Trajectory Scheduling for Timely Data Report in Underwater Wireless Sensor Networks

Ning Wang and Jie Wu
Department of Computer and Information Sciences, Temple University, USA
Email: {ning.wang, jiewu}@temple.edu

***Abstract*—This paper considers underwater wireless sensor networks (UWSNs) for surveillance and monitoring. Sensors are distributed in several key sections along the seafloor to record the surrounding environment, for example, monitoring oil pipelines and submarine volcanoes. Due to the need for timely data reporting and the fact that underwater communications suffer from a significant signal attenuation, homogeneous autonomous underwater vehicles (AUVs) are sent to retrieve information from the sensors, and periodically surface to report the collected data to the sink. In this paper, considering the huge energy consumption of surfacing and diving, our objective is to determine a trajectory schedule for the AUVs so that the total amount of surfacing for all the AUVs are minimized, and the data is reported to sink within the deadline. We first investigate the influence of different movement directions of AUVs, and provide the optimal solution to minimize the amount of surfacing for multiple AUVs within the same sensor section. Then, we propose a greedy detouring scheme to collaboratively schedule the AUVs in adjacent sensor sections. Extensive experiments show that our trajectory scheduling improves performance significantly.**

***Index Terms*—Underwater wireless sensor networks, trajectory scheduling, autonomous underwater vehicle.**

I. Introduction

Underwater monitoring has emerged as a vital part of ocean research. Its applications range among the oil industry, telecommunications, science, and the military [1]. Research project such as the North East Pacific Time-series Underwater Networked Experiment (Neptune Project) [2] does regional-scale underwater ocean observation. They collect data on physical, chemical, biological, and geological aspects of the ocean over long time periods, supporting research on complex earth processes in ways that were not previously possible. Military and homeland security applications involve securing or monitoring port facilities or ships in foreign harbors, as well as communicating with submarines and divers.

Underwater wireless sensor networks (UWSNs) appear to be a promising technique for underwater monitoring [1]. They collect data at spatial and temporal scales that are not feasible with existing instrumentation. Sensors are spatially distributed to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. Currently, sensors are usually connected to a buoy by a cable, which is expensive. Thanks to recent advances in the field of robotics, various mobile robots with superior capabilities such as autonomous underwater vehicles (AUVs) have been introduced. AUVs are endowed with multiple communication and navigation devices, as well as powerful computing abilities to collect data. The combination of the above two techniques makes new underwater monitoring applications feasible and cost-effective.

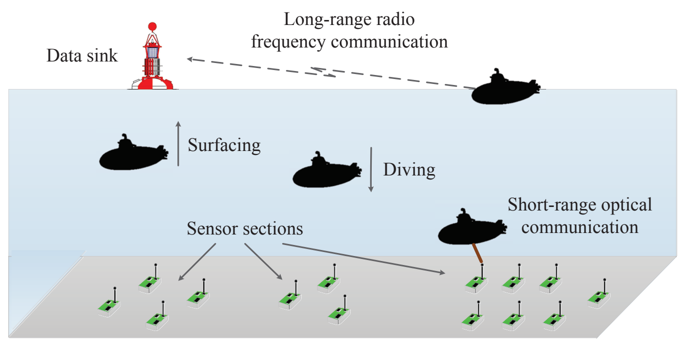
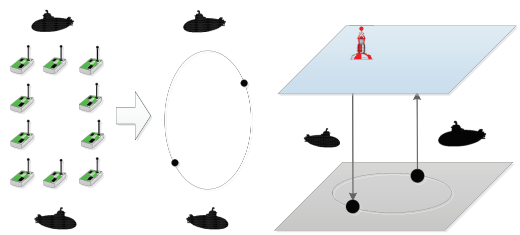


