = N_t(f_c) + N_s(f_c) + N_w(f_c) + N_{th}(f_c)$$
(5)

Then, the normalized signal-to-noise ratio (SNR) at the receiver of the underwater acoustic communication link can be calculated as Equation 6 (Hollinger et al., 2011):

$$\gamma(d, f_c) = \frac{1}{A(d, f_c)N(f_c)} \tag{6}$$

It can be seen that the SNR depends on the transmission distance under a fixed carrier frequency. Assuming additive white

#### TABLE 1 Notations summary.

Notation	Description							
fc	Communication frequency							
ζ	Propagation loss coefficient							
5	Shipping activity factor							
w	Surface wind speed							
Ν	Number of sensor nodes							
K	Number of AUV							
S	Set of N nodes							
ν	Set of K nodes							
$P_{tr}^{n}$	Transmitting power of sensor node $s_n$							
L	Packet length of sampled data							
$D_n$	The number of data packets sampled by $s_n$							
$t_n^G$	Packet time stamp of <i>s<sub>n</sub></i>							
Va	Navigation velocity of AUVs							
$\lambda_{i,j}$	AUV navigation time between HP $p_i$ and $p_j$							
$ heta_{m,n}$	Data uploading time between node $s_n$ and AUV when it hovers at HP $p_m$							
$P_a^h$	Hovering power of AUV							
$P_a^n$	Propelling power of AUV							
$E_a^{ini}$	Initial energy of AUV							
$E_{s_n}^{ini}$	Initial energy of $s_n$							

Gaussian noise (AWGN) channels, the capacity of the underwater acoustic channel between the AUV and sensor nodes can be given by Equation 7,

$$R(d) = \mathcal{B}\log_2\left(1 + 10^{\frac{p_{sL}^{p_{sL}}}{10}}\gamma(d, f_c)/\mathcal{B}\right)$$
(7)

where  $P_{tr}^{SL}$  is the source level of the transmitted signal in dB re  $\mu$ Pa and  $\mathcal{B}$  is the communication bandwidth in Hz.

## 3.2 Energy consumption model

The energy consumption of the sensor node  $s_n$  is mainly caused by sending its data to the AUV. We assume that the transmitting power of sensor nodes is fixed; then, the energy consumption is dependent on the data transmission time. If the sensor node  $s_n$  is associated with  $\mathbf{h}_{m}$ , the data transmission time of  $s_n$  is given by Equation 8:

$$\theta_{m,n} = \frac{D_n L}{R(d_{m,n})} + \frac{d_{m,n}}{v_s} \tag{8}$$

where  $D_n$  is the number of data packets that  $s_n$  planned to upload to the AUV. *L* is the packet length.  $d_{m,n} = ||\mathbf{h}_m - \mathbf{q}_n||_2$ denotes the transmission distance between  $h_m$  and  $s_n$ .  $v_s$  is the sound speed. We use a matrix  $\Gamma$  with binary variables  $\gamma_{m,n}$  to

#### TABLE 2 Abbreviation table.

Abbreviation	Full form				
AUV	Autonomous underwater vehicle				
AoI	Age of information				
MILP	Mixed integer linear programming				
MADCS	Multi-AUV-assisted underwater data collection scheme				
UASN	Underwater acoustic sensor network				
IoUT	Internet of underwater things				
СК	Canopy-Kmeans				
HP	Hovering point				
PSO	Particle swarm optimization				
GA	Genetic algorithm				
SA	Simulated annealing				
LRO	Light ray optimization				

indicate whether sensor node  $s_n$  is associated with  $h_m$ . In particular,  $\gamma_{m,n} = \{0, 1\}$  with  $\gamma_{m,n} = 1$  meaning that  $s_n$  belongs to the group of  $\mathbf{h}_m$  and  $\gamma_{m,n} = 0$  otherwise. We can get the following constraints:

$$\gamma_{m,n} = \begin{cases} 1, \ d_{m,n} \le r_n^c \\ 0, \ \text{otherwise} \end{cases}$$
(9a)

$$\sum_{m=1}^{M} \gamma_{m,n} = 1, \forall n \in [1, N]$$
(9b)

$$\sum_{n=1}^{N} \gamma_{m,n} \ge 1, \forall m \in [1, M]$$
(9c)

where  $r_n^c$  is the communication range of sensor node  $s_n$ . Equation 9a indicates that the distance between the HP and its cluster members should be shorter than the communication range of sensor nodes. Equation 9b means that each sensor only associates with one HP and Equation 9c shows that each HP is associated with at least one sensor node. Through the above analysis, the energy consumed by sensor node  $s_n$  can be denoted by Equation 10:

$$E_{s_n} = \sum_{m=1}^M \gamma_{m,n} P_{tr}^n \theta_{m,n}$$
(10)

where  $P_{tr}^n$  is the transmitting power of  $s_n$ . The total energy consumption of sensor nodes in *S* is represented as Equation 11:

$$E_{S} = \sum_{n=1}^{N} E_{s_{n}} = \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_{m,n} P_{tr}^{n} \theta_{m,n}$$
(11)

In addition, the energy consumption standard deviation  $\sigma_s$  is defined to measure the balance of sensor energy consumption in the network and can be calculated by Equation 12:

$$\sigma_{S} = \sqrt[2]{\frac{\sum_{n=1}^{N} (E_{s_{n}}^{res} - \bar{E}_{S}^{res})^{2}}{N-1}}$$
(12)

where  $E_{s_n}^{res}$  and  $\overline{E}_{S}^{res}$  represent the residual energy of sensor  $s_n$  and the average residual energy of N sensors in S after data collection, respectively.

As for the AUVs, their energy consumption mainly includes the navigation energy and hovering energy. The communication energy consumption of AUVs can be neglected compared with motion energy (Hou et al., 2022). The hovering energy of AUVs can be denoted by Equation 13:

$$E_{\mathcal{V}}^{h} = \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_{m,n} P_{a}^{h} \theta_{m,n}$$
(13)

where  $P_a^h$  is the hovering power of the AUV. The navigation time between  $\mathbf{h}_i$  and  $\mathbf{h}_j$  is  $\lambda_{i,j} = \|\mathbf{h}_i - \mathbf{h}_j\|_2 / v_a$ ,  $\forall i \neq j$ , where  $v_a$  is the navigation velocity of the AUV. From the AUV's perspective, the network can be seen as a graph  $\mathcal{G} = (\mathcal{H}^+, \mathcal{E})$ , where  $\mathcal{H}^+ = \mathcal{H} \cup s_0$ , while  $\mathcal{E}$  refers to the set of edges. We define a pathfinder matrix X, where  $[X]_{ij} = x_{i,j}$  is the path indicator.  $x_{i,j} = 1$  means that the subpath  $\mathbf{h}_i \rightarrow \mathbf{h}_j$  is traveled by the AUV; otherwise,  $x_{i,j} = 0$ . Then, the navigation energy of the AUV can be formulated as Equation 14:

$$E_{\mathcal{V}}^{n} = \sum_{i=0}^{M} \sum_{j=0}^{M} \lambda_{i,j} P_{a}^{n} x_{i,j}$$
(14)

where  $P_a^n$  is the propelling power of the AUV. The total energy of AUV is defined by Equation 15:

$$E_{\mathcal{V}} = E_{\mathcal{V}}^{n} + E_{\mathcal{V}}^{h} = \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_{m,n} P_{a}^{h} \theta_{m,n} + \sum_{i=0}^{M} \sum_{j=0}^{M} \lambda_{i,j} P_{a}^{n} x_{i,j}$$
(15)

### 3.3 Age of information model

The freshness of information is defined by the AoI. Let  $A_n(t)$  denote the instantaneous AoI of data retrieved from sensor  $s_n$ . We assume  $s_n$  adds timestamp  $T_n^G$  to its packets.  $A_n(t)$  is shown as Equation 16:

$$A_n(t) = \max\left\{(t - t_n^G), 0\right\}, n = 1, \cdots, N.$$
 (16)

In our scenario, AUVs offload the collected data to the data center for further analysis immediately when they return back to the data center. Thus, the AoI of data from  $s_n$  is determined by the sensor's 226 uploading periods and AUV's navigation periods. Let us take the sensor nodes visited by AUV  $v_k$  as an example to derive the AoI. Suppose there are  $M^k \left( \sum_{k=1}^K M^k = M \right)$  clusters assigned to AUV  $v_k$ for data collection and vector  $\mathbf{f}_k = [\mathbf{h}_{(0)}^k, \mathbf{h}_{(1)}^k, \cdots, \mathbf{h}_{(M^k+1)}^k]$  is used to denote the trajectory of  $v_k$ , where  $h_{(0)}^k = h_{(M^{k+1})}^k = s_0$ . Meanwhile,  $\pi_m = [s_{(1)}^m, \cdots, s_{(|\mathcal{J}_m|)}^m]$  is defined to represent the uploading sequence vector of sensors, where  $\mathcal{J}_m = \{s_n \mid \gamma_{m,n} = 1, s_n \in S\}$  is the set of sensors that are associated with  $\mathbf{h}_m$ . If  $s_n$  is associated with  $\mathbf{h}_{(m)}^k$  and is scheduled to be the *j*th one to send its data, the AoI of data uploaded by  $s_n$  when  $v_k$  completes data collection at  $\mathbf{h}_{(m)}^k$  can be represented as  $\sum_{i=j}^{|\mathcal{J}_{(m)}|} \theta_{(m),(i)}$ . We also label the AoI of  $s_n$  as  $A_{(m),(j)}^k$  and its final expression when  $v_k$  arrives at  $s_0$  is shown as Equation 17:

$$A_{(m),(j)}^{k} = \sum_{i=j}^{|\mathcal{J}_{(m)}|} \theta_{(m),(i)} + \sum_{i=m}^{M^{k}} \lambda_{(i),(i+1)} + \sum_{l=m+1}^{M^{k}} \sum_{i=1}^{|\mathcal{J}_{(l)}|} \theta_{(l),(i)}$$
(17)

where the second and third terms are the navigation time and the uploading time of all the clusters after  $v_k$  leaves  $\mathbf{h}_m$ . Thus, the sum AoI of sensors collected by  $v_k$  can be expressed as Equation 18:

$$A^{k} = \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} A_{(m),(j)}^{(k)}$$

$$= \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} \left( \sum_{l=m+1}^{M^{k}} \sum_{i=1}^{|\mathcal{J}_{(l)}|} \theta_{(l),(i)} + \sum_{i=j}^{|\mathcal{J}_{(m)}|} \theta_{(m),(i)} + \sum_{i=m}^{M^{k}} \lambda_{(i),(i+1)} \right)$$

$$= \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} \left( \sum_{i=m}^{M^{k}} \lambda_{(i),(i+1)} + \sum_{l=m}^{M^{k}} \sum_{i=1}^{|\mathcal{J}_{(m)}|} \theta_{(l),(i)} - \sum_{i=1}^{j-1} \theta_{(m),(i)} \right)$$
(18)

