Reinforcement Learning-based Trajectory Optimization for Data Muling with Underwater Mobile Nodes

QIANG FU1, AIJUN SONG 1, (Member, IEEE), FUMING ZHANG 2, (Senior Member, IEEE), AND MIAO PAN 3, (Senior Member, IEEE)

1Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL 35487 USA
2School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: fumin@gatech.edu).
3Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA (e-mail: mpan2@uh.edu)

Corresponding author: Aijun Song (e-mail: song@eng.ua.edu).

The efforts presented in this paper were partially supported by the NSF (CNS#: 1801925, 2048188, and 2016726).

ABSTRACT This manuscript addresses the trajectory optimization for underwater data muling with mobile nodes. In the underwater data muling scenario, multiple autonomous underwater vehicles (AUVs) explore or sample a mission area and autonomous surface vehicles (ASVs) visit underway AUVs to retrieve collected data. The optimization objectives are to simultaneously maximize fairness in data transmissions and minimize the travel distance of the surface nodes. We propose a nearest-K reinforcement learning algorithm. In the algorithm, we choose only from the nearest-K AUVs as candidates for the next node for data transmissions. We choose the distance between AUVs and the ASV as the state, selected AUVs as the action. A reward is designed as the function of both data volume transmitted and the ASV travel distance. In the scenario with multiple ASVs, an AUV association strategy is proposed to support the use of multiple surface nodes. We conduct computer simulations for performance evaluation. The effects from the number of AUVs, the size of the mission area, and state selection are investigated. Simulation results show that the proposed algorithm outperforms traditional methods in terms of fairness and the ASV travel distance.

INDEX TERMS Underwater acoustic communications, data muling, autonomous underwater vehicles, autonomous surface vehicles, trajectory optimization, reinforcement learning

I. INTRODUCTION

The mobile platforms, such as autonomous underwater vehicles (AUVs) and autonomous surface vehicles (ASVs), have been used as effective tools in ocean monitoring and exploration [1]. Compared with fixed ocean monitoring networks, AUVs and ASVs have clear advantages in operational costs and mission flexibility [2]. In this paper, we mainly focus on the underwater data muling using these mobile nodes, AUVs and ASVs.

Underwater data muling with mobile nodes promises a new way to achieve data collection in the oceans [3]. In this scenario, a mission area is divided into several submissions, which are assigned to individual AUVs. Surface vehicles, ASVs, ferry data from AUVs using underwater acoustic communications and then transfer the data to the control center using terrestrial wireless communications. Due to the limited communication range of underwater acoustic communications, it is impossible for an ASV to cover a large geographic area. The surface node needs to visit each AUV so that these AUVs can obtain a reasonable amount of time for data transmissions.

The use of ASVs to retrieve the AUV data provides several benefits. First, low latency can be achieved. AUV measurements can be accessed before the vehicle recovery. Second, the AUV energy is conserved since AUVs do not need to surface for communications with the control center. AUVs can work underwater for a longer period of time. Third, multiple surface vehicles provide the possibility to expedite data transmissions.

The trajectory of ASVs needs to be optimized for minimized travel distance. Due to limited energy, ASVs are required to select the shortest route to approach each AUV. In this way, energy efficiency is achieved and the ASV mission time is extended. Access fairness among AUVs needs to be ensured too. Often AUVs are expected to transmit equal data volumes. Unfairness among users can cause large package
delay [4]. This requires the surface vehicle to balance transmitted data volume among all AUVs.

The requirements of energy saving and fairness impose a trade-off in ASVs trajectory planning. On one hand, for the limited energy resource, ASVs need to design a visiting sequence in an energy-efficient way. Therefore, ASVs prefer to move as little as possible to save energy. On the other hand, for the low latency requirement, ASVs should visit each AUV and fetch an equal amount of data among all AUVs. It requires each AUV not only to be given the same communication time but also the same visiting frequency. It is not desirable to approach one AUV for a long time and leave the rest barely connected. That is, ASVs are required to hop among AUVs.

Multiple applications of mobile nodes were reported in Wireless networks [5]–[8]. In [5], mobile sinks were used to collect the data to bypass the hot-spot problem. In [6], an ant colony-based path determination algorithm was proposed for wireless sensor networks. In [7], instead of visiting each sink node, the mobile sink visited each rendezvous point. One common characteristic in these wireless applications was that sensor node locations remained stationary. Therefore, the developed solutions are not applicable to the problem of interest in this paper.

One related application of underwater mobile platforms was the AUV-aided data muling in underwater acoustic sensor networks [9]–[14]. The network consisted a variable number of fixed sensors that are deployed to perform collaborative data collection over a wide area. The AUV-aided data muling was proposed to extend the lifetime of fixed sensors [12]. Survey data were collected by using one or multiple AUVs rather than being transmitted among fixed nodes.

Multiple schemes were proposed in the AUV-aided underwater data muling [13], [15]–[21]. Early research focused on hardware design and simple path planning algorithms [13]. Several energy-efficient protocols were proposed to address this problem. Those protocols reduced AUV energy used by either designing the trajectory of the AUV [15]–[19] or grouping the underwater acoustic sensor into several clusters [20], [21]. None of the efforts in the literature used the multiple AUVs cooperation or the mobile sensor problem in the data muling protocol.

The data muling was formulated as the traveling salesmen problem (TSP) [22]. As one of the most widely studied optimization problems [23], the objective of the TSP was to find the shortest route from a list of cities. There were two types of algorithms: exact algorithms and heuristic algorithms [24]. Exact algorithms were impractical for a large number of cities. Heuristic algorithms, such as the nearest neighbor algorithm and the ant colony algorithm, were able to find a sub-optimal solution for large-scale problems within the fair time expense and acceptable accuracy (97%–98%) [25]. The multiple objectives TSP optimization was explored in [12], [26]. Those research only considered the scenario with static nodes. Research about mobile TSP problems mainly assumed the target moves along a straight line in a two-dimensional space [27]. Mobile TSP problems only focus on a single optimization objective, the shortest route [28], [29].

