sensors

MDPI

Article

HAS^4: A Heuristic Adaptive Sink Sensor Set Selection for Underwater AUV-Aid Data Gathering Algorithm

Xingwang Wang [1,2,*], Debing Wei [3], Xiaohui Wei [1,2,*], Junhong Cui [2] and Miao Pan [3]

[1] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

[2] College of Computer Science & Technology, Jilin University, Changchun 130012, China; junhong_cui@jlu.edu.cn

[3] Department of Electrical & Computer Engineering, University of Houston, Houston, TX 77204, USA; dwei3@uh.edu (D.W.); mpan2@uh.edu (M.P.)

* Correspondence: xww@jlu.edu.cn (X.W.); weixh@jlu.edu.cn (X.W.);
Tel.: +86-155-9054-5413 (Xingwang Wang)

Received: 30 September 2018; Accepted: 15 November 2018; Published: 23 November 2018

Abstract: In this paper, we target solving the data gathering problem in underwater wireless sensor networks. In many underwater applications, it is not quick to retrieve sensed data, which gives us the opportunity to leverage mobile autonomous underwater vehicles (AUV) as data mules to periodically collect it. For each round of data gathering, the AUV visits part of the sensors, and the communication between AUV and sensor nodes is a novel high-speed magnetic-induction communication system. The rest of the sensors acoustically transmit their sensed data to the AUV-visit sensors. This paper deploys the HAS^4 (Heuristic Adaptive Sink Sensor Set Selection) algorithm to select the AUV-visited sensors for the purpose of energy saving, AUV cost reduction and network lifetime prolonging. By comparing HAS^4 with two benchmark selection methods, experiment results demonstrate that our algorithm can achieve a better performance.

Keywords: underwater data gathering; underwater wireless sensor network; AUV

1. Introduction

Ocean Big Data (OBD) is an emerging research area that benefits ocean environmental monitoring, offshore exploration, disaster prevention, and military surveillance. Since the oceans, rivers and lakes cover 71% of our planet, and the traditional "deploy, sense, retrieve, and post-process" routine for these sensing activities is difficult and costly for humans in large area applications. Deploying an underwater sensor network is very efficient for monitoring physical factors (e.g., temperature, salty, light) by a multitude of sensing modalities. Since it is impractical to connect every possible underwater sensor by wire, wireless communication is currently the dominant data delivery technique for numerous underwater applications. Current underwater communication fails in three categories, acoustic, electromagnetic and optical [1]. Benefits and drawbacks of these methods are compared in Table 1.

Table 1. Comparison of underwater transmission modes.

Communication Method	Data Rates	Communication Ranges	Channel Dependency
acoustic	Kbps	km	Multi-path, Doppler, temperature, pressure, salinity, environment sound noise
electromagnetic	Mbps	10–100 m	Conductivity
optical	Mbps	10–100 m	Light scattering, Line-of-sight communication, Ambient light noise

Due to serious attenuation, except for acoustic waves, the effective communication range in sea water is no more than 100 m, which leads to acoustic communication being the dominant data delivery technique for numerous underwater applications. However, acoustic communication is the most difficult obstacle to the realization of these networks, which is caused by limited bandwidth (currently, few hundreds of bits per second), long propagation delay and unchangeable power-battery [2–5].

Fortunately, for underwater environments, magnetic-induction (MI) as an alternative wireless transmission mode besides acoustic communications obtains high bandwidth connectivity (up to 10 Mbps) with energy per-bit orders of magnitude lower than that of acoustic communications [6]. However, for MI, it allows nodes to robustly communicate only when they are a few meters from each other (usually less than 10 m) [7]. Since mobile autonomous underwater vehicles (AUVs) can hover close enough to a sensor node [8,9], using an AUV and multi-modal communications (e.g., acoustic and MI) can enhance the performance of UWSNs and can enable many critical applications. In this scenario, as shown in Figure 1, which leverages the advantages of different wireless underwater transmission modes and the mobility of AUVs, AUVs could be made aware of the sensing task and visit part of the sensors (sink nodes) for data gathering, while the rest of the sensors could acoustically forward their sensing data to sink nodes.

