Received: September 18, 2023. Revised: December 11, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimal Data Collection Path Finding for AUV in Internet of Underwater Things

Lara Riyadh Alnajjar[1] **A. E. Abdelkareem[1]***

[1]*College of Information Engineering, Al-Nahrain University, Baghdad, Iraq*
Corresponding author's Email: ammar.algassab@nahrain univ.edu.iq

Abstract: The Internet of Underwater Things (IoUT) systems using Autonomous Underwater Vehicle (AUV) have significant issues with data collecting latency. This work explores the utilization of heuristic approaches and reinforcement learning. Specifically, we employ ant colony optimization (ACO) and Q-learning methods by implementing them using NS3 simulation. Based on the traveling salesman problem (TSP), adopting two objectives, determining the shortest path and achieving a balance between the length of the AUV's tour and increasing the value of information (VoI) of the entire network, moreover, adopting 7 scenarios, every one with a specific style of ordering visiting the sensor nodes. Additionally, the study investigates the integration of ACO and Q-learning algorithms. The results prove the suggested algorithms can obtain the desired paths planning of AUVs dealing with various numbers of SNs (10, 30, and 50) by considering the length of the path, energy consumption, and period of time for data transfer and the computation time. In contrast to previous studies that employed the branch and bound (BB), genetic algorithms (GA), and ant colony algorithm (ACA), the distance obtained is smaller by 7.5%, and the VoI increased by 0.3%. In reference to the examined algorithm's computational time, in ACO using 10 SNs (less by 0.96% for BB method and 0.32% for ACA method, and increased by 0.2% for GA method), however, the utilisation of 30 SNs exhibits a decrease of 0.99%, 52.688%, and 30.37% for BB, ACA, and GA, respectively. Additionally, Q-learning with 10 SNs takes less time than BB, ACA, and GA methods, by 95.4%, 90%, and 82.22%, respectively, as well, with 30 SNs decreased by (99.95%, 97.6%, and 96.47%) for (BB, ACA and GA) methods, respectively.

Keywords: IoUT, Autonomous underwater vehicle (AUV), Data collection, Path planning, Optimization, Ant colony optimization, Q-learning, TSP.

Notation list

N	Number of sensor nodes.
E_{tx}	Energy consumption to send a single packet
E_{rx}	Energy consumption to receive a single packet
$E_{k,i}$	Sensor i detects event k.
T_{tx}	The transmission time for single packet
P_r	Receiving power consumption
L_{travel}	The length of the AUV's travel path.
$A(d,f)$	Attenuation factor
$N(f)$	Ambient noise
$N_s(f)$	Shipping noise
$N_{th}(f)$	Thermal noise
$N_w(f)$	Waves noise
w, s	The wind speed and the shipping activity factor
z	An ACO algorithm artificial ant is constructed from possible solutions to an issue, like the AUV path.
P_{ij}^z	The likelihood that ant z chooses sensor node S_j
$T_{s,i}$	Sending event with timestamp
d_{ij}	The distance between the first point and the second point
$\mathcal{N}_{ieff}^z$	The set of SNs not visited so far by ant z in its current tour.
$\mathcal{S}^{ib}$	The ant colony's best resolve.
$\mathcal{V}_{AUV}$	The velocity of AUV.
$\mathcal{V}_{S,i}(t)$	Function to trace the VoI gathered from S_i at time t.

$\mathcal{V}(P)$	The total VoI of the data collected from the N SNs
α_q , β_q	The learning rate and the discount factor to Q-learning algorithm, respectively.
α_c , β_c	These factors determine how pheromone track values and heuristic information affect P_{ij}^{z}
η_{ij}	Heuristic value as input to ACO algorithm
τ_{ij}	The value of the pheromone trail.
ω_1, ω_2, $\acute{\omega}_1, \acute{\omega}_2$, ω_1^q , ω_2^q, $\acute{\omega}_1^q$, $\acute{\omega}_2^q$, $\mathfrak{z}^q$, and $\mathfrak{z}$	Weighting factors
ʤ	Threshold of distance used in a special style of SNs
v_k	A relationship equation between the event importance and the time that AUV takes to reach the next hop, used by E-ACO
OB	Inverse of η_{ij} used in E-ACO
E_0	Initial energy
P_{opt}	The optimal path of AUV
P_{sht}	The shortest path of AUV
R_{max}	Maximum iterations.
r_t	Transmission distance.
t_{ij}, $t_{i\to i+1}$ $t_{sail\to i+1}$	The amount of time taken for the AUV to relocate between any two points, from S_i to S_{i+1}
$Q(s_l, a_l)$	The Q value for current state and current action.
$R(s_l, a_l)$	The reward function in Q-learning algorithm, that, s_l and a_l are the current state and the current action.
α	The absorption coefficient
f	Carrier frequency
$\boldsymbol{S}^z$	One ACO algorithm path solution for ant (z)
β	The trade-off factor for VoI measure.
ρ	The pheromone evaporation factor in ACO algorithm.
ϑ	The pheromone decay rate.

1. Introduction

The internet of things (IoT) aims to quickly connect all the things around us so they can interact with one another with little support from humans. Researchers have recently started looking at the IoT possibilities in underwater environments due to the rising need for maritime discovery and usage. The internet of underwater things (IoUT) has been suggested as a way to examine the intelligent relationships between underwater things all around the world [1].

Although the underwater wireless sensor network (UWSN) has a bright future, it also presents novel difficulties for IoUT [2]. Underwater acoustic sensor networks (UASN) are one of the most promising technologies for data collecting [3]. However, there are a number of distinctions between UASNs and wireless sensor networks (WSNs) that have a significant impact on data collecting. Problems arise when a lot of data in the UWSN transferred across long distances. In addition, batteries used in underwater sensors are more difficult to refuel, and underwater acoustic communication requires a transmission power that is significantly higher than radio wave communication. Additionally, the speed of acoustic transmission is around 1.5×10^3 m/s slower than the speed of light, which is (3×10^8 m/s), and is slower than radio wave communication, which causes a significant delay [4–6]. The doppler effect poses a greater vulnerability to the sent signal [7]. Due to these limitations, it is not possible to upload the algorithms created for terrestrial sensor networks to the UASN [4].

"Path planning (PP)" is often used term to describe motion planning. PP involves identifying a sequence of points the AUV must traverse to reach the predetermined destination from the beginning place [8]. PP is a task that is entirely geometric in nature. When the plan is created previously, it is deemed offline; however, when the plan is created progressively while the robot is running, it is identified online[9]. Thus, PP for AUV is a challenge, it needs effectiveness and flexibility of the evolutionary computation paradigms to find time- and energy-efficient pathways. The genetic algorithm (GA), the particle swarm optimization (PSO), the differential evolution (DE), and the ant colony optimization (ACO) algorithms are a few examples of typical works. Even though evolutionary algorithms are highly in a constrained optimization problem in small-scale or coarse-grained working scenarios, they frequently struggle or fail to find a workable solution in large-scale or fine-grained environments [10]. One of the most well-known swarm intelligence methods in the scientific community is ACO. This technique solves combinatorial optimization issues like vehicle routing and scheduling. Finding the best collection of values to maximize or reduce an objective is the goal. For underwater path procedures, other intelligent algorithms have been proposed, such as artificial intelligence (AI) methods, machine learning (ML) techniques are typically favored because they offered faster, more accurate results with lower processing costs[11]. The reinforcement learning (RL) approach is more practical for path problem exploration. The Q-learning algorithm is a markov decision process (MDP) and a RL state. Due to learning, the experiment requires varied actions to gain different rewards and determine the next action with the greatest reward [12–14].

The value of information (VoI) metric is measure how important and timely data deteriorates with time. The following example illustrates VoI's importance. Caution zones are designated in intrusion detection frameworks. When a target enters the caution zone, installed sensors might report higher priority for event detection than regular observance reports [4]. The VoI is greatest immediately after an event is detected, and it subsequently gets less over time [15].

The kind of UWSNs we take into account in this work are shown in Fig. 1. This essay will describe the basic concept and algorithm logic, then create and simulate the mobile robot. In conclusion, the identification of optimal route plans will be found, and followed by a systematic process of verification, comparison, and evaluation of the obtained outcomes.

1.1 Problem statement and objectives

The use of AUV route planning for data collecting has been investigated in earlier research projects. But these researches focus on path length and one particular node order style using optimization methods, which require a long computation time, in addition to the performance degradation in large search networks. Furthermore, not many researchers try to employ alternative machine learning techniques and take the priority of stored data into account and solve the forementioned problem, and most regarded the path distance as the most important component.

Defining the main problems of UWSN, it appears that two issues are still present: First, conserving energy is important in IoUT because inadequate batteries storage and charge-receiving abilities, this reduces network service life. The length of the AUV route passively influences energy consumption. Second, the QoI and the AUV's traversal path are strongly associated. Effective AUV path planning is required to raise QoI. In the preceding instance, the geographic dependency sometimes fails to accurately reflect the value of the data, this means the validity of the data is not taken into account. Given that the values associated with the data generated by the nodes degrade over time, it is important to take into account the importance and priority of the collected data [1]. As an illustration, each application has different needs for the quantity, valuation, and urgency of the data provided by a node that senses an event relevant to the corresponding application. It is necessary to strike a trade-off between the energy consumption as well as the VoI of the IoUT system [16].

This study prioritizes the importance of nodes by utilizing the event packet as an initial step. ACO and Q-learning are utilized in the second step to enhance the objectives of the selected path. The third step involves the AUV following a predetermined path to collect data from nodes. In some cases, AUV has the capability to modify its trajectory and afterward drive within an alternative adaptive route.

The aim of this work is to develop an efficient algorithm for determining the shortest travel distance of the AUV through the collection process in IoUT, taking into consideration the importance of data provided by each sensor node. To achieve this aim, the following objectives have been considered.