Fig. 1. An illustration of the network model

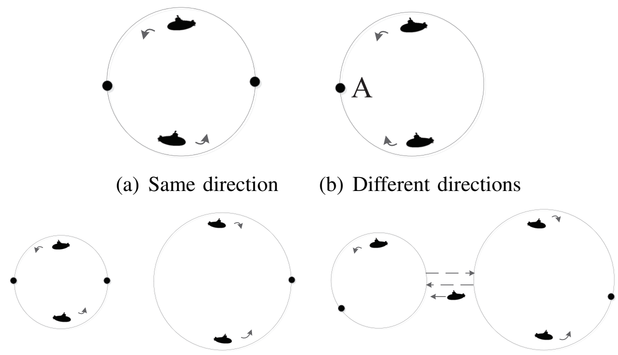
In this paper, we consider a UWSN with homogeneous AUVs. Sensor nodes are deployed on the sea-floor and keep monitoring the environment. For example, every sensor generates a new data per second. Since the wireless communication suffers a significant signal attenuation, AUVs are assigned to move around sensors and collect the data in sensors' buffers. AUVs re-surface periodically, and send all the generated data to the sink within the deadline. Fig. 1 illustrates the scenario. Since sensors keep generating data, AUVs have to conduct round-trips to collect data from a certain sensor repeatedly. We abstract the trajectory of an AUV into a circle, called a *cyclic tour*. Multiple AUVs might share the same cyclic tour, with different starting positions to reduce the waiting delay for data. For example, in Fig. 2(a), the trajectory of two AUVs is abstracted into a circle, where the two dark points represent the surfacing points. A better illustration of the surfacing points is shown in Fig. 2(b).

According to [1], surfacing and diving is costly. In this paper, our objective is to minimize the amount of resurfacing, and simultaneously satisfy the deadline constraint of data. It is a trajectory scheduling problem. Specifically, we study the following three problems in sequence. (1) Given the cyclic tour and the number of corresponding AUVs in this cyclic tour, we determine the homogeneous movement of each AUV that minimizes the amount of surfacing. (2) We study a more general case, in which the movement of AUVs can be different in a cyclic tour. (3) Given a monitoring area that includes multiple cyclic tours, we study a collaborative AUV trajectory



(a) The circle abstraction (b) The surfacing points

Fig. 2. The trajectory abstraction of AUVs, where the trajectory of AUVs can be abstracted into a circle, called cyclic tour. The dark point in Fig. 2(a) is the surfacing point. A 3-D representation is shown in Fig. 2(b)



(a) Same direction (b) Different directions

(c) Without detouring (d) With detouring

Fig. 3. Different scheduling strategies

planning, where the cyclic tours are merged, through AUVs' detouring, to further reduce the amount of surfacing.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to propose this multiple AUV trajectory scheduling problem. The different movement directions of AUVs and the multiple cyclic tours merging are all considered.
- The optimal schedule for minimizing the amount of surfacing for data collection in one cyclic tour is provided, considering the same or different movement directions.

The remainder of the paper is as follows. The related work is introduced in Section II. The overview of both the network model and the problem formulation is shown in Section III. The trajectory scheduling problem and proposed solution are presented in Section IV. The experimental settings and results are shown in Section V. We conclude the paper in Section VI.

II. Related Work

In this section, we capture some important issues arising from the design of trajectory scheduling optimization. This trajectory schedule evolves from mobile sinks in wireless sensor networks (WSNs), and from data ferries in delay tolerant networks (DTNs) for data collection and routing [3].

1) Single mobile vehicle: [4, 5] is focused on the scheduling of individual mobile sink. Specifically, in [4], the authors considered a similar network model as the model in this paper, but only one AUV is used. An AUV needs to collect data from the 3-D underwater sensors and then periodically surfaces to send the data to the sink. Clearly, the collaborative routing problem does not exist in this scenario.

2) Multiple mobile vehicles with collaboration: To the best of our knowledge, limited work has been done in the sub-area of the collaborative trajectory schedule. In [6, 7], the authors considered the collaborative mobile charging and coverage problem in WSNs. That is, mobile vehicles take the role of chargers to provide energy to sensors. In [6], their trajectories can overlap, so that the sensor is global and collaboratively covered by several mobile vehicles. The key difference between our work and their works is that the AUV resurfacing process brings one more factor while scheduling.

