Ad Hoc Networks 176 (2025) 103889

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A fire hawk optimizer-based energy-efficient clustering scheme in underwater acoustic sensor networks (UASNs)

Sang-Woong Lee [a], Musaed Alhussein [b], Khursheed Aurangzeb [b], Mohammad Sadegh Yousefpoor [c], Efat Yousefpoor [c], Mehdi Hosseinzadeh [d,e,*]

[a] Pattern Recognition and Machine Learning Lab, School of Computing, Gachon University, Seongnam 13120, Republic of Korea
[b] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh, 11543, Saudi Arabia
[c] Center of Research and Strategic Studies, Lebanese French University, Kurdistan Region, Iraq
[d] School of Computer Science, Duy Tan University, Da Nang, Vietnam
[e] Jadara Research Center, Jadara University, Irbid 21110, Jordan

ARTICLE INFO

Keywords:
Underwater acoustic sensor networks (UASNs)
Clustering
Energy efficiency
Optimization
Artificial intelligence (AI)

ABSTRACT

An underwater acoustic sensor network (UASN) is suitable for gathering data from aquatic environments, including lakes, rivers, seas, and oceans. This network faces several issues due to the distinct features of underwater environments and the limitations of acoustic channels. These challenges include energy limitations, unreliable communication links, and dynamic network topologies. Additionally, the difficulty of recharging or replacing batteries in underwater conditions makes energy optimization essential for prolonging the network lifespan. Currently, many energy-efficient approaches in UASNs emphasize node clustering and multi-hop communication, but most of these methods rely on distributed algorithms. This paper introduces a novel energy-efficient clustering framework called FHOEEC (**F**ire **H**awk **O**ptimization-based **E**nergy-**E**fficient **C**lustering), which integrates both distributed and centralized strategies. The clustering process is divided into three stages: (1) cluster formation, (2) selection of cluster heads, and (3) cluster maintenance. During the periodic neighbor discovery phase, FHOEEC examines two key aspects: the format of the hello packet and its propagation process. FHOEEC aims to create an energy-efficient, cluster-based network structure. To achieve this, the sink node utilizes the fire hawk optimization (FHO) algorithm to decide on the optimal range and number of clusters. To establish these clusters, a fitness function considers a weighted combination of three sub-functions: intra-cluster and inter-cluster distances, the proportion of isolated clusters compared to others, and cluster density. In the final stage, intra-cluster and inter-cluster communication paths are established by focusing on energy balance. This ensures that nodes with energy levels below a specified threshold are excluded from serving as intermediate nodes. Simulation results and performance evaluations show that FHOEEC outperforms three existing clustering methods–CCCS, GTC, and EULC–in terms of energy efficiency and network performance. Therefore, FHOEEC significantly enhances network lifespan, balances energy usage among the nodes, and offers better scalability than other schemes.

1. Introduction

Underwater acoustic sensor networks (UASNs) are self-organizing and multi-hop networks composed of acoustic sensor nodes. These networks represent a specialized form of Internet of Things (IoT) technology for marine environments [1,2]. Since traditional electromagnetic radio waves face some limitations in underwater environments, acoustic waves are preferred for communication in UASNs. UASNs play a vital role in areas such as marine resource management, ocean ecosystem monitoring, and environmental conservation [3,4]. Compared to terrestrial wireless sensor networks, UASNs require a larger number of nodes to continuously collect and transmit data to base stations [5,6]. However, the unique challenges of underwater environments and the difficulty of replacing or recharging batteries limit the nodes' lifespan [7,8]. Once a sensor node's energy is depleted, it can disrupt the normal functioning of the network [9,10]. One of the primary causes of rapid node failure is the issue of "hot spots", where some sensor nodes take a lot of load from the network. This leads to

* Corresponding author.
E-mail addresses: slee@gachon.ac.kr (S.-W. Lee), musaed@ccis.ksu.edu.sa (M. Alhussein), kaurangzeb@ksu.edu.sa (K. Aurangzeb), mohammad.sadegh@lfu.edu.krd (M.S. Yousefpoor), efat.yousefpoor@lfu.edu.krd (E. Yousefpoor), mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).

https://doi.org/10.1016/j.adhoc.2025.103889
Received 11 March 2025; Received in revised form 16 April 2025; Accepted 21 April 2025
Available online 8 May 2025

increased energy consumption and a shorter network lifespan [11,12]. To enhance scalability and extend network lifetime, designing efficient clustering protocols has become an emerging and necessary research topic for UASNs [13,14]. Given the energy constraints inherent to these networks, it is crucial to balance energy consumption across the nodes to maximize data collection and increase the overall lifespan of the network. Clustering has emerged as one of the most important energy-efficient strategies for UASNs [15,16].

In the clustering strategy, cluster heads (CHs) aggregate data packets, which helps to lower the number of transmissions required throughout the network. This strategy minimizes the number of active nodes in data transfer, thereby decreasing energy usage and extending the network lifespan [17,18]. However, when a node repeatedly serves as a CH, it expends more energy than other nodes. It leads to faster energy depletion [19,20]. This imbalance creates an energy hole around the cluster head and reduces the network lifetime. Hence, it is necessary to periodically change the role of CH among different nodes to ensure more balanced energy consumption [21,22]. Traditional clustering techniques typically involve selecting a cluster head through a agreement process among the nodes in the network [23,24]. Once the cluster head is selected, the nodes connect to the nearest cluster and forward their data to this CH, which then cumulates the information and forwards it to the subsequent CH [25,26]. Nevertheless, this clustering strategy encounters three key challenges:

- Choosing cluster heads incurs substantial communication expenses because of the use of control packets.
- Additional communication overhead when the network is dynamic and requires frequent updates to maintain the clusters.
- Cluster heads consume a lot of energy due to their continuous communication with other cluster heads.

Ergo, the existing clustering schemes need to be adapted to the UASN environment and consider the specific limitations inherent in these networks [27,28].

In this paper, a fire hawk optimization-based energy-efficient clustering (FHOEEC) framework is offered for UASNs. The framework combines both distributed and centralized approaches to optimize energy consumption. The clustering process is divided into three phases: (1) cluster creation, (2) selection of cluster heads, and (3) cluster maintenance. The periodic neighbor discovery process in FHOEEC examines two components: the structure of the hello packet and its dissemination scheme. Likewise, the sink node designs a centralized FHO-based cluster formation process. Ultimately, energy-efficient intra-cluster and inter-cluster paths are established. Subsequently, the main contributions of this paper are outlined below.

- FHOEEC emphasizes the structure of a hello message and its dissemination process in the neighbor discovery phase so that each node gathers information about its surrounding nodes to gain awareness of the local network topology.
- FHOEEC employs a centralized FHO-based cluster formation process to decide on the optimal range and number of clusters. This strategy aims to distribute energy usage evenly across the nodes, thereby prolonging the network lifespan. As a result, the sink node takes a set of sensor nodes as input and generates a set of the best clusters.
- In the FHO-based cluster formation phase, each hawk represents a potential set of cluster centers. To determine the best solution, the sink node assesses the fitness of each hawk through a weighted linear combination of three sub-functions: intra-cluster and inter-cluster distances, the proportion of isolated clusters, and cluster density.
- The CH selection process in FHOEEC is decentralized. It determines the priority of each cluster member node based on energy, average distance, and neighborhood degree. The node having the maximum priority accepts the role of CH.
- In FHOEEC, intra-cluster and inter-cluster paths are formed by focusing on the energy balancing issue, such that nodes with energy below the average energy are not selected as intermediate nodes in communication paths.

The organization of the present study is as follows: Section 2 consists of the latest studies in UASNs. Section 3 illustrates a summarized explanation of the fire hawk optimizer (FHO). Section 4 includes network configuration. Section 5 describes the proposed clustering scheme in UASNs. Section 6 compares FHOEEC with three clustering schemes–including CCCS, GTC, and EULC–and assesses the simulation outcomes. Section 7 concludes this paper.

2. Pertinent works

In [29], a centralized control approach is proposed to optimize energy usage in UASNs from a global viewpoint. The authors introduce the centralized control-based clustering technique (CCCS), which adapts the clustering technique based on the density of nodes within UASNs. This method generates intra-cluster controllers within each cluster and optimizes the selection of relay nodes and relay clusters to balance energy consumption and improve routing efficiency. Likewise, sink nodes oversee and facilitate inter-cluster communication across the network. In each cluster, a controller is responsible for assigning the roles of either CH or a cluster member (CM) and keeping communication both within and between the clusters. Selecting CHs depends on the energy consumed by nodes, aiming to create an energy balance by filtering relay nodes and clusters. In CCCS, communication paths are categorized into two types: (1) inter-cluster paths and (2) intra-cluster paths. The sink node optimizes inter-cluster paths, while intra-cluster controllers manage intra-cluster paths. This improves energy consumption in routing. Simulation results show that CCCS effectively enhances energy balance and extends the network lifetime.

In [30], a clustering method inspired by game theory called GTC is suggested to optimize energy consumption in UASNs. Cluster-based routing, as an effective routing method, organizes sensor nodes into several clusters, and each cluster has a cluster head (CH) responsible for collecting and aggregating data sent by other cluster nodes and then forwarding it to the desired node. In GTC, sensor nodes are modeled as independent and rational players. During the selection of CHs, each node aims to maximize its payoff based on the principles of Nash equilibrium. In addition, an incentive system is designed to encourage nodes to make collaborative decisions. Also, the CH role rotates between sensor nodes to balance energy consumption. Additionally, the network is divided into non-uniform areas to guarantee an even distribution of energy usage across cluster heads. The simulation results indicate that GTC effectively balances the energy load within the network and improves its overall lifespan.

In [31], an energy-efficient unequal layering clustering algorithm (EULC) is introduced for UASNs. EULC optimizes energy usage while addressing the energy imbalance issues, such as “hot spots”, by grouping sensor nodes into unequal layers based on their depth. This results in clusters with different sizes within each layer. EULC offers several improvements over previous methods. Firstly, the network is organized into layered based on the depth of nodes and forms clusters in each layer. In addition, the selection of CHs takes into account the remaining energy of each node, its number of connections, and its proximity to the sink node. This helps to uniformly distribute CHs in the network. Secondly, this algorithm adjusts the size of each cluster based on the distance to the sink node. Hence, clusters nearer to the sink are smaller. This modification reduces energy consumption and resolves the issue of ”hot spots”. Thirdly, EULC uses a mix of single-hop and multi-hop routing for communication within and between clusters. Likewise, to conserve energy during inter-cluster communication, the subsequent node is selected according to its available energy and its depth. Simulation results confirm that EULC enhance energy distribution among sensor nodes and enhance the network lifetime.

In [32], a Q-learning-based hierarchical routing protocol with unequal clustering (QHUC) is presented in UASNs. The goal of QHUC is to establish an efficient communication path that enhances the network lifespan. This process first creates a hierarchical network structure, where the whole network is divided into different regions according to their hop distance from the sink. The region closest to the sink node employs a direct transmission method to reduce the isolation problem. Meanwhile, other regions use an unequal clustering strategy to achieve a better energy balance. Furthermore, the integration of Q-learning with unequal clustering helps to distribute the energy usage more evenly throughout the network. A reward function, which considers the network structure and residual energy, is applied to choose the most suitable relay nodes and cluster heads. Unlike traditional greedy approaches, QHUC uses Q-learning to identify an optimal cluster head and the next-hop relay. Likewise, Q-values can be calculated without extra costs and guarantee optimal routing decisions due to the integration of Q-learning and clustering.

In [33], a reliable cluster-based routing protocol (RCRP) is offered for underwater environments. In this clustering strategy, CHs are tasked with forwarding the aggregated data to the sink via multi-hop communication. In RCRP, a Markov chain-based forecast system solves the routing holes during data forwarding. This system identifies void nodes and notifies nearby nodes of these holes. Additionally, this system improves network reliability by preventing nodes with insufficient energy from interfering in transmission processes. Furthermore, RCRP designs a connection prediction mechanism by considering the movement of nodes and the received signal-to-noise ratio (SNR) to deal with changes in the network topology. To optimize data forwarding, a waiting mechanism inspired by the opportunistic routing (OR) model chooses the optimal path based on the probability of successful connections between nodes. The evaluation outcomes confirm that RCRP is energy-efficient and maximizes the efficiency of UASNs.