By recursive computation, it can be further derived as follows. PROPOSITION 1. *The sum AoI of all sensors collected by AUV*  $v_k$  can be expressed as Equation 19 (Liu et al., 2020b):

$$A^{k} = \sum_{m=2}^{M^{k}} \xi^{(m-1)} \psi^{(m)} + \sum_{i=1}^{M^{k}} \xi^{i} \lambda_{(i),(i+1)} + \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j\theta_{(m),(j)}$$
(19)

where  $\xi^{(m)} = \sum_{i=1}^{m} |\mathcal{J}_{(i)}|$  is the number of sensors visited from the first cluster to the mth cluster in the trajectory of  $v_k$  and  $\psi^{(m)} = \sum_{j=1}^{|\mathcal{J}_{(m)}|} \theta_{(m),(j)}$  denotes the total amount of uploading time of the sensors in the mth cluster visited by  $v_k$ .

PROOF. The details of the proof are provided in Appendix 1.

For all the sensors in S, their average AoI can be formulated as Equation 20:

$$\bar{A} = \frac{1}{N} \sum_{k=1}^{K} A^k \tag{20}$$

Observing Equation 19, we can easily find that the average AoI is jointly determined by the clustering results of sensors, the navigation trajectory of AUVs, and the uploading sequence of sensors.

### 3.4 Problem formulation and analysis

In this work, both the energy consumption and the average AoI are considered. The objective is to minimize the total energy consumption of the whole network while simultaneously minimizing the average AoI  $\overline{A}$  as follows:

P1): 
$$\min_{\mathcal{M}\mathcal{H}\Gamma\mathbf{X}} E_{\mathcal{V}} + E_{\mathcal{S}}$$
(21a)

$$\min_{M,\mathcal{H},\Gamma,\mathbf{X},\mathbf{f}_k,\pi_m} \bar{A}$$
(21b)

$$s.t.$$
  $1 \le K \le M$  (21c)

$$E_{s_r} \le E_{s_r}^{ini}$$
 (21d)

$$\sum_{i=1}^{M^k} P_a^n \lambda_{(i),(i+1)} + \sum_{m=1}^{M^k} \sum_{i=1}^{|\mathcal{J}_{(m)}|} \theta_{(m),(i)} P_a^h \le E_a^{ini}$$
(21e)

$$\sum_{j=1}^{M} x_{0,j} = \sum_{i=1}^{M} x_{i,0} = K$$
(21f)

$$\sum_{i=0}^{M} x_{i,j} = \sum_{j=0}^{M} x_{i,j} = 1, \forall i, j = 1, ..., M$$
(21g)

$$\sum_{m=1}^{M} \gamma_{m,n} = 1, \ \forall \ n = 1, ..., N$$
(21h)

$$\sum_{n=1}^{N} \gamma_{m,n} \ge 1, \forall m = 1, \cdots, M$$
(21i)

$$x_{i,j}, \gamma_{m,n} \in \{0,1\} \tag{21j}$$

where  $E_{s_n}^{ini}$  and  $E_a^{ini}$  are the initial energy of each sensor and AUV, respectively. Equation 21c indicates that the number of clusters should be less than that of AUVs. Equations 21d and 21e impose the energy constraints of sensor nodes and AUVs. Equation 21f means that all the AUVs are released from the data center and then return back after completing data collection tasks. Equation 21g reflects that each sensor is visited only once. The meaning of Equations 21h, 21i and 21j has been demonstrated in Section 3.2. Obviously, Equation 21 is a MILP problem, which is NP-hard.

Observing the expression of the optimization functions,  $E_S$  is determined by the association  $\gamma_{m,n}$  and the position of  $\mathbf{h}_m$ . Besides the above factors,  $E_V$  is also affected by the path of AUVs.  $\overline{A}$  is determined by the data uploading time, the navigation time of AUV, the number of sensors in each cluster, and the uploading sequence of sensors. The uploading time of  $s_n$  is influenced by the association  $\gamma_{m,n}$  and the location of  $\mathbf{h}_m$ . Thus, we can find that the optimization problem is determined jointly by the clustering result, the location of HPs, the navigation path of AUVs, and the uploading sequence of sensors in each cluster.

## 4 Proposed scheme

In this section, we propose a two-stage solution to solve the optimization problem formulated in Equation 21a. First, we design a cluster-based AUV scheduling data transmission protocol to optimize the clustering result, the locations of HPs, and the transmission sequence of sensors in each cluster. Then, considering the complexity of the problem, we propose a combinatorial heuristic algorithm to perform cluster allocation and path planning for AUVs. Based on the transmission protocol and path planning, the AoI and energy consumption in multi-AUV-assisted data collection are jointly optimized.

# 4.1 Cluster-based AUV scheduling data transmission protocol

In general, sensor nodes are deployed randomly to imitate the practical scenario in UASNs. To reduce the path length of AUVs, sensor nodes are divided into groups so that AUVs can visit each group to collect data efficiently. Kmeans clustering is the ideal method for clustering groups. However, two problems often arise in the practical application of the Kmeans algorithm. The first issue is that the Kmeans clustering algorithm requires the number of clusters as an input parameter, which is not easily known in advance. The second issue is that initial clustering centers must be selected before Kmeans clustering. These two parameters affect the final clustering results. In UASNs, the coordinates of the sensors are available, but the number of clusters and the initial clustering centers are difficult to estimate. In order to solve this problem, we adopt the Canopy algorithm as a preliminary step to the Kmeans clustering algorithm. The Canopy algorithm is a rough clustering method, which only requires the coordinates of sensor nodes as input and outputs the number of clusters and cluster centers. Nevertheless, the clustering result of the Canopy algorithm is with low accuracy. To this end, we use the number of clusters and cluster centers from the Canopy algorithm as input parameters to the Kmeans algorithm to obtain the final clustering results. We call the proposed algorithm as the CK clustering algorithm, which combines the advantages of Canopy and Kmeans algorithms. The details of the CK clustering algorithm are shown below.

The Canopy algorithm is an unsupervised pre-clustering algorithm. As shown in Figure 2, for the given sensor set S, the Canopy algorithm sets two area radius thresholds  $Z_1$  and  $Z_2$ , where  $(Z_1 > Z_2)$ . Then, a sensor  $s_i$  is randomly selected as the first canopy center. Next, the sensor that is set as center would be removed from S, i.e.,  $S' = S \setminus s_i$ . For each sensor in S', calculate the distance between it and the current canopy center. There are three possible cases. If the distance is less than  $Z_2$ , the sensor is considered to be a member of the corresponding canopy and cannot belong to another canopy. Afterwards, the sensor will be removed from S' to avoid adding to another canopy. If the distance is greater than  $Z_1$ , the sensor is regarded as a new canopy center and removed from S'. If the distance is greater than  $Z_2$  and less than  $Z_1$ , the sensor is kept in S' to participate in the next round of clustering process. The above process should be repeated until the dataset reaches empty.

Through the Canopy algorithm, we can get the number of canopies and their centers. For the sake of precise clustering, the result of the Canopy algorithm is used as input to Kmeans algorithm. The canopy centers are selected as initial cluster centers of Kmeans algorithm. Then, calculate the distance between the remaining sensor nodes with each initial cluster center. The sensor nodes will be associated with the nearest cluster center. After completing the association of all sensor nodes with cluster centers, the cluster centers are updated according to Equation 22:



$$C'_{i} = \frac{1}{|S_{i}|} \sum_{\mathbf{q}_{n}^{s} \in S_{i}} \mathbf{q}_{n}^{s}$$
(22)

where  $C'_i$  is the new center of cluster  $S_i |S_i|$  and  $\mathbf{q}^s_n$  denote the number of sensors in  $S_i$  and the coordinates of  $s_n$ , respectively. For each cluster  $S_i$ , we check whether  $C'_i$  is equal to  $C_i$ , where  $C_i$  is the clustering center from the last computation. If all the cluster centers remain unchanged, the clustering process should be stopped and output the Kmeans clustering results, i.e., the location of the cluster centers and the association between the sensor nodes and the cluster centers. Otherwise, the steps of dividing clusters and updating the clustering centers would be performed cyclically. The specific steps of the CK clustering algorithm are depicted in Algorithm 1. The Canopy algorithm determines the number of canopies and the initial centers from lines 1 to 13. Kmeans algorithm performs precise clustering from lines 15 to 21.