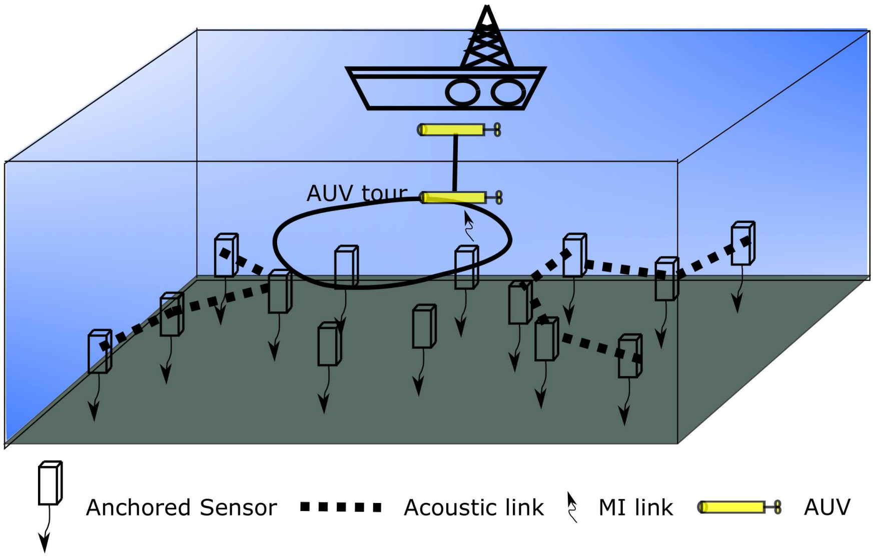

Figure 1. AUV(autonomous underwater vehicles)-aided data gathering.

Although AUVs can act as data mules and travel from node to node across a sparsely deployed sensor network to collect data [10], the slow mobility and limited energy that constrain them means that they can only visit a very small number of the sensors. Thus, much sensing data should be acoustically forwarded sensor by sensor. In such a procedure, packets may experience multiple relay nodes before reaching a sink node, which means that an enormous amount of energy is consumed on data forwarding along the path [11]. In addition, with the increasing number of acoustic communications, the probability of package collision and transmission energy consumption could be notably magnified, which will reduce the network performance. Moreover, reducing the total energy consumption and acoustic communications is not enough for prolonging the network lifetime, as some popular sensors, e.g., some neighbours around the sink, may run out of the energy faster than the others, which will cause energy consumption unbalance. Furthermore, due to the channel interference and limitation of acoustic channel capacity, a single sink node may not be able to receive all of the sensing data. Hence, multiple sink nodes are required to collect the whole sensing data. Due to the limitation of AUVs' energy, sink nodes should be carefully selected.

This paper addresses all of these problems mentioned above to solve the underwater data gathering problem. Targeting the reduction of the total energy consumption on acoustic communication and balancing the energy consumption among the networks, we propose HAS^4, a Heuristic Adaptive sink sensor set selection algorithm for underwater AUV-aid data gathering. HAS^4 has both centralized and distributed versions. They both achieve energy efficient by decreasing the possible acoustic communication or reducing the maximum acoustic link length. HAS^4 prolongs the network lifetime by adaptively selecting short lifetime sensors as sink nodes to eliminate the energy consumption on data transmission.