In [34], an adaptive clustering routing technique named MLAR, which leverages multi-agent reinforcement learning, is suggested for UASNs. In MLAR, nodes work together and employ reinforcement learning to determine the most efficient communication paths. Likewise, MLAR incorporates an adaptive CH selection algorithm to minimize the occurrence of hot spots. This approach avoids adding extra overhead and does not need a consensus between sensor nodes. Instead, each node independently determines its suitability to serve as a CH based on its routing and environmental conditions. In addition, a biased reward function is introduced to evaluate how the adaptive selection of CHs affects routing performance. Simulation results indicate that MLAR provides significant improvements in routing efficiency, energy conservation, and network lifetime.

In [35], a clustering-based depth source selection routing approach (CBDS2R) is presented in UASNs. This approach enhances energy efficiency and extends the network lifetime by focusing on the depth source selection based on link quality between sensor nodes and mobile sinks. The main principle behind CBDS2R is to cluster sensor nodes based on the link quality between adjacent nodes and a mobile sink node. In this approach, data transmission is prioritized based on the good-quality link rather than the highest-quality one. The clustering begins by assigning the sensor nodes closest to the sink into the first cluster. The node that includes a relatively good-quality link, is chosen as “head node”. Additionally, other nodes assess their connection quality relative to this head node, and those with strong connections are included in the second cluster. This procedure continues until sensor nodes join appropriate clusters based on their link quality and proximity to the mobile sink. Each sink node is centrally located within its network environment according to the connection quality. CBDS2R is scalable, and the evaluations clearly highlight the superior efficiency of CBDS2R.

In [36], a distance- and energy-constrained k-means clustering approach (DEKCS) is designed. It decides on CHs by considering both distance and energy constraints. In addition, the energy thresholds of these cluster heads are dynamically adjusted. It guarantees that the network completely depletes its energy before any disconnection. Likewise, the number of clusters is dynamically determined using the elbow method. In DEKCS, the autonomous underwater vehicles (AUVs) gather data from clusters, where CHs are chosen based on the intra-cluster distance and residual energy. In the CH selection procedure, DEKCS employs dynamic energy thresholds to maximize the energy utilization within that network. Also, the underwater channel inversion is applied to estimate the required transmission power to avoid unnecessary energy consumption. The simulation evaluations show the superior performance of DEKCS in comparison with other clustering techniques.

Table 1 outlines the strengths and weaknesses of the methods discussed in this section.

3. Basic concepts

The fire hawk optimizer (FHO) is a nature-inspired algorithm proposed by Azizi et al. FHO mimics the prey-seeking and hunting behavior of each fire hawk in the wild [37]. This algorithm is a suitable option for use in the clustering procedure since it includes different benefits, like simplicity, good convergence speed, lack of parameters, and global search ability. The mathematical framework of the algorithm is briefly explained below.

- **First phase:** An initial population (X), consisting of a group of fire hawks and prey, is randomly generated and expressed in Eq. (1). In FHO, each element in X, like X_p, is a potential solution to the given problem.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \\ \vdots \\ X_P \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \dots x_1^q \dots x_1^d \\ x_2^1 x_2^2 \dots x_2^q \dots x_2^d \\ \vdots \\ x_p^1 x_p^2 \dots x_p^q \dots x_p^d \\ \vdots \\ x_P^1 x_P^2 \dots x_P^j \dots x_P^d \end{bmatrix}, \begin{cases} p = 1, 2, \dots, P. \\ q = 1, 2, \dots, d. \end{cases} \tag{1}$$

$$x_p^q(0) = x_{p,\min}^q + rand.\left(x_{p,\max}^q - x_{p,\min}^q\right), \begin{cases} p = 1, 2, \dots, P. \\ q = 1, 2, \dots, d. \end{cases} \tag{2}$$

so that each potential solution, like X_p, contains d dimensions, and x_p^q indicates its qth dimension. Furthermore, P, $x_p^q(0)$, $x_{p,\min}^q$, $x_{p,\max}^q$, and $rand \in [0,1]$ represent the total number of responses, the initial position of x_p^q, its lower bound, its upper bound, and a random number, respectively.

- **Second phase:** An objective function is defined to assess the responses and divide them into two groups, namely hawks (better responses) and preys (weaker responses), which are stated in Eqs. (3) and (4). Likewise, the best response is selected as the main fire (GB).

$$FH = \begin{bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_l \\ \vdots \\ FH_g \end{bmatrix}, l = 1, 2, \dots, g. \tag{3}$$

$$PR = \begin{bmatrix} PR_1 \\ PR_2 \\ \vdots \\ PR_u \\ \vdots \\ PR_h \end{bmatrix}, u = 1, 2, \dots, h. \tag{4}$$

Here, PR_u, FH_l, h, and g correspond to uth prey, lth hawk, the number of preys, and the number of hawks, respectively.

Table 1
Comparison of the pertinent literature.

Method	Pros	Cons
CCCS [29]	Improving energy efficiency and network lifetime, upgrading packet transmission rate, high compatibility with the UASN environment	Ignoring isolated clusters in the network, low scalability
GTC [30]	Enhancing energy efficiency, improving network lifetime, reducing packet loss rate, decreasing delay, high adaptability with dynamic UASN environment, increasing communication stability, appropriate distribution of nodes in the network	Low scalability, ignoring the isolated clusters in the network
EULC [31]	Increasing energy efficiency, optimizing network lifetime, reducing packet loss rate, suitable adaptability with UASNs, maximizing communication stability, paying attention to the depth information of underwater nodes	Ignoring isolated clusters in the network
QHUC [32]	Improving energy consumption and the network lifetime, optimizing the data transfer process, suitable adaptability to the dynamic environment of UASN, high scalability, enhancing the reliability of communication, paying attention to the number of hops to the sink node, and dividing the network environment.	Ignoring isolated clusters in the network and the local optimum issue, slow convergence speed of the algorithm
RCRP [33]	Optimizing energy consumption and enhancing network life, improving the data transmission process, compatibility with the UASN environment, high scalability, increasing communication reliability, recognizing void nodes and preventing the formation of routing holes, taking into consideration the mobility of nodes in the network	High computational complexity, low convergence speed of the routing algorithm
MLAR [34]	Optimized energy consumption and increased network lifespan, improving data transfer process, high adaptability with the dynamic environment of UASNs, high scalability, communication reliability.	Slow convergence speed of the algorithm, failure to identify void nodes and prevent the formation of routing holes.
CBDS2R [35]	Optimal energy consumption and enhanced network lifetime, high adaptability to UASNs, high scalability, increased communication reliability	Not considering suitable criteria to evaluate the link quality, the probability of being trapped in the void area due to the underwater environmental conditions
DEKCS [36]	Energy efficiency and improved network lifetime, increased data transfer rate, high scalability, enhanced communication reliability	Ignoring the movement of underwater nodes and their depth information, the possibility of being trapped in the void region due to the environmental conditions, ignoring routing holes and data waste, low compatibility with the dynamic environment of UASNs

- **Third phase:** To identify the area controlled by each hawk, FH_l, Eq. (5) calculates the distance from FH_l to each prey like PR_u (i.e., D_u^l), and PR_u is joined to the nearest adjacent hawk.

$$D_u^l = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2}, \begin{cases} l = 1, 2, \dots, g. \\ u = 1, 2, \dots, h \end{cases} \tag{5}$$

  Here, $\left(x_1, y_1\right)$ and $\left(x_2, y_2\right)$ represent the coordinates of FH_l and PR_u within the search domain, respectively.
- **Fourth phase:** Each hawk, such as FH_l, sets fire to its territory by collecting burning woody material from the central fire (GB) or the adjacent hawk's territory (FH_{Near}) to chase the prey and hunt them. By imitating this behavior, the position of FH_l in FHO is updated according to Eq. (6).

$$FH_l^{new} = FH_l + \left(r_1 \times GB - r_2 \times FH_{Near}\right), l = 1, 2, \dots, g. \tag{6}$$

  Here, $r_1, r_2 \in [0, 1]$ are random numbers.
- **Fifth phase:** After a territory is set on fire by FH_l, a prey, for example PR_u, is forced to hide, run away, and move towards the hawk FH_l. By imitating this behavior, the position of PR_u in FHO is updated according to Eq. (7).

$$PR_u^{new} = PR_u + \left(r_3 \times FH_l - r_4 \times SP_l\right), \begin{cases} l = 1, 2, \dots, g. \\ u = 1, 2, \dots, h. \end{cases} \tag{7}$$

  So that SP_l is the territory of FH_l and $r_3, r_4 \in [0, 1]$ are random numbers.
  Additionally, PR_u might escape towards another hawk's territory (FH_{Alter}) after igniting a territory. By imitating this behavior, the position of PR_u in FHO is updated according to Eq. (8).

$$PR_u^{new} = PR_u + \left(r_5 \times FH_{Alter} - r_6 \times SP\right), \begin{cases} l = 1, 2, \dots, g. \\ u = 1, 2, \dots, h. \end{cases} \tag{8}$$

  Here, SP refers to a secure area located outside the boundaries of FH_l's territory. $r_5, r_6 \in [0, 1]$ are random numbers. Eqs. (9) and (10) calculate SP_l and SP.

$$SP_l = \frac{\sum_{u=1}^{h} PR_u}{r}, \begin{cases} u = 1, 2, \dots, h. \\ l = 1, 2, \dots, g. \end{cases} \tag{9}$$

$$SP = \frac{\sum_{u=1}^{h} PR_u}{m}, u = 1, 2, \dots, h. \tag{10}$$

- **Sixth phase:** A condition is defined to stop the FHO algorithm. After this condition is fulfilled, FHO is finished, and GB is returned as the final result.

4. Network settings

In this section, network settings are described in three sub-sections, i.e., network model, channel model, and energy consumption model.

4.1. Network model

Here, a three-dimensional UASN is considered in the form of a cube-shaped region. In this network, there is a sink node with unlimited energy at the center of its water surface. Sensor nodes with constrained energy (i.e., $S = \left\{s_1, s_2, \dots, s_i, \dots, s_N\right\}$ so that N is the number of sensor nodes) are distributed throughout this cube-shaped region. This network model is displayed in Fig. 1. The nodes in this network are connected to an anchor node located on the sea floor through an anchor chain so that the movement of water caused by the ocean current, the movement of vessels, etc., cannot affect this anchor chain. This ensures that the movement pattern of the nodes remains steady over a given time and spatial range. Thus, FHOEEC follows the movement pattern presented in [38]. Accordingly, the network structure can be viewed as relatively unchanged over a certain duration. In this network, the movement prediction-based location algorithm introduced in [38] is employed to obtain location information. Sensor nodes are homogeneous in terms of energy capacity and communication radius. The

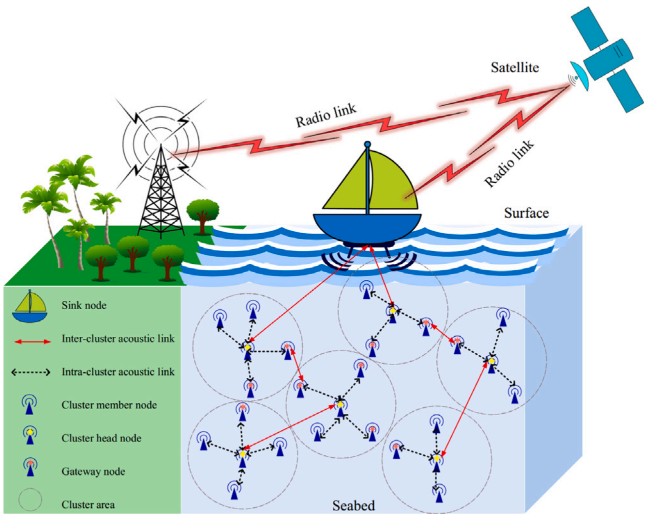

Fig. 1. Network model in FHOEEC.

transmission power of a node like s_i is freely determined according to the communication distance. As a result, the received signal strength can be utilized to measure the distance from the sender to the receiver. Each node, such as s_i, has a unique identifier and can play three roles, namely cluster head (CH), cluster member (CM), and gateway (GT). These roles are described below:

- **Cluster head node (CH):** The cluster head node handles the management of the cluster, the collection of the information about cluster members, the aggregation of this information, and the transmission of the aggregated data to the sink through an inter-cluster path.
- **Cluster member node (CM):** The cluster member node senses and observes its surroundings, and its task is to forward the collected data to the CH node.
- **Gateway node (GT):** Gateway nodes are a type of cluster member node. In addition to carrying out tasks related to each cluster member node, GTs create communication with nearby clusters.