#### Input:

The sensor set  ${\cal S}$  and their coordinates  $q_n^s$ ;

#### Output:

The number of clusters: *M*, the positions of cluster centers:  $C_i$ , the association matrix:  $\Gamma = [\gamma_{m,n}]$ ;

1: Set the initial distance threshold  $T_1$  and  $T_2$  based on GridSearchCV, where  $T_1 > T_2$ ;

- 2: while  $\mathcal{S} \neq \emptyset$  do
- 3: Randomly select  $s_i \in S$  as canopy center  $C_i$ , update  $S = S \setminus s_i$ ;

- 4:  $\forall s_j \in S$ , calculate the Euclidean distance  $d_{ij}$ between  $s_j$  and  $C_i$ ;
- 5: if  $d_{ij} \leq T_2$  then
- 6: Associate  $s_j$  to  $C_i$  and remove  $s_j$  from  $S: S = S \setminus s_j$
- 7: else if  $d_{ij} > T_1$  then

8: Regard  $s_j$  as a new canopy center  $C_j$  and update  $S = S \setminus s_j$ ;

- 9: **else if**  $T_2 < d_{ii} \le T_1$  **then**
- 10: Keep  $s_i$  in S;
- 11: end if
- 12: end while
- 13: Print the canopy centers.
- 14: Kmeans input: the canopy centers as initial cluster centers;
- 15: for Each sensor  $s_i \in S$  do
- 16: Associate it with the nearest cluster;
- 17: end for
- 18: Compute new center according to Equation 22;
- 19: while New center ≠ Original center do
- 20: Repeat steps 15-18.

21: end while 22: return *M*, {*C*<sub>1</sub>,...,*C*<sub>M</sub>},  $\Gamma^* = [\gamma_{mn}^*]$ ;

Algorithm 1. The CK clustering algorithm.

In previous works (Han et al., 2021; Omeke et al., 2021), the sensor node closest to the cluster center is appointed as the cluster head, which is responsible for gathering the data from the cluster members and forwarding it. However, as we have stated in Section 3, this approach introduces the problem of unbalanced energy consumption and unnecessary secondary forwarding. In this paper, we directly select the cluster center as the virtual cluster head of the cluster, which is not a sensor entity. The virtual cluster head is a hovering position reserved for the AUV, i.e.,  $\mathbf{h}_m = C_m$ . When the AUV arrives at  $\mathbf{h}_m$ , it establishes communication links with the sensor nodes in the cluster to collect data. Through CK clustering and virtual cluster head selection, we associate each sensor node with the nearest hover point.

To collect data efficiently, we design a scheduling-based data transmission protocol. For simplicity, we take the data transmission process in one cluster as an example to illustrate the protocol, which is shown in Figure 3. Upon hovering at  $\mathbf{h}_m$ , the AUV immediately sends an ANNUNCIATION packet (*AnP*) to wake up cluster members. Since the locations of cluster members are known, the propagation delay can be easily computed. *AnP* contains the AUV's ID and the propagation delays from the AUV to each cluster member. Note that the propagation delay has an impact on determining when the sensor should send its REQUEST packet (*ReP*) and DATA packet (*DaP*). After the *AnP* is transmitted, the AUV starts a timeout  $T_{ReP}^{TO}$  to wait for

replies from sensors. Ideally, the *ReP* from sensors that are associated with  $\mathbf{h}_m$  should arrive at the AUV before  $T_{ReP}^{TO}$  expires. The rule of thumb is that the sensor responds *ReP* immediately upon receiving the *AnP*. However, *ReP* collisions occur occasionally, which reduces the channel utilization and throughput. Fortunately, propagation delays included in *AnP* can be used to avoid *ReP* collisions. First, the sensor ranks the propagation delays in ascending order. Suppose that the sensor finds itself with order *l*, the time at which it should send its own *ReP* can be calculated as:

$$t_{(l)}^{ReP} = \max \left\{ t_{(l)}^{AnP}, t_{(l-1)}^{ReP} + d_{m,(l-1)} / v_s + T_g + T_{ReP} - d_{m,(l)} / v_s \right\}$$
(23)

where  $t_{(l)}^{AnP}$  is the time at which the sensor receives the AnP,  $d_{m,(l-1)}$  is the propagation delay between AUV and the sensor with order l - 1,  $T_g$  is a small guard time, and  $T_{ReP}$  is the packet duration of *ReP*. In Equation 23,  $t_{(l-1)}^{ReP}$  is involved in the calculation of  $t_{(l)}^{ReP}$ . The  $t_{(l)}^{ReP}$  can be easily calculated step by step when  $t_{(1)}^{ReP}$  is known. Since the AnP is sent from the AUV at the same time,  $t_{(1)}^{ReP}$  can be calculated as Equation 24:

$$t_{(1)}^{ReP} = t_{(1)}^{AnP} = t_{(l)}^{AnP} - d_{m,(l)} / v_s + d_{m,(1)} / v_s$$
(24)

From the perspective of AUV, the length of  $T_{ReP}^{AUV}$  needs to cover from sending AnP to the arrival of the last ReP. For the sensor with order  $|\mathcal{J}_m|$ , the maximum back-off interval could be  $(|\mathcal{J}_m| - 1)(T_g + T_{ReP})$ . Thus, the expression of  $T_{ReP}^{TO}$  is  $2d_{m,(|\mathcal{J}_m|)}/v_s + (|\mathcal{J}_m| - 1)(T_g + T_{ReP})$ .

The number of data packets (i.e.,  $D_n$ ) that sensor  $s_n$  plans to transmit and its own ID should be included in the *ReP*. After sending the *ReP*,  $s_n$  starts a timeout  $T_{OP}^{TO}$  within which it expects to receive an ORDER packet (*OrP*) from the AUV. Naturally, for the sensor with order *l*, its  $T_{OP}^{TO}$  can be calculated as  $t_{|\mathcal{J}_m|}^{ReP} - t_l^{ReP} + T_{ReP} + 2d_{m,(|\mathcal{J}_m|)}$ . As we have analyzed, uploading the sequence of sensor



10.3389/fmars.2025.1580751

nodes influences the AoI. When  $T_{OrP}^{TO}$  expires, the AUV is supposed to determine the uploading sequence and transmission time for all sensor nodes in the cluster. Based on the  $D_n$  extracted from the received *RePs*, the AUV can calculate the sensor's uploading time  $\theta_{m,n}$  according to Equation 8. We find that the descending order of  $\theta_{m,n}$  is the optimal uploading sequence. Specifically, for any given navigation trajectory  $\mathbf{f}_k$  and association matrix  $\Gamma$ , the first and second items of Equation 19 are fixed. The AoI is solely determined by the third item  $\sum_{m=1}^{M^k} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j\theta_{(m),(j)}$ , which depends on the sensor's uploading sequence in each cluster. Giving higher uploading priority to sensor nodes with greater uploading time can minimize the third term, i.e., Equation 25:

$$\theta_{(m),(1)} \ge \theta_{(m),(2)} \ge \dots \ge \theta_{(m),(\left|\mathcal{J}_{(m)}\right|)}$$

$$(25)$$

It can be proved by the law of contradiction. We assume that the uploading sequence is not in the descending order of uploading time, then it certainly satisfies  $\exists j > i$ ,  $\theta_{m,(j)} \ge \theta_{m,(i)}$ . Thus,  $j(\theta_{m,(j)} - \theta_{m,(i)}) \ge i(\theta_{m,(j)} - \theta_{m,(i)})$ , then shift the terms to get  $i\theta_{m,(i)} + j\theta_{m,(j)} \ge i\theta_{m,(j)}$ , which contradicts our goal of minimizing  $\sum_{m=1}^{M^k} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j\theta_{(m),(j)}$ . The result can be easily extended to all the nondescending sequences. Obviously, when the sensors in each cluster upload data in the descending order of their uploading time, the value of  $\sum_{m=1}^{M^k} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j\theta_{(m),(j)}$  gets minimum.

With the determination of uploading sequence, the AUV can calculate when the sensor should start transmitting its *DaP*. In this article, we compute the scheduling time for each sensor, which refers to the expected time interval from receiving the *OrP* to the start of *DaP* transmission. For the *i*th sensor, its scheduling time  $t_i^{sch}$  is denoted as Equation 26.

$$t_{(i)}^{sch} = \begin{cases} 0, & i = 1 \\ d_{m,(1)}/\nu_s + (d_{m,(1)}/\nu_s + \theta_{(m),(1)} + T_g) - d_{m,(2)}/\nu_s, & i = 2 \\ t_{i-1}^{sch} + d_{m,(i-1)}/\nu_s + \theta_{(m),(i-1)} + T_g - d_{m,(i)}/\nu_s, & 3 \le i \le |\mathcal{J}_m| \end{cases}$$
(26)

Then, the scheduling time is embedded into the *OrP* and sent to the sensors. After that, the AUV starts a timeout  $T_{DaP}^{TO}$  to receive *DaPs*. The expression for  $T_{DaP}^{TO}$  is shown as Equation 27:

$$T_{DaP}^{TO} = t_{(\mathcal{J}_m)}^{sch} + \theta_{m,(\mathcal{J}_m)} + d_{m,(|\mathcal{J}_m|)} / \nu_s + T_g$$
(27)

If the sensor receives the *OrP* before  $T_{OrP}^{TO}$  expires and finds its ID in the packet, it stops  $T_{OrP}^{TO}$  and starts  $t_{(i)}^{sch}$ . Otherwise, it returns to an IDLE state and waits for another *AnP*. When  $t_{(i)}^{sch}$  expires, the sensor transmits *DaP* to the AUV. All the *DaPs* will form a packet train at the AUV. Through the cluster-based AUV scheduling data transmission protocol, the AUV accomplishes data collection from sensors within a cluster.