Recent successes in the application of the reinforcement learning to optimization problems created interest in many areas [30]. In the area of underwater wireless sensor networks (UWSNs), reinforcement learning-based methods were widely used in oceanographic data collection. Energy efficiency [31]–[37] and end-to-end delay [38], [39] were two main concerns in UWSNs protocol design. In [40], a reinforcement learning-based congestion-avoided routing (RCAR) protocol was designed to reduce the end-to-end delay and energy consumption. In [38], the Q-learning based energy-efficient and balanced data gathering routing protocol (QL-EEBDG) was proposed to enhance the network lifespan.

Unmanned aerial vehicles (UAVs) play a similar role in terrestrial communications to ASVs in underwater environments. Reinforcement learning-based algorithms were proposed in UAV-aided terrestrial communications [41]–[44]. Current applications of the reinforcement learning often focus on a single optimization objective, such as minimizing energy consumption, minimizing delay and maximizing coverage area. The role of UAVs in the protocol, the characteristics of the communication channel, the speed of UAVs and ASVs are different. Those algorithms cannot be applied to the underwater environment. The reinforcement learning-based trajectory optimization for ASVs has not been investigated yet.

In this paper, we propose a nearest-\( K \) reinforcement learning-based trajectory optimization for the data muling with underwater mobile nodes. ASV tracks are optimized for the surface vehicles to visit AUVs in a certain sequence. We use the distance between ASVs and AUVs as states; and the selected AUVs as actions. We design the reward as a function of the ASV travel distance and the data volume of AUVs. In this way, we achieve a balanced optimization objective: maximizing fairness among AUVs and minimizing the ASV travel distance. The reinforcement learning algorithm is simplified by limiting only the nearest \( K \) AUVs as candidates, which is selected as the next target AUV to be approached. We also design a user association algorithm such that multiple AUVs can be assigned to different ASVs. The multiple ASVs can work together to serve a group of AUVs.

The major efforts of this paper are summarized as follows. First, we propose a nearest-\( K \) reinforcement learning-based trajectory optimization for data muling with a single ASV. Second, we design an AUVs association strategy for a multiple ASVs scenario. Third, we demonstrate the performance advantage of the proposed algorithm in a multi-AUVs and ASVs cooperation scenario with four and eight AUVs. The proposed algorithm is able to design an optimized track for ASVs, achieve a balanced optimization objective: minimizing the ASV travel distance and maximizing fairness.

The paper is organized as follows. In Section 2, we describe the underwater data muling scenario and related problem statement. In Section 3, we introduce the nearest-\( K \) reinforcement learning-based trajectory optimization al-
algorithm. In Section 4, we demonstrate the performance of the proposed algorithm. In Section 5, we provide concluding remarks.

II. SYSTEM MODEL AND PROBLEM STATEMENT

A. SCENARIO DESCRIPTION

We consider an oceanographic mission with multiple ASVs. The environmental characteristics of a mission area such as temperature, water depth, and salinity need to be collected. These measurements are required to send to a control center onshore. A group of AUVs and ASVs are operated to execute this mission, as illustrated in Fig. 1.

The mission is often divided into multiple sub-missions, which are carried out by individual AUVs. Each AUV is assigned a small area to explore. ASVs are used to collect the survey data from AUVs via acoustic communications. Those data are sent to the control center by terrestrial wireless communications. In this scenario, AUVs do not need to surface or to directly transfer the collected data to the control center. Instead, ASVs approach each AUV and collect data by using underwater acoustic communications and underwater optical communications. The AUV mission time underwater can be extended. If ASVs visit each AUV more frequently, the control center gets data with less delay.

We first consider a scenario with $N_a$ AUVs supported by a single ASV. We define an episode as a period time in which the ASV visits one or multiple AUVs for data transmissions. Each episode has two phases: capturing and trailing phases. During the capturing phase, the ASV chases an selected AUV with a full speed, $v_i$, which is higher than the AUV speed. To maximize the data volume, the selected AUV sends data to the ASV in the capturing phase via acoustic communications. AUVs and ASVs use acoustic communication with a low data rate. Data rate changes based on communication distance. The AUV is considered captured when the ASV-AUV distance reduces to a threshold, for example, 50 m. In the trailing phase, the ASV trails the AUV at a lower speed of $v_c$ for data transmissions. The ASV and ASVs use optical communications with a high data rate. The ASV can communicate with multiple AUVs to fully utilize the communication resources within the operating range of optical communications.

Different from the application in [5], the scenario in this paper only includes moving nodes such as AUVs and ASVs. The AUVs in our scenario move along a designed track to collect data for a target area. Because deploying these AUVs at different depths does not bring additional benefits, we assume that AUVs are deployed in the same depth in the ocean.

B. PROBLEM FORMULATION

We define the equivalent data volume as:

$$B_i'(k) = \omega_i(k) \times b_i(k),$$

(1)

where $b_i$ is successfully received bits from the $i$-th AUV in the $k$-th episode. We define the average data volume as the average value of data volume in a time window length of $T_c$.