The contributions of this paper are as follows:

1. We use a mix integer linear problem (MILP) to formulate the underwater AUV-aid data gathering problem. In the formulation, we take the energy consumption on acoustic communication, AUV traveling distance, and energy balance into account to prolong the network lifetime and reduce the AUV traveling cost.
2. We propose HAS^4 to solve the MILP (mix integer linear problem). HAS^4 is independent from network topologies, and has parameters for trading off between AUV traveling cost and network lifetime. By providing both centralized and distributed versions, HAS^4 widens the applicable scenarios.
3. We conduct simulations to verify the efficiency of HAS^4. Experiment results show both centralized and distributed HAS^4 prolong the network lifetime with low AUV traveling cost.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3, we introduce the problem of AUV-aid underwater data gathering problem. Centralized and distributed algorithms are proposed in Section 4 to solve the problem. Section 5 reports some simulation results. Finally, Section 6 concludes the paper.

2. Related Work

Generally, underwater data gathering systems fall into two categories: multi-hop and AUV-aid approaches.

2.1. Multi-Hop Approaches

In these approaches, underwater sensing data are forwarded hop-by-hop along routing paths. Thus, routing technique plays the key role in these kinds of approaches. Different from terrestrial wireless sensor networks (WSN), the underwater nodes move with water flows involuntarily in the underwater environment [12–16], and routing protocols designed for WSNs can not be directly applied to UWSNs. Thus, the research focus in this area is to design energy-efficient and reliable routing protocols.

Based on the information required in the protocols, most underwater routing can be classified into location-based and depth-based. Location-based routing protocols, such as vector-based forwarding [17], hop-by-hop vector-based forwarding [18], depth-controlled routing [19], etc., assume that each node knows its location and each packet contains the position of the sender. Depth-based routing protocols, such as depth-based routing [20], directional depth-based routing [21], and depth-based multi-hop routing [22], etc., use depth information to route the packets instead of the full location coordinates.

Due to the energy-extensive consumption of underwater acoustic communication and limited battery power, multi-hop approaches will reduce the lifetime of the network.

2.2. AUV-Aid Approaches

In these approaches, AUVs act as data mules to deliver underwater data. Vasilescu et al. illustrated the feasibility of leveraging AUVs for underwater data collection [10]. Tekdas et al. demonstrated the increase of the network lifetime via AUV gathering data from sensor nodes [23].

Research on AUV-aid data gathering has focused primarily on how to reduce the length of AUV's traveling path [24]. Basagni et al. take the Value of Information (VoI) that value of sensing data decreases over time into account and propose a Greedy and Adaptive AUV Path-planning (GAAP) algorithm [25]. Moreover, the cooperation among AUVs is also studied in 3D underwater environments [26]. However, these works are not using MI communications to improve the reliability and throughput.

In this paper, we leverage the advantages of different wireless underwater transmission modes and the mobility of AUVs to provide a cyber interconnection scheme that enables distributed and efficient data delivery from underwater sensors to the surface stations.

3. AUV-Aid Underwater Data Gathering Problem

In this section, we formulate the AUV-aid data gathering problem with the objectives of minimizing the AUV travel distance, minimizing the total energy consumption and maximizing the network lifetime.

3.1. Motivation and Overview

Leveraging the sub-sea MI communication model in [27], as shown in Figure 2, the MI channel capacity can exceed 5×10^6 bps. Moreover, we have built up a sub-sea MI testbed as shown in Figure 3. A data transmission based on Quadrature Phase Shift Keying (QPSK) modulation scheme with symbol rate equal to 100 kHz is demonstrated in Figure 4. Limited by the size of the water tank, we were only able to separate the transmitter and receiver at the distance of 0.7 m. The testing results show that the data transmission is 100% successful. Thus, leveraging the advantages of reliable and high speed MI communication, long distance acoustic communication and the mobility of AUVs, UWSNs can dramatically reduce energy cost on data gathering with high reliability.