## 4.2 PGS-based AUV trajectory planning

Based on the CK-clustering and scheduling protocol above, the association between the HPs and the sensor nodes, the location of

HPs, and the uploading sequence of the sensor nodes in each cluster are determined. In Equations 21a and 21b, the items related to these variables are also determined. Specifically, they are the  $E_{S}$ , the first item of  $E_{V}$ . In addition, the first and the third items of  $\bar{A}$  are included. The second item of  $E_{V}$ , i.e.,  $\sum_{i=0}^{M} \sum_{j=0}^{M} \lambda_{i,j} P_a^n x_{i,j}$ , and the second item of  $\bar{A}$ , i.e.,  $\sum_{k=1}^{K} \sum_{i=1}^{M^k} \xi^i \lambda_{(i),(i+1)}$  could be further optimized. The objective of the subproblem is modeled as Equation 28:

$$(P2): \min_{\mathbf{X}} \quad \sum_{i=0}^{M} \sum_{j=0}^{M} \lambda_{i,j} P_{a}^{n} x_{ij}$$

$$\min_{\mathbf{X}, f_{k}} \quad \sum_{k=1}^{K} \sum_{i=1}^{M^{k}} \xi^{i} \lambda_{(i),(i+1)}$$

$$s.t. \quad (22c), (22e), (22f), (22g), x_{ij} \in \{0,1\}.$$

$$(28)$$

Both objective functions are only affected by the AUVs' trajectories. Moreover, the pathfinder matrix **X** is decided by the trajectory of each AUV, i.e.,  $[\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_K]$ . Then, the next step is to plan trajectories for the *K* AUVs to traverse all the HPs. The problem (*P2*) can be reformulated as a special multi-AUV traveling salesman problem (MTSP):

$$(P3): \min_{\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K\}} \sum_{k=1}^{K} \sum_{i=1}^{M^k} (\xi^i + P_a^n) \lambda_{(i), (i+1)}$$

$$s.t. \quad (22c), (22e), (22f), (22g).$$
(29)

MTSP is NP-hard. In order to reduce the computational complexity, we propose a combinatorial heuristic algorithm to solve this problem. The algorithm is named PGS, which combines the advantages of particle swarm optimization (PSO) algorithm, genetic algorithm (GA), and simulated annealing (SA) algorithm. On the one hand, considering the characteristics of the problem, we apply the particles updating method of discrete PSO and the crossover and mutation operation of GA to PGS (Roberge et al., 2012; Abhishek et al., 2020). On the other hand, PGS retains the strategies of particle information exchange and collaboration in PSO and introduces the annealing operation to enhance the global optimization ability of particles (Lin et al., 2023).

The goal of P3 is to find a path  $\mathbf{f}_k = [\mathbf{h}_{(0)}^k, \mathbf{h}_{(1)}^k, \dots, \mathbf{h}_{(M^k)}^k, h_{(M^{k+1})}^k]$ for AUV  $v_k$ ,  $(k = 1, \dots, K)$  while minimizing Equation 29. We adopt the integer sequence as the path encoding in PGS. For example, path  $2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 0$  can be expressed as [0, 2, 5, 4, 3, 1, 0]. Since the initial solution has a significant impact on the quality of the final solution in the heuristic algorithm, we apply the light ray optimization (LRO) algorithm to obtain high-quality initial solutions. We first introduce several concepts: (1) Current HP's density: the shortest path from the current HP to another HP that has not been visited; (2) Frontier HP's density: the average path length among the remaining HPs except the HPs that have been visited; (3) Reflection: an unvisited HP is randomly selected as the next one; and (4) Refraction: select the unvisited HP with the shortest distance from the current HP as the next one. The procedure of LRO to produce the initial solution is as follows: (1) Randomly select an HP as the first visited one and calculate the current HP's density and the frontier HP's density respectively; (2) Compare them; if the current HP density is greater than the frontier HP density, perform reflection; otherwise, perform refraction; (3) Repeat the second step until every HP is included in the trajectory; and (4) Let each HP serve as the first visited point and repeat the above three steps to obtain M initial solutions.

In the classic PSO algorithm, the particles update themselves by inheriting the speed of the previous moment, self-learning, and information interaction of the population, which is shown as:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (\mathbf{p}_{i,best}^k - \mathbf{z}_i^k) + c_2 r_2 (\mathbf{G}_{best}^k - \mathbf{z}_i^k)$$
  
$$z_i^{k+1} = \mathbf{z}_i^k + v_i^{k+1}$$
(30)

where  $z_i^k$  represents the position of particle *i* at time *k*,  $v_i^k$  denotes the velocity of particle *i* at time *k*.  $p_{i,best}^k$  prefers to the local optimal solution searched by particle *i* at time *k*,  $G_{best}^k$  refers to the global optimal solution searched by the whole population at time *k*.  $\omega$  is the inertia factor.  $c_1$  and  $c_2$  are the local learning factor and global learning factor, which represent the ability to learn from their own experience and population experience, respectively.  $r_1$  and  $r_2$  are random numbers in [0,1], increasing the randomness of the searching process. However, Equation 30 is designed for continuous domain and cannot be applied to discrete problems. For the proposed problem *P*3, we combine the crossover and mutation operations with the particles' dating mode of PSO to achieve optimization searching in discrete problems:

$$\begin{aligned} v_i^{k+1} &= \omega \odot v_i^k \oplus r_1(\mathbf{p}_{i,best}^k \otimes \mathbf{z}_i^k) \oplus r_2(\mathbf{g}_{best}^k \otimes \mathbf{z}_i^k) \\ \mathbf{z}_i^{k+1} &= \mathbf{z}_i^k \oplus v_i^{k+1} \end{aligned}$$
(31)

where  $\otimes$  and  $\odot$  denote the crossover and mutation operations, respectively.  $\oplus$  represents accumulation of operations performed on particles. The item  $\omega \odot v_i^k$  indicates that the mutation operation occurs with probability  $\omega$ . The items  $r_1(\mathbf{p}_{i,best}^k \otimes \mathbf{z}_i^k)$  and  $r_2(\mathbf{g}_{best}^k \otimes \mathbf{z}_i^k)$  indicate that the outcomes resulting from the crossover operation of particle *i* with  $\mathbf{p}_{i,best}^k$  and  $\mathbf{g}_{best}^k$  will be adopted with probabilities  $r_1$  and  $r_2$ , respectively.

In the PSO algorithm, most of the particles tend to gather near the local optimal area when the current optimal positions of particles approach the local optimal solution, i.e., falling into local optimum. In order to strengthen the global search ability of particles and augment the probability of escaping from local optimum, the annealing method in the SA algorithm is introduced. The Metropolis sampling criterion is the core idea of annealing, which is used to calculate the probability that the new solution can be accepted:

$$P_{i \to j} = \begin{cases} 1, & E_j < E_i, \\ \exp\left(-\frac{E_j - E_i}{T}\right), & E_j \ge E_i. \end{cases}$$
(32)

where  $E_i$  and  $E_j$  are the energy of the particles in state *i* and *j*, respectively. *T* represents the current temperature.  $P_{i\rightarrow j}$  denotes the probability of the system state transferring from state *i* to state *j*. In the PGS, we define the fitness function as  $f = \sum_{k=1}^{K} \sum_{i=1}^{M^k} (\xi^i + P_n^a) \lambda_{(i),(i+1)}$ . Referring to Equation 32, the probability of the new solution being accepted is:

$$P = \begin{cases} 1, & f(z_i^{k+1}) < f(z_i^k) \\ \exp\left(-\frac{f(z_i^{k+1}) - f(z_i^k)}{T}\right), & f(z_i^{k+1}) \ge f(z_i^k) \end{cases}$$
(33)

The details of the proposed PGS algorithm are shown in Algorithm 2.

### Input:

The coordinates of *M* HPs and the number of AUVs *K*;

PGS-related parameters:  $\omega$ ,  $r_1$ ,  $r_2$ , maxiterations  $l_{max}$ ;

- 1: Initialize the population through the LRO algorithm;
- 2: **for** each particle in the initial population **do**
- 3: Assign the path to K AUVs;

4: Calculate the fitness value of particle according to the fitness function  $f = \sum_{k=1}^{K} \sum_{i=1}^{M^k} (\xi^i + P_n^a) \lambda_{(i),(i+1)}$ 

#### 5: end for

- 6: Get  $p_{i,best}^0$  and  $g_{best}^0$ ;
- 7: Iteration *l* = 1;
- 8: while  $l \leq l_{max}$  do
- 9: for each particle *i* do
- 10: Compute the velocity and update the location of particle according to Equation 31;
- 11: Calculate the fitness value of updated particle;
- 12: Calculate the probability of new solution being accepted by Equation 33 and get a Random(0,1);
- 13: **if** Random(0,1) < the probability **then**
- 14: Accept the new solution
- 15: end if
- 16: **end for**
- 17:  $l \leftarrow l + 1$

18: end while l > l<sub>max</sub>
Output:

The navigation trajectory of K AUVs  $\{\mathbf{f}_1^*, \dots, \mathbf{f}_K^*\}$ , where  $\mathbf{f}_k^* = [\mathbf{h}_{(0)}^k, \mathbf{h}_{(1)}^k, \dots, \mathbf{h}_{(M^k)}^k, \mathbf{h}_{(M^{k-1})}^k]$ 

Algorithm 2. PGS algorithm.