The average equivalent data volume of the $i$-th user can be calculated as:

$$B_i(k) = \begin{cases} (1 - \frac{1}{T_c}) B_i(k - 1) + \frac{1}{T_c} B_i'(k) & \text{$i$-th user} \\ (1 - \frac{1}{T_c}) B_i(k - 1) & \text{otherwise}. \end{cases}$$

(2)

A weighting factor $\omega_i(k)$ is defined to account for the user priority and fairness. It is the inverse of the equivalent transmitted data volume in the previous episode,

$$\omega_i(k) = \frac{1}{B_i(k - 1)}.$$  

(3)

The total equivalent data volume is the summation of the equivalent data volume from all AUVs:

$$B = \sum_{k=1}^{K} \sum_{i=1}^{N_a} \tau_i(k) B_i(k),$$

(4)

where $N_a$ is the number of AUVs and $\tau_i(k)$ is an indicator function. When an AUV is selected at $k$-th episode, $\tau_i(k)$ is equal to 1, otherwise, $\tau_i(k)$ is equal to 0. Therefore,

$$\tau_i(k) = \begin{cases} 1 & \text{$i$-th AUV is selected} \\ 0 & \text{otherwise}. \end{cases}$$

(5)

The total ASV travel distance can be expressed as:

$$D = \sum_{k=1}^{K} \sum_{i=1}^{N_a} \tau_i(k) (d_{c,i}(k) + d_{t,i}(k)),$$

(6)

where $d_{c,i}(k)$ is the distance that the ASV uses to capture a target AUV, $d_{t,i}(k)$ is the distance that the ASV trails with the AUV in $k$-th episode. In the Eq. (6), $D_i(k) = d_{c,i}(k) + d_{t,i}(k)$.
We define a metric $R$ below to represents that how much data volume can be transmitted when an ASV travels one unit of distance,

$$
R = \frac{B}{D} = \sum_{k=1}^{K} \sum_{i=1}^{N_{v}} \tau_{i}(k) \times \omega_{i}(k) \times b_{i}(k)
$$

(7)

The optimization objective is to minimize $R$, which is equivalent to maximizing the equivalent data volume while minimizing the travel distance. So the optimization objective becomes to select suitable AUVs at all episodes so that the value of $R$ is minimized.

The data volume at the $i$-th AUV for all $K$ episodes can be calculated as

$$
b_{i}' = \sum_{k=1}^{K} \tau_{i}(k) b_{i}(k).
$$

(8)

The total transmitted data volume from all AUVs during all $K$ episodes is

$$
b' = \sum_{k=1}^{K} \sum_{i=1}^{N_{v}} \tau_{i}(k) b_{i}(k).
$$

(9)

C. ENERGY CONSUMPTION MODEL

In this paper, we focus on designing the ASV track with considerations of energy savings and communication connectivity. The energy consumption of the ASVs comes from two parts. The first part is related to vehicle movements. Let $f_{c}$ and $f_{t}$ be the resistant forces for the capture and trailing phase. The energy consumption for the ASV to capture and trail the $i$-th underwater node during the $k$-th episode is expressed as:

$$
E_{m,i}(k) = f_{c} d_{c,i}(k) + f_{t} d_{t,i}(k),
$$

(10)

which increases linearly with the travel distance. In the capturing phase, the ASV move at a high speed to catch up with an AUV. We consider the $f_{c} = 300 N$ for the speed of 10 $kn$ based on [45]. In the trailing phase, the ASV moves at a lower speed. We consider $f_{t} = 50 N$ at the speed of 2 $kn$ and $f = 100 N$ at the speed of 4 $kn$.

The other part of energy consumption comes from the data transmission. We assume a constant power level for acoustic transmissions, $P_{c}$. The associated energy consumption can be expressed as:

$$
E_{c} = P_{c} \times T_{c},
$$

(11)

where the value of $P_{c}$ is set up as 10 $W$.

Combining (10) and (11), the total energy consumption of the ASV is:

$$
E = \sum_{k=1}^{K} \sum_{i=1}^{N_{v}} \tau_{i}(k) E_{m,i}(k) + E_{c}.
$$

(12)

III. Q-LEARNING BASED ALGORITHM FOR THE ASV TRAJECTORY OPTIMIZATION

A. PROPOSED ALGORITHM

We adopt the Q-learning algorithm to optimize the trajectory of the ASV. The Q-learning algorithm includes an agent, a set of state, actions, and a reward. The agent executes an action and gets a reward, while the state of the agent transits from one to another. The Q-learning algorithm selects actions to maximize the total reward. The following parameters need to be defined based on our scenario:

1) State: The state is defined as the distance that the ASV needs to move in the capturing and trailing phases. Therefore, the state in the $k$-th episode is defined as a vector $D(k)$:

$$
D(k) = [D_{1}(k) \quad D_{2}(k) \quad \cdots \quad D_{N_{v}}(k)],
$$

(13)

where $D_{i}(k)$ is the distance that the ASV needs to travel when the $k$-th AUV is selected. $D_{i}(k)$ is the distance that the ASV needs to travel when the $i$-th AUV is selected. The value $D_{i}(k)$ includes two parts, the ASV distance in the capturing phase $(d_{c,i}(k))$ and the distance in the trailing phase $(d_{t,i}(k))$.

The process of the ASV capturing AUVs is formulated as a differential equation. The location of the ASV is set as the coordinate origin. We assume the $i$-th AUV is located at $a_{i}(k) = (a_{i,1}(k), a_{i,2}(k))$ in the $k$-th episode. The AUV moves along the $y$ axis at the speed of $v$. At time $\Delta_{t}$, this AUV arrives at $(a_{i,1}(k) + v\Delta_{t})$. At the same time, the ASV moves to the point $(x(\Delta_{t}), y(\Delta_{t}))$, traveling at the direction towards to the AUV. Therefore, the differential equation model can be express as,

$$
\frac{dy}{dx} = \frac{y(\Delta_{t}) - a_{i,2}(k) - v\Delta_{t}}{x(\Delta_{t}) - a_{i,1}(k)}.
$$

(14)

The ASV travel distance in the capturing phase $d_{c,i}(k)$ is calculated by the ASV track $((x(t), y(t)))$ determined in this differential equation. The ASV travel distance in the trailing phase $d_{t,i}(k)$ is calculated by using data transmission and ASV-AUV trailing speed.

2) Reward: The reward is defined as:

$$
R' = \sum_{k=1}^{K} \gamma^{k} \sum_{i=1}^{N_{v}} \tau_{i}(k) \times \omega_{i}(k) \times b_{i}(k),
$$

(15)

where $0 < \gamma < 1$ is the discount rate.