1) Developing an efficient algorithm with low complexity and computation time; the AUV path planning problem as a matter of combinatorial optimization with two objectives: the first objective function being the minimum shortest distance of AUV path planning. The second objective maximizes the value of information by minimizing the latency that critical data sensed by network nodes experience when they reach sink; this involves prioritizing the sensor nodes based on their importance's value.
2) To solve the objectives, low-complexity algorithms were utilized, ACO and Q-learning algorithms; explaining a comparison between them in the obtained results.
3) Presenting 7 scenarios, each one with style of arrange the visiting node and different mathematics, further the point of online changing path, and the SN that doesn't have data also considered.
4) Introducing a hybrid method by merging ACO and Q-learning algorithms in a novel way.
5) A particular style was employed, similar to previous studies, leading to a decrease in computation time and improved balance between path length and VoI.
6) For the purpose of assessing the effectiveness of the suggested algorithms, extensive simulations utilizing the NS3 simulator are carried out.

The remainder of the paper is structured so that section 2 will explore similar works. Section 3 reviews the system model of our work. Section 4 presents the different steps and mathematical models of the suggested methods. Section 5 presents simulation research. Finally, Conclusions and recommendations for more research are offered in section 6.

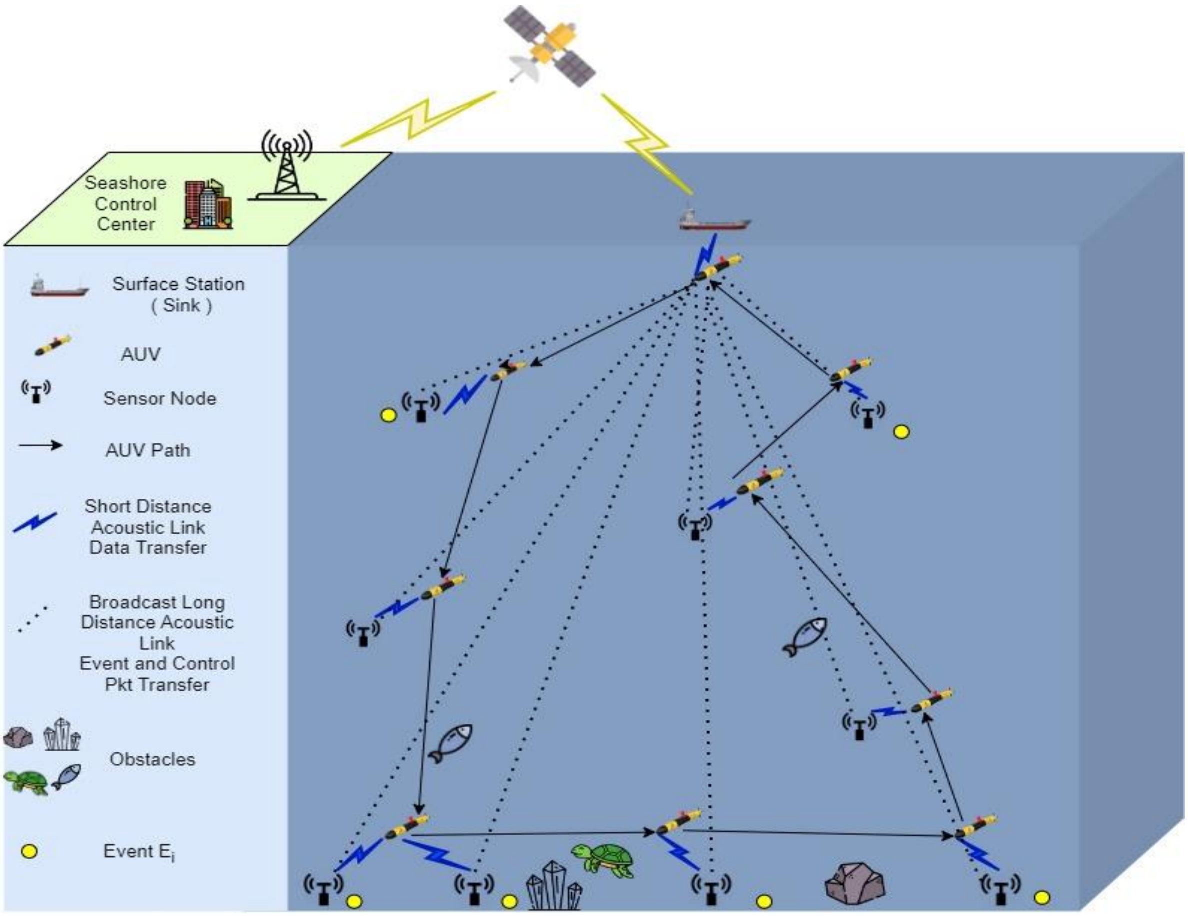


Figure. 1 The model of AUV-enabled IoUT/UWASN visits the deployed sensor node

2. Related works

Planning the AUV's course has a significant influence on the efficiency of data collecting. Numerous academics have suggested a number of ways to improve data collecting with AUVs. In [17], Dijkstra's algorithm, which shortens the AUV's path distance, is still used despite the authors' integration of a multi-hop transmission scheme with the AUV for data gathering, however, it is still severe the collection delay and reduction of the value of information. The authors of [18] developed the alarm pheromone aided ant colony system (AP-ACS), which integrates the alarm pheromone with the standard guiding pheromone, to enhance the algorithm's resilience by mitigating fruitless searches, however, the standard deviation decreases compared to the other methods, moreover, it necessitates a substantial amount of traffic. Also [19] combine AUV with cluster-based routing, and they adopt GAs with objective of reduce length of AUV path and the propagation delay. One other typical indicator of quality of information is the time it takes for data to be transmitted. For instance, the authors of studied the delay optimal data collection problem in [5], which they implemented through a decentralized prediction-based plan was put out using the Kernel Ridge Regression technique. Moreover, the method in [20] Only the impact of geographic location on AUV data collecting is taken into account, the accuracy of the data is not. It is vital to take into account the value information of the acquired data since the values linked with the data supplied by the nodes depreciate with time. The authors of [21] introduced A four-layer software-defined smart underwater edge drone internet (EdgeIoUT), the AUV's path was determined based on the distance and energy.

Unfortunately, these measurements, don't always accurately capture the full worth of the data. VoI has been presented as a novel measure to enhance QoI by concurrently taking into consideration event significance and timeliness in order to address this issue and then optimize the real value of the whole data gathering [1, 4, 15, 22]. In [15] and [22], the authors examined instances in that AUVs made regular trips back to base to offload data, with the VoI being the time metric of choice. They solved ILP

AUV route planning difficulties, and heuristics methods have been provided. However, they failed to adequately convey the significance of the data, and the AUV path planning failed to take the length of trip between nodes into account. Zhao $et\ al.$ In [4] Using VoI, they tackled the AUV route planning problem with a dynamic planning method as an unconstrained optimization problem. Their definition did not consider node timeliness sensitivity. About [1] In order to optimize VoI, the authors investigated the AUV route planning problem. The issue is modelled as an integer linear programming (ILP), and the branch and bound (BB) approach is presented for finding the best solution. Two near-optimal heuristic methods, one of them built on the ideas of the ant colony algorithm (ACA) and the other on those of the GA, are provided. Therefore, all methods used are heuristic methods and take time and complexity overhead in large networks. The route of the AUV was calculated in [12] using simply the Q-learning method, which was based on the energy and VoI of the sensors. Nevertheless, because of the translation of acoustic linkages and group VoI, each sensor's actual VoI is not committed. Now in order to decrease the travel distance of AUV and reduce delay for transmission data some specialists use multiple AUVs [16, 23, 24]. In [16] The process of exchanging information was simulated using the M/G/1 vacation queueing system, however, determine the nearly best path by applying the GA.

3. System model

3.1 Architecture for underwater sensor network

This study considers a limited area represents a portion of the ocean from the surface to the bottom, with a sink node located in the center of the water's surface outfitted with radio and acoustic transceivers. At random distribution isometry nodes with unique IDs, use acoustic modems and may communicate with every other node and the AUV system. The starting energy, cache size, and processing capability of these nodes are all constrained, also locations are fixed. Each node is assigned the task of recording and packs them into a packet with a timestamp of $T_{S,i}$ and relying it upon the arrival of AUV. The expression $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ is used to identify the N sensor nodes. A node S_i's observations at a certain moment have a value of importance E_i. Every time a new event occurs, the node that discovered it generates a short event packet explaining the importance of the observation, naturally, this package much too small, and send it to AUV using single-hop communication. More so, the AUVs are sailing on a tour, gathering data from various sensor nodes. Also, data may be sent to AUV through acoustic connections, then regularly surface to offload data to a sink acoustically. Initially, the AUV follows the planned route in order to gather data, and then modifies its course depending on the algorithm. Effective AUV route design is required to optimize the energy consuming and the VoI of the network. Assuming that the surface station is node S_0, the AUV's route should begin and terminate at S_0 and comprise a series of nonrepeating SNs. For instance, $S_0 \to S_1 \to S_2 \to \cdot\cdot\cdot \to S_M \to S_0$ is one such path. The shortest path and the optimal path denoted by P_{sht} and P_{opt} respectively. Noteworthy is the fact that, it is difficult to determine the precise dynamic characteristics of an AUV, because of the impact of ocean currents. This problem may be solved by model-free tracking, which allows an AUV to follow the target path with accuracy.