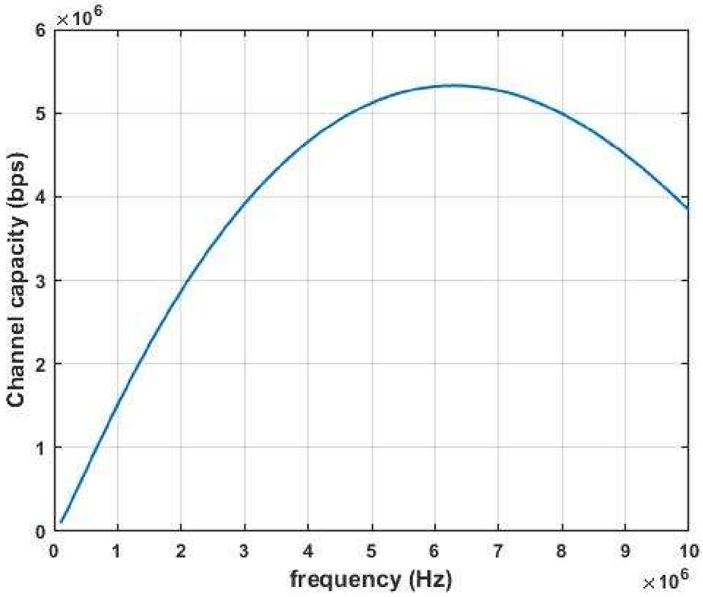

Figure 2. Channel capacity of subsea MI (magnetic-induction) wireless communications with transmission distance = 1 m and coil radius = 0:05 m.

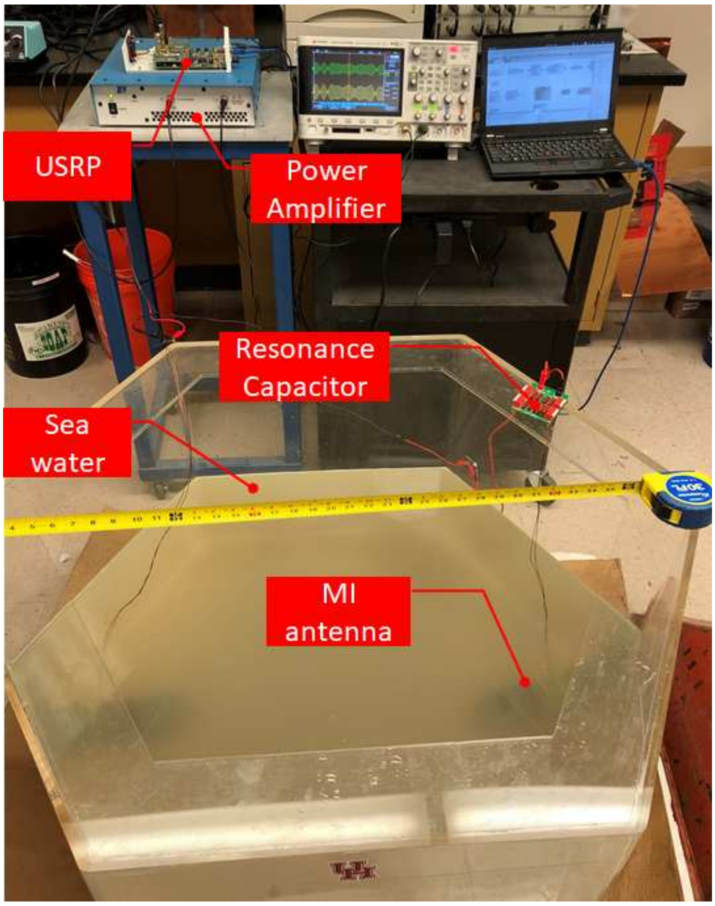

Figure 3. Sub-sea MI communication testbed.