## 4.3 MADCS design

Based on the above solutions, we propose a MADCS to solve the *P*1, an AoI and energy joint optimization problem. The procedure of MADCS is shown in Algorithm 3 and the main steps of MADCS are as follows. First, we construct the cluster-based network by Algorithm 1 to determine associations between sensor nodes and HPs, and then find the position of HPs. In the second phase, we plan the navigation path for each AUV via Algorithm 2 according to the different numbers of AUVs. Upon arriving at HPs, the AUV starts the scheduling protocol described in Section 4.1 to complete data transmission. After traversing the HPs assigned to it, the AUV returns to the surface data center to offload all the collected data.

#### Input:

The sensor set  ${\cal S}$  and their coordinates  $q^{\rm S}_{\it n}\,;$  The number of AUVs: K;

Canopy parameters:  $T_1$  and  $T_2$ ; PGS-related parameters:  $\omega$ ,  $r_1$ ,  $r_2$ , max iterations  $l_{max}$ ;

1: Run Algorithm 1 to get the number of clusters *M* and the location of HPs:  $\left\{\mathbf{h}_{1}^{*}, \mathbf{h}_{2}^{*}, \cdots, \mathbf{h}_{M}^{*}\right\}$ , and the association matrix between sensor nodes and HPs:  $\Gamma^{*} = [\gamma_{m,n}^{*}]$ ;

2 : Run Algorithm 2 to find the navigation path for K AUVs :  $\{f_1^*, \cdots f_k^*\}$ ;

3: Apply AUV scheduling data transmission protocol when the AUVs arrive at the corresponding HPs.

#### Output:

The solution for Problem P1: M,  $\mathcal{H}$ ,  $\Gamma$ , X,  $\mathbf{f}_k$ ,  $\pi_m$ .

Algorithm 3. MADCS.

# 5 Simulation results and analysis

## 5.1 Simulation setup

In this section, extensive simulations are carried out to evaluate the performance of the proposed MADCS algorithm. We first introduce the scenario setup of simulations. Consider an AUVassisted UASN that consists of a data center, *M* AUVs, and *N* sensor nodes. The value of *N* is set to 150 unless otherwise specified. The sensor nodes are randomly deployed in a 3D underwater environment with a size of 5,000 m × 5,000 m × 1,000 m. The data center is located at the sea surface with coordinate (0, 0, 1,000). All AUVs start from the data center and return back after traversing the assigned HPs. The parameters of underwater acoustic modems are chosen from technical specifications of the EvoLogics S2C modems (Favaro et al., 2013; EvoLogics, 2023). The transmission power is set to 154 dB re  $\mu$ Pa<sup>2</sup>, which yields a transmission range of approximately 1,000 m. The carrier frequency and the bandwidth of the PHY layer are 26 and 16 kHz, respectively. Other detailed parameters are listed in Table 3.

# 5.2 Analysis of hyper-parameters in clustering

In simulations, the CK algorithm is firstly used to obtain the association matrix between the sensor nodes and the HPs, and the positions of HPs.  $T_1$  and  $T_2$  are two important hyper-parameters in clustering and have a great impact on the clustering results. On the one hand, if  $T_1$  is too large, it will result in a high overlap of canopies, where one node belongs to multiple canopies. On the other hand, if  $T_2$  is excessively large, it will lead to a low number of canopies. Conversely, when  $T_2$  is too small, the number of canopies increases, leading to an increase in computational time. In this paper, we adopted the GridSearchCV method to choose the values of  $T_1$  and  $T_2$ . The mean distance of node pairs among S is denoted as  $\bar{d} = \frac{1}{N(N-1)} \sum_{i,j=1}^{N} d(s_i, s_j)$ . The minimum distance between nodes in the S is denoted as  $d_{min} = \min \{ d(s_i, s_j) | i \neq j \}$ . We set the grid search range of  $T_1$  and  $T_2$  are  $\left[\frac{d+d_{\min}}{4}, \overline{d}\right]$  and  $\left[d_{\min}, \frac{d+d_{\min}}{2}\right]$ , respectively. The search steps of  $T_1$  and  $T_2$  are  $\frac{\overline{d}}{10}$  and  $\frac{\overline{d}}{20}$ . In addition, three classical cluster validity indices are applied to evaluate the effectiveness of  $T_1$  and  $T_2$ , i.e., the silhouette coefficient, the Calinski-Harabasz index, and the Davies-Bouldin index. The specific meanings and equations of the three indices are listed in Appendix 2.

Take the number of 150 sensor nodes as an example; the results of three evaluation indices are summarized in Tables 4–6. Observed from the three tables, we can get that when  $T_1 = 1,150$  and  $T_2 = 650$ , the three evaluation indices reach optimal values. Through the CK clustering algorithm with  $T_1 = 1,150$  and  $T_2 = 650, 150$  sensor nodes are grouped into 25 clusters, which is shown in Figure 4. We use different colors to distinguish different clusters; however, the same color is used in several non-adjacent clusters due to the lack of color.

## 5.3 Performance comparison and analysis

To evaluate the performance of the proposed MADCS, we compare it with three benchmark algorithms, CKA-DPSO, LEACH-TSP, and DEKCS. CKA-DPSO is based on the clusterbased AUV scheduling data transmission protocol in Section 4.1,

#### TABLE 3 Simulation parameters.

Parameters	Value					
Network size	5,000 m ×5,000 m ×1,000 m					
The number of sensor nodes (N)	150					
Carrier frequency (f <sub>c</sub> )	26 kHz Favaro et al. (2013)					
Bandwidth (B)	16 kHz Favaro et al. (2013)					
Communication range $(r_n^c)$	1,000 m					
Packet length (L)	1,024 bit					
Cruise speed of AUV (V)	6 m/s Han et al. (2021)					
Spreading factor (k)	1.5 Fang et al. (2021)					
Shipping activity factor (s)	0.5 Fang et al. (2021)					
Wind speed (w)	0 m/s Fang et al. (2021)					
Initial energy of sensors $(E_{s_n}^{ini})$	40 kJ					
Initial energy of AUV $(E_a^{ini})$	600 kJ					
Unit energy consumption of AUV	7 J/m Han et al. (2021)					
Navigation power of AUV $(P_a^n)$	42 W					
Hovering power of AUV $(P_a^h)$	15 W Cruz (2019)					
Transmission power of sensor $(P_{tr}^n)$	2.8 W EvoLogics (2023)					

while applying DPSO algorithm to solve Problem P3 rather than the PGS algorithm. Given that LEACH (Heinzelman et al., 2002) is a data collection protocol without path planning, thus we add a path planning phase in LEACH. The path planning is formulated as a TSP problem that is solved by PGS. It is called LEACH-TSP. Moreover, the details of DEKCS are described in Omeke et al. (2021).

Obviously, the number of AUVs affects the average AoI. Since the CK algorithm divides the 150 sensor nodes into 25 clusters, we first study the trend of the average AoI with the number of AUVs varying from 1 to 25, as shown in Figure 5. In Figure 5A, it

	$T_2$	/1	65
--	-------	----	----

TABLE 4 Silhouette score

is difficult to distinguish the values of all average AoI under 1-25 AUVs in one graph because of the large span of average AoI values. For a clearer observation, we segregate Figure 5A into three figures, Figures 5B-D. They show the variation of average AoI at AUV numbers of 1-5, 6-15, and 16-25, respectively. As seen in Figure 5B, the average AoI decreases sharply when the number of AUVs increases from 1 to 2. It is because nearly half of the clusters are traversed by the other AUV, which means nearly half of the navigation time and uploading time is reduced, so that the average AoI curve is the steepest there. As the number of AUVs increases, the average AoI of four algorithms decreases. The explanation is that each AUV visits fewer HPs in its trajectory, so the packets wait for less time before arriving at the data center. Among the four data collection algorithms, the MADCS algorithm achieves the smallest average AoI under any number of AUVs, implying that the data are the freshest. In addition, we can find in Figure 5D that the average AoI is the same when 25 AUVs are dispatched to collaborate on data collection. Back to Figure 4, the sensors are grouped into 25 clusters, and each AUV visits one HP with a straight-line path when 25 AUVs are dispatched. Consequently, there is no difference between four algorithms in the case of 25 AUVs. Naturally, the average AoI is the same.

Figure 6 shows how the AUVs' total energy consumption is affected by the number of AUVs. We only chose several typical numbers to depict the tendency. It is seen that as the number of AUVs increases, the AUVs' total energy consumption increases and tends to the same maximum value. Because more AUVs navigate greater distances, the total energy consumption increases. As the number of AUVs increases, the navigation energy consumption between HPs is constant, while each AUV consumes more energy to depart from and return to the data center. On the other hand, in the case of 25 AUVs, the energy consumption is the same among four algorithms, which has been analyzed before. Since the shortest path is considered in LEACH-TSP, it always consumes minimal energy, while MADCS comes in a close second. However, the proposed MADCS performs much better in average AoI than LEACH-TSP.

$T_2$ $T_1$	650	900	1,150	1,400	1,650	1,800	2,050	2,300	2,500
150	0.2752	0.2057	0.1742	0.1605	0.1296	0.1373	0.1379	0.1275	0.1249
275	0.3411	0.3364	0.3261	0.3175	0.3050	0.3000	0.4302	0.3005	0.4206
400	0.3499	0.3217	0.3477	0.3556	0.3527	0.3311	0.3314	0.3445	0.3425
525	0.4445	0.4494	0.3841	0.4167	0.4106	0.3852	0.3881	0.3812	0.3563
650	0.4963	0.4876	0.5012	0.4873	0.4733	0.4695	0.4745	0.4766	0.4664
775	*	0.4586	0.4966	0.4562	0.4533	0.4967	0.4963	0.4812	0.4812
900	*	0.4436	0.4491	0.4485	0.4688	0.4493	0.4631	0.4846	0.4599
1,025	*	*	0.4414	0.4293	0.4529	0.4490	0.4469	0.4451	0.4420
1,150	*	*	*	0.4267	0.4402	0.7949	0.4444	0.4495	0.4231

The bold values represent the optimal values of three indices. And symbol \* means that there are no values, because the values of T1 and T2 are illegal to compute the cluster validity indices.