4.2. Channel model

According to the model presented in [39], the high bit error rate (BER) in the channel causes data loss. It is related to the signal-to-noise ratio (SNR) described in Eq. (11).

$$SNR\left(d_{ij}, f\right) = \frac{P/A\left(d_{ij}, f\right)}{N(f)\,\Delta f} \tag{11}$$

Here, P, Δf, $N(f)$, and $A\left(d_{ij}, f\right)$ represent the acoustic signal strength, the noise bandwidth, the power spectral density of noises, and the path loss of the acoustic signal, respectively.

Accordingly, Eq. (12) calculates $A\left(d_{ij}, f\right)$ according to the signal frequency f (in the scale of kHz) and the distance d_{ij} between s_i and s_j.

$$A\left(d_{ij}, f\right) = A_0 {d_{ij}}^k \alpha(f)^{d_{ij}} \tag{12}$$

Here, A_0 means the normalization constant and the diffusion factor k ranges from 1 to 2, where $k = 2$ indicates spherical diffusion, $k = 1$ represents cylindrical diffusion, and $k = 1.5$ denotes experimental diffusion. Likewise, $\alpha(f)$ (in the scale of dB/km) means the absorption coefficient computed via Eq. (13).

$$10 \log \alpha(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\times 10^{-4} f^2 + 0.003 \tag{13}$$

Here, $N(f)$ represents the effective noise level at the frequency f. It denotes the total noise sources in the underwater environment, i.e., turbulence noise ($N_t(f)$), shipping noise ($N_s(f)$), wave movement noise ($N_w(f)$), and thermal noise ($N_{th}(f)$). $N(f)$ is calculated via Eq. (14).

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{14}$$

so that $N_t(f)$, $N_s(f)$, $N_w(f)$ and $N_{th}(f)$ are experimentally estimated according to Eq. (15).

$$\begin{cases} 10\log N_t(f) = 17 - 30\log f \\ 10\log N_s(f) = 40 + 20(s-0.5) + 26\log f - 60\log(f+0.03) \\ 10\log N_w(f) = 50 + 7.5w^{1/2} + 2\log f - 40\log(f+0.4) \\ 10\log N_{th}(f) = -15 + 20\log f \end{cases} \tag{15}$$

So that $s \in [0,1]$ expresses the surface vessel activity frequency and w denotes the wind speed (in the scale of m/s).

Lastly, the average bit error rate (BER) is directly related to SNR and the modulation mode, i.e., $BER\left(d_{ij}, f\right) = \Phi^M\left(SNR\left(d_{ij}, f\right)\right)$. Hence, the data loss rate resulting from the channel ($PLR_{Channel}$) is calculated through Eq. (16).

$$PLR_{Channel} = \Psi^F\left(b, BER\left(d_{ij}, f\right)\right) \tag{16}$$

here, F represents the forward error correction mechanism (FEC) and b denotes the packet length. For simplicity, the binary phase shift keying (BPSK) modulation method [40] is considered. Therefore, $PLR_{Channel}$ can be estimated through Eq. (17).

$$PLR_{Channel} = \left(1 - \frac{1}{2} erfc\sqrt{SNR}\right)^b \tag{17}$$

4.3. Energy model

FHOEEC uses the energy consumption model introduced in [41] to evaluate the energy required by sensor nodes in the data transmission process. Accordingly, Eq. (18) expresses the energy required by a node, like s_i, to send a packet to another node, like s_j.

$$E_{tx}\left(m, d_{ij}\right) = P_{tx}T_{tx} = P_0 A\left(d_{ij}, f\right) T_{tx} \tag{18}$$

where P_{tx} means the transfer power defined based on the distance from s_i to s_j. P_0 indicates the minimum receiving power of sensor nodes. $A\left(d_{ij}, f\right)$ is the power attenuation coefficient that is obtained through Eq. (12). Eq. (19) calculates T_{tx}, which means the time required to forward this packet.

$$T_{tx} = \frac{m}{\lambda} \cdot B(m) \tag{19}$$

Here, m means the number of bits in the data packet or its size; $B(m)$ also represents the bit rate, and λ means the coding efficiency in the scale of bps/Hz.

Likewise, the energy consumed by s_j to get a packet, which includes m bits, is obtained through Eq. (20).

$$E_{rx} = P_{rx}T_{rx} \tag{20}$$

Here, P_{rx} means the receiver power, i.e., the energy needed for receiving one bit. It corresponds to the minimum signal power for receiving this packet. T_{rx} means the receiving time and is obtained via Eq. (21).

$$T_{rx} = \frac{m}{\lambda} \cdot B(m) \tag{21}$$

Additionally, the energy required to aggregate m bits is calculated through Eq. (22).

$$E_{\text{int}\,x} = E_{\text{int}\,x} \times m \tag{22}$$

Here, $E_{\text{int}\,x}$ refers to the energy required to aggregate one bit.

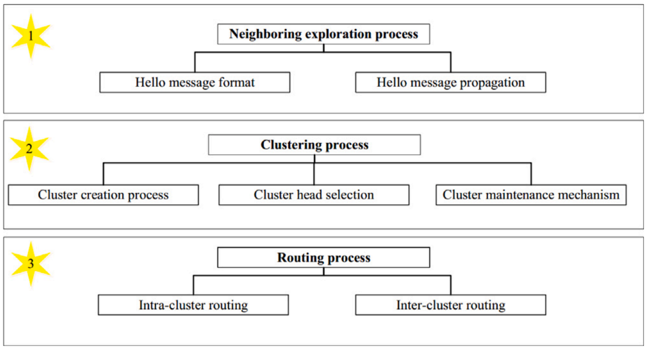

Fig. 2. General framework in FHOEEC.

5. Proposed method

The different steps of FHOEEC are explained in detail. The general framework of this method is shown in Fig. 2. According to this framework, sensor nodes periodically exchange their status information with each other. This information is used to update clusters and find communication paths. Moreover, the clustering process includes three elements: (1) cluster creation, (2) CH selection, and (3) cluster maintenance. Lastly, paths within and between the clusters are formed in the network. In the following, these three processes, including neighbor discovery, clustering, and routing are described in detail.

5.1. Neighbor discovery process

To implement the clustering and routing processes in UASNs, each underwater node, like s_i, gathers information about its surrounding nodes to gain awareness of the local network topology. Thus, s_i performs a periodic neighbor discovery process and stores the collected information in its neighboring table, i.e., NT_i. In this process, FHOEEC examines two steps, namely the format of hello packet and its dissemination process.

5.1.1. Hello message format

Here, each sensor node, like s_i, exchanges hello messages to detect its nearby nodes. As shown in Fig. 3, a hello message consists of two main parts: packet information and routing information. The first part contains information about the hello message, and the second part includes information about the node that transmits this message. This information is used in the clustering and routing processes.

In the following, the details related to the different fields of the hello packet are explained.

- **Packet type:** When this field is equal to one, it means that this packet is a hello message, and thus it is separated from other packets exchanged in the network, such as data packets and ACK packets.
- **Packet ID:** This field is used to identify repeated messages in the network.
- **Dissemination period:** This field is related to the packet information part and includes the dissemination period of the hello packet. In FHOEEC, it is a fixed time interval chosen based on the conditions of UASNs. When the network is highly dynamic and many changes are made in the network topology, this dissemination period should be chosen as a small number so that the neighboring tables are updated on time and the changes made in the network topology are accurately recorded. On the other hand, if few changes are made in the network topology, the dissemination period should be larger to reduce the communication overhead and the energy consumption in the network.
- **Source identification number:** This field refers to the ID of s_i that transmits the hello message to its neighbors in the network.

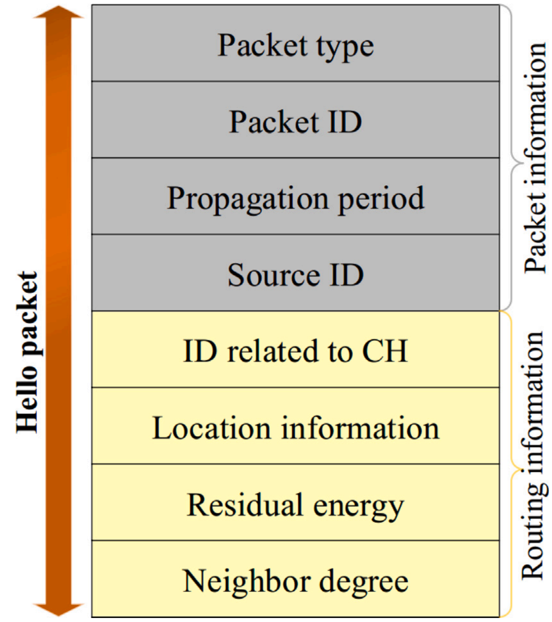

Fig. 3. Hello packet format.

- **Cluster head ID:** This ID is related to the cluster head which is connected to s_i.
- **Location information:** This field specifies the coordinates of s_i, i.e., $Loc_i = (x_i, y_i, z_i)$ in the network.
- **Residual energy:** This field is used to determine the remaining energy of s_i (i.e., E_i).
- **Neighboring degree:** This field indicates the number of nearby nodes of s_i (NB_i).

5.1.2. Hello dissemination process

For performing clustering and routing processes in UASNs, every node, like s_i, needs to collect information about its adjacent nodes to gain awareness of the local network topology. Consequently, s_i periodically generates a hello message according to the format introduced in Fig. 3 and broadcasts it to its nearby nodes in the network, which shown in Fig. 4. No node is allowed to replay this message in the network and is only permitted to extract the information from this message to complete its own neighbor table, i.e., NT_i. The format of the table is illustrated in Table 2. According to this format, the entry lifetime (EL_j) means the validity period of a neighbor node in NT_i so that this node will be removed from NT_i when EL_j is expired. To refresh or register information about a neighboring node, such as s_j, in NT_i, s_i and s_j need to exchange hello messages with each other and act according to the following pattern. Algorithm 1 demonstrates the pseudocode related to this process.

- If s_i has old information about s_j in NT_i and obtains a new hello message from it, s_i replaces the old information s_j in NT_i with the information extracted from this new message and updates the entry lifetime (EL_j) again.
- If s_i does not have any information about s_j in NT_i and receives a hello message from it, s_i considers a new entry in NT_i, and records the information extracted from this message in this new entry and adjusts EL_j.
- If EL_j expires and s_i does not receive any hello packets from its old neighbor, s_j, in the network, then, s_i deletes its information from NT_i.

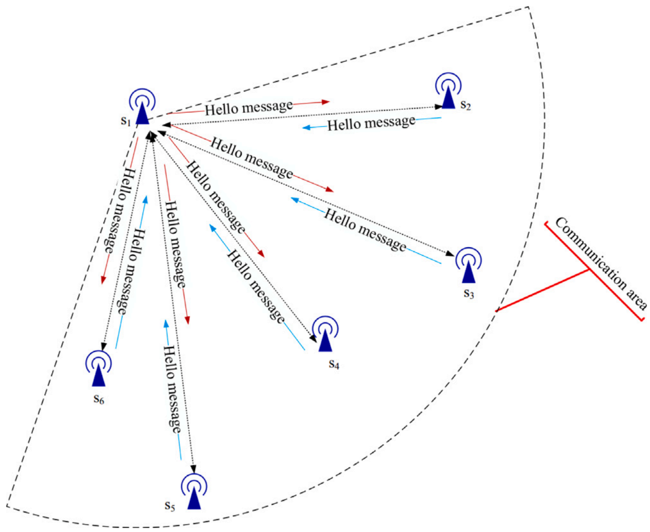

Fig. 4. The neighbor discovery process related to s_1.