III. Network Model and Problem Formulation

A. Network Model

We consider a wireless sensor network to monitor the sea floor by deploying sensors at a particular area. The sensors keep generating data, and the data should be transmitted to the sink within the deadline. Multiple AUVs are assigned to this area and endlessly collect data from the uniformly distributed sensors. An applied scenario is that of submarine oil pipeline monitoring [8]: sensors are deployed in several key sections of this area, such as the pipe joints. To predict the oil leaking and fix the pipeline in a timely manner, the data generated by the sensors should be sent to the sink within the deadline. Another applied scenario which we consider is to monitor a submarine volcano [2]. Sensors are deployed in the crater of the volcano. The data are collected to predict the tsunamis caused by the volcanic explosion. Compared to the depth of the sea, the variation of terrain can be ignored. It is reasonable to assume that the sensors are distributed in 2-D space, which is parallel to the water surface. AUVs surface periodically to transmit the collected data to the sink through wireless communication.

B. Problem Formulation

Since the cost of surfacing and diving is costly [1], the schedule objective is to minimize the amount of surfacing in the whole network. In [7], they provide a trajectory planning method to generate the cyclic tours for AUVs. Therefore, we assume that the cyclic tours and the corresponding number of AUVs in each cyclic tour are given, and focus on the trajectory schedule of AUVs. This multiple AUVs trajectory schedule problem can be formalized as follows:

Assume that there exist n cyclic tours with lengths $\{c_1, c_2, \cdots, c_n\}$. Sensors in these cyclic tours keep generating data endlessly. The number of homogeneous AUVs in this n cyclic tours is $\{k_1, k_2, \cdots, k_n\}$, respectively, and the whole number of AUV is N, $\sum_{i=i}^{n} k_i = N$. The trajectory scheduling problem is to minimize the whole amount of surfacing of N AUVs, under the constraint that all the data generated by sensors can be transmitted to the sink within the deadline, T.

In the following paper, if the cyclic tour is not specified, we denote the length of a tour and the number of AUVs in this

(a) Any node in Fig. 3(a) (b) Node A in Fig. 3(b)

Fig. 4. An illustration of the different movement schedules to the maximal delay. For simplicity, the depth of the sea is 0. The length of the tour is c, and the speed of AUVs is v. In Fig. 3(a), AUV will collect data from sensors every $\frac{c}{2v}$. However, node A in Fig. 3(b) will be visited by AUVs every $\frac{c}{v}$.

tour as c and k, respectively, for simplicity of explanation. The depth of the sea is denoted as d, and the surface frequency for each AUV is denoted as m in one cyclic tour. We assume that the AUV has the same speed, v, whether they are moving in 2-D space, surfacing, or diving. Though these three speeds are not the same in reality, we can do sea depth transformations. For example, suppose the surfacing speed is 1 m/s, the collecting data speed of AUVs is 2 m/s , and the depth of the sea is 1000 m. This is the same as considering that the speed of AUVs is always 2 m/s, but the depth of the sea is changed into 2000 m.

IV. Trajectory Schedule Problem

In this section, we start with a trajectory schedule in the single cyclic tour. Basically, the AUVs can move in the same direction or different directions as shown in Figs. 3(a) and 3(a). Then, we extend the schedule in the multiple cyclic tours, by exploring the AUV detouring as shown in Figs. 3(c) and 3(d).

A. Scheduling AUV in the same direction

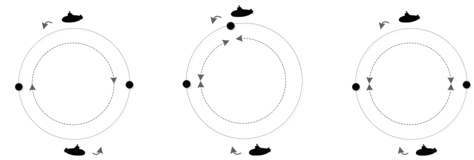
Starting with the most basic case, we assume all the AUVs in one cyclic tour all move in the same manner. An example is shown in Fig. 3(a), where each AUV moves in the same direction and surfaces after a distance of $\frac{c}{m}$. Then, the actual travel length for each AUV is $c + 2md$ to arrive at the same sensor again. From the view of data, its delay is made up of three parts: the delay from data generation to AUV collection, the delay from AUV collection to AUV surfacing, and the surfacing delay. If the k AUV are uniformly distributed, all sensors wait $\frac{2md+c}{kv}$ before an AUV reaches the sensor again. Any other unbalanced distribution of AUVs will cost a larger waiting delay for some sensors. Besides, after data is collected by an AUV, it takes $\frac{c}{mv}$ at most to reach the surfacing position. Another $\frac{d}{v}$ interval is needed for surfacing. So, the data delay can be bounded by using the following equation:

$$\frac{1}{v}\left(\frac{2md+c}{k}+\frac{c}{m}+d\right) \tag{1}$$

When $m = \sqrt{\frac{ck}{2d}}$, the data delay is minimized. That is, AUVs surface, when they traverse $\sqrt{\frac{2cd}{k}}$ distance. By calculating the smallest m by Eq. 1 to make sure the maximal delay is within the deadline, the amount of surfacing is minimized.