It is assumed that a set E of occurrences, where $|E|$ stands for the number of events, $E_1, \ldots, E_{|E|}$. when $1 \leq k \leq |E|$, means sensor i detects event E_k at time t_k, node i combine information about itself and the event in a small control packet with timestamp $T_{s,i}$, and deliver it. We use $\mathcal{V}_{S,i}(t)$ to trace the VoI gathered from S_i at time t, at the completion of the AUV tour to the sink, the measure of the $VoI_{k,i}$ for received data by sensor i of event E_k, as follows:

$$\mathcal{V}_{S,i}(t) = \mathrm{VoI}_{k,i} = \begin{cases} \beta E_{k,i} + (1-\beta)E_{k,i}f(t), & for\ t \geq T_{s,i} \\ 0, & otherwise \end{cases} \quad (1)$$

Where β is the trade-off factor that contributing in balancing between both the severity of the event and how quickly it decays over time. f(t) is a monotonically declining function of time, if $t > T_{s,i}$. In this work, we suppose $f(t) = e^{-\frac{t-T_{s,i}}{\alpha}}$, where α is scaling factor calculates the level of sensitivity with the timeliness [1]. At the end of The AUV path P When the AUV returns to the surface station at time T_{N+1}, the AUV has visited all of the N SNs, and t denotes the journey time of AUV path, completely determines $\mathcal{V}_i(N+1)$, hence $\mathcal{V}_i(N+1) = \mathcal{V}_i(P)$. As thus, the following equation gives the total VoI of the data collected from the N SNs:

$$\mathcal{V}(P) = \sum_{i=1}^{N} \mathcal{V}_i(P) = \beta \sum_{i=1}^{N} E_i + (1-\beta)\sum_{i=1}^{N} E_i\, e^{-\sum_{i+1}^{N}\frac{t-T_{s,i}}{\alpha}} \quad (2)$$

3.2 Energy consumption of underwater environment

The distinct characteristics of underwater acoustic signal propagation undoubtedly produce significant influences on the transmission of data. In this research, we examine a shallow-water acoustic propagation environment and make the simplifying assumption that it exhibits time and spatial homogeneity.

When the environment beneath is rough, there is significant attenuation during transmission that is caused by the acoustic signal through single path attenuation. Across a distance of d, a signal with a frequency f, is attenuated based on [5]:

$$A(d,f) = d^{\lambda} a(f)^{d} \quad (3)$$

Where, "d" indicates the distance, while the variable "f" reflects the carrier frequency in kHz. The parameter $\boldsymbol{\lambda}$ practically is assumed to be 1.5, While the equivalent values for spherical spreading and cylindrical spreading are set as 2 and 1, respectively. Additionally, the variable $a(f)$ provides the absorption coefficient given in dB/km. which is determined using Thorp's expression [25] as follows:

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

Assuming there is no site-specific noise, the power spectrum density of the atmospheric, ambient noise in oceans can be determined employing the following four sources: thermal noise $N_{th}(f)$, shipping $N_s(f)$, waves $N_w(f)$, and turbulence $N_t(f)$ [26] by:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \quad (5)$$

The four noise components' power spectral densities are given in the empirical formulas below as a function of frequency in kHz and expressed in dB re μ Pa per Hz.

$$\begin{aligned}
&10logN_t(f) = 17 - 30logf, \\
&10logN_s(f) = 40 + 20(s - 0.5) \\
&\qquad +26log\, f - 60\, log\, (f + 0.03), \\
&10logN_w(f) = 50 + 7.5\sqrt{w} + 20logf \\
&\qquad - 40log\, (f + 0.4), \\
&10logN_{th}(f) = -15 + 20logf
\end{aligned} \quad (6)$$

Where w is the wind speed in m/s and s is the shipping activity factor, $0 \leq s \leq 1$.

We conclude by applying a standard underwater energy model to the issue of acoustic energy consumption, which is influenced by bandwidth, transmission loss, and propagation delay. The general definition of packet transmission energy usage is as follows [17, 27].

$$E_{tx}(x,d) = xP_rT_p\, A(d,f) \quad (7)$$

Where x represents bits packet; P_r is the power consumption and T_p is data transmission time.

In a similar manner [5], the calculation of a node's energy consumption to receive a data packet is as follows:

$$E_{rx}(x,d) = xP_rT_p \quad (8)$$

4. AUV path planning algorithms

This section focuses on the analysis of algorithms with low complexity that are applicable to the problem of path planning for AUVs.

4.1 ACO based algorithm

ACO approach is used for AUV missions that are limited in both time and space, or per the assigned task in a dynamic underwater environment. Optimizing travel length, energy utilization, etc. In this case we conclude P_{sht} meaning the shortest path. Any ACO solutions are constructed iteratively by deploying ants upon random moves and tours over a linked graph. Artificial ants are constructed from a collection of n possible solutions. Each ant z, where z = 1, ..., M, begins on its journey from the sink node S_0 is shown by $\boldsymbol{S}^z$, which is one of the solutions to the issue. Ant z chooses the S_j that best fits its present location as the S_i and begins building a solution, until all SNs have been visited once and end at S_0, depending on the relative strengths of pheromones and other heuristic factors among the SNs. At each stage of the tour-building process the likelihood that ant z, now at SN S_i, i = 1, 2, … |N|, will choose sensor node S_j, j = 1, 2, … |N|, to visit next is:

$$P_{ij}^{z} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha_c}[\eta_j]^{\beta_c}}{\sum_{S_j \in N_i^z}[\tau_{ij}]^{\alpha_c}[\eta_j]^{\beta_c}}, if S_j \in \mathcal{N}_i^z, q > q_0 \\ [\tau_{ij}]^{\alpha_c}[\eta_j]^{\beta_c}, \qquad if\, d_j \in \mathcal{N}_i^z, q \leq q_0 \\ 0, \qquad otherwise \end{cases} \quad (9)$$

Where τ_{ij} represents the value of the pheromone trail between SNs S_i (S_0 if at the start of tour from the sink) and S_j. η_j denotes the heuristic value associated with the addition of the SN by ant z. $\boldsymbol{\mathcal{N}}_i^z$ represents the available neighborhood of ant z at its current position. Additionally, α_c and β_c are parameters that govern the impact of the pheromone trail values and heuristic information on P_{ij}^z. According to our case study of ACOs based on the ant colony system (ACS), ants use a decision policy, ants in ACS follow the pseudorandom proportional rule in order to transition from city i to city j, which states that given a random value uniform distribution of q between [0,1], and a parameter q_0 , if $q \leq q_0$, then P_{ij}^z be as $[\tau_{ij}]^{\alpha_c}[\eta_j]^{\beta_c}$ Otherwise, the first term in Eq. (9) is used [28]. The definition of the available neighborhood $\boldsymbol{\mathcal{N}}_i^z$ of ant z at a given current position S_i to select the next S_j, is described as follows.

$$\boldsymbol{\mathcal{N}}_i^z = \mathcal{N}_{\mathrm{i}eff}^z, \qquad \mathcal{N}_{\mathrm{i}eff}^z \subseteq \mathcal{N}_{\mathrm{i}full}^z, \quad (10)$$

Where the collection of SNs that make up our network, $\mathcal{N}_{ifull}^z$, is described. Whereas, the set $\mathcal{N}_{ieff}^z$, is a subset of SNs belonging to $\mathcal{N}_{ifull}^z$ that would be the set of SNs not visited so far by ant z in its current tour, and would offer a coverage. Let g_j^z serve as a symbol for the coverage gain of an SN $S_j \in \mathcal{N}_{ifull}^z$. We define g_j^z as the inverted value of distance, or according any another target. Where the start of the tour, i = 0 and $\mathcal{N}_{full} = \mathcal{N}_{eff}$. Due to all ants' path start from S_0, the neighborhood rule guarantees the possibility of ants creating pathways that match to practical solutions to the optimal path for AUV. An edge's desirability annotated by η_{ij} is determined by a function that is basically heuristic and indicates how excellent between SNs S_i and S_j. If the shortest distance of the entire tour is our goal, by using the Euclidean distance between two locations d_{ij}, the definition of η_{ij} is as follows.

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (11)$$

The following cost formula is used to assess the quality of the solution to the optimum route corresponding to the tour built by ant z:

$$C(\mathcal{S}^z) = L_z \quad (12)$$

Where L_z represents the total distance traveled by ant z. Once all ants have built their paths and solutions have been found, the values of pheromone traces are updated according to the ACS, details of updating pheromone adopted from the research [28].

4.2 Event importance based ACO algorithm

The importance of the next event is the primary factor in selecting the best action to take. The period should be as small as possible between sampling data at SN to reach the sink for analysis. An integer number between [1,7] is chosen to signify the event importance. In order to minimize the latency associated with the transmission of real-time event data to the sink, it is suggested to prioritize visiting the SNs. SNs with larger importance values and poor timeliness might result in more value loss, and therefore these are likely to be visited later. Our aim is to explore AUV's route that optimizes the total VoI delivered to the sink as possible and shorten the path length. In this part, we'll go through how to design the ACO algorithm to achieve the goal, which is also known as a multi-objective ACO (MO-ACO) since we'll be utilizing more than one measurement. The multi-constraint circumstances of route planning in such a complex environment challenge solver, particularly in large problem spaces. The complete strategy of ACO algorithm explained in previous sub-section. Here we will specialize to explain how to compute the heuristic information and the cost function. We consider the importance of events for nodes and the distance between nodes i and j and use the weighting factors ω_1 and ω_2 to specify their role importance respectively, hence the heuristic η_{ij} for edge desirability given by the weight function can be calculated by theoretical and empirical formulate:

$$\begin{aligned} t_{ij} &= d_{ij}/\mathcal{V}_{auv} \\ v_k &= 2 \cdot (E_j) + 0.001 \cdot (E_j) \cdot t_{ij} \\ OB &= (\omega_1 \cdot v_k + \omega_2 \cdot d_{ij}) \\ \eta_{ij} &= \frac{1}{OB} \end{aligned} \quad (13)$$

Where t_{ij} is the amount of time taken for the AUV to travel between any two points. $\mathcal{V}_{AUV}$ is the speed of AUV. The time distance is derived from node positions due to the calculation of the Euclidean distance in addition to the AUV's speed. While, v_k is a relationship equation between the event importance and the time that AUV takes to reach the next hop. The weights (2, 0.001) have been specified using trial and error, according to multi runs (1000 tries), in order to meet the best balance, this formula is like the computation of VoI in Eq. (1), however, here this formulation is more convenient, because the minimum path value was examined. From Eq. (13)

It's obvious that η_{ij} is meant to give preference to a node that is closer together and has a lower degree of importance when deciding between two or more candidates, so as to improve the timeliness for applications by checking the importance priority demands and balancing the length of the tour, E_j is the importance value of the node j, d_{ij} is the distance between node i and node j. The weighting factors ω_1 and ω_2, if ω_1 set as $\frac{1}{N} \cdot 100$ and it has an inverse relation with N because the d_{ij} is big in a few spaced sensors. ω_2 is set as a number between 0 to 1 to converge the equation to the first term and it has a positive correlation with N because the d_{ij} value is already tiny in nearby sensors. The utilization of single pheromone information has been considered in the framework of the MO-ACO algorithm.