Algorithm 1 Neighboring exploration process

Input: s_i: i-th sensor node available in $S = \{s_1, s_2, ..., s_i, ..., s_N\}$
ID_i: ID corresponding to s_i
ID_{CH_k}: ID related to CH_k
$Loc_i = (x_i, y_i, z_i)$: Position information of s_i
E_i: Remaining energy of s_i
NB_i: Neighboring degree of s_i
ID_{Hello}: ID related to the hello packet
PP: The propagation period of hello packets in the network.
T_{net}: A timer for counting the network time.
Output: NT_i: Neighbor table related to s_i

```
Begin
for ∀ s_i ∈ S do
    if (T_net mod PP) = 0 then
        s_i: Prepare a hello packet according to the format illustrated in Fig. 3;
        s_i: Propagate the hello packet for all sensor nodes into its communication range;
    end if
end for
for ∀ sensor node such as s_j into the communication rang of s_i do
    if s_j dispatches a hello packet to s_i and s_j ∈ NT_i then
        s_i: Extract the information of this hello packet;
        s_i: Replace the new information with the old one of s_j in NT_i;
        s_i: Reset the entry lifetime of s_j (EL_j) in NT_i;
    else if s_j dispatches a hello packet to s_i and s_j ∉ NT_i then
        s_i: Add a new entry to NT_i;
        s_i: Extract the information of this hello packet;
        s_i: Insert the information of s_j into NT_i;
        s_i: Reset the entry lifetime of s_j (EL_j) in NT_i;
    else if s_j does not dispatch a hello packet to s_i and s_j ∈ NT_i and EL_j = 0 then
        s_i: Remove s_j from NT_i;
    end if
end for
End
```

5.2. Clustering process

Here, FHOEEC seeks to find an energy-efficient cluster-based hierarchical network structure. In this regard, the sink node should specify the range and number of clusters in a centralized manner. Then, nodes in a cluster find the best cluster head node in their cluster with a distributed method. As a result, the clustering process includes three important subsections.

- Cluster construction
- Cluster head selection
- Cluster maintenance

5.2.1. Cluster creation process

In FHOEEC, the sink node finds the optimal range and number of clusters using the fire hawk algorithm (FHO). The cluster creation process balances the energy used by the nodes and increases the network longevity. In FHOEEC, every node in the cluster periodically serves as CH in UASN. To achieve this goal, a common hypothesis is that the sink has access to the spatial coordinates of the nodes in the network. Hence, it can decide on the best clusters in the network. As a result, the sink takes a set of nodes and executes the clustering process to produce a set of the best clusters, i.e., $C = \{c_1, c_2, \ldots, c_f, \ldots, c_M\}$, so that M represents the number of clusters. If this parameter is not chosen correctly, an uneven energy distribution among the nodes will drastically lower the network lifetime. If the sink node assigns a small value to the parameter M, the size of each cluster grows, the probability of early death of CHs is high due to imposing high intra-cluster routing overhead, and the clustering structure must be rebuilt quickly. Whereas, if the sink node allocates a large value to M, the size of the clusters is reduced. In this case, the inter-cluster routing overhead will increase significantly. To select M correctly, the sink node calculates the best clusters in $C = \{c_1, c_2, \ldots, c_f, \ldots, c_M\}$ in such a way that each sensor node in $S = \{s_1, s_2, \ldots, s_i, \ldots, s_N\}$ becomes a member of a cluster. This membership is evaluated according to the distance to the cluster center. Thus, if the Euclidean distance of s_i to the cluster center c_f is less than its communication radius, r (i.e., $d_{if} \le r$), then s_i can be a member of the cluster c_f. Likewise, if s_i can connect to more than one cluster, s_i becomes the member of the closest cluster center. Thus, the nodes' inclusion in clusters is determined based on a criterion called coverage level, i.e., cm_{if}. For each cluster center, like c_f, if $d_{if} \le r$, then $cm_{if} = 1$; otherwise, $cm_{if} = 0$. According to this definition, cm_{if} is obtained through Eq. (23).

$$cm_{if} = \begin{cases} 1, & if\ s_i\ lies\ within\ the\ range\ of\ c_f \\ 0 & ,\qquad otherwise \end{cases} \tag{23}$$

Consequently, the sink node must consider two important conditions (Eqs. (24) and (25)) to determine the optimal number of cluster centers, i.e., M. First, each node in $S = \{s_1, s_2, \ldots, s_i, \ldots, s_N\}$ must belong to a cluster in $C = \{c_1, c_2, \ldots, c_f, \ldots, c_M\}$. Second, s_i is only allowed to be a member of one cluster in the network.

$$\sum_{\forall\ s_i \in S} cm_{if} = N - M \tag{24}$$

$$\sum_{\forall\ c_f \in C} cm_{if} = 1, \qquad \forall\ s_i \in S \tag{25}$$

The FHO-based cluster creation procedure is described below.

- **Phase 1:** First, the sink node allocates the values related to the initial parameters of the fire hawk algorithm, namely the number of initial population ($\mathbb{C}$), the number of hawks (g), the number of preys ($h = P - g$), and the stopping condition ($\max_{iter}$).
- **Phase 2:** The sink node is responsible for initializing the population (a set of hawks and preys) randomly, so that each hawk plays the role of a solution to the problem, i.e., a set of cluster centers ($C = \{c_1, c_2, \ldots, c_f, \ldots, c_M\}$). The initial population is defined in Eq. (26).

$$\mathbb{C} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_p \\ \vdots \\ C_P \end{bmatrix} = \begin{bmatrix} c_1^1 c_1^2 \ldots c_1^f \ldots c_1^M \\ c_2^1 c_2^2 \ldots c_2^f \ldots c_2^M \\ \vdots \\ c_p^1 c_p^2 \ldots c_p^f \ldots c_p^M \\ \vdots \\ c_P^1 c_P^2 \ldots c_P^f \ldots c_P^M \end{bmatrix}, \begin{cases} p = 1, 2, \ldots, P. \\ f = 1, 2, \ldots, M. \end{cases} \tag{26}$$

$$c_p^f(0) = c_{p,\min}^f + rand.\left(c_{p,\max}^f - c_{p,\min}^f\right), \begin{cases} p = 1, 2, \ldots, P. \\ f = 1, 2, \ldots, M. \end{cases} \tag{27}$$

Here, C_p means the pth solution, which has M dimensions (the number of cluster centers), and c_p^f indicates the cluster center c_f in the solution C_p. Also, P is the initial population size. $c_p^f(0)$, $c_{p,\min}^f$, and $c_{p,\max}^f$ represent the initial position, lower and upper bounds of c_p^f, respectively, and $rand$ is a random number in $[0, 1]$.

Table 2
The structure of the neighbor table.

Identifier	ID related to CH	Location information	Residual energy	Neighboring degree	Entry lifetime
ID_j	ID_{CH}	Loc_j	E_j	NB_j	EL_j
…	…	…	…	…	…
…	…	…	…	…	…

- **Phase 3:** Now, the sink node evaluates the fitness value of each hawk, like C_p in $\mathbb{C}$ using Eq. (28) and determines the best solution or the main fire (GB). Then, it separates the hawks (Eq. (3)) and preys (Eq. (4)) based on the fitness value of the solutions in $\mathbb{C}$.

$$F\left(C_p\right) = \eta_1 f_1 + \eta_2 f_2 + \eta_3 f_3 \tag{28}$$

Here, the fitness function is the weighted linear combination of three sub-functions, namely f_1, f_2, and f_3, so that f_1 is responsible for examining two factors, i.e., intra-cluster and inter-cluster distances; f_2 measures the ratio of isolated clusters to others; and f_3 evaluates the density of clusters. The weight coefficients, η_1, η_2, and η_3 express the influence of these coefficients on $F\left(C_p\right)$, so that if one weight coefficient has a larger value than the others, then the corresponding sub-function has the greatest effect on $F\left(C_p\right)$. When determining these weight coefficients, note that $\sum_{i=1}^{3} \eta_i = 1$. Calculating the exact values of these weight coefficients is beyond the scope of this paper, and FHOEEC assumes that $\eta_1 = \eta_2 = \eta_3 = \frac{1}{3}$.

  - f_1: This sub-function, which is stated in Eq. (29), is responsible for checking the two factors of intra-cluster and inter-cluster distances. The ideal situation in the response $C_p = \left[c_p^1 c_p^2 \dots c_p^f \dots c_p^M\right]$ is to minimize the distance of each cluster center, like c_p^f (where $f = 1, \dots, M$), to its cluster member nodes. In this case, C_p can reduce the intra-cluster transmission distance and prevent energy loss in the data transfer process. On the other hand, the favorable status in $C_p = \left[c_p^1 c_p^2 \dots c_p^f \dots c_p^M\right]$ is to maximize the distance of both cluster centers, such as c_p^f and c_p^q (so that $c_p^f, c_p^q \in C_p$). In this case, C_p guarantees that the selected cluster centers are distributed throughout the network and are easily accessible to nodes in different regions of the network.

$$f_1 = \Upsilon\left(1 - \text{intra}-\text{dis}\right) + (1 - \Upsilon)\,\text{inter}-\text{dis} \tag{29}$$

$$\text{intra}-\text{dis} = \frac{1}{M}\left(\sum_{f=1}^{M}\left(\frac{1}{\left|c_p^f\right|}\sum_{\forall\, CM_j^f \in c_p^f} \frac{d\left(CM_j^f, c_p^f\right)}{\max\limits_{\substack{\forall\, CM_j^f \in c_p^f \\ AND \\ \forall\, c_p^f \in C_p}} \left\{d\left(CM_j^f, c_p^f\right)\right\}}\right)\right) \tag{30}$$

$$\text{inter}-\text{dis} = \frac{2}{M(M+1)} \sum_{q=1}^{M}\sum_{f=q+1}^{M} \frac{d\left(c_p^q, c_p^f\right)}{\max\limits_{\forall\, c_p^f \text{ and } c_p^q \in C_p}\left\{d\left(c_p^q, c_p^f\right)\right\}} \tag{31}$$

Here, $\Upsilon \in [0,1]$ is a weight coefficient and $\Upsilon = \frac{1}{2}$. Moreover, M and $\left|c_p^f\right|$ are the number of clusters and the number of CMs in c_p^f, respectively. $d\left(c_p^q, c_p^f\right) = \sqrt{\left(x_q - x_f\right)^2 + \left(y_q - y_f\right)^2 + \left(z_q - z_f\right)^2}$ denotes the distance between two cluster centers, c_p^f and c_p^q, and $d\left(CM_j^f, c_p^f\right) = \sqrt{\left(x_j - x_f\right)^2 + \left(y_j - y_f\right)^2 + \left(z_j - z_f\right)^2}$ indicates the distance between c_p^f and the cluster member node CM_j^f. $\left(x_q, y_q, z_q\right)$, $\left(x_f, y_f, z_f\right)$, and $\left(x_j, y_j, z_j\right)$ indicate the locations of c_p^q, c_p^f, and CM_j^f, respectively.

  - f_2: This sub-function, which is shown in Eq. (32), is responsible for evaluating isolated clusters (i.e., clusters with one or two members). In $C_p = \left[c_p^1 c_p^2 \dots c_p^f \dots c_p^M\right]$, the desirable situation is that the clustering process produces the least number of isolated clusters. If the sink node allocates a large value to the parameter M, the size of the clusters is small and the probability of forming isolated clusters is high. In this case, the inter-cluster overhead will be increased.

$$f_2 = e^{-\left(\frac{M_{isolated}}{M}\right)} \tag{32}$$

Here, M and $M_{isolated}$ are the number of clusters and the number of isolated clusters, respectively.

  - f_3: This sub-function, which is presented in Eq. (33), is responsible for evaluating the intra-cluster density. In $C_p = \left[c_p^1 c_p^2 \dots c_p^f \dots c_p^M\right]$, the desired situation is that the density of clusters established in the network is almost uniform, i.e., $C_{opt} = \frac{N}{M}$, so that N and M indicate the number of nodes and the number of clusters in the network, respectively. In this case, C_p balances the intra-cluster routing overhead related to CHs and manages their energy usage to enhance the network lifespan.

$$f_3 = e^{-\left(\frac{1}{M}\right)\sum_{f=1}^{M}\left(\frac{\left|C_{opt} - \left|c_p^f\right|\right|}{\max\limits_{\forall\, c_p^f \in C_p}\left\{\left|C_{opt} - \left|c_p^f\right|\right|\right\}}\right)} \tag{33}$$

So that $\left|c_f^p\right|$ is the size of the cluster c_p^f.