To compare the proposed algorithm with single-objective optimizations, we define another reward function to minimize the ASV travel distance,

$$
D' = \sum_{k=1}^{K} \gamma^{k} \sum_{i=1}^{N_{v}} \tau_{i}(k) \times (-D_{i}(k)).
$$

(16)

3) Action: The ASV catches different AUVs to fetch the data in $k$-th episode. Therefore, the action here is defined as the selection of AUVs. Here we set the maximum number that the ASV can transmit
data simultaneously as two. So the action candidates are \(AUV_1, AUV_2, ..., AUV_{N_v}, AUV_1 + AUV_2,..., AUV_{N_v-1} + AUV_{N_v}\).

### B. NEAREST-\(K\) MODIFICATION

To implement the Q-learning algorithm, we need to discretize the vector \(D(k)\) as the state. In later implementation, \(D_i(k)\) values are discretized in the step size of \(\Delta_d = 80\) m. Even with this quantization step, there are a large number of states when the number of AUVs is moderate number. With eight AUVs in a medium mission area, for example, \(3\ km \times 3\ km\), the number of possible states is \(38^8\). This is impractical to implement.

We solve this issue by integrating a nearest-\(K\) algorithm. Instead of using the distance between ASV and all AUVs, we use the distance between ASV and the nearest-\(K\) AUVs as the state:

\[
D(k) = \left[D_1(k)\ D_2(k)\ \cdots\ D_K(k)\right]. \quad (17)
\]

The number of states can be drastically decreased for a small number for \(K\).

### C. USER ASSOCIATION FOR MULTIPLE ASVS

When there are a large number of AUVs, multiple ASVs can be used to reduce the access delay. We propose a geometry-based association algorithm to divide the AUVs into sub-groups, each of which is associated with a single ASV. When there are \(N_v\) AUVs and \(M\) ASVs, the algorithm separates \(N_v\) AUVs into \(M\) sub-groups. AUVs with similar location are assigned into the same group, associated with an ASV.

The \(N_v\) AUVs have their location vectors: \(a_1, a_2, ..., a_N\) (\(a_i \in \mathbb{A}\)). The \(M\) ASVs have the initial locations: \(c_1, c_2, ..., c_M\) (\(c_i \in \mathbb{C}\)). Each AUV is associated to an ASV based on the Euclidian distance \(d(a_i, c_i)\),

\[
\arg\min_{c_i \in \mathbb{C}} d(a_i, c_i) = \arg\min_{c_i \in \mathbb{C}} \sqrt{\sum_{j=1}^{2} (a_{i,j} - c_{i,j})^2}. \quad (18)
\]

The strategy of the association algorithm is to divide the AUVs into \(M\) sub-groups \(S_i\) with pre-determined group sizes, \(N_{v_i} = |S_i|\). Each sub-group has a centroid, which can be considered the starting location of an surface vehicle. The AUVs in a particular sub-group have the shortest distances to its centroid. The association algorithm uses the following procedures:

1) Preset the sizes \(N_{v_i}\) for subgroups of AUVs so that \(\sum_{i=1}^{M} N_{v_i} = N_v\). Randomly choose the locations of \(M\) AUVs as the centroids.
2) Calculate the Euclidian distances between each AUV and the \(M\) centroids.
3) Assign each AUV to the closest centroid using the Euclidean distance calculated in Step 2. If the closest centroid has associated an adequate number of AUVs \(N_{v_i}\), assign this AUV to the second closest centroid.
4) Update the centroid location of each sub-group by

\[
c_i = \frac{1}{|S_i|} \sum_{a_i \in S_i} a_i. \quad (19)
\]

5) Repeat Steps 2 to 4 until the centroids do not change. Those centroids are the initial locations of ASVs.

#### Algorithm 1: User association

**Data:** number of ASVs \(M\), size number of AUVs in each group \(N_{v_i}\), the AUVs location vectors: \(a_1, a_2, ..., a_N\) (\(a_i \in \mathbb{A}\))

**Result:** \(M\) ASVs locations: \(c_1, c_2, ..., c_M\) (\(c_i \in \mathbb{C}\)) at random;

1) Initial \(c_1, c_2, ..., c_M\) \((c_i \in \mathbb{C})\) at random;
2) while \(C\) has not converged do
   for \(i \leftarrow 1\) to \(N_v\) do
     if \(|S| < N_{v_i}\) then
       \(l \leftarrow \arg\min_{c_i \in \mathbb{C}} d(a_i, c_i)\);
       \(S_i = S_i \cup \{a_l\}\);
     end
     else
       \(w \leftarrow \arg\min_{c_i \in \mathbb{C} \setminus \{c_i\}} d(a_i, c_i)\);
       \(S_i = S_i \cup \{a_w\}\);
     end
   end
   \(c_i = \frac{1}{|S_i|} \sum_{a_i \in S_i} a_i\);
end

#### D. NON-LEARNING BASED ASV TRACK PLANNING ALGORITHM

For comparison purposes, we implement the ASV trajectory planning based on the non-learning nearest-\(K\) algorithm in the single ASV scenario. The surface vehicle select the nearest \(K\) AUVs as the candidate for their next visits. The nearest-\(K\) algorithm is described below:

1) Initialization: Select an AUV randomly as the first target to be visited. This AUV is referred to the current stop.
2) At the \(k\)-th episode: calculate the distance \(D_i(k)\), from the currently selected AUV, or the current stop, to unvisited AUVs.
3) Choose the nearest \(K\) AUVs to form an unvisited AUV pool based on the distance \(D_i(k)\).
4) Select one, from the unvisited AUV pool, which has a minimum distance. Mark the selected AUVs as visited.
5) Repeat Steps 2 to 5 until all AUVs are marked as visited.
6) Mark all AUVs as unvisited. Repeat Steps 2 to 6.