The cost function that is employed by using summation of OB equation calculated above in moving to any SN within ant z' path:

$$C(\mathcal{S}^z) = \Sigma_{k=1}^{N_z}\left(\omega_1 \cdot v_k + \omega_2 \cdot d_{ij}\right) \tag{14}$$

Where N_z is number of SNs in path of current ant. The description of ACO steps by Algorithm 1.

In another goal ordering approach, priority is given to visiting SNs located near the sink with low importance first, while those with high importance are visited at the end of the tour. This is motivated by the need to ensure that important data does not have to wait for the entire tour duration before being received by the sink for analysis. On the other hand, SNs that are located at greater distances from the surface tend to prioritize the sequence of visitors primarily based on distance compared to importance. We handle this objectively while maintaining a balanced distance. The heuristic factor η_{ij} calculated by the following formula:

$$\eta_{ij} = \begin{cases} \frac{1}{(\acute{\omega}_1 \cdot E_j + \acute{\omega}_2 \cdot d_{ij})}, & if\ d_0 \leq ʤ \\ \frac{1}{((\acute{\omega}_1 - ʒ) \cdot E_j + (\acute{\omega}_2 + ʒ) \cdot d_{ij})}, & otherwise \end{cases} \tag{15}$$

Where $\acute{\omega}_1$ and $\acute{\omega}_2$ are the weighting factors, ʒ represents the differentiation of weights. Specifically, while considering nodes in close proximity to the sink, we provide a higher value to their importance. ʤ is the area defined as being close to the sink, d_0 is the distance to sink. And the cost function has been formed as the following equation.

$$C(\mathcal{S}^z) = \Sigma_{i,j \in N}\left(\frac{1}{\eta_{ij}}\right) \tag{16}$$

Algorithm 1: ACO Based AUV path planning algorithm

1. **Input:** Sensor nodes' position, the position p_i and p_j of sensor i and j, respectively, the event importance vector $\boldsymbol{E_s} = [E_{s_1}, E_{s_2}, \cdots, E_{s_N}]$. The number of ants N_{ant}, the volatilization coefficient ρ, the maximum iteration rounds R_{max}.
2. Calculate the travel distance between nodes by Euclidean distance, Find Lnn by calculate adjacent distance of a random node.
3. Set r = 1, $\tau_0 = \frac{1}{N \cdot Lnn}$.
4. For each edge (i,j), initialize trail intensity as the pheromone matrix to $\tau_{ij} = \tau_0$.
5. Calculate the heuristic factor matrix η according to Eq. (11), (13), or (15).
6. For each ant z:
7. Place ant z on SN S_0 as starting point and store this information in a set.
8. Set $\mathcal{S}_{remain} = \mathcal{S} \setminus \{S_0\}$ as the set of unvisited nodes.
9. For r =1 to R_{max} do
10. For z = 1 to N_{ant} do
11. While $\mathcal{S}_{remain} \neq \emptyset$ do
12. Calculate the probability to choose the next city j according to: $j = Eq. (9)$
13. Store the chosen node, remove this node from $\mathcal{S}_{remain}$.
14. Local update of trail for chosen edge (i,j).
15. Calculate the total cost Eq. (12), (14), or (16) of path of ant z. Choose the best ant.
16. End while
17. End for
18. Choose the best path of existing paths that have minimum length P_{opt}.
19. Update r = r + 1, update the pheromone intensity of P_{opt}.
20. End for
21. **Output:** the optimal path as P_{opt}.

4.3 Shortest path based on Q-learning

We briefly demonstrate the Q-learning approach which is the art of optimal decision-making based on interactive learning logic, the agent (here, the AUV) chooses the best course of action by interacting with the surrounding world. To characterize the network learning process as a Markov decision process with a finite state and action space, we may write M = (S, A, R), where S is the state set i.e., the collection of sensor nodes and the start node, A is the action set i.e., a collection of optional nodes, and R is the reward function. The next node to be visited is selected as the destination depending on the expected the highest rewards to the next location, hence, each state in this section contains N+1 potential actions. At the lth episode step, the reward function $R(s_l, a_l)$ is constructed as:

$$R(s_l, a_l) = \begin{cases} 1/d_{ij}, & if\ i \neq j \\ 0, & otherwise, \end{cases} \quad (17)$$

Where d_{ij} is the distance between the nodes i and j. Besides that, s_l and a_l are the current state and the current action, respectively. According to $R(s_l, a_l)$, the Eq. (18) used to update the Q function [12–14]:

$$\mathrm{Q}(s_l, a_l) = \mathrm{Q}(s_l, a_l) + \alpha_q \left(R(s_l, a_l) + \beta_q max(\mathrm{Q}(s_{l+1}, a_{l+1})) - \mathrm{Q}(s_l, a_l)\right) \quad (18)$$

The variables s_l and s_{l+1} denote the state in the environment, s_{l+1} and a_{l+1} being the subsequent state and action, respectively. The learning rate, α, determines how much of the new Q an old Q value learns, convergence of Q value occurs more rapidly as increases of α. The discount factor $\beta \in [0,1]$, this factor quantifies the significance of future (long-term) rewards. The term $max(\mathrm{Q}(s_{l+1}, a_{l+1})$ means the chosen action a_l achieves the new state s_{l+1}, that has maximum Q. Since the AUV tour is completed by returning back to the starting point, that will be more efficient when add 1 to $\mathrm{Q}(s_l, 0)$ equation at end of each episode when s_{l+1} is the start point but that fitting for small network. Initialize the Q value table at the start of the process. The AUV chooses the node with the highest Q value as the next hop. AUVs can explore the shortest distance. The Q-table may help the agent reach an optimal state by predicting future rewards for each state-action combination. There are benefits to using the Q-table, for example, if the AUV is at a different location than the surface point and we need to use that location as the starting position, we can quickly calculate the path using the Q-table without performing a complicated computation.

4.4 Event importance based Q-learning algorithm

The importance of SNs as it is achieved and explained in section iv.B, here is will implemented using Q-learning, and will be mentioned specific points. The AUV chooses the next node to be visited as the destination based on the balance between the distance and importance of consequences nodes. The calculation using the Q-learning we called it E-Q-learning. The definition of the Q-learning algorithm was presented in the previous sub-section. The reward function $R(s_l, a_l)$ at l th episode step is designed based on the parameters of distance between states s_l and the s_{l+1} and the importance of each event:

$$R(s_l, a_l) = \omega_1^q \left(\frac{1}{d_{ij}}\right) + \omega_2^q \left(\frac{1}{E_j}\right) \quad (19)$$

The weights ω_1^q and ω_2^q are set as the following criterion. According to how much weight is given to the parameters, if ω_1^q is set to a large number, the solution will ignore the importance, and if ω_2^q is set to a large number, the opposite will occur and will be too long length path.

In a situation where one of the SNs did not sense any event, there is no need for AUV to visit them since there is no new information to be taken, therefore AUV neglect sensor nodes without importance (NSWI). Simultaneously, since the SN's event, denoted as Ej, had an initial value of zero at the beginning of the tour, it was possible for the SN to detect new data during the tour, and that was an important event, then, SN should not wait until AUV finishes the current round and starts a new round. The Q-learning algorithm is helpful in this point by using the Q-table directly without the need for long recalculations during of rounding to perform adaptive path planning, especially, this is more helpful in a case using vertical and horizontal diagrams of AUVs [29], however here we depend on only the distance in ordering these remaining SNs. We can limit the degree of value by threshold, if its high value then AUV goes to it in the current round, whereas, if it is low value, then keep it for the next round. All concepts are explained in algorithm 3. The reward function $R(s_l, a_l)$ is given according the formulation:

$$R(s_l, a_l) = \begin{cases} \omega_1^q \left(\frac{1}{d_{ij}}\right) + \omega_2^q \left(\frac{1}{E_j}\right), & if\ E_j \neq 0, d_{ij} \neq 0, \\ \left(\frac{1}{d_{ij}}\right), & if\ E_j = 0, \\ 0, & otherwise, \end{cases} \quad (20)$$

Furthermore, another style target of SNs may be achieved by using a Q-learning algorithm, as outlined in sub-section IV achieving a balance between distances and the importance of events, while the nodes that are in close proximity to the sink node (defined as a threshold ʤ found by distance to the sink d_0) are assigned a higher value in terms of their relevance. The formula for the reward function $R(s_l, a_l)$ defines it as:

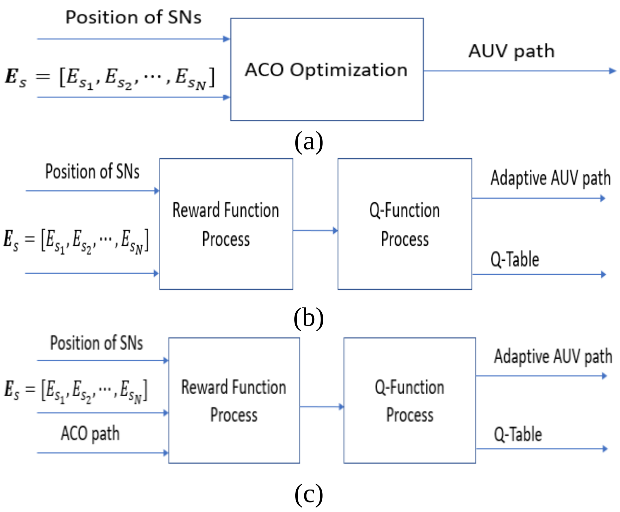


Figure. 2: (a) Block diagram of ACO, (b) Block diagram of Q-learning, and (c) Block diagram of QL-ACO

$$R(s_l, a_l) = \begin{cases} \acute{\omega}_1^q \left(\frac{1}{d_{ij}}\right) + \acute{\omega}_2^q \left(\frac{1}{E_j}\right), if\ E_j \neq 0\,, \\ \qquad\qquad d_{ij} \neq 0, d_0 \leq ʤ \\ \left(\acute{\omega}_1^q + ʒ^q\right) \cdot \left(\frac{1}{d_{ij}}\right) + \left(\acute{\omega}_2^q - ʒ^q\right) \cdot \left(\frac{1}{E_j}\right), \\ \qquad if\ E_j \neq 0\,, d_{ij} \neq 0, d_0 > ʤ \\ 0, \qquad\qquad otherwise, \end{cases} \tag{21}$$