- **Phase 4:** Now, the sink node defines the territory of the hawks and the preys based on Eq. (5).
- **Phase 5:** The sink node utilizes Eq. (6) to recalculate the position of fire hawks and Eqs. (7) and (8) to update the new position of the prey.
- **Phase 6:** Finally, the sink node evaluates the stopping condition ($\max_{iter}$) to terminate the FHO algorithm. When this condition is fulfilled, the FHO algorithm will return the best clustering solution, i.e., $C = \left\{c_1, c_2, \dots, c_f, \dots, c_M\right\}$, and then the sink node will send this result to the sensor nodes in UASN. Ultimately, each node is connected to the nearest cluster center. Also, the nodes located in the range of several cluster centers are known as gateway nodes and play the role of intermediate nodes to communicate with adjacent clusters.

Algorithm 2 illustrates the pseudocode related to this process.

5.2.2. Cluster head selection process

In FHOEEC, each cluster member in c_f such as CM_j^f must decide on its own cluster head, CH_f, based on a distributed CH selection process. In this way, CM_j^f obtains its priority (P_j^f) to become the cluster head through Eq. (34), and inserts it in the hello message to share P_j^f with other members in the cluster c_f. Finally, one of the members with the maximum priority serves as CH.

$$P_j^f = \psi_1 Eng_j + \psi_2 Dis_j + \psi_3 Ngh_j \tag{34}$$

where ψ_1, ψ_2, and ψ_3 denote the weights to specify the effect of energy (Eng_j), average distance (Dis_j), and connection degree (Ngh_j) on P_j^f so that $\sum_{k=1}^{3} \psi_k = 1$. FHOECC assumes that $\psi_1 = \psi_2 = \psi_3 = \frac{1}{3}$.

Furthermore, Eng_j is calculated through Eq. (35).

$$Eng_j = \begin{cases} 1 \quad , & E_j \geq \overline{E_f} \\ e^{E_j - \overline{E_f}} \;, & E_j < \overline{E_f} \end{cases} \tag{35}$$

Algorithm 2 FHO-based cluster creation process

Input: s_i: i-th sensor node available in $S = \{s_1, s_2, ..., s_i, ..., s_N\}$
ID_i: ID corresponding to s_i
r: Communication radius of sensor nodes
$Loc_i = (x_i, y_i, z_i)$: Position information of s_i
E_i: Remaining energy of s_i
NB_i: Neighboring degree of s_i
M: Optimal number of clusters in the network.
T_{net}: A timer for counting the network time.
Output: $C = \{c_1, c_2, ..., c_f, ..., c_M\}$: The set of clusters in the network.
Begin
Sink: Calculates the optimal number of clusters (M) based on Eqs. (23), (24), and (25);
Sink: Demonstrate the FHO parameters, the initial population $\mathbb{C}$, number of fire hawks g, number of preys ($h = \mathbb{C} - g$), and stop condition $\max_{iter}$;
Sink: Determine the initial population using Eqs. (26) and (27) randomly;
while $iter \leq \max_{iter}$ **do**
  for $\forall\, C_p \in \mathbb{C}$ **do**
    Sink: Obtain the fitness value of C_p(i.e. $F(C_p)$) from Eq. (28);
    Sink: Classify the solutions in $\mathbb{C}$ into two groups, fire hawks and preys, using Eqs. (3) and (4);
    Sink: Obtain the distance between fire hawks and preys according to Eq. (5);
    Sink: Calculate the territory of fire hawks;
    Sink: Re-calculate the new position of fire hawks according to Eq. (6);
    Sink: Re-calculate the new position of preys according to Eqs. (7) and (8);
    Sink: Sort fire hawks and preys according to their fitness values;
    Sink: Specify the best solution i.e. the main fire (GB);
  end for
  Sink: $iter = iter + 1$;
end while
Sink: Return $C = \{c_1, c_2, ..., c_f, ..., c_M\}$;
for $\forall\, s_i \in S$ **do**
  $\mathbf{s_i}$**:** Connect to the nearest cluster in $C = \{c_1, c_2, ..., c_f, ..., c_M\}$;
  if s_i is the cluster range of different clusters such as c_1 and c_2 **then**
    $\mathbf{s_i}$**:** Play the role of the gateway node (GT) between c_1 and c_2;
  end if
end for
End

so that E_j means the remaining energy of CM_j^f. $\overline{E_f}$ is the average remaining energy of the members in the cluster c_f.

$$\overline{E_f} = \frac{1}{|c_f|} \sum_{\forall\, CM_k^f \in c_f} E_k \tag{36}$$

where $|c_f|$ represents the size of the cluster c_f. Also, Dis_j is obtained through Eq. (37).

$$Dis_j = \begin{cases} 1 \quad , & d_j \leq \overline{d_f} \\ e^{-\left(d_j - \overline{d_f}\right)} \ , & d_j > \overline{d_f} \end{cases} \tag{37}$$

where d_j means the average distance from CM_j^f to its neighbors and is calculated through Eq. (38).

$$d_j = \frac{1}{NB_j} \left(\sum_{\forall\, s_q \in NT_j} \sqrt{(x_j - x_q)^2 + (y_j - y_q)^2 + (z_j - z_q)^2} \right) \tag{38}$$

Additionally, NB_j shows the number of its adjacent nodes in the neighbor table NT_j. (x_j, y_j, z_j) and (x_q, y_q, z_q) are two spatial coordinates of CM_j^f and its neighbor, s_q.

$$\overline{d_f} = \frac{1}{|c_f|} \sum_{\forall\, CM_k^f \in c_f} d_k \tag{39}$$

Also, Ngh_j is obtained through Eq. (40).

$$Ngh_j = \begin{cases} 1 \quad , & NB_j \geq \overline{NB_f} \\ e^{NB_j - \overline{NB_f}} \ , & NB_j < \overline{NB_f} \end{cases} \tag{40}$$

so that NB_j displays the number of adjacent nodes of CM_j^f in NT_j. $\overline{NB_f}$ is the average neighbor degree of CMs in the cluster c_f.

$$\overline{NB_f} = \frac{1}{|c_f|} \sum_{\forall\, CM_k^f \in c_f} NB_k \tag{41}$$

Algorithm 3 demonstrates the pseudo-code related to the CH selection procedure.

Algorithm 3 CH election process

Input: CM_j^f: j-th cluster member in the cluster c_f.
$C = \{c_1, c_2, ..., c_f, ..., c_M\}$: The set of clusters in the network.
Output: CH_f: Cluster head in c_f
Begin
for $\forall\, CM_j^f \in c_f$ **do**
  $\mathbf{CM_j^f}$**:** Calculate its priority P_j^f using Eq. (34);
  $\mathbf{CM_j^f}$**:** Record P_j^f in the hello packet;
  $\mathbf{CM_j^f}$**:** Share the hello packet with other cluster members in c_f;
end for
for $\forall\, CM_j^f \in c_f$ **do**
  if CM_j^f has the highest priority in c_f **then**
    $\mathbf{CM_j^f}$**:** Accept the role of CH_f in c_f;
  else
    $\mathbf{CM_j^f}$**:** Play the role of a cluster member in c_f;
  end if
end for
End

5.2.3. Cluster maintenance process

This process consists of three steps: connecting a new node, controlling the clusters, and changing the cluster head. Algorithm 4 describes the pseudo-code of the phase.

- **Connecting a new node:** A new node such as s' transmits a membership request to its nearby clusters in the network. Upon receiving this request, the adjacent cluster head nodes respond to this request and s' connects to the nearest cluster.
- **Controlling the clusters:** In a cluster like c_f, the corresponding cluster head, CH_f, and the cluster member node, CM_j^f, periodically transfer hello messages to each other. If the exchange of this message is successful, this means that the connection between CH_f and CM_j^f is still valid, and the membership of CM_j^f in c_f is confirmed. If the exchange of this message is unsuccessful, it means that the connection between CH_f and CM_j^f is broken and the membership of CM_j^f in c_f is invalid. As a result, CM_j^f must transmit a membership request to a new cluster.
- **Changing the cluster head:** In FHOEEC, the CH role in the clusters rotates periodically to provide the energy balance issue in UASNs. If the cluster head change period arrives, each sensor node belonging to c_f, such as CM_j^f, must decide on a new cluster head, i.e., CH_f. In this way, CM_j^f obtains its priority (P_j^f) to become cluster head through Eq. (35), and adds it in the hello message. Finally, one of the members with the highest priority accepts the role of cluster head.

5.3. Routing process

After deciding on cluster heads, each cluster member node, like CM_j^f, transmits its collected data to the cluster head, CH_f. In the cluster c_f, CM_j^f forwards data packets to CH_f through the shortest path. To achieve energy balance in the cluster, intermediate nodes in routing paths are not chosen if their energy is below the average.

- **Intra-cluster route construction:** For creating a route from CM_j^f to CH_f, if the distance between them is less than their communication radius, CM_j^f sends its data packets directly to CH_f. Otherwise, CM_j^f selects the closest cluster member node to CH_f, which has more energy than the average energy (Eq. (36)), as the next-hop node and forwards its data to it. Refer to the pseudo-code of the procedure in Algorithm 5.

Algorithm 4 Cluster maintenance process

Input: $C = \{c_1, c_2, ..., c_f, ..., c_M\}$
CH_f: Cluster head within the cluster c_f
CM_j^f: Cluster member node within the cluster c_f
Output: Refreshing the cluster members
Begin
1: **if** s' enters to the network **then**
2: **s′:** Disseminate a connection request into its adjacent clusters;
3: **for** ∀ Cluster head such as CH_f receives this request **do**
4: $\mathbf{CH_f}$**:** Answer to this request;
5: **end for**
6: **s′:** Connect to the nearest CH;
7: **end if**
8: **for** $\forall\ c_f \in C$ **do**
9: **for** $\forall\ CM_j^f \in c_f$ **do**
10: $\mathbf{CH_f}$**:** Exchange its hello packet with CM_j^f periodically;
11: $\mathbf{CM_j^f}$**:** Exchange its hello packet with its cluster head CH_f periodically;
12: **if** lines 10 and 11 successfully are run **then**
13: $\mathbf{CH_f}$**:** Connection between CH_f and CM_j^f is valid;
14: **else**
15: $\mathbf{CH_f}$**:** Remove CM_j^f from the list of cluster members;
16: $\mathbf{CM_j^f}$**:** Disseminate a connection request message to its adjacent clusters;
17: **if** ∀ Cluster head such as CH_k receives this request **then**
18: $\mathbf{CH_k}$**:** Answer to this request;
19: **end if**
20: $\mathbf{CM_j^f}$**:** Connect to the nearest CH;
21: **end if**
22: **end for**
23: **end for**
24: **if** The period of CH change reaches **then**
25: $\mathbf{CM_j^f}$**: Call** Algorithm 3;
26: **end if**
End

Algorithm 5 Intra-cluster path construction

Input: CH_f: Cluster head in c_f
CM_j^f: j-th cluster member in c_f
Output: Constructing an path from CM_j^f to CH_f
Begin
1: **if** distance between CM_j^f and CH_f is smaller than r **then**
2: $\mathbf{CM_j^f}$**:** Send its data packets to CH_f using a one-hop path;
3: **else**
4: $\mathbf{CM_j^f}$**:** Decide on the next forwarder like CM_k^f based on the nearest distance to CH_f and the energy (Eq. (36));
5: $\mathbf{CM_j^f}$**:** Send its data packets to CM_k^f using a multi-hop path;
6: $\mathbf{CM_k^f}$**: Go to** line 1;
7: **end if**
End

- **Inter-cluster route construction:** For creating the route from CH_k to CH_f, if the distance between them is less than their communication radius, CH_k forwards its data packets directly to CH_f. Otherwise, CH_k chooses the node closest to CH_f, which has more energy than the average energy, as the next-hop node and forward its data to it. The pseudo-code of the procedure is presented in Algorithm 6.

Algorithm 6 Inter-cluster path construction

Input: CH_f: Cluster head corresponding to c_f
CH_k: Cluster head corresponding to c_k
Output: Constructing an inter-cluster path from CH_k to CH_f
Begin
1: **if** distance between CH_k and CH_f is smaller than r **then**
2: $\mathbf{CH_k}$**:** Send its data packets to CH_f using a one-hop path;
3: **else**
4: $\mathbf{CH_k}$**:** Decide on the next forwarder among its adjacent CHs and gateways (such as CH_l) based on the nearest distance to CH_f and the highest energy;
5: $\mathbf{CH_k}$**:** Send its data packets to CH_l using a multi-hop path;
6: $\mathbf{CH_l}$**: Go to** line 1;
7: **end if**
End

Table 3
Simulation parameters.