In the single ASV scenario, the algorithm calculate the distance \(D_i(k)\). The AUV which has the shortest distance is chosen as the new target AUV, which is expressed in Eq. (20). In the multiple ASVs scenario, AUVs are associated to
A ASV first. The ASV visit the associated AUVs by using the nearest-\( K \) algorithm as below:

\[
D = \sum_{k=1}^{K} \min\left(D_1(k), D_2(k), \ldots, D_K(k)\right). \tag{20}
\]

**Algorithm 2: Nearest-\( K \) algorithm**

**Data:** number of ASV \( M \), number of episodes \( K \), \( N_v \)

AUVs location vectors: \( a_1, a_2, \ldots, a_N \) 

\( (a_i \in A) \)

**Result:** \( M \) ASVs locations: \( w_1, w_2, \ldots, w_K \) 

\( (w_i \in W) \)

1. Select the \( j \)-th AUV arbitrarily. Initial \( S = \{1, 2, \ldots, N_v\} \setminus \{j\} \), \( k \leftarrow 1; \)

2. while \( k \leq K \) episodes do 
   3. while \( S \neq \emptyset \) do 
      4. \( l \leftarrow \arg \min_{l \in S} d(a_j, a_l); \)
      5. \( w_k \leftarrow l; \)
      6. \( S \leftarrow S \setminus \{l\}; \)
      7. \( j \leftarrow l; \)
      8. \( k \leftarrow k + 1; \)
   end 

9. Initial \( S = \{1, 2, \ldots, N_v\} \); 
10. Select the \( j \)-th AUV arbitrarily. Initial \( S = \{1, 2, \ldots, N_v\} \setminus \{j\} \); 
11. \( k \leftarrow k + 1; \)
end 

Note that we deal with a new application where the ASVs are working with mobile platforms to perform data muling, which is different from wireless sensor networks [46]. No learning-based algorithms are applicable to this new application. We use a non-learning-based algorithm as the comparison. The complexity of the non-learning algorithm is lower than our proposed learning-based algorithm.

**IV. SIMULATION AND RESULTS**

We conducted computer simulations to evaluate the performance of the proposed algorithm. Standard datasets are critical in performance evaluation [47]. However, no standard datasets are available for our research problems. Therefore, to evaluate algorithm performance, we created multiple scenarios with different mission areas and different numbers of AUVs.

The ASV-AUV communication data rates were simulated based on the transmitter-receiver distance. It was assumed that the product of the bit rate and distance remained constant. This constant number \( C \) was \( 5 \) kbps \( \times \) km. In the capturing phase, the ASV speed \( v_t \) equaled to 10 km. The data rate decreased with the increase of the ASV-AUV range. We assumed the maximum communication distance was 1 km with the lowest bit rate of 5 kbps. In the trailing phase, the data rate was 200 kbps within the communication range of 50 m. The ASV used the same speed with target AUVs, \( v_c \). The discretization step size \( \Delta_d \) was 80 m. We calculated the total ASV travel distance \( D \) for 100 episodes.

![FIGURE 2: Scenario 1: small mission area, 2 km × 0.4 km, with four AUVs. Four AUVs used the lawn-mowing tracks with a track spacing of 20 m. Three vehicles, AUV-1, AUV-3 and AUV-4, had a higher speed, 2 kn, while the remaining one had a lower speed, AUV-2 at 1 kn.](image)

**A. SCENARIO 1: SMALL MISSION AREA WITH FOUR AUVS**

First, we considered a small mission area, 2 km by 0.4 km, with a small number of AUVs, as shown in Fig. 2. Four AUVs used the same lawn-mowing track pattern with a track space of 20 m. Three vehicles, AUV-1, AUV-3 and AUV-4, had a higher speed, 2 \( kn \), while the remaining one had a lower speed, AUV-2 at 1 \( kn \).

To validate the proposed algorithm, we used it to optimize for a single objective, minimizing the ASV travel distance \( D \). The reward was set based on Eq. (16). The nearest-\( K \) algorithm were used for comparison.

Two initial ASV locations were tested. The results is shown in Table 1. With the initial location as ASV \#01, the ASV traveled 81.9 km when using the Q-learning algorithm. The ASV traveled 87.1 km when using the nearest-\( K \) algorithms. The ASV travel distance was shortened by 5.2 km when the proposed algorithm was used. Similar results were obtained for the initial location of ASV \#02. Two ASV initial locations generated similar ASV travel distance for each of two algorithms. The proposed algorithm generated a shortened distance of 5.8 km. Correspondingly, the energy consumption decreased 5.1%. The results show that the proposed algorithm is effective in optimizing the track distance of the ASV. It performed better than the nearest-\( K \) algorithms.

Next, the Q-learning algorithm was tested with the combined objective of the ASV travel distance and data transmission fairness, which was achieved based on the reward function Eq.(15). Fig. 3 shows the transmitted data volume...
for each AUV. In the nearest-\( K \) algorithms, the data volume of AUV-4 increased fastest among all AUVs. The data volume of AUV-1 had the lowest increase. In comparison, the Q-learning algorithm generated balanced data volumes across four AUVs.

**TABLE 2: AUV data volume under the single-objective optimization (Scenario 1)**

| Data volume (kbits) | AUV-1 | AUV-2 | AUV-3 | AUV-4 | Total |
|---------------------|-------|-------|-------|-------|-------|
| Nearest-\( K \)     | 6,400 | 16,260| 14,400| 22,800| 59,880|
| Q-learning          | 12,380| 13,320| 19,420| 10,920| 66,040|

The two algorithms generated different ASV travel distances. The nearest-\( K \) algorithm produced an ASV travel distance of 26.4 km. The Q-learning algorithm led to an ASV travel distance of 30.1 km, 3.7 km increase. The proposed algorithm optimized for balanced data transmissions. Each AUV had more fairly transmission opportunities. The ASV visited the AUV with lower data volume more frequently to improve fairness. Therefore the ASV travel distance was longer compared with the nearest-\( K \) algorithm. It confirms that the proposed algorithm was effective in implementing the combined objective of the ASV travel distance and data transmission fairness.