4.5 Q-Learning based ACO

Q-learning is an off-policy reinforcement learning algorithm, as such, it takes a greedy approach by looking for the most straightforward action. ACO is a probabilistic strategy to finding high-quality pathways to computational issues. Therefore, the Q-learning algorithm is a non-probabilistic technique, and it use greedy approach, while ACO is probabilistic technique so it has a full future perspective. TSP one of processes needs a full future perspective, so ACO is more suitable for TSP, but in small network, and this algorithm lacks the useful feature of owning a table like Q-table, that why ACO distance result is lower than q-learning but ACO is very overhead with large networks, and if we want to take advantage of the Q-table and the result of ACO, then it is possible by calculation the ACO algorithm and obtain its path for Q-learning algorithm, so on, we can benefit from the Q-table to perform adaptive path planning. The details of this method same to algorithm 2 unless take the best path of ACO as input. The path of ACO denoted by $P_{sht}^{ACO} = \{S_1^A, S_2^A, \ldots, S_k^A, S_{k+1}^A, S_N^A\}$, so we can make the next formulation to calculate the reward function $R(s_l, a_l)$:

Algorithm 2: Qlearning based AUV path planning algorithm

1. **Input:** Sensor nodes' position, the position p_i and p_j of sensor i and j, respectively, the event importance vector $\boldsymbol{E_s} = [E_{s_1}, E_{s_2}, \cdots, E_{s_N}]$, the maximum iteration rounds R_{max} and number of sensor nodes N.
2. Set the parameters: $\sigma, \gamma\ and\ \epsilon$.
3. Initialize the matrix Q(s, a) = 0.
4. Calculate the Euclidean distance between nodes.
5. Find the reward matrix R(s, a) according to Eq. (17), (19), or (21).
6. **Repeat**
7. Observe the state S_0 = 0 as initial city, set possible nodes = N.
8. **Repeat**
9. Select the action a (destination node) using ϵ-greedy method.
10. Take the action a.
11. Receive immediate reward R(s, a).
12. Observe the new state s' (new city).
13. Update Q(s, a) with Eq. (18).
14. s = s'.
15. Store the chosen node, remove it from possible nodes.
16. **Until** complete the tour.
17. Set action a to a = 0.
18. Update Q(s, a) according to Eq. (18).
19. Store a node, update ϵ.
20. **Until** the stopping criterion R_{max} is satisfied.
21. Set possible nodes = N.
22. **Repeat**
23. Take the action by which node give max value in the Q-table.
24. Store the chosen node in a set, remove it from possible nodes.
25. **Until** complete the tour.
26. Set action a to a = 0.
27. Store node a in the set, and define the set as the best path $P_{opt.}$
28. **Output:** the optimal path as P_{opt}.

$$R(s_l, a_l) = \begin{cases} \left(\frac{1}{d_{ij}}\right) + x, if\ i = S_k^A, j = S_{k+x}^A\,, \\ \qquad\qquad x \in \{1, \cdots, N-1\}, \\ 0, \qquad if\ i = j, \\ \left(\frac{1}{d_{ij}}\right), \quad otherwise, \end{cases} \tag{22}$$

Moreover, initializing the Q-table using Eq. (18). Fig. 2 explains the difference in the block diagram of the methods.

Algorithm 3: E-Qlearning based AUV path planning algorithm considering $E_j = 0$

1. **Input:** Sensor nodes' position, the position p_i and p_j of sensor i and j, respectively, the event importance vector $\boldsymbol{E_s} = [E_{s_1}, E_{s_2}, \cdots, E_{s_N}]$, the maximum iteration rounds R_{max} and number of sensor nodes N.
2. Apply Algorithm 2 with the reward matrix R(s, a) according to Eq. (20) until complete step 20.
3. Set possible nodes, remove the SNs have $E_j = 0$.
4. Apply Algorithm 2 from step 22 until complete step 28.
5. **Repeat**
6. AUV travel from S_i to S_j
7. If received event packet from a sensor node its event was 0 and it sensed data with value > threshold then
8. Save visited nodes in vector, remove new visited nodes from possible Nodes. Go to 3.
9. Break
10. **Until** complete P_{opt}.

Table 1. The setting of parameters

Parameters	Value	Parameters	Value
AUV's velocity $\mathcal{V}_{auv}$	4 m/s	Volatilization coefficient	0.1
Transmitted power PT	30 mW	Maximum iterations R_{max} (E-ACO)	$30 \cdot N$
Received power PR	10 mW	Maximum iterations R_{max} (ACO)	$20 \cdot N$
Maximum iterations R_{max} (Q-learning)& (E-Qlearning)	$50 \cdot N$	Number of ants (ACO)	N
α_c (ACO)&(E-ACO)	2	Number of ants (E-ACO)	$5 \cdot N$
β_c (ACO)&(E-AC0)	3	Pheromone factor k1	$1/(N \cdot Lnn)$
Transmission distance r_t	1000 m	Node Initial Energy(E_0)	100 J
AUV Initial Energy(E_0)	70 KJ	Data rate	25 Kbps
Event packet length	80 bit	Number of Nodes(N)	10, 30, 50
Data length	80-200 Kbits		
α_q	0.15	β_q	0.15

5. Simulation setup and result evaluation

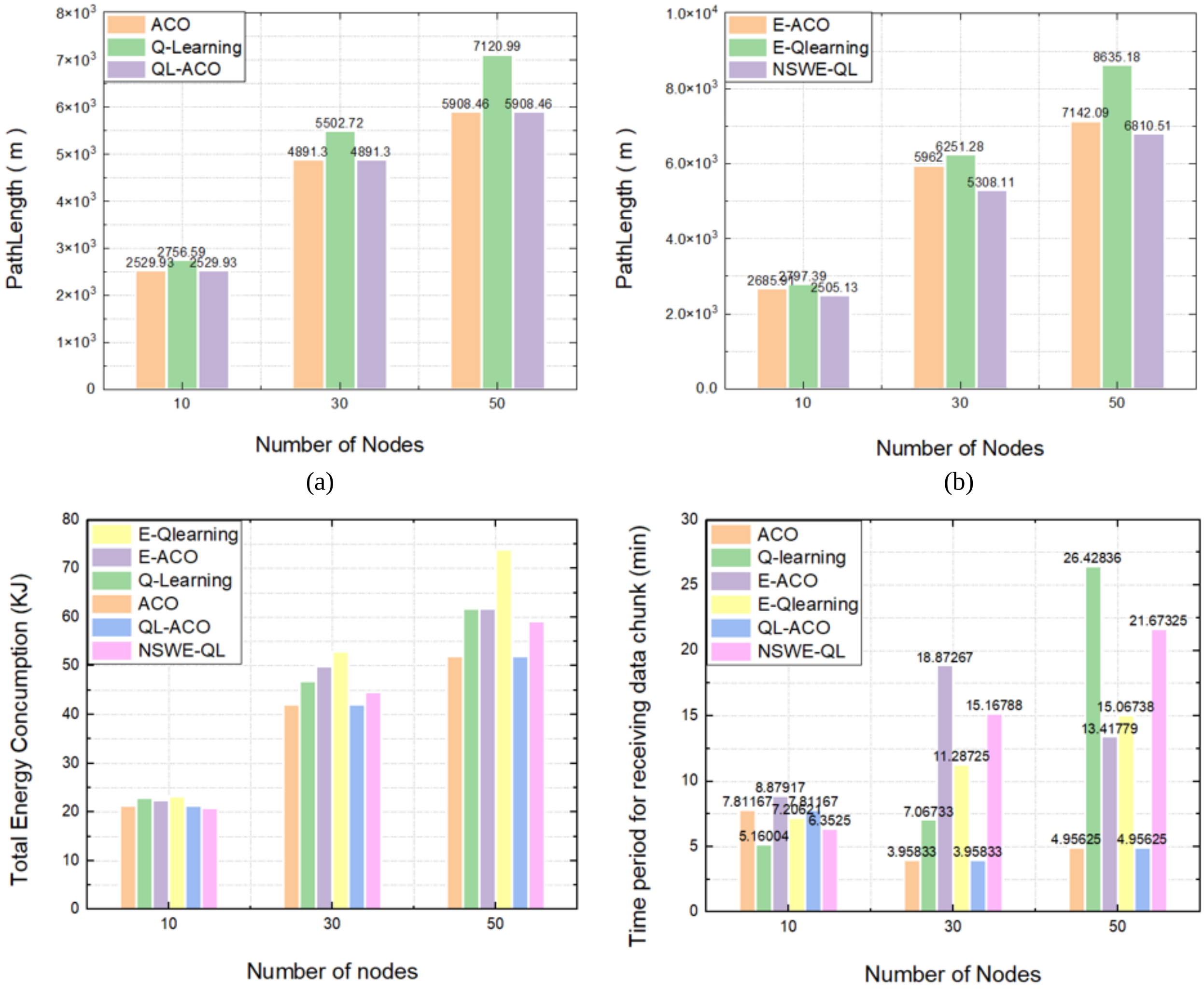
Real-world underwater simulation experiments are challenging. The NS3 simulation environment is an excellent option for rapidly and easily assessing the performance of the suggested algorithm. The network size in the simulated environment is 1000m1000m. In the 2D region, a vast and varied number of sensor nodes are randomly distributed. A sink node, which is deployed in the middle of the water's surface and is thought to be static. To gather data, the AUV travels at a steady speed. Table 1 provides an overview of more specific factors.