Parameter	Value
Simulation tool	NS3
Compared methods	FHOEEC, CCCS, GTC, and EULC
Simulation environment	$100 \times 100 \times 100\ \text{m}^3$
Number of sensor nodes	100
The location coordinates of the sink	(50, 50, 100)
Initial energy of nodes	0.5 J
Communication radius of nodes	20 m
Data packet size	2000 bits
Hello packet size	200 bits
E_{rx}	50 nJ/bit
$E_{\text{int } x}$	5 nJ/bit
K	1.5
P_0	3 mW
f	10 kHz

6. Simulation and evaluation of results

This section evaluates the efficiency and performance of FHOEEC in accordance with the network simulation tool, version 3, (NS3) [42] and compares its results with three methods, namely CCCS [29], GTC [30], and EULC [31]. The reasoning behind selecting these methods is that FHOEEC, CCCS, and GTC are centralized clustering approaches. However, FHOEEC selects CHs in a distributed manner. In contrast, EULC is a distributed clustering method. Additionally, FHOEEC, CCCS, GTC, and EULC focus on energy efficiency and energy balance in UASN. Table 3 presents the simulation parameters.

6.1. Energy consumption and the network lifetime

Fig. 5 displays the residual energy of nodes across different simulation rounds. In all clustering approaches, the average remaining energy of nodes lowers as the simulation rounds grow. Nevertheless, the energy reduction rate of sensor nodes under FHOEEC is significantly lower than in other methods. On average, FHOEEC can improve the average residual energy of nodes by 19.81%, 40.91%, and 87.87% compared to CCCS, GTC, and EULC, respectively. Additionally, Fig. 6 illustrates the energy consumption of nodes across various simulation rounds. This evaluation indicates that energy consumption in centralized clustering methods, namely FHOEEC, CCCS, and GTC, is lower than that in the distributed clustering method, EULC. Furthermore, the average energy usage of sensor nodes in FHOEEC is significantly lower than in that in other methods. Energy consumption in FHOEEC is approximately 21.53%, 35.79%, and 55.15% less than that in CCCS, GTC, and EULC, respectively. In UASNs, evaluating the number of alive nodes can provide a good criterion of the network lifespan. Fig. 7 evaluates the count of alive nodes based on simulation rounds. In this experiment, FHOEEC displays the best network lifetime and improves the first node died (FND) compared to other methods. Accordingly, in FHOEEC, CCCS, GTC, and EULC, the death of the first node occurs at rounds 1400, 1350, 869, and 747, respectively. This means that FHOEEC improves FND by 3.7%, 61.10%, and 87.42% compared to CCCS, GTC, and EULC, respectively. The reason for this successful performance is that FHOEEC conducts a centralized cluster formation process, avoiding the imposition of additional computational load on the nodes. During the CH selection process, nodes with energy above the average energy have higher priority to be selected as CHs, while other nodes are penalized and have lower priority. In the routing process, paths within and between clusters are formed with a focus on the energy balance issue. This strategy promotes balanced energy consumption, ultimately enhancing the network lifetime.

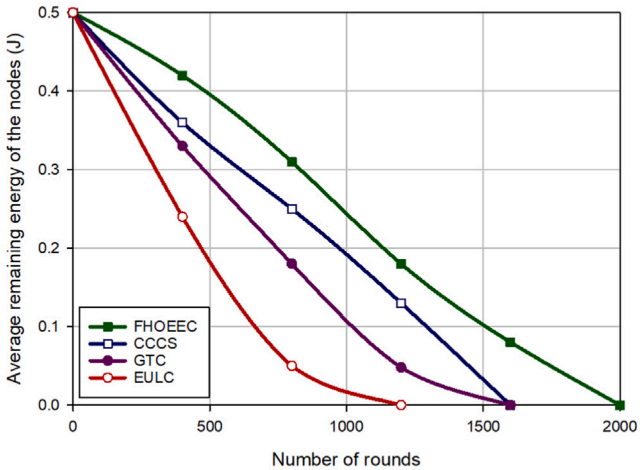

Fig. 5. Average remaining energy of nodes in different clustering methods.

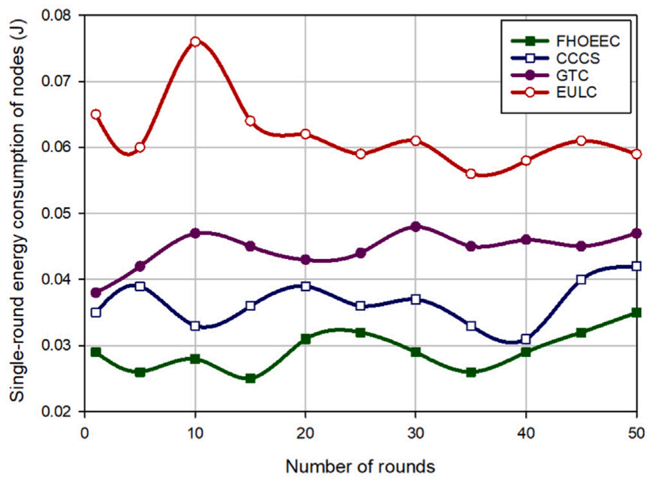

Fig. 6. Average energy consumption of nodes in different clustering methods.

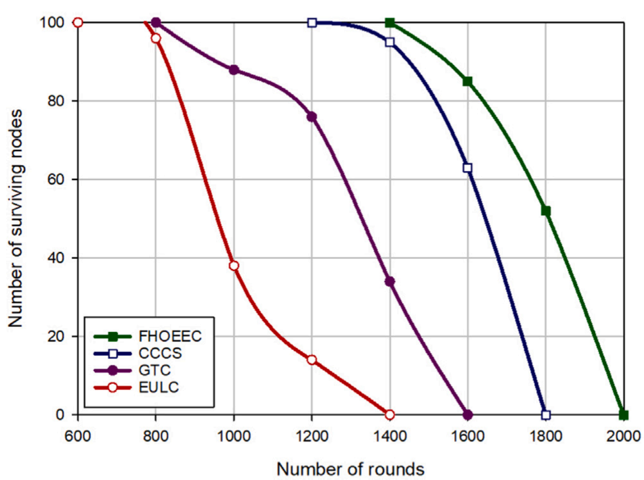

Fig. 7. Network lifetime in different clustering methods.

6.2. Number of received packets

Fig. 8 displays the number of packets received by the sink at different rounds. According to this figure, FHOEEC has the highest packet delivery rate to the sink. It increases the packet reception rate by approximately 10.78% compared to CCCS. However, CCCS also performs better than other clustering methods. As observed in this figure, when simulation rounds increases, the packet reception rate in different clustering methods also increases. This is because in the initial simulation rounds, the sink node carries out a cluster formation operation and determines the membership of sensor nodes in each cluster, as well as their roles, such as cluster member, cluster head, and gateway. Likewise, the stable and energy-efficient path discovery process in the network requires time. After this, the number of packets delivered to the sink gradually increases until it reaches a stable level. Furthermore, when nodes die in the network, the PDR decreases. Among the different methods, FHOEEC can maintain the highest packet delivery rate because it has the most alive nodes and experiences the longest network lifetime.

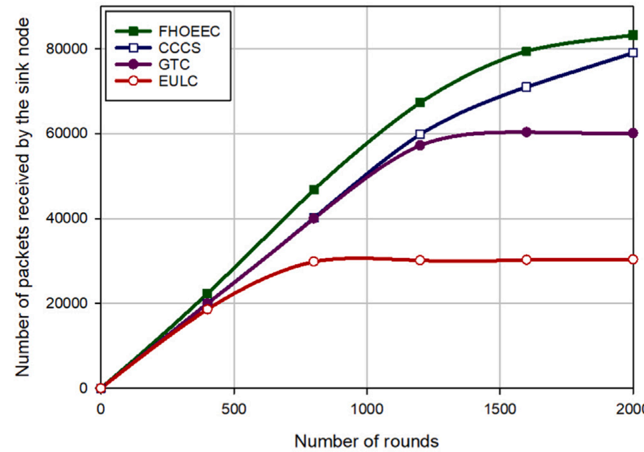

Fig. 8. The number of received packets in different methods.

6.3. Node death rate

Figs. 9 to 12 illustrate the death rate of nodes based on their initial energy in various clustering schemes. Naturally, sensor nodes with higher initial energy have a longer lifespan. Here, FHOEEC exhibits the longest network lifetime. There are three reasons for this issue. First, in FHOEEC, the sink node evaluates the fitness value of each hawk based on a weighted linear combination of three sub-functions: intra-cluster and inter-cluster distances, the ratio of isolated clusters to others, and cluster density. According to the first sub-function, namely intra-cluster and inter-cluster distances, the distance from each cluster center to the cluster member nodes is minimized, resulting in reduced intra-cluster distance and preventing energy wastage during the data transmission process. Furthermore, based on the third sub-function, which assesses intra-cluster density, FHOEEC can uniformly distribute the density of clusters in the network, balance the intra-cluster routing overhead imposed on CHs, and manage their energy consumption to improve the network lifespan. Second, in FHOEEC, the distributed CH selection process is based on three factors, namely energy, average distance, and neighbor degree. Two energy and average distance factors contribute to balancing energy consumption across the network and enhancing the network lifespan. Finally, the intra-cluster and inter-cluster path discovery processes help balance energy consumption in the network, as nodes with energy levels below the average energy are not selected as intermediate nodes in the routing process.

6.4. Scalability

Figs. 13 to 15 evaluate the death rate of nodes across different network sizes. As shown in these figures, when the network dimen-

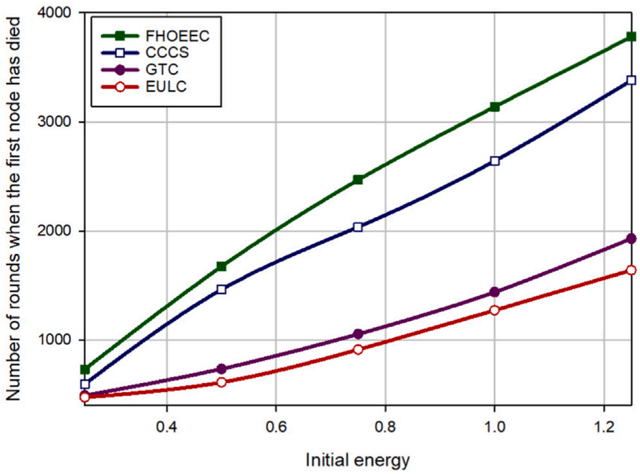

Fig. 9. Evaluation of the death time of the first node in different methods.

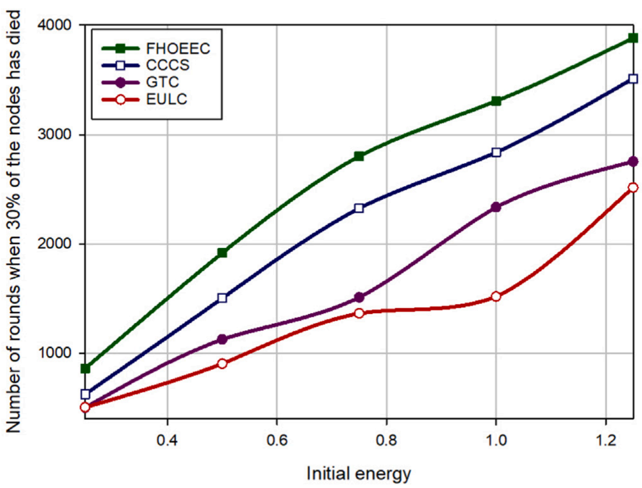

Fig. 10. Evaluation of the death time of 30% of nodes in different methods.