Table 2 shows individual AUV data volume \( b'_i \). In the nearest-\( K \) algorithms, AUV-1 achieved the lowest data volume, 6,400 kbits. AUV-4 achieved the highest data volume, 22,800 kbits which was 3.56 times the data volume of AUV-1. In the Q-learning algorithm, AUV-3 transmitted 19,420 kbits data, which was the highest among four AUVs. AUV-4 achieved the lowest data volume, only 10,920 kbits. And AUV-3 transmitted 1.7 times more data than AUV-4.

Table 2 also shows the total data volume \( b'_i \) for all AUVs. In the nearest-\( K \) algorithm, the total data volume was 17,699 kbits. And the total data volume was 16,299 kbits in the Q-learning algorithm, 1,400 kbits lower than the nearest-\( K \) algorithms.

The results show that the proposed algorithm can keep a more evenly data volume among AUVs. The AUV which has a longer distance has lower data volume in the traditional algorithm. The proposed algorithm assigned more visiting opportunities to those AUVs so that they have more opportunities to transmit the data. Because of this, the total data volume decreased a little compared with the traditional algorithm.

Next we investigated the effects of the distance discretization. Three step sizes, \( \Delta_d = 400, 90, \) and 80 m were tested. The results are shown in Fig. 4. When \( \Delta_d = 400 \text{ m} \), the Q-learning algorithm did not work properly. Only two AUVs were given opportunities to transmit data. With a smaller discretization step size, the performance of the algorithm improved. The fairness among AUVs was best ensured when the discretization step size was 80 m. In the following simulations, the discretization step size was setup as 80 m.

**B. SCENARIO 2: LARGE MISSION AREA WITH EIGHT AUVs**

In this subsection, we tested the proposed algorithm with a large group of eight AUVs in a relatively large mission area. Scenario 2 had a survey area of 3 km by 1 km, as illustrated in Fig. 5. The eight AUVs had same lawnmower tracks with a track space of 20 m. Five AUVs, AUV-1, AUV-3, AUV-4, AUV-6, and AUV-8, had the speed of 2 kn. The other three, AUV-2, AUV-5 and AUV-7, had the speed of 1 kn.

We first compared the AUV data volume \( b'_i \) as the episode count increased, as shown in Fig. 6. The Q-learning algorithm used the reward in Eq. (15). In the nearest-\( K \) algorithms, the data volume of AUV-8 increased fastest. The data volume of...
AUV-1 had the lowest increasing speed. In the Q-learning algorithm, the data volume of each AUV increased at a similar speed. Table 3 shows the AUV data volume $b'_i$ under the combined objective optimization (Scenario 2).

Table 3: AUV data volume $b'_i$ under the combined objective optimization (Scenario 2)

| Data volume (kbits) | AUV-1 | AUV-2 | AUV-3 | AUV-4 | AUV-5 | AUV-6 | AUV-7 | AUV-8 | Total |
|---------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Nearest- $K$        | 1,680 | 10,440| 11,760| 15,760| 19,600| 21,960| 21,000| 32,480| 154,680|
| Q-learning          | 21,260| 11,100| 3,200 | 14,740| 7,099 | 11,960| 5,760 | 8,919 | 84,038|

The results show that the proposed algorithm was effective for a larger group of eight AUVs. The ASV visited farther underwater nodes more often to achieve the fairness in the data volume. Therefore, the ASV traveled a longer distance than in the nearest-$K$ algorithms. The nearest-$K$ modifications decreased the number of states while not bringing visible negative impacts on algorithm performance.

Compared with the performance in Scenario 1, the data volume became more uneven among the AUVs. The reason was that the Q-learning algorithm only considered the nearest four AUVs in each episode to plan the track of ASV. It increased discrepancies of data volumes among the AUVs.

C. SCENARIO 3: LARGE MISSION AREA WITH TWO ASVS

In this subsection, we tested the proposed AUVs association strategy in the scenario of two ASVs and eight AUVs, as shown in Fig. 7. AUVs tracks and the optimization objective remained the same as Scenario 2. The node number, track spacing, and vehicle speed were kept the same with Scenario 2.

The AUV data volumes are shown in Fig. 8. In the nearest-$K$ algorithms, the data volume of eight AUVs had a similar trend with that for a single ASV. In the Q-learning algorithm, the data volume of four AUVs increased more evenly than that for a single ASV.

Table 4 shows the AUV data volume $b'_i$ and the total data volume $b'$. In the nearest-$K$ algorithms, AUV-1 achieved the lowest data volume, 7,660 kbits. AUV-4 achieved the highest data volume, 30,080 kbits, which was 3.9 times the data volume of AUV-1. In the Q-learning algorithm, AUV-3 travel distance increased 17% when using the Q-learning algorithm. The energy expense of nearest-$K$ algorithm was $2.5 \times 10^4$ KJ. The Q-learning algorithm led to a higher energy expense, $3.0 \times 10^4$ KJ.

The travel distance increased 17% when using the Q-learning algorithm. The energy expense of nearest-$K$ algorithm was $2.5 \times 10^4$ KJ. The Q-learning algorithm led to a higher energy expense, $3.0 \times 10^4$ KJ.

The results show that the proposed algorithm was effective for a larger group of eight AUVs. The ASV visited farther underwater nodes more often to achieve the fairness in the data volume. Therefore, the ASV traveled a longer distance than in the nearest-$K$ algorithms. The nearest-$K$ modifications decreased the number of states while not bringing visible negative impacts on algorithm performance.