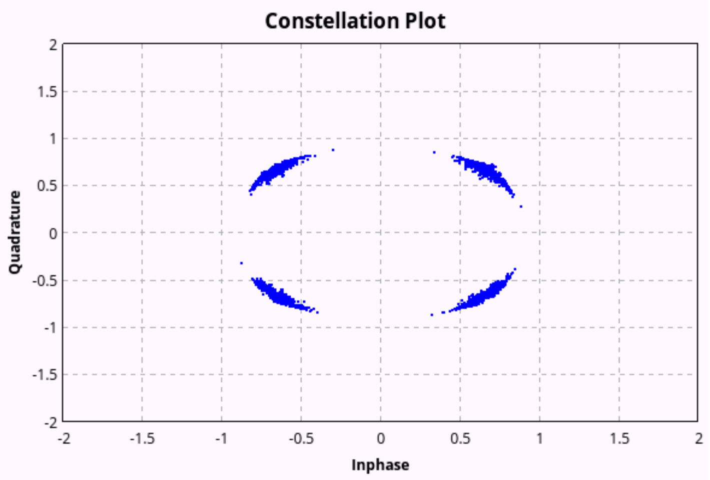

Figure 4. QPSK (quadrature phase shift keying) data transmission results for sub-sea MI (magnetic-induction) wireless communications.

Consider an AUV and many sensors form a underwater wireless sensor network. Each sensor is anchored to the target area for ocean monitoring/sensing applications. Both magnetic-induction (MI) and acoustic communication modules are equipped to every sensor for short distance high speed communication and long distance low speed communication, respectively. We assume that the perceived data are periodically gathered and delivered to the designated surface station. During each period, the AUV travels from the surface station, dives into the water and visits a subset of the sensors for data collection, brings the collected data back to the surface station, and prepares for the next round data collection. Since the MI communication range is quite short, the AUV should visit the sensor position exactly for MI communication. These sensors that are not visited by the AUV transmit their sensing data to chosen sensors via acoustical channels.

Suppose there are $S = \{s_1, s_2, \cdots, s_n\}$ sensors, and the i-th sensor s_i is anchored at the coordinate $c_i = [x^i, y^i, z^i]$. Let $M = \{m_1, m_2, \cdots, m_k\}$ denote the acoustic sub-channel set. The transmission range on the m acoustic sub channel is m^t, and the interference range of this channel is m^i. The neighbor sensor set for each sensor can be obtained according to their coordinates and communication range. For simplicity, in this paper, we assume the AUV's speed is constant at V, and the MI communication rate is constant at r_{mi}.

3.2. Problem Formulation

As mentioned above, we have three objectives for this problem, and the first objective is to minimize the AUV traveling distance:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}, \tag{1}$$

where variable d_{ij} is the distance between sensor i and sensor j and x_{ij} is a binary variable that indicate the AUV travel path. If the AVU travel from sensor i to sensor j, $x_{ij} = 1$. Otherwise, $x_{ij} = 0$.

The second objective is to minimize the total energy consumption:

$$\sum_{i=1}^{n} ({p_i}^{in} \sum_{j \in N_i} D_{ji} + {p_i}^{out} \sum_{j \in N_i} D_{ij} + p_i^{out}(mi) \sum_{j \in N_i} D_{ij}), \tag{2}$$

where D_{ij} is the data amount from sensor i to sensor j. p_i^{out} and p_i^{in} denote the energy consumption of sensor i on transmitting and receiving through acoustic channels, respectively. N_i is the neighbour sensor set of sensor i.

The final objective is to maximize the network lifetime. We define the network lifetime as the sensing round that the first sensor runs out of energy. Therefore, the objective of maximizing the network lifetime can be seen as maximizing the lifetime that is the minimum lifetime in the network:

$$\max \min \frac{e_i}{p_i^0 + p_i^{out}(mi)}, \quad \forall i \in S, \tag{3}$$

where p_i^0 is the energy consumption on this round predefined sensing task, e_i is the residual energy of sensor i and $p_i^{out}(mi)$ is the energy consumption on transmitting its own data through MI.