This section provides numerical results to assess how well the suggested AUV path planning methods. The results are shown in Fig. 3 by comparing the outcomes of different approaches, all of which operate in the same surroundings. and have the same number of sensors. Specifically, the 'ACO', 'Q-learning', 'E-ACO', 'E-Qlearning', 'QL-ACO' and NSWI represent the paths that found by Algorithm 1, Algorithm 2, and Algorithm 3, then how online resolve the problem of change path easily.

As illustrated in Fig. 3 (a), the path length calculated from the paths resulted from the ACO, Q-learning and QL-ACO algorithms based on the distance between the AUV location and the location of any sensor, the AUV analyses the distance and path planning, where the path of ACO is shorter than Q-learning and still shorter with increasing the number of nodes. Of note, ACO with an increasing number of nodes it becomes unstable and falls in local optimal solution, increasing the overhead and computation time. On the other hand, the Q-learning is much less overhead. The last column "QL-ACO" is for the result of merging the two algorithms, depending on ACO's best paths to plan the rewards for the Q-learning algorithm, it demonstrates that the outcomes almost match the ACO optimal pathways, but this is useful in order to take advantages of the Q-table.

Fig. 3 (b) shows the length of the collecting path with various number of nodes based on case 2 of our objectives. The trade-off between SNs' distances and the importance of the event, here the path length more than the previous case that depends on the distance between nodes. In this study will consider the values of importance in network of N=10 as: $E_1 = 2, E_2 = 4, E_3 = 3, E_4 = 4, E_5 = 2, E_6 = 3, E_7 = 4, E_8 = 6, E_9 = 7, E_{10} = 3$. SNs with higher event importance are visited later, and, this assumption was implemented in [1] to plan the trajectory of AUV, however, they did it using the branch and bound (BB) method which is a method very time-consuming, also they had used a type of ACA and GA, therefore we can solve the idea using other better algorithms like another type ACS of ACO algorithm and Q-learning algorithm. In this result, also the ACO is the shorter path. This figure, which includes the result of

Received: September 18, 2023. Revised: December 11, 2023.



(a) (b)

(c) (d)

Figure. 3 Simulation results: (a) Path length based on distance, (b) Path Length based on distance and importance, (c) Energy consumption, and (d) Time period for data of SN5 reach the sink

Algorithm 3 by neglecting sensor nodes without importance (NSWI) using Q-learning algorithm, proves how to decrease the total length of the tour compared to E-Qlearning and E-ACO. Without loss of generality, nodes with E_k=0 were made when N=10 with two nodes, when N=30 with six nodes, and when N=50 with eleven nodes. This concept is instead of methods in [5] and [12].

The energy consumption of the data collection process is divided into multiple parts and devices, the sensors nodes broadcast the importance directly to AUV by small event packet with a long distance to AUV location. Also, SNs start transferring their chunk data of captured events to AUV when it becomes near by the SN. The consumed energy for moving AUV is very influenced by the path length in order to increase the time of mobility. Also AUV receive the data chunks from SNs and when back to the starting location transmit all the collected data. Of note, the transmission energy is higher than receiving one and depends on the transmitted distance.

As shown in Fig. 3 (c), the consumed energy of ACO is lower because it has shortest path which means less time is taken for complete the collection process. The Qlearning-ACO that depends on the ACO path has the same values, therefore, the same order of path length is positively correlated with consumes energy, after that, the result by Q-learning, then E-ACO, finally E-Qlearning.

The time of receiving the data chunks by the surface station varies for an increasing number of nodes in the network. There isn't a specified rule, it depends on how each algorithm arranges the nodes. The Si was visited first by the AUV, its data has a longer time period until reaching the surface station. This long period causes high value loss, which means it becomes less important if we know about it now. especially with critical time application. Therefore, we assign every event with the event's importance,

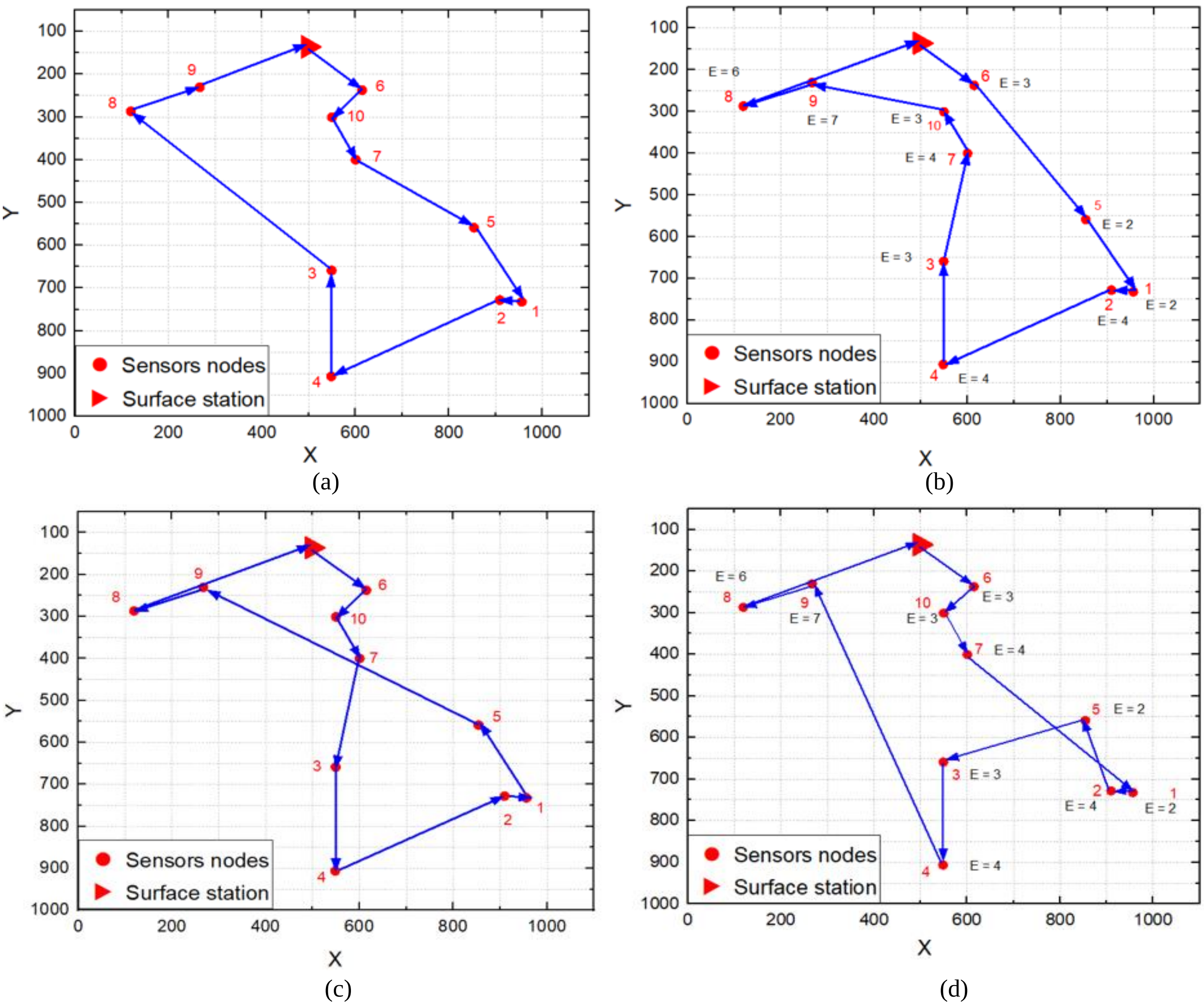


Figure. 4 An overview of the optimal path of each algorithm with N = 10 randomly distributed, where the red numbers are the nodeID of SNs, and E denotes the importance of events recording by each SN: (a) ACO, (b) E-ACO, (c) Q-learning, and (d) E-Qlearning

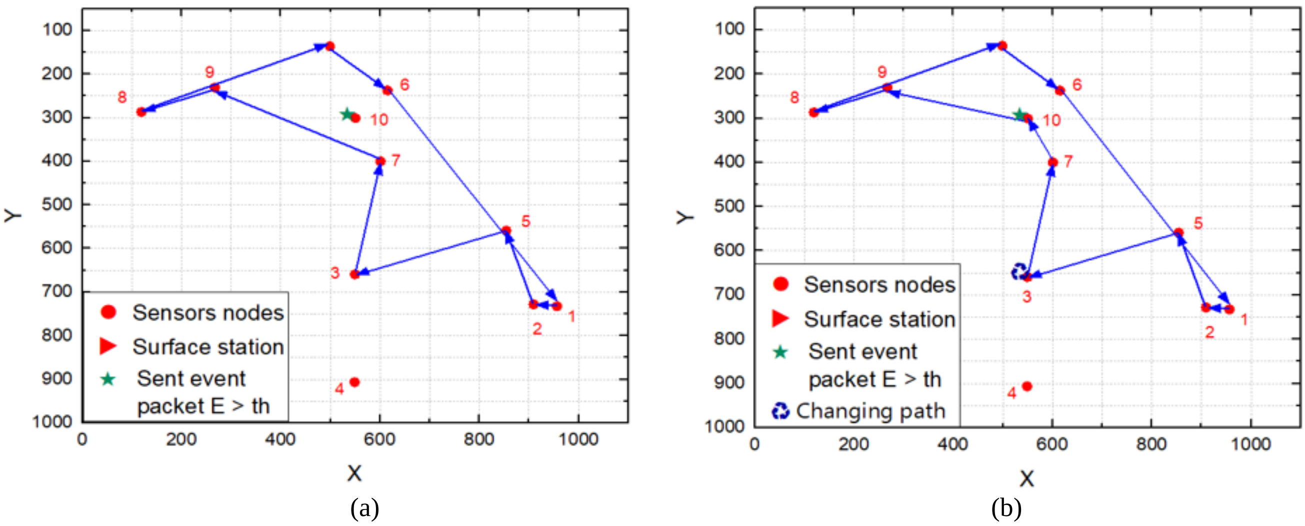


Figure. 5 The path trajectory of (NSWE-QL): (a) The path trajectory and (b) Changing path during the AUV tour

Received: September 18, 2023. Revised: December 11, 2023.