TABLE 5 Calinski Harabasz	score
---------------------------	-------

< -									
$T_2$ $I_1$	650	900	1,150	1,400	1,650	1,800	2,050	2,300	2,500
150	297.22	369.20	406.43	427.33	456.82	459.91	450.66	462.45	463.56
275	232.53	245.48	271.17	271.87	276.94	272.19	261.52	268.33	271.19
400	216.85	210.35	222.18	213.03	224.37	217.22	222.87	232.71	217.86
525	256.22	253.81	236.79	241.82	230.92	236.49	225.78	225.58	226.42
650	259.83	258.32	280.87	275.87	267.95	262.79	266.65	265.95	260.02
775	*	219.64	258.65	213.29	215.37	258.65	259.83	261.18	241.02
900	*	199.95	204.57	203.29	218.63	209.15	212.78	244.70	215.57
1,025	*	*	200.19	190.69	212.82	201.14	197.41	202.51	192.68
1,150	*	*	*	182.98	189.84	189.26	197.22	201.38	188.23

The bold values represent the optimal values of three indices. And symbol \* means that there are no values, because the values of T1 and T2 are illegal to compute the cluster validity indices.

Next, we plot the relationship between the collection utility and the number of AUVs in Figure 7. We find that the proposed MADCS algorithm always holds a higher collection utility among four algorithms. This illustrates the effectiveness of the MADCS algorithm. Interestingly, the collection utility of different approaches increases with the rise of the number of AUVs and then decreases after reaching a peak value. The peak value is reached when the number of AUVs is 5. The justification behind this is that the collection delay decreases with the rise of AUVs' number, leading to an improvement in the collection utility. However, more AUVs bring more energy consumption. Therefore, when continuing to increase the number of AUVs, the increase in energy consumption will outweigh the decrease in collection delay, which, in turn, degrades the collection utility.

We have investigated the performance of the four algorithms in terms of average AoI, energy consumption, and collection utility, and the proposed MADCS algorithm performs the best overall. In order to more intuitively demonstrate how the AUVs travel in the 3D scenario when the MADCS algorithm is applied, we show the trajectories of 1, 8, 16, and 25 AUVs under MADCS in Figures 8A,

B, C, D, respectively. As shown in Figure 8, the number of HPs traversed by each AUV decreases with the AUVs increasing. Hence, the navigation time and hovering time of each AUV are reduced, which results in a smaller AoI. In the case of fewer than 25 AUVs, the trajectory of each AUV can be optimized, including the number of HPs visited by it and the traversal order. Yet, in the case of 25 AUVs, the trajectory of each AUV is fixed, i.e., every AUV visits only one HP, collecting data and then returning to the data center. It is the evidence why the above-mentioned indicators are the same in the case of 25 AUVs.

In the previous analysis, we have mentioned that reducing energy consumption and average AoI simultaneously is difficult to do because they compete with each other. It is necessary to find the optimal trade-off between energy consumption and average AoI. Figure 9 depicts the Pareto frontiers for average AoI and energy consumption in MADCS. The results are consistent with the intuition that average AoI and energy consumption compete with each other. It is worth noticing that the Pareto points distribute sparsely and overlap, because both average AoI and energy consumption are functions of the AUVs' trajectory, which is

$T_2$ $T_1$	650	900	1,150	1,400	1,650	1,800	2,050	2,300	2,500
150	0.3689	0.2669	0.2219	0.2125	0.1850	0.1804	0.1910	0.1722	0.1776
275	0.5941	0.5635	0.5063	0.4614	0.4488	0.4193	0.4302	0.4528	0.4206
400	0.7911	0.8568	0.7654	0.7418	0.6945	0.7376	0.7042	0.6705	0.7382
525	0.7131	0.7452	0.8545	0.7736	0.7867	0.8258	0.7782	0.81162	0.8627
650	0.6406	0.6746	0.6432	0.6606	0.6811	0.6811	0.6967	0.6799	0.7056
775	*	0.7099	0.6448	0.7103	0.7465	0.6448	0.6406	0.6735	0.6781
900	*	0.7557	0.6899	0.7506	0.6923	0.7612	0.7008	0.6698	0.7263
1,025	*	*	0.7076	0.7613	0.7172	0.7356	0.7314	0.7376	0.7445
1,150	*	*	*	0.7649	0.7749	0.7949	0.7600	0.7354	0.7893

The bold values represent the optimal values of three indices. And symbol \* means that there are no values, because the values of T1 and T2 are illegal to compute the cluster validity indices.

TABLE 6 Davies Bouldin score.







determined by discrete integer variables and thus cannot vary smoothly. Furthermore, generating an operating solution from the Pareto front is always meaningful in practice. As shown in Figure 9, the closest solution to the utopian point is dispatching five AUVs. Therefore, we choose five AUVs as the operating point and will analyze the impact of other factors on the performance of the MADCS algorithm in the case of five AUVs. The location and the performances of the operating point are marked with a black circle and rectangle, respectively, in Figure 9.

The velocity of AUVs affects their navigation time, which is correlated with AoI. Faster velocity requires higher propulsion power, leading to increased energy consumption. In the following



16







experiment, we investigate the impact of AUV's velocity on the performance of MADCS and the benchmark schemes. As shown in Figure 10, we compare the average AoI in different AUV velocities. Apparently, the average AoI of the four algorithms decreases as the velocity of AUVs increases. The greater the velocity of AUVs, the shorter the navigation time. Thus, the average AoI becomes smaller. In addition, the MADCS algorithm always holds the freshest average AoI under any velocity. It can be seen that the average AoI decreases sharper under small velocity than at large velocity. The reason is that the average AoI consists of navigation time and hovering time, while only the navigation time could be affected by the velocity of AUVs. When the velocity is small, the navigation time accounts for the majority of average AoI, whereas the collection time only accounts for a small portion. However, the situation reverses when in large velocity.



To demonstrate the effectiveness of the cluster-based AUV scheduling data transmission protocol, the performance of network lifetime and the residual energy standard deviation of sensors are explored. Network lifetime is defined as the duration until the first node runs out of its energy. CKA-DPSO differs from MADCS only in terms of path planning, so the performance of both metrics is the same. The network lifetime with different numbers of sensor nodes is shown in Figure 11, where MADCS, LEACH-TSP, and DEKCS are compared. From Figure 11, it can be seen that the network lifetime of MADCS increases with the number of sensor nodes, while the network lifetime of LEACH-TSP and DECKS decreases. The reason is as follows. In LEACH-TSP and DECKS, all the sensor nodes in the cluster transmit data to the cluster head, and then the cluster head transmits the data to the sink node and AUV, respectively. Sending the aggregated data of the whole cluster consumes a lot of energy of the cluster head. Moreover, when the network coverage area is fixed, the increase in the number of sensor nodes increases the node density. Then, the number of sensors in a cluster grows, which leads to the cluster heads spending more energy to collect and transmit data, thus reducing lifetime, whereas in MADCS, the AUV hovers at the cluster center position to act as a cluster head and the nodes within the cluster transmit data directly to the AUV. The energy consumption for secondary forwarding of data is saved. Therefore, the nodes consume less energy for data transmission than LEACH-TSP and DECKS and the network lifetime is extended. The distance between the nodes and AUVs is shortened as the number of nodes increases. Thus, the communication energy consumed by the nodes is reduced and the network lifetime is slightly increased.

The residual energy standard deviation of sensors [i.e.,  $\sigma_{Sin}$ Equation 12] is depicted in Figure 12. The performance of the proposed MADCS is relatively steady and also smaller than the other two algorithms. This is due to the fact that the nodes only transmit their data to AUVs directly, which not only saves but also balances the energy consumption of the sensor nodes. There are cluster heads in both LEACH-TSP and DECKS, which inevitably leads to uneven energy consumption.

# 6 Conclusion

In this paper, we investigate a multi-AUV-assisted underwater data collection framework for UASNs. In order to jointly optimize the AoI and energy consumption, we modeled it as a multi-objective MILP. A two-stage scheme MADCS, which consists of the clusterbased AUV scheduling data transmission protocol and the PGSbased trajectory planning algorithm, is proposed to solve the problem. Aiming at balancing and reducing the energy consumption of underwater sensors, the CK clustering algorithm is designed to make clustering more efficient and practical. The AUV scheduling-based transmission protocol is proposed to avoid packet collision and enhance channel utilization. According to the cluster-based network, trajectory for optimizing average AoI and energy consumption of AUV is planned through the proposed PGS algorithm. Simulation results validate that the proposed multi-AUV-assisted underwater data collection scheme outperforms three benchmark algorithms and achieves a proper trade-off between average AoI and energy consumption. We believe that the proposed multi-AUV-assisted underwater data collection scheme has potential application prospects in UASNs with demand for large-scale communication, large system capacity, long-term monitoring, and high data throughput. In future work, collaboration among multiple AUVs and trajectory optimization under real-time ocean currents will be considered.



## Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

# Author contributions

WC: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Validation, Visualization. KC: Writing – review & editing, Methodology, Funding acquisition. EC: Funding acquisition, Supervision, Writing – review & editing.

## Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62271425, U24A20215, and 62071402.