To achieve the above objectives, we have the following constraints:

$$d_{ij} = \left\| c_i - c_j \right\|_2 \quad 1 \le i, j \le n, \tag{4}$$

$$\sum_{k \in N_i} \chi_{ki}^m(t) + \sum_{j \in N_i} \chi_{ij}^m(t) \le 1 \quad \forall t, i \in S, m \in M, \tag{5}$$

$$\chi_{ij}^m(t) + \sum_{l \in N_k} \chi_{kl}^m(t) \le 1 \quad \forall i \in S, j \in N_i, k \in \mathcal{I}_m^j, k \ne i, \tag{6}$$

$$\chi_{ki}^m(t) \in \{0, 1\} \quad 1 \le i, j \le n, m \in M, \tag{7}$$

$$\sum_{k \in N_i} \chi_{ki}^m(t) = 0 \quad \forall t, i \in S_U, m \in M, \tag{8}$$

$$\sum_{\mathrm{i} \in N_k} D_{ik} = 0 \quad \forall k \in S_U, \tag{9}$$

$$\sum_{i \in N_k} \chi_{ki}^m(t) = 0 \quad \forall t, k \in S_A, m \in M, \tag{10}$$

$$\sum_{k \in N_i} D_{ik} = 0 \quad \forall i \in S_A, \tag{11}$$

$$\sum_{k \in N_i} D_{ik} = v_i \Delta t \quad \forall i \in S_U, \tag{12}$$

$$\sum_{k \in N_i} D_{ik} = \sum_{j \in N_i} D_{ji} + v_i \Delta t \quad \forall i \in S, \tag{13}$$

$$D_{ij} \leq \sum_{t} \sum_{m \in M} c_{ij}^m \chi_{ij}^m(t) \quad \forall i, j \in S, \tag{14}$$

$$\begin{array}{l} e_i - {p_i}^{out} \sum\limits_{j \in N_i} D_{ij} - {p_i}^{in} \sum\limits_{j \in N_i} D_{ji} - \\ p_i^{out}(MI) \sum\limits_{j \in N_i} D_{ij} - p_i^0 \Delta t \geq 0 \end{array} \quad \forall i \in S, \tag{15}$$

$$y_i d_i + \sum_{j \in N_i} D_{ij} - v_i \Delta t - \sum_{j \in N_i} D_{ji} \geq 0 \quad \forall i \in S, \tag{16}$$

$$\sum_{j \in N_i} x_{ij} = y_i \quad \forall i \in S, \tag{17}$$

$$\sum_{i \in N_j} x_{ij} = y_j \quad \forall j \in S, \tag{18}$$

$$x_{ij}, y_i \in \{0, 1\} \quad 1 \leq i, j \leq n. \tag{19}$$

Notations used in above equations are listed in Table 2. Equations (5) and (6) are acoustic-channel interference constraints. Because of the interference in acoustical communication, each sensor can only transmit to or receive from another sensor through a specific acoustic channel at a time (Equation (5)). We use a binary variable $\chi_{ij}^m(t)$ to indicate the sub-channel state. If node i transmits data to node j on sub-channel m at time t, $\chi_{ij}^m(t) = 1$. Otherwise, $\chi_{ij}^m(t) = 0$. In Equation (6), $\mathcal{I}_m^j$ is the sensor set within the jth sensor's interference range on channel m. If sensor j is within the interference range of sensor k on sub-channel m, and sensor k is transmitting data to its neighbor l via m sub-channel, then sensor j's neighbor i will fail to transmit to sensor j through sub-channel m at this time because of signal interference (Equation (6)).

Table 2. Notations.

d_{ij}	The distance between node i to node j.
c_i	The coordinate of node i.
$\chi_{ki}^m(t)$	A binary variable, that equals to 1 if node i transmits data to node j on sub-channel m at time t.
N_i	The neighbor node set of node i.
D_{ik}	The data amount from node i to node k.
c_{ij}^m	The channel capacity between node i and node j on sub channel m.
y_i	An indicator that equals 1 if node i is an S_A node.
S_A	The node set that will be visit by an AUV.
S_U	The node set that will not be visit by an AUV.
S_C	The node set that need to be confirm whether will be visit by AUV.