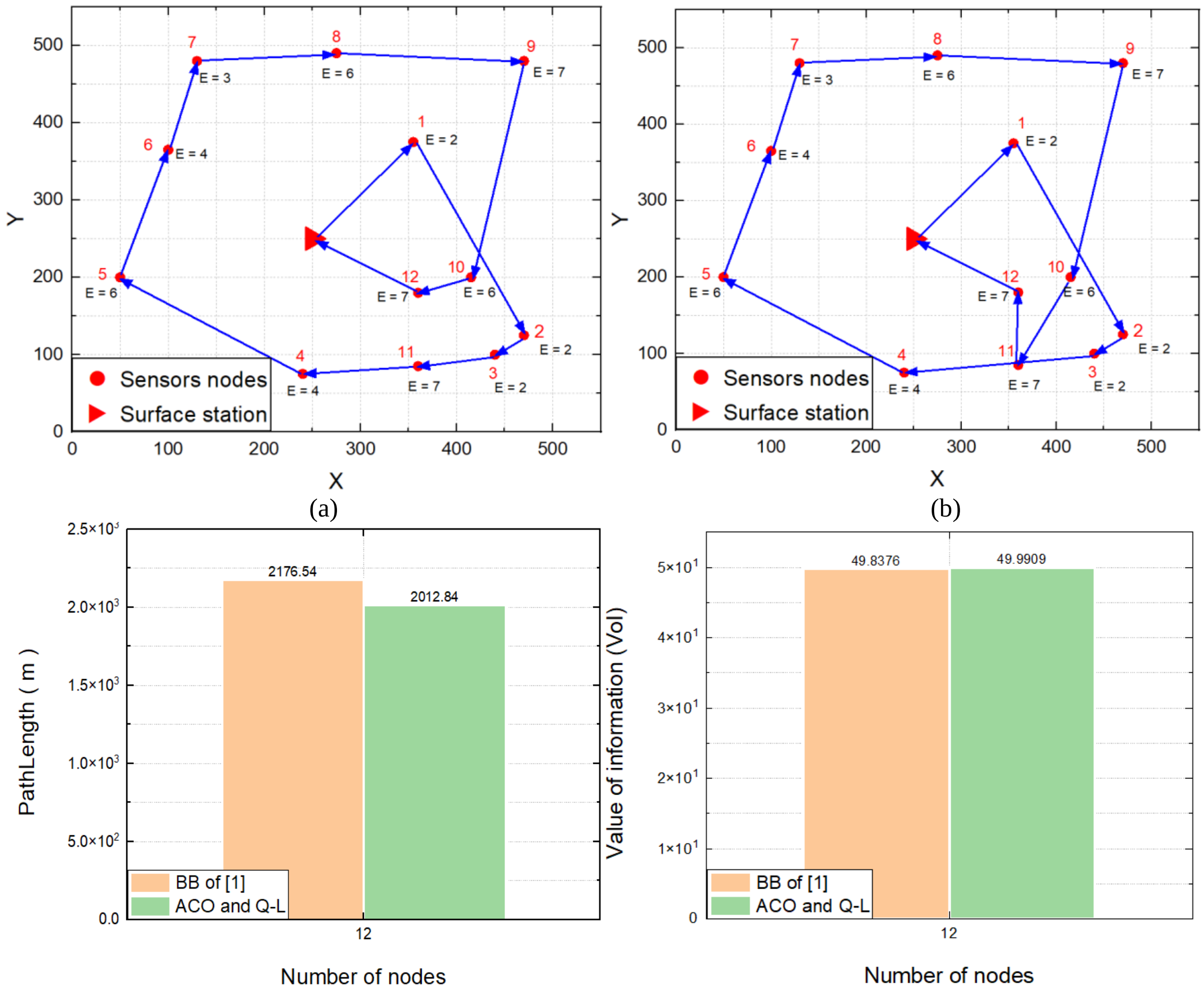


Figure. 6 Results of optimal path under objectives of SNs near sink: (a) Our path obtained by both ACO and Q-learning algorithms, (b) Path in [1] obtained by BB algorithm, (c) Distance, and (d) The VoI

meaning that it is more critical with time. In Figure 4.4, we chose a random node SN5 to examine the time period for its data, we set value 2 to SN5, then it is obvious the time of ACO lower than E-ACO, and the time of Q-learning is lower than E-Qlearning, due to the assumption of every node has lower Importance then first visited according to the trade-off with distance then it has more time.

As a consequence, Fig. 4 explain the desired trajectory of an AUV using the techniques presented in this article. It does so by overview of the best path produced by each algorithm in the case of N=10. The SNs are randomly arranged in a square region of this picture, and the event relevance of their data is shown beside to the SNs. In Fig. 4 (a), the result of ACO algorithm distinguish the distance between nodes and find the greedy shortest path, whether, in Fig. 4 (b), using E-ACO this balance between a shortest path and maximize the VoI by event importance, in mean, where the SNs are close in distance then take the best importance but if they are far then not traverse the distance, cumulated of near nodes is considering more than one near node, as we mentioned, ACO has better full future perspective. after node 6 the node 10 is chose to be visited in Fig. 4 (a) but the node 5 is chose in Fig. 4 (b) because it has a lower importance with near SNs have also low importance.

The Q-learning equation states that each node is chosen based on the rewards offered by it and the options that are available to it subsequently, in Fig. 4 (c), after SN 6, the SN 10 is chosen, and after SN 7, then the agent moves to SN 3, because they are near and have more rewards. Whereas, in Fig. 4 (d), after the SN 6 also select the 10 and 7 because they are near and have convergent importance values, but the SN 1 is selected after SN 7 in order to it have more rewards by its small importance and comparable distance and also has appropriate adjacent SN choices.

However, Fig. 5 (a) shows the path trajectory of neglecting sensor nodes without event NSWE using Q-learning, with E4 = 0 and E10 = 0, there is no requirement for the AUV to visit the sensor nodes 4 and 10. As shown in Fig. 5 (b) if there is a node (SN 10) sensed important data as desired threshold value of importance, it is difficult to wait for the second round. AUV received the event packet from SN10 when it is located near SN 3 and then it changed its path directly according to the Q-table without long computation. Now, AUV will visit this new SN according distance of current location.

The proposed algorithms is.evaluated against the optimal path in [1] obtained by the high-complexity algorithms IPL and BB algorithms with regard to the trajectory of SNs in the optimal path and the length of the path. For this comparison use the same deployment coordinates of SNs as [1] under itsobjectives but more balancing between the distance and the importance of events, using the Eqs. (15) and (21) for ACO and Q-Learning, respectively.

The path planning by this objective in this paper is shown in Fig. 6 (a). The optimal path of [1] is shown in Fig. 6 (b). We were able to get a very near path by the ACO and Qlearning algorithms, we are more concerned about balancing with distance so it is a main factor that affects the energy consumption of AUV. In Fig. 6 (a) SN1 is taken at the start because it has low importance, while SNs 10 and 12 have a high importance so leave them for the last. SN 11 locate between SNs 3 and 4, and the AUV will visit it on the go. In Fig. 6 (b) the authors of [1] were concerned with maximizing the VoI of the path so they leave SN11 at last, they obtained their optimal path by BB algorithm and they got near-optimal path by ACA and GA algorithms. Fig. 6 (c) shows how the path length varies and is less in our procedure. Additionally, calculate the total VoI of the path according Eq. (2) to prove our path has no impact on this crucial parameter. As a result, in Fig. 6 (d), the VoI of our path is a bit more 0.3% than the optimal path of [1], achieved by the low computing time techniques ACO and Q-Learning.

In addition, Table 2 evaluates the calculation times of applied AUV path planning techniques. The time computation for the methods BB, ACA and GA in [1] is mentioned first, followed by the examination of our results. Obviously, The BB method is not capable of dealing with cases regarding plenty of SNs, so we are concerned about other methods. The more SNs there are, the longer it takes to calculate. Since the ACO and Q-learning approaches require the least amount of time, they developed as the most affordable choices. As demonstrated, implementing the trade-off with importance requires more time than the situation without importance consideration, which depends on the number of iterations. It is feasible to apply a smaller number of iterations, but the results won't be stable. If you increase the number of iterations, you'll get results that are more precise.

Table 2. The computational time of applied methods

. Number SNs / Methods	10	30	50
BB method in [1]	0.35 s	300.76 s	-
ACA method in [1]	0.16 s	5.43 s	-
GA method in [1]	0.09 s	3.69 s	-
ACO method in this paper	0.009 s	0.341 s	2.44 s
E-ACO method	0.108 s	2.569 s	18.41 s
Q-learning method	0.016 s	0.13 s	0.631 s
E-Qlearning method	0.016 s	0.13 s	0.631 s

Table 3 presents a comparison between our proposed technique and other previous research.

6. Conclusion and future works

In this study, we looked at the information gathering issue in an IoUT with an AUV. We undertook a detailed examination of the AUV path planning problem to determine the order in which nodes are visited for the data collection, concluded:

1. The ACO algorithm is considered as ACS type and is utilized in this study to enhance AUV path planning for data collection. This is achieved by implementing one objective function and two objectives function.
2. The Q-learning is depended on the reward function, in this work the reward function implemented to work as one and two objectives, summation of both objectives and also scale each to avoid dominant of one objective.
3. The prioritization of importance necessitates visiting high-value nodes later, as critical data cannot be stored for extended periods by AUV without incurring value loss. This approach aims to maximize the VoI.
4. ACO can find the shortest path more than Q-learning, especially with the small network by few numbers of nodes.
5. ACO in large networks become not constant every run with different results, the average more repeating close numbers were chosen, and it becomes much overhead, complexity and the computation time than Q-learning.