B. Scheduling AUV in the different directions

Suppose k is an even number, $\frac{k}{2}$ AUVs move in one direction and $\frac{k}{2}$ AUVs move in the opposite directions as shown Fig. 3(b). A more general case is our future work. The



(a) Schedule 1 (b) Schedule 2 (c) Schedule 3

Fig. 5. An example to explain the surfacing interval for different movement models within the deadline. The deadline is $\frac{c+3d}{v}$, and $4d = c$. If two AUVs travel in the same direction, as shown in Fig. 5(a), they have to surface no more than $\frac{c/2+2d}{v}$ interval. If they keep moving in the opposite directions, and surface when they encounter, as shown in Fig. 5(b), the surfacing interval is no more than $\frac{c/2+d}{v}$. However, if one AUV always waits for the surfacing AUV until it comes back, then they traverse the tour together; with each traversing half of the tour, they can accomplish the surfacing interval $\frac{c/2+2d}{v}$, but can save one surfacing as shown in Fig. 5(c).

Algorithm 1 AUVs Schedule for One Cyclic Tour

Input: The depth of sea, d, the length of cyclic tour, c, the number of AUV, k, and the data deadline, T.
Output: The trajectory schedule of AUVs in this tour.

```
if k is even then
    distribute k/2 pair of AUVs with different movement directions evenly in the tour.
    Schedule k/2 AUVs to always surface; the another k/2 AUVs wait for them to come back. These k AUVs begin to move just to ensure that the oldest data can be sent to the sink within the deadline.
else
    Schedule the k − 1 AUVs as above to get a schedule.
    Schedule the k AUVs in the same direction.
    Compare these two schedules, and use the better one.
```

advantage of this type of schedule is that when two AUVs encounter, one AUV can surface, carrying the data collected by two AUVs. The other AUV keeps moving to collect data, thus, saves one surfacing time.

To minimize the amount of surfacing, we should minimize the maximal delay of data to sink. The reason is that, if the maximal delay of data to sink is minimized, after surfacing, the AUVs can wait as long as possible before the next surfacing, and thus the amount of surfacing is minimized. However, if AUVs move in the different direction, as shown in Fig. 3(b), some sensors will be visited by two AUVs together, which causes a long waiting delay for the next visit, and it is bad to minimize the maximal delay in a tour. If AUVs move in the same direction as shown in Fig. 3(a), all the sensors will be visited every half interval of the tour. A more detailed illustration is shown in Fig. 4.

Theorem 1. *To minimize the maximal delay of all the data to sink, AUVs moving in the same direction are always better than those moving in different directions.*

Proof. For movement with the same direction, when $m = \sqrt{\frac{ck}{2d}}$, it achieves the minimal value, $\frac{1}{v}(2\sqrt{\frac{2cd}{k}} + \frac{c}{k} + d)$. As for movement in the opposite directions, all the $\frac{k}{2}$ pairs of

AUVs are evenly distributed in the tour at the beginning. Any other distribution is worse by contradiction. If $d < \frac{c}{k}$, the minimal time interval for k AUVs to encounter each other is $\frac{1}{v}((\frac{\frac{2c}{k}-2d}{2}+2d)+d)$ seconds, which means that k AUVs keep moving. It is also the oldest data in the network. The oldest data still need the same amount of time to be collected by some AUVs. So, the minimum maximal delay that we can accomplish is as follows:

$$\frac{1}{v}(2(\cdot\frac{c}{k}+d)+d) = \frac{1}{v}(\frac{2c}{k}+3d) \geq \frac{1}{v}(2\sqrt{\frac{2cd}{k}}+\frac{c}{k}+d)$$