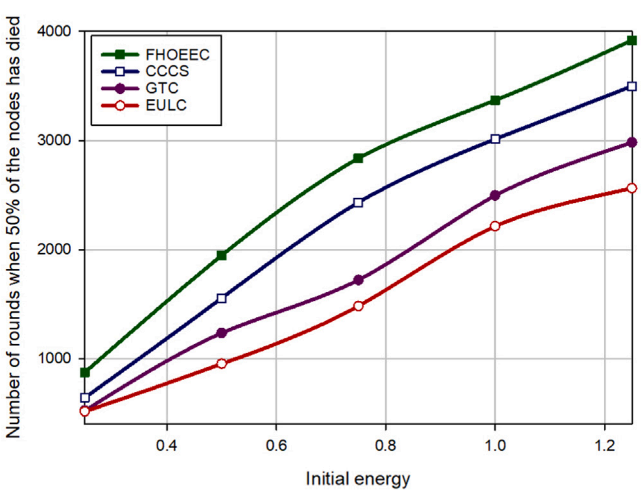

Fig. 11. Evaluation of the death time of 50% of the nodes in different methods.

sions are small, sensor nodes in the network have a long lifespan because the nodes are closer to each other, resulting in less energy consumption during the data transmission process. In contrast, when the network dimensions are larger, nodes die sooner because they are

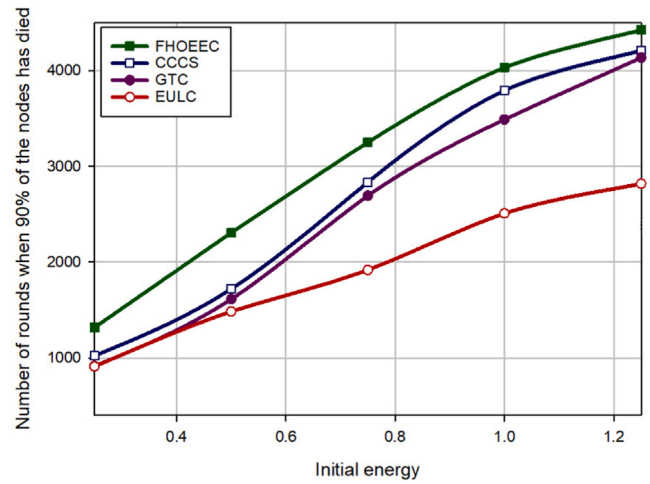

Fig. 12. Evaluation of the death time of 90% of the nodes in different methods.

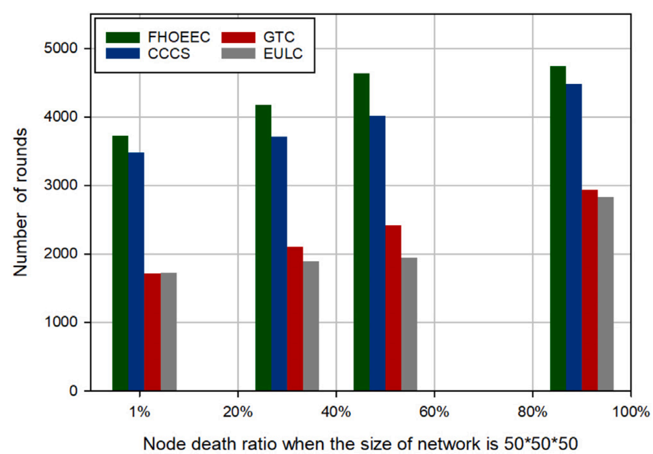

Fig. 13. Number of rounds based on the node death rate in different clustering schemes in a network with the size of 50 × 50 × 50.

farther apart and consume more energy for data transmission. As shown in Figs. 13 to 15, FHOEEC provides the best survival time for nodes in networks with different sizes since FHOEEC is a centralized clustering method that imposes a lower computational load on the sensor nodes. Additionally, the proposed method focuses on energy efficiency and balanced energy consumption in all phases, namely cluster formation, CH selection, and the selection of intra- and inter-cluster paths (see Figs. 16 and 17).

7. Conclusion

In this paper, an FHO-based energy-efficient clustering method (FHOEEC) is proposed for UASNs. This scheme introduces a centralized cluster creation process to determine the optimal range and number of clusters using the fire hawk optimization (FHO) algorithm. This process involves a fitness function focused on three criteria: intra-cluster and inter-cluster distances, the proportion of isolated clusters compared to others, and cluster density. Ultimately, FHOEEC designs an energy-efficient intra-cluster and inter-cluster path discovery process to ensure that nodes with energy levels below the average energy do not participate in the routing process. Simulation results and evaluations confirm the efficiency and effective performance of FHOEEC compared to three clustering methods, namely CCCS, GTC, and EULC. Accordingly, FHOEEC improves the average residual energy of nodes by 19.81%, 40.91%, and 87.87% compared to CCCS, GTC, and EULC,

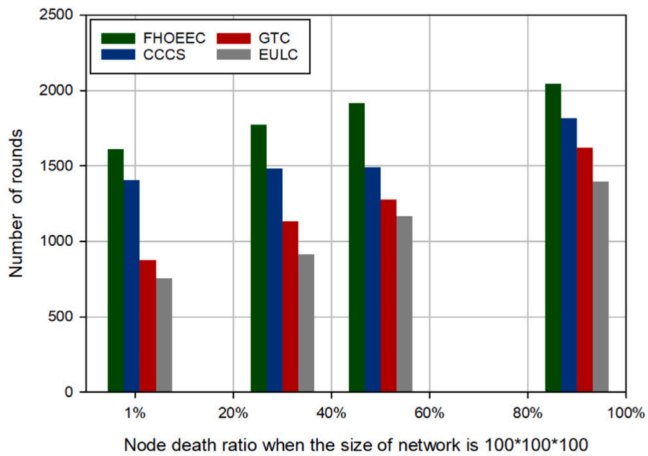

Fig. 14. Number of rounds based on the node death rate in different clustering methods in a network with the size of 100 × 100 × 100.

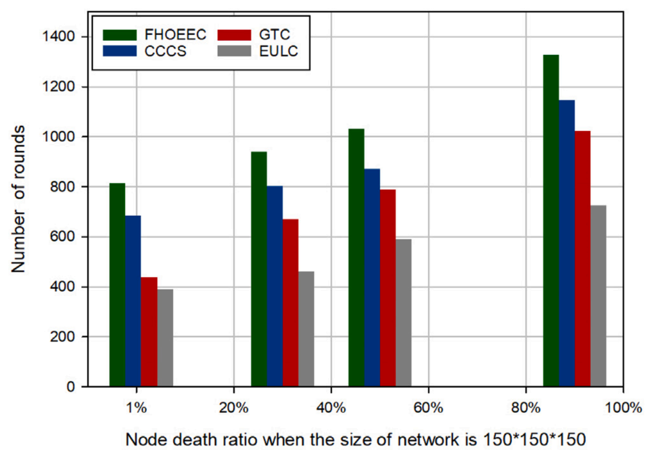

Fig. 15. Number of rounds based on the node death rate in different clustering methods in a network with the size of 150 × 150 × 150.

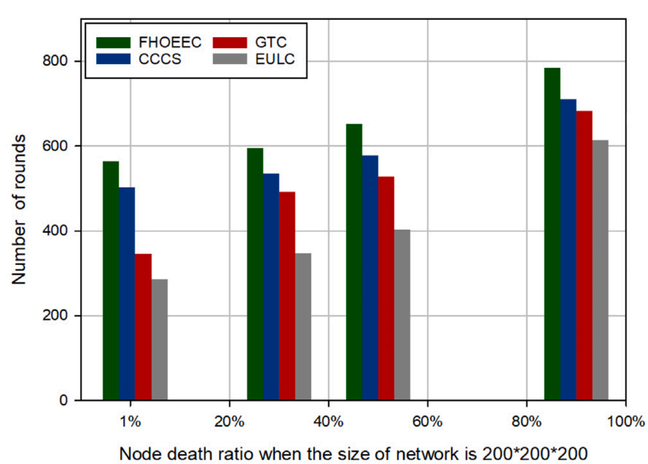

Fig. 16. Number of rounds based on the node death rate in different clustering methods in a network with the size of 200 × 200 × 200.

respectively. Also, energy consumption in FHOEEC is approximately 21.53%, 35.79%, and 55.15% less than that in CCCS, GTC, and EULC, respectively. In addition, FHOEEC displays the best network lifetime and improves FND by 3.7%, 61.10%, and 87.42% compared to CCCS, GTC, and EULC, respectively. FHOEEC increases the packet reception

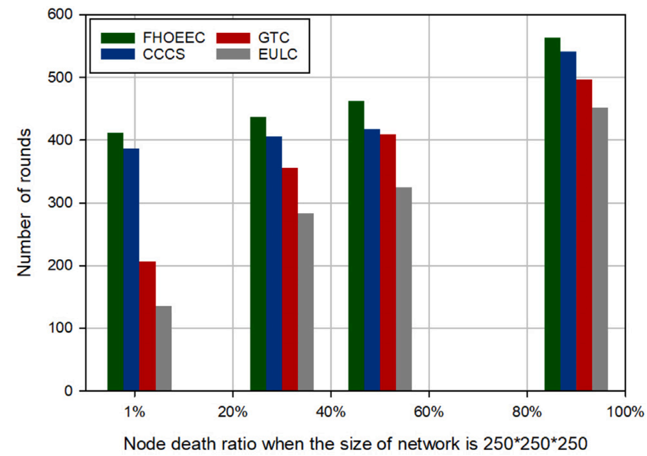

Fig. 17. Number of rounds based on the node death rate in different clustering methods in a network with the size of 250 × 250 × 250.

rate by approximately 10.78% compared to CCCS. Additionally, this method demonstrates scalability compared to other approaches.

Note that FHOEEC is an efficient and effective clustering method in UASNs. However, there are some potential limitations in this scheme that must be considered in the future.

- ***Dynamic environmental conditions***
  - *Limitation:* The performance of FHOEEC may be sensitive to dynamic underwater environments, such as fluctuating water currents, varying depths, or changes in salinity and temperature, which can affect acoustic signal propagation and node mobility. These factors could lead to unreliable communication or suboptimal clustering decisions.
  - *Possible addressing:* Note that environmental variability can impact the stability of the network, and further research could incorporate adaptive clustering techniques to dynamically adjust to these changes.
- ***Limited validation in real-world scenarios***
  - *Limitation:* The results presented in the proposed scheme are based on simulation studies, and real-world underwater environments may present additional challenges not captured in the simulations, such as unpredictable physical conditions.
  - *Possible addressing:* Note that future work could involve real-world deployment and testing of the FHOEEC framework to assess its robustness and performance in actual underwater settings.

Hence, in future research directions, the performance and efficiency of FHOEEC will be evaluated in networks with dynamic topologies. In this regard, efforts will be made to enhance the cluster formation algorithm using machine learning techniques such as Q-learning and deep neural networks like Deep Q-Networks (DQNs). In the future, cluster formation and maintenance mechanisms can consider load balancing and adaptive re-clustering to prevent early energy depletion of CHs. Also, the effect of environmental factors like salinity, temperature, and depth variations on acoustic signal propagation can be considered. To bridge the gap between theory and application in the future, this work can be improved by incorporating real-world case studies that illustrate how the discussed concepts are applied in practice.

CRediT authorship contribution statement

Sang-Woong Lee: Writing – review & editing, Visualization, Data curation. **Musaed Alhussein:** Writing – review & editing, Visualization, Resources. **Khursheed Aurangzeb:** Writing – review & editing, Resources, Data curation. **Mohammad Sadegh Yousefpoor:** Writing – original draft, Validation, Methodology, Conceptualization. **Efat Yousefpoor:** Writing – original draft, Validation, Methodology, Conceptualization. **Mehdi Hosseinzadeh:** Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This Research is Funded by Researchers Supporting Project Number (RSPD2025R947), King Saud University, Riyadh, Saudi Arabia.

Data availability

No data was used for the research described in the article.

References

[1] M.S. Yousefpoor, H. Barati, Dynamic key management algorithms in wireless sensor networks: A survey, Comput. Commun. 134 (2019) 52–69, http://dx.doi.org/10.1016/j.comcom.2018.11.005.

[2] N. Goyal, M. Nain, A. Singh, K. Abualsaud, K. Alsubhi, A. Ortega-Mansilla, N. Zorba, An anchor-based localization in underwater wireless sensor networks for industrial oil pipeline monitoring, IEEE Can. J. Electr. Comput. Eng. 45 (4) (2022) 466–474, http://dx.doi.org/10.1109/ICJECE.2022.3206275.