# References

Abd-Elmagid, M. A., and Dhillon, H. S. (2018). Average peak age-of-information minimization in uav-assisted iot networks. *IEEE Trans. Vehicular Technol.* 68, 2003–2008. doi: 10.1109/TVT.25

Abhishek, B., Ranjit, S., Shankar, T., Eappen, G., Sivasankar, P., and Rajesh, A. (2020). Hybrid pso-hsa and pso-ga algorithm for 3d path planning in autonomous uavs. *SN Appl. Sci.* 2, 1–16. doi: 10.1007/s42452-020-03498-0

Al-Habob, A. A., Dobre, O. A., and Poor, H. V. (2021). Age-optimal information gathering in linear underwater networks: A deep reinforcement learning approach. *IEEE Trans. Vehicular Technol.* 70, 13129–13138. doi: 10.1109/TVT.2021.3117536

Anuradha, D., Subramani, N., Khalaf, O. I., Alotaibi, Y., Alghamdi, S., and Rajagopal, M. (20222867). Chaotic search-and-rescue-optimization-based multi-hop data transmission protocol for underwater wireless sensor networks. *Sensors* 22. doi: 10.3390/s22082867

Brekhovskikh, L. M., Lysanov, Y. P., and Lysanov, J. P. (2003). Fundamentals of ocean acoustics (New York, NY, USA: Springer Science & Business Media).

Chaitanya, D. E., Sridevi, C. V., and Rao, G. S. B. (2011). "Path loss analysis of underwater communication systems," in *IEEE technology students' Symposium* (Kharagpur, India: IEEE), 65–70.

Cruz, N. A. (2019). "Optimizing the power budget of hovering auvs," in 2019 IEEE underwater technology (UT) (Kaohsiung, Taiwan: IEEE), 1–6.

de Souza, F. A., Chang, B. S., Brante, G., Souza, R. D., Pellenz, M. E., and Rosas, F. (2016). Optimizing the number of hops and retransmissions for energy efficient multihop underwater acoustic communications. *IEEE Sensors J.* 16, 3927–3938. doi: 10.1109/ JSEN.2016.2532384

EvoLogics (2023). *Evologics acoustic modems technical specifications*. Available online at: https://www.evologics.com/product/s2c-r-18-34-1.

Fang, Z., Wang, J., Jiang, C., Zhang, Q., and Ren, Y. (2021). Aoi-inspired collaborative information collection for auv-assisted internet of underwater things. *IEEE Internet Things J.* 814571. doi: 10.1109/JIOT.2021.3049239

Favaro, F., Brolo, L., Toso, G., Casari, P., and Zorzi, M. (2013). "A study on remote data retrieval strategies in underwater acoustic networks," in 2013 OCEANS-san diego (San Diego, CA, USA: IEEE), 1–8.

Gjanci, P., Petrioli, C., Basagni, S., Phillips, C. A., Bölöni, L., and Turgut, D. (2017). Path finding for maximum value of information in multi-modal underwater wireless sensor networks. *IEEE Trans. Mobile Computing* 17, 404–418. doi: 10.1109/TMC.2017.2706689

Guan, Q., Ji, F., Liu, Y., Yu, H., and Chen, W. (2019). Distance-vector-based opportunistic routing for underwater acoustic sensor networks. *IEEE Internet Things J.* 6, 3831–3839. doi: 10.1109/JIOT.6488907

## **Conflict of interest**

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

# **Generative AI statement**

The author(s) declare that no Generative AI was used in the creation of this manuscript.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Han, G., Gong, A., Wang, H., Martínez-García, M., and Peng, Y. (2021). Multi-auv collaborative data collection algorithm based on q-learning in underwater acoustic sensor networks. *IEEE Trans. Vehicular Technol.* 70, 9294–9305. doi: 10.1109/ TVT.2021.3097084

Han, G., Long, X., Zhu, C., Guizani, M., Bi, Y., and Zhang, W. (2019). An auv location prediction-based data collection scheme for underwater wireless sensor networks. *IEEE Trans. Vehicular Technol.* 68, 6037–6049. doi: 10.1109/TVT.25

Han, G., Shen, S., Song, H., Yang, T., and Zhang, W. (2018). A stratification-based data collection scheme in underwater acoustic sensor networks. *IEEE Trans. Vehicular Technol.* 67, 10671–10682. doi: 10.1109/TVT.2018.2867021

Hao, Z., Li, W., and Zhang, Q. (2023). "Efficient clustering data collection in auvaided underwater sensor network," in *OCEANS 2023-MTS/IEEE US gulf coast* (Biloxi, MS, USA: IEEE), 1–6.

Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. wireless Commun.* 1, 660–670. doi: 10.1109/TWC.2002.804190

Hollinger, G. A., Yerramalli, S., Singh, S., Mitra, U., and Sukhatme, G. S. (2011). "Distributed coordination and data fusion for underwater search," in 2011 IEEE international conference on robotics and automation (Shanghai, China: IEEE), 349–355.

Hou, X., Wang, J., Bai, T., Deng, Y., Ren, Y., and Hanzo, L. (2022). Environment-aware auv trajectory design and resource management for multi-tier underwater computing. *IEEE J. Selected Areas Commun.* 41, 474–490. doi: 10.1109/JSAC.2022.3227103

Hwang, D., and Kim, D. (2008). "Dfr: Directional flooding-based routing protocol for underwater sensor networks," in *OCEANS*, vol. 2008. (Quebec City, QC, Canada: IEEE), 1–7.

Jiang, B., Du, J., Jiang, C., Han, Z., and Debbah, M. (2024). Underwater searching and multiround data collection via auv swarms: An energy-efficient aoi-aware mappo approach. *IEEE Internet Things J.* 11, 12768–12782. doi: 10.1109/ JIOT.2023.3336055

Kaul, S., Yates, R., and Gruteser, M. (2012). "Real-time status: How often should one update?," in *In* 2012 proceedings IEEE INFOCOM (Orlando, FL, USA: IEEE), 2731–2735.

Khan, Z. U., Aman, M., Rahman, W. U., Khan, F., Jamil, T., and Hashim, R. (2023). "Machine learningbased multi-path reliable and energy-efficient routing protocol for underwater wireless sensor networks," in 2023 international conference on frontiers of information technology (FIT) (Islamabad, Pakistan: IEEE), 316–321.

Khan, M. T. R., Jembre, Y. Z., Ahmed, S. H., Seo, J., and Kim, D. (2019). "Data freshness based auv path planning for uwsn in the internet of underwater things," in

2019 IEEE global communications conference (GLOBECOM) (Waikoloa, HI, USA: IEEE), 1–6.

Khan, S. U., Khan, Z. U., Alkhowaiter, M., Khan, J., and Ullah, S. (2024). Energyefficient routing protocols for uwsns: A comprehensive review of taxonomy, challenges, opportunities, future research directions, and machine learning perspectives. *J. King Saud University-Computer Inf. Sci.* 102128. doi: 10.1016/j.jksuci.2024.102128

Li, Y., Huang, H., Zhuang, Y., Zhong, Z., Wang, C.-X., and Wang, X. (2023). An auvassisted data collection scheme for uwsns based on reinforcement learning under the influence of ocean current. *IEEE Sensors J.* 24, 3960–3972. doi: 10.1109/JSEN.2023.3339155

Li, N., Martínez, J.-F., Meneses Chaus, J. M., and Eckert, M. (2016). A survey on underwater acoustic sensor network routing protocols. *Sensors* 16, 414. doi: 10.3390/s16030414

Lin, S., Liu, A., Wang, J., and Kong, X. (2023). An intelligence-based hybrid pso-sa for mobile robot path planning in warehouse. J. Comput. Sci. 67, 101938. doi: 10.1016/j.jocs.2022.101938

Liu, J., Tong, P., Wang, X., Bai, B., and Dai, H. (2020a). Uav-aided data collection for information freshness in wireless sensor networks. *IEEE Trans. Wireless Commun.* 20, 2368–2382. doi: 10.1109/TWC.2020.3041750

Liu, J., Tong, P., Wang, X., Bai, B., and Dai, H. (2020b). Uav-aided data collection for information freshness in wireless sensor networks. *IEEE Trans. Wireless Commun.* 20, 2368–2382. doi: 10.1109/TWC.2020.3041750

Luo, H., Xu, Z., Wang, J., Yang, Y., Ruby, R., and Wu, K. (2022). Reinforcement learning-based adaptive switching scheme for hybrid optical-acoustic auv mobile network. *Wireless Commun. Mobile Computing* 2022. doi: 10.1155/2022/9471698

National Research Council, Division on Earth, Life Studies, Ocean Studies Board and Committee on Exploration of the Seas. (2003). *Exploration of the seas: voyage into the unknown* (Washington, DC, USA: National Academies Press).