If $d \geq \frac{c}{k}$, the minimal time for two AUVs to encounter costs $\frac{1}{v}(\frac{2c}{k} - \lfloor\frac{2d}{\frac{2c}{k}}\rfloor \cdot \frac{2c}{k}+2d)+d)$, since in this case, once the surfacing AUV comes back, all the AUVs traverse together to collect data in the tour. It is the fastest way. The oldest data also need the same time to be carried by one surfacing AUV to the surfacing point. Another $\frac{d}{v}$ is the surface delay.

$$\frac{1}{v}(2\cdot(\frac{2c}{k} - \lfloor\frac{2d}{\frac{2c}{k}}\rfloor \cdot \frac{2c}{k}+2d)+d) \geq \frac{5d}{v} \geq \frac{1}{v}(2\sqrt{\frac{2cd}{k}}+\frac{c}{k}+d).$$

Thus, the minimum maximal delay of moving with different directions is worse than that with the same direction. □

Though scheduling the movement of AUVs in different directions cannot reduce the minimum maximal delay, the amount of surfacing of this strategy is less than that moving in the same direction. An example of three scheduling strategies for two AUVs is shown in Fig. 5, where $c = 4d$ and $T = \frac{c+3d}{v}$. If two AUVs move in the same direction, they have to surface twice every $\frac{c/2+2d}{v}$ time interval to meet the deadline. If two AUVs move in opposite directions without waiting, two AUVs only need to surface once every $\frac{c/2+d}{v}$. However, the optimal schedule is to surface once every $\frac{c/2+2d}{v}$ interval. For a general situation, we can get the following theorem:

Theorem 2. *For a tour with an even number, k, of AUVs, the optimal schedule for minimizing the amount of surfacing is to assign $\frac{k}{2}$ AUVs to surface every time of $T - \frac{c/k+d}{v}$.*

Proof. To minimize the amount of surfacing, which is the same as to maximize the surfacing interval, the oldest data after surfacing should be minimized. Clearly, it is equivalent to traversing the tour in the minimum time. For k AUVs, it takes at least $\frac{c}{kv}$ to traverse the tour, when the k AUVs are assigned the same length to cover the whole tour. Any other assignment will cause some AUVs to have a longer surfacing interval. So, the longest surfacing interval for k AUVs is to surface every $T - \frac{c/k+d}{v}$ interval. □

With Theorem 2, the optimal schedule for minimizing the surfacing number is that the k AUVs should be distributed $\frac{k}{2}$ pairs uniformly around the tour. The movement direction of AUVs will be staggered so that if one AUV moves clockwise, the next moves anti-clockwise, and so on for all k AUVs. Once an AUV encounters with another AUV, one AUV will pause and wait, while the other AUV surfaces. After the surfacing AUV dives back, all the AUVs will continue on their path.

Algorithm 2 Greedy Cyclic Tour Merging

Input: The sensor distribution $S = \{S_1, S_2, \cdots, S_n\}$

Output: The cyclic tour merging result

Calculate the optimal schedule for each cyclic tour, S_i.
while Merging can reduce surfacing numbers **do**
    Find S_i and S_j, which can reduce the amount of surfacing most by back and forth merging.
    if S_i and S_j are original cyclic tours **then**
        They are merged into cyclic tour M_i, $S \setminus \{S_i, S_j\}$, and $S = S \cup M_i$
    else
        Denote the cost of merged cyclic tour, $Cost_1$.
        Backtrack the later merged cyclic tour, M_i.
        Do circle merging; calculate the cost, $Cost_2$, for the three cyclic tour.
        Compare $Cost_1$ and $Cost_2$; use the smaller one.

C. Trajectory Schedule between Cyclic Tours

Instead of scheduling the AUVs within their cyclic tour, AUVs in adjacent tours can be scheduled collaboratively to further minimize the total amount of surfacing with detouring.

Definition 1. ***Sensor arc***. *The original cyclic tour, where the sensors are evenly distributed.*

Definition 2. ***Non-sensor arc***. *The cyclic tour generated by detouring, where no sensor exists.*

In Fig. 3(d), the sensor arcs and the non-sensor arcs are represented by the solid line and the dotted line, respectively.