Compared with the performance in Scenario 1, the data volume became more uneven among the AUVs. The reason was that the Q-learning algorithm only considered the nearest four AUVs in each episode to plan the track of ASV. It increased discrepancies of data volumes among the AUVs.
transmitted 21,060 kbits data, which was the highest number among eight AUVs. AUV-7 achieved the lowest data volume, only 6,799 kbits. And AUV-3 transmitted 3.09 times data volume of AUV-6.

The total data volumes $b'_i$ of two algorithms showed a minor difference. The nearest-$K$ algorithm led of total data volume of 134,740 kbits. In comparison, the Q-learning algorithm had 126,399 kbits, only 6.1% percent lower than the nearest-$K$ algorithms.

The ASV travel distances $D$ had a significant difference. The ASV travel distance was 73.4 km in the nearest-$K$ algorithm. The two ASVs traveled 124.1 km in the Q-learning algorithm. The distance increased by 69% when using the Q-learning algorithm. The energy expense of the nearest-$K$ algorithm is $2.1 \times 10^4$ KJ. The energy expense of the Q-learning algorithm is $3.7 \times 10^4$ KJ.

The ASVs visited AUVs with lower data volume more so that the data volume of each AUV increased more evenly. And the ASV moved a longer distance compared to the nearest-$K$ algorithm. It shows that the proposed algorithm assigned more opportunities to the AUV which has a longer distance from the ASV. Same as before, the results show that each AUV had more fairly transmission opportunities in this scenario. Compared with the performance in one ASV scenarios, the AUV data volumes were more evenly distributed among the AUVs.

The usage of two ASVs brought two benefits with the Q-learning algorithm. First, the total data volume increased from 84,038 kbits to 126,399 kbits, which is a 50.4% increase with one more ASV introduced. Second, the ratio between the highest and lowest data volumes decreased from 3.73 to 3.09. This meant the data volumes among eight AUVs became more evenly. The introduction of multiple ASVs made each ASV to serve fewer AUVs than in Scenario 2. The AUVs had more communication opportunities.

**V. CONCLUSION**

In this paper, we proposed a nearest-$K$ reinforcement learning-based trajectory optimization for underwater data muling with mobile nodes. The main idea was to use reinforcement learning algorithm to optimize the track of ASVs. ASVs approached each AUVs along its track and collected data from them. This optimized track maximized the fairness among AUVs and simultaneously minimized the ASV travel distance. We simplified the reinforcement learning algorithm by limiting only the nearest-$K$ AUVs as candidates when the reinforcement learning algorithm calculated optimized ASV trajectories. We also designed a user association strategy, which supported multiple ASVs works together.

**REFERENCES**

[1] H. Zheng, N. Wang, and J. Wu. Minimizing deep sea data collection delay with autonomous underwater vehicles. Journal of Parallel and Distributed Computing, 104:99–113, 2017.
FIGURE 8: AUV data volume $b_i^t$ in Scenario 3 for two algorithms: (a) Nearest-$K$ algorithm and (b) Q-learning algorithm. The Nearest-$K$ algorithm use the optimization objective in Eq. (20). The Q-learning algorithm used the reward in Eq. (15).

[12] R. Kolar, J. Cronin, P. Hartwick, A. C. Sanderson, J. S. Bonner, L. Hotaling, R. F. Ambrosio, Z. Liu, M. L. Passow, and M. L. Reath. Complex real-time environmental monitoring of the hudson river and estuary system. IBM Journal of Research and Development, 53(3):4:1–4:10, 2009.

[13] J. Yuh, G. Marani, and D. Blidberg. Applications of marine robotic vehicles. Intelligent Service Robotics, 4:221–231, 10 2011.

[14] P. Donta, T. Amgoth, and C. Annavarapu. Machine learning algorithms for wireless sensor networks: A survey. Information Fusion, 49,1–25, 2019.

[15] R. Coutinho and A. Boukerche. Exploiting mobility to improve underwater sensor networks. In Proceedings of the 16th ACM International Symposium on Mobility Management and Wireless Access, pages 89–94, 2018.

[16] B. Gutam, P. Donta, C. Sekhar, and Y. Hu. Optimal rendezvous points selection and mobile sink journey construction for data collection in WSNs. Journal of Ambient Intelligence and Humanized Computing, pages 1–12, 10 2021.

[17] P. Donta, T. Amgoth, and C. Annarapu. Machine learning algorithms for wireless sensor networks: A survey. Information Fusion, 49,1–25, 2019.

[18] R. Coutinho and A. Boukerche. Exploiting mobility to improve underwater sensor networks. In Proceedings of the 16th ACM International Symposium on Mobility Management and Wireless Access, pages 89–94, 2018.

[19] E. Simetti and G. Casalino. Manipulation and transportation with cooperative underwater vehicle manipulator systems. IEEE Journal of oceanic engineering, 42(4):782–799, 2016.

[20] N. Tsogkas, V. De Carolis, and D. Lane. Energy-constrained informative routing for AUVs. In OCEANS 2016-Shanghai, pages 1–5, IEEE, 2016.

[21] K. Li, C. Shen, and G. Chen. Energy-constrained bi-objective data muling in underwater wireless sensor networks. In The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), pages 332–341. IEEE, 2010.

[22] D. Munbabin, P. Corke, I. Vasilescu, and D. Rus. Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., pages 2091–2098, 2006.

[23] 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 Transactions on Mobile Computing, 17(2):404–418, 2018.

[24] M. Doniec. Autonomous underwater data muling using wireless optical communication and agile AUV control. PhD thesis, Massachusetts Institute of Technology, 2013.

[25] H. Nam. Data-gathering protocol-based AUV path-planning for long-duration cooperation in underwater acoustic sensor networks. IEEE Sensors Journal, 18(21):8902–8912, 2018.

[26] G. Han, S. Shen, H. Wang, J. Jiang, and M. Guizani. Prediction-based delay optimization data collection algorithm for underwater acoustic sensor networks. IEEE Transactions on Vehicular Technology, 68(7):6926–6936, 2019.