[3] A.M. Rahmani, J. Tanveer, A. Mutairi, M. Altulyan, E. Gemeay, M. Alam, M.S. Yousefpoor, E. Yousefpoor, M. Hosseinzadeh, A Q-learning-based hierarchical routing protocol in underwater acoustic sensor networks, Comput. Electr. Eng. 123 (2025) 110211, http://dx.doi.org/10.1016/j.compeleceng.2025.110211.

[4] M. Hosseinzadeh, S. Ali, A.M. Rahmani, J. Lansky, V. Nulicek, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, S.W. Lee, A smart filtering-based adaptive optimized link state routing protocol in flying ad hoc networks for traffic monitoring, J. King Saud Univ.- Comput. Inf. Sci. 36 (4) (2024) 102034, http://dx.doi.org/10.1016/j.jksuci.2024.102034.

[5] M.S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, M. Hosseinzadeh, Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review, J. Netw. Comput. Appl. 190 (2021) 103118, http://dx.doi.org/10.1016/j.jnca.2021.103118.

[6] A.M. Rahmani, A. Haider, K. Aurangzeb, M. Altulyan, E. Gemeay, M.S. Yousefpoor, E. Yousefpoor, P. Khoshvaght, M. Hosseinzadeh, A novel cylindrical filtering-based greedy perimeter stateless routing scheme in flying ad hoc networks, Veh. Commun. (2025) 100879, http://dx.doi.org/10.1016/j.vehcom.2025.100879.

[7] H. Mohapatra, A.K. Rath, Fault tolerance in WSN through PE-LEACH protocol, IET Wirel. Sens. Syst. 9 (6) (2019) 358–365, http://dx.doi.org/10.1049/iet-wss.2018.5229.

[8] H. Mohapatra, A.K. Rath, Fault-tolerant mechanism for wireless sensor network, IET Wirel. Sens. Syst. 10 (1) (2020) 23–30, http://dx.doi.org/10.1049/iet-wss.2019.0106.

[9] T. Luo, B. Zhang, J. Li, J. Xiao, C. Li, Y. Liu, Y. Zhang, J. Zhou, An innovative cluster routing method for performance enhancement in underwater acoustic sensor networks, IEEE Internet Things J. (2024) http://dx.doi.org/10.1109/JIOT.2024.3394498.

[10] M.U. Khan, P. Otero, M. Aamir, An energy efficient clustering routing protocol based on arithmetic progression for underwater acoustic sensor networks, IEEE Sensors J. 24 (5) (2024) 6964–6975, http://dx.doi.org/10.1109/JSEN.2024.3354252.

[11] J. Yan, X. Guan, X. Yang, C. Chen, X. Luo, A survey on integration design of localization, communication and control for underwater acoustic sensor networks, IEEE Internet Things J. (2025) http://dx.doi.org/10.1109/JIOT.2025.3525482.

[12] Y. Wang, S. Song, J. Liu, X. Guo, J. Cui, Efficient AUVs-aided localization for large-scale underwater acoustic sensor networks, IEEE Internet Things J. (2024) http://dx.doi.org/10.1109/JIOT.2024.3420448.

[13] Y. Qin, H. Liu, R. Yin, M. Dong, S. Zhao, Y. Deng, Robust multi-model mobile target localization scheme based on underwater acoustic sensor networks, Ocean Eng. 291 (2024) 116441, http://dx.doi.org/10.1016/j.oceaneng.2023.116441.

[14] P. Jiang, H. Wang, Z. Xiong, A dynamic routing scheme for underwater acoustic sensor networks in submarine disaster applications, Front. Mar. Sci. 11 (2024) 1400586, http://dx.doi.org/10.3389/fmars.2024.1400586.

[15] M. Hosseinzadeh, F.M. Husari, M.S. Yousefpoor, J. Lansky, H. Min, A local filtering-based energy-aware routing scheme in flying ad hoc networks, Scientific Reports 14 (1) (2024) 17733, http://dx.doi.org/10.1038/s41598-024-68471-y.

[16] M. Hosseinzadeh, J. Yoo, S. Ali, J. Lansky, S. Mildeova, M.S. Yousefpoor, O.H. Ahmed, A.M. Rahmani, L. Tightiz, A cluster-based trusted routing method using fire hawk optimizer (FHO) in wireless sensor networks (WSNs), Sci. Rep. 13 (1) (2023) 13046, http://dx.doi.org/10.1038/s41598-023-40273-8.

[17] M. Hosseinzadeh, O.H. Ahmed, J. Lansky, S. Mildeova, M.S. Yousefpoor, E. Yousefpoor, J. Yoo, L. Tightiz, A.M. Rahmani, A cluster-tree-based trusted routing algorithm using grasshopper optimization algorithm (GOA) in Wireless Sensor Networks (WSNs), Plos One 18 (9) (2023) e0289173, http://dx.doi.org/10.1371/journal.pone.0289173.

[18] Z. Jin, H. Li, Y. Wang, J. Liang, S. Cheng, Energy-balanced routing protocol with nonuniform clustering for underwater acoustic sensors networks, IEEE Sensors J. (2024) http://dx.doi.org/10.1109/JSEN.2024.3471878.

[19] U. Draz, T. Ali, S. Yasin, M.H. Chaudary, I. Yasin, M. Ayaz, E.H.M. Aggoune, Hybrid underwater localization communication framework for blockchain enabled IoT-underwater acoustic sensor network, IEEE Internet Things J. (2025) http://dx.doi.org/10.1109/JIOT.2025.3535375.

[20] X. Wang, Y. Su, R. Fan, SFDM: A time-synchronization-free detection mechanism for mobile underwater acoustic sensor networks, Ocean Eng. 311 (2024) 118983, http://dx.doi.org/10.1016/j.oceaneng.2024.118983.

[21] A.M. Rahmani, D. Hussain, R.J. Ismail, F. Alanazi, S. Belhaj, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, M. Hosseinzadeh, An adaptive and multi-path greedy perimeter stateless routing protocol in flying ad hoc networks, Veh. Commun. 50 (2024) 100838, http://dx.doi.org/10.1016/j.vehcom.2024.100838.

[22] M. Hosseinzadeh, J. Tanveer, F. Alanazi, K. Aurangzeb, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, S.W. Lee, A.M. Rahmani, An intelligent clustering scheme based on whale optimization algorithm in flying ad hoc networks, Veh. Commun. 49 (2024) 100805, http://dx.doi.org/10.1016/j.vehcom.2024.100805.

[23] J. Yan, Y. Zheng, X. Yang, C. Chen, X. Guan, Privacy-preserving localization for underwater acoustic sensor networks: A differential privacy-based deep learning approach, IEEE Trans. Inf. Forensics Secur. (2024) http://dx.doi.org/10.1109/TIFS.2024.3518069.

[24] A.V. Jha, B. Appasani, M.S. Khan, H.H. Song, A novel clustering protocol for network lifetime maximization in underwater wireless sensor networks, IEEE Trans. Green Commun. Netw. (2024) http://dx.doi.org/10.1109/TGCN.2024.3375011.

[25] Y. Zhao, Y. Chen, X. Wang, X. Zhu, Y. Tao, X. Xu, PUCRC: A PSO-based unequal clustering routing protocol for underwater acoustic cooperative networks, IEEE Internet Things J. (2025) http://dx.doi.org/10.1109/JIOT.2025.3545761.

[26] S. Raghu, M. Harshavarthan, S. Thangavelu, A cluster based routing for maximizing the lifetime of underwater wireless sensor network using gravitational search algorithm, Results Eng. (2025) 104470, http://dx.doi.org/10.1016/j.rineng.2025.104470.

[27] M. Hosseinzadeh, S. Ali, H.J. Ahmad, F. Alanazi, M.S. Yousefpoor, E. Yousefpoor, A. Darwesh, A.M. Rahmani, S.W. Lee, DCFH: A dynamic clustering approach based on fire hawk optimizer in flying ad hoc networks, Veh. Commun. 47 (2024) 100778, http://dx.doi.org/10.1016/j.vehcom.2024.100778.

[28] M. Hosseinzadeh, J. Tanveer, A.M. Rahmani, K. Aurangzeb, E. Yousefpoor, M.S. Yousefpoor, A. Darwesh, S.W. Lee, M. Fazlali, A Q-learning-based smart clustering routing method in flying ad hoc networks, J. King Saud Univ.- Comput. Inf. Sci. 36 (1) (2024) 101894, http://dx.doi.org/10.1016/j.jksuci.2023.101894.

[29] W. Tian, Y. Zhao, R. Hou, M. Dong, K. Ota, D. Zeng, J. Zhang, A centralized control-based clustering scheme for energy efficiency in underwater acoustic sensor networks, IEEE Trans. Green Commun. Netw. 7 (2) (2023) 668–679, http://dx.doi.org/10.1109/TGCN.2023.3249208.

[30] G. Xing, Y. Chen, R. Hou, M. Dong, D. Zeng, J. Luo, M. Ma, Game-theory-based clustering scheme for energy balancing in underwater acoustic sensor networks, IEEE Internet Things J. 8 (11) (2021) 9005–9013, http://dx.doi.org/10.1109/JIOT.2021.3055857.

[31] R. Hou, L. He, S. Hu, J. Luo, Energy-balanced unequal layering clustering in underwater acoustic sensor networks, IEEE Access 6 (2018) 39685–39691, http://dx.doi.org/10.1109/ACCESS.2018.2854276.

[32] Y. Yuan, M. Liu, X. Zhuo, Y. Wei, X. Tu, F. Qu, A Q-learning-based hierarchical routing protocol with unequal clustering for underwater acoustic sensor networks, IEEE Sensors J. 23 (6) (2023) 6312–6325, http://dx.doi.org/10.1109/JSEN.2022.3232614.

[33] R. Zhu, A. Boukerche, Y. Chen, Q. Yang, A reliable cluster-based opportunistic routing protocol for underwater wireless sensor networks, Comput. Netw. 251 (2024) 110622, http://dx.doi.org/10.1016/j.comnet.2024.110622.

[34] Y. Sun, M. Zheng, X. Han, S. Li, J. Yin, Adaptive clustering routing protocol for underwater sensor networks, Ad Hoc Networks 136 (2022) 102953, http://dx.doi.org/10.1016/j.adhoc.2022.102953.

[35] S. Vahabi, A. Daneshvar, M. Eslaminejad, S.E. Dashti, CBDS2r: A cluster-based depth source selection routing for underwater wireless sensor network, IEEE Trans. Signal Inf. Process. over Networks 9 (2023) 468–476, http://dx.doi.org/10.1109/TSIPN.2023.3299108.

[36] K.G. Omeke, M.S. Mollel, M. Ozturk, S. Ansari, L. Zhang, Q.H. Abbasi, M.A. Imran, DEKCS: A dynamic clustering protocol to prolong underwater sensor networks, IEEE Sensors J. 21 (7) (2021) 9457–9464, http://dx.doi.org/10.1109/JSEN.2021.3054943.

[37] M. Azizi, S. Talatahari, A.H. Gandomi, Fire hawk optimizer: A novel metaheuristic algorithm, Artif. Intell. Rev. 56 (1) (2023) 287–363, http://dx.doi.org/10.1007/s10462-022-10173-w.

[38] W. Zhang, G. Han, X. Wang, M. Guizani, K. Fan, L. Shu, A node location algorithm based on node movement prediction in underwater acoustic sensor networks, IEEE Trans. Veh. Technol. 69 (3) (2020) 3166–3178, http://dx.doi.org/10.1109/TVT.2019.2963406.

[39] M. Stojanovic, On the relationship between capacity and distance in an underwater acoustic communication channel, ACM SIGMOBILE Mob. Comput. Commun. Rev. 11 (4) (2007) 34–43, http://dx.doi.org/10.1145/1347364.1347373.

[40] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, K. Ball, The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms, in: Proceedings of OCEANS 2005 MTS/IEEE (1086–1092), IEEE, 2005, http://dx.doi.org/10.1109/OCEANS.2005.1639901.

[41] E.M. Sozer, M. Stojanovic, J.G. Proakis, Underwater acoustic networks, IEEE J. Ocean. Eng. 25 (1) (2000) 72–83, http://dx.doi.org/10.1109/48.820738.

[42] G.H. Adday, S.K. Subramaniam, Z.A. Zukarnain, N. Samian, Investigating and analyzing simulation tools of wireless sensor networks: a comprehensive survey, IEEE Access 12 (2024) 22938–22977, http://dx.doi.org/10.1109/ACCESS.2024.3362889.