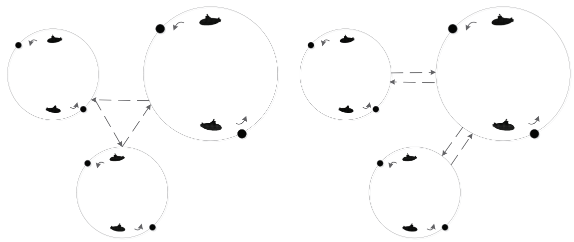
We propose a greedy merge strategy as follows: we calculate the amount of surfacing of each cyclic tour. Then, we try all the merging combinations, and merge the two tours, which can bring the most benefit, reducing the amount of surfacing mostly. Then, we keep doing this type of selection until the merging process cannot bring any benefit.

To avoid the exhaustive search for sensor section merge, a general criterion is given as follows: if p tours are merged together, the length of the merged tour is $L_p = \sum_{i=1}^{p}(c_i + l_i)$, where l_i is the detour distance for cyclic tour i. The number of AUVs in the merged tour is $K_p = \sum_{i=1}^{p} k_i$; then, the cyclic tour merging brings benefit, which is equal to

$$\begin{aligned}
&\frac{K_p}{T - \frac{1}{v}(\frac{L_p}{K_p}+d)} + \frac{k_{p+1}}{T - \frac{1}{v}(\frac{c_{p+1}}{k_{p+1}}+d)} - \frac{K_{p+1}}{T - \frac{1}{v}(\frac{L_{p+1}}{K_{p+1}}+d)} \geq 0 \\
&= K_p(\frac{1}{T - \frac{1}{v}(\frac{L_p}{K_p}+d)} - \frac{1}{T - \frac{1}{v}(\frac{L_{p+1}}{k_{p+1}}+d)}) \\
&+ k_{p+1}(\frac{1}{T - \frac{1}{v}(\frac{c_{p+1}}{k_{p+1}}+d)} - \frac{1}{T - \frac{1}{v}(\frac{L_{p+1}}{K_{p+1}}+d)}) \geq 0
\end{aligned}$$

By separating the l_{p+1} from the above function, we get

$$\begin{aligned}
&(c_{p+1}\frac{K_p}{k_{p+1}} + L_n\frac{k_{p+1}}{K_p})\frac{L_p + c_{p+1}}{K_{p+1}} - L_p c_{p+1}(\frac{1}{K_p} + \frac{1}{k_{p+1}}) \\
&> l_{p+1}(Tv - d - \frac{1}{K_{p+1}}(c_{p+1}\frac{K_p}{k_{p+1}} + L_p\frac{k_{p+1}}{K_p}))
\end{aligned}$$



(a) Back and forth merging (b) Circle merging

Fig. 6. Two types of the three-component merging method

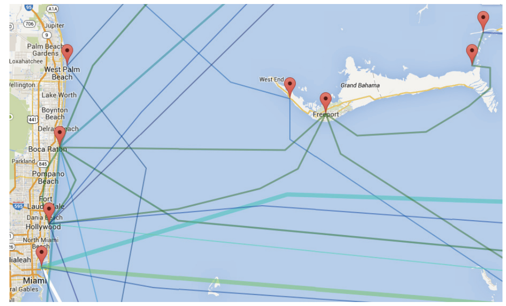


Fig. 7. The oil pipes in Florida, USA

Then, we can estimate the longest possible l_{p+1} by merging the current cyclic tour with other cyclic tours.

Due to the non-sensor arcs, the optimal schedule in the merged cyclic tour might not be the schedule that we discuss in Section IV-B. This is because the AUVs do not need to collect the data in non-sensor arcs. However, we can simply regard that the merged cyclic tour is made up by sensor arcs and we use the Algorithm 1 to schedule AUVs. Its performance can be bound by the following theorem.