Equation (8) to (14) are data flow constraints. We define S_A and S_U to represent the sensor sets that will be visited by AUV and will not be visited by AUV, respectively. Then there is no data flow to S_U sensors (Equations (8) and (9)) and no data flow from S_A sensors (Equations (10) and (11)). Moreover, the data flow from an S_U sensor equals the data it sensed (Equation (12)), and the outgoing data of each sensor is the sum of the incoming data and its sensing data (Equation (13)). Variable v_i in Equation (12) is the sensing rate of sensor i. In addition, the data flow on each sub-channel should not exceed the link capacity (Equation (14)). Variable c_{ij}^m in Equation (14) is the channel capacity between sensor i and sensor j on sub channel m.

Equation (15) means that, during the data gathering process, each node should not spend more energy than its remained energy. Variable d_i in Equation (16) is the buffer size of the ith sensor, and Equation (16) constrain the data amount stored on an S_A sensor should not exceed the buffer size of this sensor. A binary variable y_i is used to indicate the sensor type. If sensor i is an S_A sensor, then

$y_i = 1$. Otherwise, $y_i = 0$. Equations (18) and (19) are the travel tour constraints that each chosen sensor must visit exactly once. In addition, the AUV stop time on each sensor for data gathering is:

$$t_i = \frac{v_i \Delta t - \sum\limits_{j \in N_i} D_{ij} + \sum\limits_{j \in N_i} D_{ji}}{m}, \forall i \in S. \tag{20}$$

Since there is a polynomial-time reduction from the traditional traveling salesman problem to the above problem by setting p_i^{in} and p_i^{out} to zero, the above problem is an non-deterministic polynomial-time hard (NP-hard) problem. We implement the problem using Matlab+YALMP+GUROBI and found that, for a small scenario with no more than eight nodes, it takes more than an hour to find the optimal solution. Considering that this problem is NP-hard, the execution time would grow exponentially, and we propose a heuristic algorithm HAS^4 to solve this problem in the following section.

4. Centralized and Distributed Algorithms

In this section, we provide HAS^4 to solve the MILP, and give a centralized and a distributed version of HAS^4.

4.1. Description of HAS^4

Since the path planning problem is well studied as the traditional travelling salesman problem (TSP) in previous works [28], in this paper, we focus on a sink sensor selection problem. In HAS^4, we define S_C to represent the sensor set that needs to be confirmed as to whether it will be visited by AUV. At first, all sensors are in S_C and then HAS^4 determines which sensor falls into S_A and which sensor falls into S_U. During this process, HAS^4 greedily chooses sensors with less energy to S_A. This is because acoustic transmission consumes more energy than acoustic reception and MI communication. By selecting these sensors to S_A, in this round of data gathering, these sensors could spend less energy to balance the network energy consumption and prolong the network lifetime.

For the objective of minimizing the AUV cost, the mobile AUV should visit as few sensors as possible. For the purpose of minimizing the network energy consumption, the network should reduce acoustic communications, which means that AUV should visit as many sensors as it can. It is easy to observe that these two objectives are opposite to each other. In HAS^4, we define a variable to make a trade-off between these objectives. We provide both centralized and distributed version of HAS^4. Centralized HAS^4 can leverage all sensors' information about remainder energy, data sensing rate, channel capacity, etc. to make a sub-optimal solution for the problem in Section 3.2. Although the surface station or AUV could be the centralized decision maker, with the long propagation delay of acoustic communications and slow speed of AUV, centralized HAS^4 may not scalable for large underwater networks. Distributed HAS^4 utilizes the information from sensor-self and its one-hop neighbours to decide which set this sensor belongs to. Thus, it is very practical with both small and large size networks.

4.2. Centralized HAS^4

In centralized HAS^4, the trade-off factor l falls in $[0, 1]$, and $1 - l$ is the proportion of S_U candidate in all sensors, which means at least $n \times l$ sensors will be visited by the mobile AUV. The whole procedure is as shown in Algorithm 1.