Omeke, K. G., Mollel, M. S., Ozturk, M., Ansari, S., Zhang, L., Abbasi, Q. H., et al. (2021). Dekcs: A dynamic clustering protocol to prolong underwater sensor networks. *IEEE Sensors J.* 21, 9457–9464. doi: 10.1109/JSEN.7361

Qiu, T., Zhao, Z., Zhang, T., Chen, C., and Chen, C. P. (2019). Underwater internet of things in smart ocean: System architecture and open issues. *IEEE Trans. Ind. Inf.* 16, 4297–4307. doi: 10.1109/TII.2019.2946618

Rahman, W. U., Gang, Q., Feng, Z., Khan, Z. U., Aman, M., and Ullah, I. (2023). "A q-learning-based multihop energy-efficient and low collision mac protocol for underwater acoustic wireless sensor networks," in *In 2023 20th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Bhurban, Murree, Pakistan, pp. 872–877. doi: 10.1109/IBCAST59916.2023.10712998

Roberge, V., Tarbouchi, M., and Labonté, G. (2012). Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Trans. Ind. Inf.* 9, 132–141. doi: 10.1109/TII.2012.2198665

Stojanovic, M. (2007). On the relationship between capacity and distance in an underwater acoustic communication channel. ACM SIGMOBILE Mobile Computing Commun. Rev. 11, 34–43. doi: 10.1145/1347364.1347373

Su, R., Zhang, D., Li, C., Gong, Z., Venkatesan, R., and Jiang, F. (2019). Localization and data collection in auv-aided underwater sensor networks: Challenges and opportunities. *IEEE Network* 33, 86–93. doi: 10.1109/MNET.2019.1800425

Talak, R., Karaman, S., and Modiano, E. (2018). Optimizing information freshness in wireless networks under general interference constraints. In. *Proc. Eighteenth ACM Int. Symposium Mobile Ad Hoc Networking Computing.*, 61–70. doi: 10.1145/ 3209582

ur Rahman, W., Gang, Q., Feng, Z., Khan, Z. U., Aman, M., and Bilal, M. (2023). "A maca-based energy-efficient mac protocol using q-learning technique for underwater acoustic sensor network," in 2023 IEEE 11th international conference on computer science and network technology (ICCSNT) (Dalian, China: IEEE), 352–355.

Wang, Y., Tian, Z., Sun, Y., Du, X., and Guizani, N. (2020). Preserving location privacy in uasn through collaboration and semantic encapsulation. *IEEE Network* 34, 284–290. doi: 10.1109/MNET.65

Wang, T., Zhao, D., Cai, S., Jia, W., and Liu, A. (2019). Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system. *IEEE Trans. Ind. Inf.* 16, 4791–4799. doi: 10.1109/TII.9424

Xu, J., Zhang, Z., Wang, Z., Wang, J., and Rent, Y. (2024). "Vol and energy-aware auv-assisted data collection for internet of underwater things," in 2024 IEEE wireless communications and networking conference (WCNC) (Dubai, United Arab Emirates: IEEE), 1–6.

Yang, Y., Xiao, Y., and Li, T. (2021). A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability. *IEEE Commun. Surveys Tutorials* 23, 815–841. doi: 10.1109/COMST.2021.3059998

Yu, W., Chen, Y., Wan, L., Zhang, X., Zhu, P., and Xu, X. (2020). An energy optimization clustering scheme for multi-hop underwater acoustic cooperative sensor networks. *IEEE Access* 8, 89171–89184. doi: 10.1109/Access.6287639

Zhang, S., Zhang, H., Han, Z., Poor, H. V., and Song, L. (2020). Age of information in a cellular internet of uavs: Sensing and communication trade-off design. *IEEE Trans. Wireless Commun.* 19, 6578–6592. doi: 10.1109/TWC.7693

Zhao, C., Feng, B., and Zhang, Q. (2021). "Mobile sink leach protocol for underwater acoustic sensor networks," in 2021 13th international conference on wireless communications and signal processing (WCSP) (Changsha, China: IEEE), 1–6.

Zhuo, X., Liu, M., Wei, Y., Yu, G., Qu, F., and Sun, R. (2020). Auv-aided energyefficient data collection in underwater acoustic sensor networks. *IEEE Internet Things J.* 7, 10010–10022. doi: 10.1109/JIoT.6488907

## Appendix 1 Proof of proposition 1

According to Equation 17, the AoI of the *j*th sensor node in the last cluster of the AUV  $v_k$ 's trajectory is:

$$A_{(M^{k}),(j)}^{(k)} = \sum_{i=j}^{\left|\mathcal{J}_{(M^{k})}\right|} \theta_{(M^{k}),(i)} + \lambda_{(M^{k}),(M^{k}+1)}$$

Thus, the overall AoI of sensors in the last cluster of the AUV  $v_k$ 's trajectory is:

$$A_{(M^{k})}^{k} = \sum_{j=1}^{\left|\mathcal{J}_{(M^{k})}\right|} \left(\sum_{i=j}^{\left|\mathcal{J}_{(M^{k})}\right|} \theta_{(M^{k}),(i)} + \lambda_{(M^{k}),(M^{k}+1)}\right) = \sum_{j=1}^{\left|\mathcal{J}_{(M^{k})}\right|} j\theta_{(M^{k}),(j)} + \left|\mathcal{J}_{(M^{k})}\right|\lambda_{(M^{k}),(M^{k}+1)}$$

Likewise, the overall AoI of sensors in the *m*th cluster of the AUV  $v_k$ 's trajectory is:

$$A_{(m)}^{k} = \sum_{j=1}^{|\mathcal{J}_{(m)}|} \left( \sum_{i=j}^{|\mathcal{J}_{(m)}|} \theta_{(m),(i)} + \sum_{i=m}^{M^{k}} \lambda_{(i),(i+1)} + \sum_{l=m+1}^{M^{k}} \sum_{i=1}^{|\mathcal{J}_{(l)}|} \theta_{(l),(i)} \right)$$
$$= |\mathcal{J}_{(m)}| \left( \sum_{l=m+1}^{M^{k}} \psi^{(l)} + \sum_{i=m}^{M^{k}} \lambda_{(i),(i+1)} \right) + \sum_{j=1}^{|\mathcal{J}_{(m)}|} j \theta_{(m),(j)}$$

where  $\Psi^{(l)} = \sum_{i=1}^{|\mathcal{J}_{(l)}|} \theta_{(l),(i)}$  denotes the total amount of uploading time of the sensors in the *l*th cluster visited by  $v_k$ . Then, the above equation can be used to derive the sum AoI of all sensors collected by AUV  $v_k$ :

$$\begin{split} A^{k} &= \sum_{m=1}^{M^{k}} A^{k}_{(m)} \\ &= \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j \theta_{(m),(j)} + \sum_{m=1}^{M^{k}} \left| \mathcal{J}_{(m)} \right| \sum_{l=m+1}^{M^{k}} \psi^{(l)} + \sum_{i=1}^{M^{k}} \left( \sum_{m=1}^{i} \left| \mathcal{J}_{(m)} \right| \right) \lambda_{(i),(i+1)} \\ &= \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j \theta_{(m),(j)} + \sum_{m=2}^{M^{k}} \left( \sum_{l=1}^{m-1} \left| \mathcal{J}_{(l)} \right| \right) \psi^{(m)} + \sum_{i=1}^{M^{k}} \left( \sum_{m=1}^{i} \left| \mathcal{J}_{(m)} \right| \right) \lambda_{(i),(i+1)} \end{split}$$

We further define  $\xi^{(l)} = \sum_{i=1}^{l} |\mathcal{J}_{(i)}|$  as the number of sensors visited from the first cluster to the *l*th cluster in the trajectory of  $\nu_k$ . Thus, we can get:

$$A^{k} = \sum_{m=2}^{M^{k}} \xi^{m-1} \psi^{(m)} + \sum_{i=1}^{M^{k}} \xi^{i} \lambda_{(i),(i+1)} + \sum_{m=1}^{M^{k}} \sum_{j=1}^{|\mathcal{J}_{(m)}|} j \theta_{(m),(j)}$$

As a result, the proof of Proposition 1 is completed.

# Appendix 2 Cluster validity indices

Let  $X = \{x_1, x_2, \dots, x_n\}$  be the dataset containing *n* sensor nodes in a 3D space, and  $x_i \in \mathbb{R}^3$ .  $Z_1, Z_2, \dots, Z_c$  are *c* disjoint subsets of *X*, i.e., clusters.  $n_1, n_2, \dots, n_c$  are the number of sensor nodes that belong to the corresponding cluster, thus  $\sum_{i=1}^{c} n_i = n \cdot o_1, o_2, \dots, o_c$  denote the centroid of  $Z_i$  and O is the global centroid. d(i,j) is defined as the Euclidean distance between node i and j.

1. The silhouette coefficient can be expressed as:

$$SI = \frac{1}{N} \sum_{k=1}^{N} \frac{b(x_i) - a(x_i)}{\max \{a(x_i), b(x_i)\}}$$

where  $a(x_i)$  represents the cohesiveness of the sensor nodes and denotes the average distance among  $x_i$  and other sensor nodes in the same cluster.  $b(x_i)$  represents the average distance from sensor node  $x_i$  to all sensor nodes in another most adjacent cluster:

$$a(x_i) = \frac{1}{n_k - 1} \sum_{x_i, x_j \in \mathbb{Z}_k, i \neq j} d(x_i, x_j)$$

$$b(x_i) = \min_{Z_k \neq Z_l} \frac{1}{|n_l|} \sum_{x_i \in Z_k, x_j \in Z_l} d(x_i, x_j)$$

Observing from the formula, we get that  $SI \in [-1,1]$  and the index closer to 1 refers to a better partition.

2. The Calinski-Harabasz index can be expressed as:

CHI = 
$$\frac{n-c}{c-1} \cdot \frac{\sum_{j=1}^{c} n_j d(o_j, O)^2}{\sum_{j=1}^{c} \sum_{i=1}^{n_j} d(x_i, o_j)^2}$$

where the numerator represents the separation among clusters and the denominator represents the compactness of a cluster. Thus, the larger index refers to a better partition.

3. The Davies-Bouldin index can be expressed as:

$$DBI = \frac{1}{c} \sum_{k,l=1}^{c} \max_{k \neq l} R_{kl}$$

where  $R_{kl}$  represents the similarity among clusters, which is  $R_{kl} = \frac{D_k + D_l}{d(O_k, O_l)}$ .  $D_k$  is the diameter of a cluster, the expression is:

$$D_k = \frac{1}{n_k} \sum_{i=1}^{n_k} d(x_i, o_k)$$

The value range of DBI is  $[0,+\infty]$  and the index closer to 0 refers to a better partition.