Theorem 3. *There exists an* $1 + \frac{2l}{d}$ *approximation ratio between the schedule in Algorithm* 1 *and the optimal solution in the merged cyclic tour.*

Proof. We already know the optimal solution in a cyclic tour without non-sensor arcs. The idea is that if we treat non-sensor arcs as sensor arcs, the minimum maximal delay might increase, due to the extra sensors, and thus causes more surfacing. Similarly, if we delete the non-sensor arcs from the merged cyclic tour, the amount of surfacing might decrease. The approximate ratio of our proposed method is bounded by the above mentioned upper bound and lower bound in estimating the amount of surfacing in the merged cyclic tour. The approximate ratio is

$$\begin{aligned}\alpha &= \frac{T - \frac{1}{v}(\frac{c}{k} + d)}{T - \frac{1}{v}(\frac{c+2l}{k} + d)}\\ &= 1 + \frac{2l}{k(Tv - d) - (c + 2l)} \leq 1 + \frac{2l}{d}\end{aligned}$$

This theorem shows that when the detouring distance is small, our estimation is close to the optimal schedule. □

The above mentioned greedy algorithm discusses the tour merging in two adjacent cyclic tours. If three or more tours are merged together, there are two types of merge methods, as shown in Fig. 6. The first method is to find the two shortest paths between the three tours, then detour along with the shortest paths, called *back and forth merging*, as shown in Fig. 6(a). It can be regarded as iteratively merging for 2 adjacent cyclic tours twice. Another method is to merge along a circle, which connects the three tours, called the *circle merging*, as shown in Fig. 6(b). In a general sensor distribution, the Monte Carlo method can be used to find the shortest path between circles. To adapt 3-component merging into our greedy algorithm, we check whether the two picked cyclic tours are the original cyclic tours or not. If any one of them is a merged cyclic tour, we would backtrack the last merging and try 3-component merging. Then, we would like to compare these two merging methods, and choose the better method.

V. Experiments

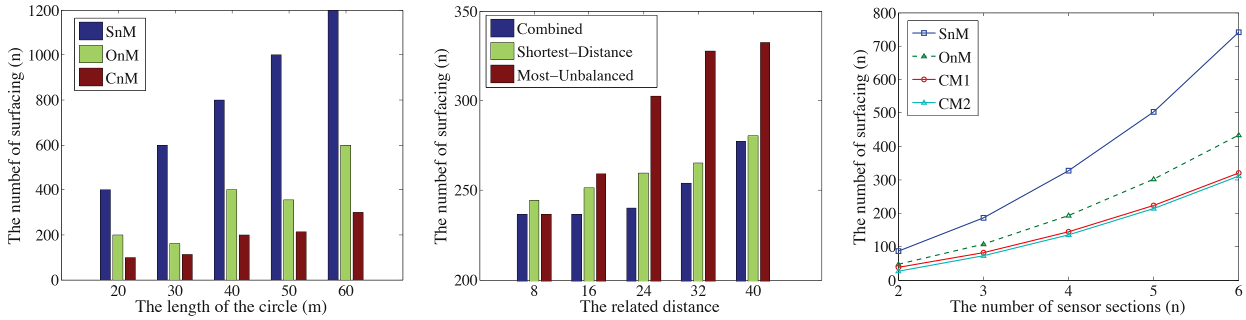
In this section, we compare several algorithms mentioned in this paper in real trace and synthetic trace. We first introduce the experiment settings and their parameters. Then, we show evaluation results.

1) Synthetic trace: We assume that the sensors are distributed evenly in rectangles. At the beginning of a second, the AUV leaves the current sensor, and arrives at a new sensor at the end of this second. In our experiment, we generate a detection area of 500m × 500m. The cyclic tours are assigned into 3 kinds of lengths (40m, 60m, and 80m) with random distances. The depth of the sea is 100m. We assume that the number of AUVs in each tour is proportional to its length.

2) Real trace: We use the data published in [9]. In this real trace, we mainly focus on the oil pipelines, BDNSi, Mid Atlantic Crossing (MAC) , GlobeNet, COLUMBUS II, III, WASACE, Americas II, cable of the Americas and BAHAMAS-2, which are among West Palm Beach, Boca Raton, and Freeport. Fig. 7 shows the network configuration. The AUV's speed is assigned as 20 knots, 16 knots, 12 knots, when they are diving, collecting data, and surfacing, respectively, according to [8]. The depth of the sea is 3682 m [10]. Initially, two AUVs are assigned in a pipe.

A. Schedule Methods

For AUVs scheduling in the single cyclic tour, we propose three schedule methods. We call them, *SnM* algorithm, same direction movement model without merging, *CnM* algorithm, different movement directions without merging, and OnM algorithm. The difference between OnM and CnM is that AUV will not wait. As for cyclic tour merging, we compare three types of merging methods, the shortest-distance-first merging, the most-unbalanced-first merging, and the greedy tour merging, called Combined algorithm, *CM*. Besides, to distinguish the two types of 3-component merging methods in CM algorithm, we denote the back and forth merging and circle merging, $CM1$ and $CM2$, respectively.