Table 3. Comparison of this paper with several previous studies

Research	Methods	Contribution	Lacks
Gjanci P and et al, in 2017 [15]	Define a Greedy and Adaptive AUV Path.	AUV path With maximizing the VoI.	Failed to take the length of trip between nodes into account.
Han G and et al, in 2018 [17]	Dijkstra's algorithm for AUV path. Integration of a multi-hop transmission scheme with the AUV for data gathering.	Shortens the AUV's path distance	Suffer from the collection delay and reduction of the value of information
Han G and et al, in 2019 [5]	The ACO as a case of TSP based on the competition coefficient of the request cluster for AUV path. Kernel Ridge Regression (KRR), for the prediction of data.	Reducing the path length and the collection delay. Prediction models will reduce the visited clusters.	The quantity of data does not accurately represent its significance. Due to, distance and the data traffic are the two parameters used.
Duan R and et al, in 2020 [1]	The heuristic methods (BB) approach, (ACA), and (GA).	Maximizing the VoI. The building of clusters.	Methods used are heuristic methods and take time, overhead, especially in large networks.
Fang Z and et al, in 2021 [16]	The process of exchanging information was simulated using the M/G/1 vacation queueing system. Determine the nearly best path by applying the genetic algorithm (GA).	AUV path trades off energy use and AoI.	Determining the path just with the traditional genetic method, based on the distance, without addition methods or considering priority of nodes.
Bhattacharjya K and De D, in 2021 [21]	A four-layer software-defined smart underwater edge drone internet (EdgeIoUT) is suggested.	Improve residual energy and the time it takes to acquire data.	The AUV's path was determined based on the distance and energy without importance of data; the algorithms are not specified.
Zhao H and et al, in 2022 [12]	Two phases: routing for SNs and path of AU V. Q-learning used.	Path Based on the VoI and energy of sensors.	Lack of real VoI for each node and shortest length. Delay.
This paper	Using of optimization method and AI method (ACO and Q-learning).	Considering the distance, the importance of data, and the computation time to develop efficient algorithms. Implementation of 7 scenarios; the SN that doesn’t have data considered; online changing adaptive path also considered.	

6. Adding more objectives when formulating in ACO is more complex than formulating them in Q-learning.
7. The Q-table feature in the Q-learning algorithm is highly beneficial, as it allows for both offline and online models to be implemented easily.
8. Neglecting the sensors without acquiring new data can enhance the path length, consequently leading to an improvement in overall performance.
9. To enhance the outcomes of Q-learning, an experiment was conducted in which the Q-learning was computed based on the path obtained from ACO, called hybrid QL-ACO. This method proves beneficial when exploiting the Q-table to perform online changing the path.
10. Ultimately, in order to establish a comparative analysis with other works inside a certain style of arranging visited sensor nodes, it has been determined that the suggested techniques demonstrate reduced distance while maintaining beneficial VoI and significantly enhanced computing speed.

However, in future works we will improve the Q-learning by using one of the algorithms, deep learning algorithm or curriculum learning (CL) method, considering the obstacles that interrupt the path of AUVs, and adopting more intricate underwater acoustic channel characteristic.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Lara R. Alnajjar and A. E. Abdelkareem; methodology, Lara R. Alnajjar; software, Lara R. Alnajjar; validation, Lara R. Alnajjar and A. E. Abdelkareem; formal analysis, Lara R. Alnajjar and A. E. Abdelkareem; investigation, Lara R. Alnajjar and A. E. Abdelkareem; resources, Lara R. Alnajjar; data curation, Lara R. Alnajjar and A. E. Abdelkareem; writing—original draft preparation, Lara R. Alnajjar; writing—review and editing, Lara R. Alnajjar and A. E. Abdelkareem; supervision, A. E. Abdelkareem.

References

[1] R. Duan, J. Du, C. Jiang, and Y. Ren, "Value-based hierarchical information collection for AUV-enabled Internet of Underwater Things", *IEEE Internet Things J*, Vol. 7, No. 10, pp. 9870–9883, 2020.

[2] E. C. Liou, C. C. Kao, C. H. Chang, Y. S. Lin, and C. J. Huang, "Internet of underwater things: Challenges and routing protocols", In: *Proc. of 2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 1171–1174, 2018.

[3] P. Braca, R. Goldhahn, G. Ferri, and K. D. LePage, "Distributed information fusion in multistatic sensor networks for underwater surveillance", *IEEE Sens. J.*, Vol. 16, No. 11, pp. 4003–4014, 2015.

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

[5] G. Han, S. Shen, H. Wang, J. Jiang, and M. Guizani, "Prediction-based delay optimization data collection algorithm for underwater acoustic sensor networks", *IEEE Trans. Veh. Technol.*, Vol. 68, No. 7, pp. 6926–6936, 2019.

[6] H. A. Naman and A. E. Abdelkareem, "Variable direction-based self-interference full-duplex channel model for underwater acoustic communication systems", *International Journal of Communication Systems*, Vol. 35, No. 7, p. e5096, 2022.

[7] W. M. Jassim and A. E. Abdelkareem, "Performance enhancement of sensor network architecture for monitoring underwater oil pipeline", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 11, No. 2, pp. 1414–1423, 2021.

[8] M. Panda, B. Das, B. Subudhi, and B. B. Pati, "A comprehensive review of path planning algorithms for autonomous underwater vehicles", *International Journal of Automation and Computing*, Vol. 17, No. 3, pp. 321–352, 2020.

[9] M. L. Muhammed, A. J. Humaidi, and E. H. Flaieh, "Towards Comparison and Real Time Implementation of Path Planning Methods for 2R Planar Manipulator with Obstacles Avoidance", *Mathematical Modelling of Engineering Problems*, Vol. 9, No. 2, 2022.

[10] Y. N. Ma, Y. J. Gong, C. F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: An ant colony algorithm incorporating alarm pheromone", *IEEE Trans. Veh. Technol.*, Vol. 68, No. 1, pp. 141–154, 2018.

[11] A. R. Nasser *et al.*, "Iot and cloud computing in health-care: A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes", *Electronics (Basel)*, Vol. 10, No. 21, p. 2719, 2021.

[12] H. Zhao, J. Yan, T. Wu, A. Li, and X. Luo, "Energy-Efficient Underwater Data Collection: A Q-Learning Based Approach", *Journal of Marine Science and Application*, Vol. 21, No. 3, pp. 204–218, 2022.

[13] Z. Jin, Y. Ma, Y. Su, S. Li, and X. Fu, "A Q-learning-based delay-aware routing algorithm to extend the lifetime of underwater sensor networks", *Sensors*, Vol. 17, No. 7, p. 1660, 2017.

[14] S. Shivani, A. Surudhi, N. Prabagarane, and L. Galluccio, "A Q-learning Approach for the Support of Reliable Transmission in the Internet of Underwater Things", In: *Proc. of 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–6, 2020.

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

[16] Z. Fang, J. Wang, C. Jiang, Q. Zhang, and Y. Ren, "AoI-inspired collaborative information collection for AUV-assisted internet of underwater things", *IEEE Internet Things J.*, Vol. 8, No. 19, pp. 14559–14571, 2021.

[17] G. Han, S. Shen, H. Song, T. Yang, and W. Zhang, "A stratification-based data collection scheme in underwater acoustic sensor networks", *IEEE Trans. Veh. Technol.*, Vol. 67, No. 11, pp. 10671–10682, 2018.

[18] Y. N. Ma, Y. J. Gong, C. F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: An ant colony algorithm incorporating alarm pheromone", *IEEE Trans. Veh. Technol.*, Vol. 68, No. 1, pp. 141–154, 2018.

[19] M. A. Bzoor, E. A. Assem, L. Alawneh, and Y. Jararweh, "Autonomous underwater vehicles support for enhanced performance in the Internet of underwater things", *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 3, p. e4225, 2021.

[20] J. McMahon and E. Plaku, "Autonomous data collection with limited time for underwater vehicles", *IEEE Robot Autom. Lett.*, Vol. 2, No. 1, pp. 112–119, 2016.

[21] K. Bhattacharjya and D. De, "IoUT: modelling and simulation of edge-drone-based software-

defined smart internet of underwater things”, *Simul. Model Pract. Theory*, Vol. 109, p. 102304, 2021.

[22] N. H. Bidoki, M. B. Baghdadabad, G. Sukthankar, and D. Turgut, “Joint value of information and energy aware sleep scheduling in wireless sensor networks: A linear programming approach”, In: *Proc. of 2018 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6, 2018.

[23] C. Lin, G. Han, J. Du, Y. Bi, L. Shu, and K. Fan, “A path planning scheme for AUV flock-based Internet-of-Underwater-Things systems to enable transparent and smart ocean”, *IEEE Internet Things J.*, Vol. 7, No. 10, pp. 9760–9772, 2020.

[24] X. Cao, H. Sun, and G. E. Jan, “Multi-AUV cooperative target search and tracking in unknown underwater environment”, *Ocean Engineering*, Vol. 150, pp. 1–11, 2018.

[25] M. Xu and L. Liu, “Sender-receiver role-based energy-aware scheduling for internet of underwater things”, *IEEE Trans. Emerg. Top Comput.*, Vol. 7, No. 2, pp. 324–336, 2016.

[26] A. Stefanov and M. Stojanovic, “Design and Performance Analysis of Underwater Acoustic Networks”, *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 10, pp. 2012–2021, 2011, doi: 10.1109/JSAC.2011.111211.

[27] K. Wang, H. Gao, X. Xu, J. Jiang, and D. Yue, “An energy-efficient reliable data transmission scheme for complex environmental monitoring in underwater acoustic sensor networks”, *IEEE Sens J.*, Vol. 16, No. 11, pp. 4051–4062, 2015.

[28] F. Valdez, J. C. Vazquez, and F. Gaxiola, “Fuzzy dynamic parameter adaptation in ACO and PSO for designing fuzzy controllers: the cases of water level and temperature control”, *Advances in Fuzzy Systems*, Vol. 2018, 2018.

[29] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, “Stochastic optimization-aided energy-efficient information collection in internet of underwater things networks”, *IEEE Internet Things J.*, Vol. 9, No. 3, pp. 1775–1789, 2021.