[27] S. Chen, Y. Chen, J. Zhu, and X. Xu. Path-planning analysis of AUV-aided mobile data collection in uwa cooperative sensor networks. In 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pages 1–5, 2020.

[28] F. Teixeira, N. Moreira, R. Campos, and M. Ricardo. Data muling approach for long-range broadband underwater communications. In 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 1–4, 2019.

[29] G. Han, Z. Zhou, T. Zhang, H. Wang, L. Liu, Y. Peng, and M. Guizani. Ant colony-based complete coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines. IEEE Transactions on Vehicular Technology, 69(8):8959–8971, 2020.

[30] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun. AUV-aided energy-efficient data collection in underwater acoustic sensor networks. IEEE Internet of Things Journal, 7(10):10010–10022, 2020.

[31] J. McMahon and E. Plaku. Autonomous underwater vehicle mine countermeasures mission planning via the physical traveling salesman problem. In OCEANS 2015 - MTS/IEEE Washington, pages 1–5, 2015.

[32] J. B. Robinson. On the Hamiltonian game (a traveling-salesman problem). RAND Corporation, Santa Monica, CA, 1949.

[33] G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59(2):231–247, 1992.

[34] J. Holland et al. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[35] Z. Wang, J. Guo, M. Zheng, and Y. Wang. Uncertain multiobjective traveling salesman problem. European Journal of Operational Research, 241(2):478–489, 2015.

[36] Q. Jiang, R. Sarker, and H. Abbass. Tracking moving targets and the non-stationary traveling salesman problem. Complexity International, 11(2005):171–179, 2005.

[37] G. Hay, N. Johnson, and A. Zelikovsky. The moving-target traveling salesman problem. Journal of Algorithms, 49(1):153–174, 2003. 1998 European Symposium on Algorithms.

[38] C. Gumbella, J. Naoum-Sawaya, and B. Ghaddar. The vehicle routing problem with floating targets: Formulation and solution approaches. INFORMS Journal on Computing, 30(3):554–569, 2018.

[39] R. S. Sutton and A. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[40] S. Wang and Y. Shin. Efficient routing protocol based on reinforcement learning for data muling with underwater mobile nodes. IEEE Access, 7:82027–82037, 2019.

[41] X. Li, X. Hu, W. Li, and H. Hu. A multi-agent reinforcement learning routing protocol for underwater optical sensor networks. In ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pages 1–7, 2019.

[42] V. Di Valerio, F. Lo Presti, C. Petrioli, L. Picari, D. Spaccini, and S. Basagni. Carma: Channel-aware reinforcement learning-based multi-
Q. Fu et al.: Reinforcement Learning-based Trajectory Optimization for Data Muling with Underwater Mobile Nodes

path adaptive routing for underwater wireless sensor networks. IEEE Journal on Selected Areas in Communications, 37(11):2634–2647, 2019.

[34] W. Xie, C. Di, H. Guo, and S. Li. Reinforcement learning based stochastic shortest path finding in wireless sensor networks. IEEE Access, 7:157807–157817, 2019.

[35] K. G. Omke, M. S. Mollel, L. Zhang, Q. H. Abbasi, and M. A. Imran. Energy optimisation through path selection for underwater wireless sensor networks. In 2020 International Conference on UK-China Emerging Technologies (UCET), pages 1–4, 2020.

[36] Y. Chen, J. Zhu, L. Wan, S. Huang, X. Zhang, and X. Xu. Acou-asfa fusion dynamic coded cooperation routing for different scale multi-hop underwater acoustic sensor networks. IEEE Access, 8:186773–186788, 2020.

[37] S. Han, L. Li, and X. Li. Deep q-network-based cooperative transmission joint strategy optimization algorithm for energy harvesting-powered underwater acoustic sensor networks. Sensors, 20(22), 2020.

[38] N. Javaid, O. A. Karim, A. Sher, M. Imran, A. U. H. Yasar, and M. Guizani. Q-learning for energy balancing and avoiding the void hole routing protocol in underwater sensor networks. In 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), pages 702–706, 2018.

[39] X. Ye and L. Fu. Deep reinforcement learning based mac protocol for underwater acoustic networks. In Proceedings of the International Conference on Underwater Networks & Systems, WUWNET’19, New York, NY, USA, 2019. Association for Computing Machinery.

[40] Z. Jin, Q. Zhao, and Y. Su. Rear: A reinforcement-learning-based routing protocol for congestion-avoided underwater acoustic sensor networks. IEEE Sensors Journal, 19(22):10881–10891, 2019.

[41] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao. Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. IEEE Journal on Selected Areas in Communications, 36(9):2059–2070, 2018.

[42] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almailta, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. IEEE Access, 7:48572–48634, 2019.

[43] C. T. Ciecik, H. Gultekin, B. Tavli, and H. Yanikomeroglu. UAV base station location optimization for next generation wireless networks: Overview and future research directions. In 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), pages 1–6, 2019.

[44] A. Sharma, P. Vanjani, N. Palival, C.M.W. Basnayaka, D.N.K. Jayakody, H. Wang, and P. Muthuchidambaranath. Communication and networking technologies for UAVs: A survey. Journal of Network and Computer Applications, 168:102739, 2020.

[45] Javier Busquets, Federico Zilic, Claudio Aron, and Ricardo Manzolli. AUV and ASV in twinned navigation for long term multipurpose survey applications. In 2013 MTS/IEEE OCEANS - Bergen, pages 1–10, 2013.

[46] P. Donga, T. Amgoth, and C. Annavarapu. Delay-aware data fusion in duty-cycled wireless sensor networks: A Q-learning approach. Sustainable Computing: Informatics and Systems, 33:100642, 2022.

[47] k. Dinesh, P. Donga, and T. Amgoth. EDGF: Empirical dataset generation framework for wireless network networks. Comput. Commun., 180:48–56, 2021.