(a) Single cyclic tour (b) Multiple cyclic tours (c) Multiple cyclic tours

Fig. 8. The amount of surfacing of different schedules in synthetic network tour

B. Experiment Results

For multiple AUVs in the single cyclic tour, we compare the amount of surfacing, when AUVs move in the same direction or different movement directions in different cyclic tour lengths. From Fig. $8(a)$, we conclude that CnM algorithm can reduce the amount of surfacing significantly, compared to the SnM algorithm. The performance of OnM algorithm is in the middle. Then, we consider the cyclic tour merging, if multiple cyclic tours exist. The results in Fig. 8(b) show that the CM algorithm accomplishes the better performance than the shortest-distance-first merging and most-unbalanced-first merge, which indicates that only considering the distance between cyclic tours and the unbalance degree of AUVs in cyclic tour is not enough. Then, we will only use Combined algorithm as the merging method for the following experiments. In Fig. 8(c), the results show that CM1 merging strategy can further reduce the surfacing time. In the real pipeline network, the experiment result is shown in Fig. 9. In different data deadline, if we consider the different movements of AUVs and the cyclic merging of adjacent tours, the amount of surfacing can be reduced into less than a half.

VI. Conclusion

This paper considers the homogeneous autonomous underwater vehicles (AUVs) trajectory schedule problem in under water sensor networks (UWSNs). Several AUVs are used to retrieve information from the sensors and periodically surface to deliver the collected data to the sink within the data deadline. In this paper, we propose a trajectory schedule for the AUVs so that the total surfacing number for all AUVs is minimized. We first investigate the movement direction of AUVs in one cyclic tour. Then we explore the benefit by collaboratively scheduling the AUVs in the adjacent cyclic tours together. We compare our trajectory scheduling strategy with trajectory scheduling without collaboration in the theory and experiments. Extensive experiment results show that our schedule is much better than that without considering the movement directions and collaboration.

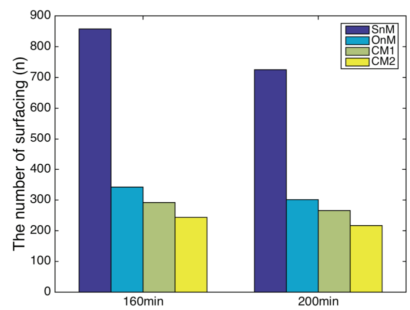


Fig. 9. The surfacing number in different deadlines in real pipe network

References

[1] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proceedings of the IEEE WCNC*, 2006, pp. 228–235.

[2] C. Barnes, M. Best, B. Bornhold, S. Juniper, B. Pirenne, and P. Phibbs, "The neptune project-a cabled ocean observatory in the ne pacific: Overview, challenges and scientific objectives for the installation and operation of stage i in canadian waters," in *Proceeding of the IEEE UT-SSC*, 2007, pp. 308–313.

[3] N. Wang and J. Wu, "A general data and acknowledgement dissemination scheme in mobile social networks," in *Proceedings of the IEEE MASS*, 2014.

[4] S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C. A. Phillips, and D. Turgut, "Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle," in *Proceeding of the IEEE INFOCOM*, 2014.

[5] L. Xue, D. Kim, Y. Zhu, D. Li, W. Wang, and A. O. Tokuta, "Multiple heterogeneous data ferry trajectory planning in wireless sensor networks," in *Proceeding of the IEEE INFOCOM*, 2014.

[6] R. Beigel, J. Wu, and H. Zheng, "On optimal scheduling of multiple mobile chargers in wireless sensor networks," in *Proceeding of the ACM MSCC*, 2014.

[7] J. Wu and H. Zheng, "On efficient data collection and event detection with delay minimization in deep sea," in *Proceedings of the ACM CHANTS*, 2014.

[8] http://en.wikipedia.org/wiki/Ocean.

[9] http://www.cablemap.info.

[10] http://en.wikipedia.org/wiki/Ohio.