Firstly, HAS^4 estimates the lifetime of each sensor to heuristically sort the sensors. The sensor lifetime is defined as the value that the remainder of the energy divides by the energy consumption in this period. The energy is consumed on data sensing and transmission, and sensing consumption is almost constant with the preassigned task. Since the energy spent on transmission varies between different modes and acoustic is much costlier than MI, without loss of generality, we use acoustic transmission cost as the communication energy consumption. In addition, due to the fact that we have not known the data flow, we average all possible values as the consumption. After sorting the sensors, a lifetime threshold is set as the lifetime of the $\lceil nl \rceil$-th sensor that sensors with smaller lifetimes can not be chosen into S_U.

Secondly, HAS^4 greedy selects the minimum lifetime sensor from S_C into S_A, and then adds all its neighbours into U. We denote U as the tuple set of candidate sensors of S_U with its upper level sensor. The upper level sensor is the sensor where the sensing data would transmit to.

Thirdly, HAS^4 heuristically chooses the sensor with the largest lifetime from U to verify if it could be added to S_U. In the verification, HAS^4 compares its lifetime with the threshold and all constraints in Section 3.2. If it satisfies all conditions, HAS^4 adds this sensor to S_U, changes channel indicators and adds its neighbours from S_C to U. Otherwise, it removes this sensor from U. HAS^4 repeats this process until U is empty and then goes to the second step.

When S_C is empty, the shortest tour among all S_A could be solved by approximately solving a TSP.

Since the outer while loop will execute at most n times, where n is the node number, and the inner while loop will execute at most $|U|$ times, the centralized HAS^4 is a $\Omega(mn)$ time algorithm, where m is neighbour number of the node with maximal neighbours.

Algorithm 1: Centralized HAS^4.

```
Input:
An underwater sensor network;
A tread off factor $l$ between delay and lifetime;
The residual energy of each sensor $e_i$.
Output:
A set of sensor that will be visited by AUV $S_A$;
Multiple data forward link;
A traveling tour.
 Add all sensor to $S_C$;
Ascending sort $S_C$ via lifetime $\frac{e_i - p_i^{out}}{p_i^{out}(mi)}$;
Define the threshold lifetime $T_l$ to be the lifetime of the $\lceil nl \rceil$-th sensor in $S_C$;
while $S_C$ is not empty do
    $u \leftarrow S_C(1)$;
    $S_A \leftarrow u \bigcup S_A$;
    $S_C \leftarrow S_C - u$;
    $U \leftarrow < j, u >, \forall j \in N_u$;
    while $U$ is not empty do
        Descending sort $U$ via lifetime $\frac{e_i - p_i^{out}}{p_i^{out}(mi)}$;
        $k \leftarrow U(1)$;
        if lifetime of $k \leq T_l$ then
            $U \leftarrow \emptyset$;
            Break;
        end
        if $\sum_{m \in M} \sum_{t \in T} f^i \geq 2\Delta t v_k$ then
            $S_U \leftarrow k \bigcup S_U$;
            $U \rightarrow < j, k >, \forall j \in N_k$;
        end
        $U \leftarrow U - k$;
    end
end
Find an approximate tour among $S_A$;
```

4.3. Distributed HAS^4

The distributed HAS^4 is a multi-round procedure. Each round consists of the following stages: self-elect inquiry, self-elect affirm, root select, root affirm, channel-use broadcast and conflict handler.

Sensors serving different roles participate in different stages and show different behaviors in each stage. Moreover, in distributed HAS^4, the trade-off factor l indicates the maximum acoustic path from any S_U sensors to S_A sensors.

In distributed HAS^4, each S_C node changes itself to be S_A node with probability 1/lifetime. When there are some S_A nodes, the relationship between S_A and S_C becomes a matching problem. Each S_C node takes the distance to each S_A node and the lifetime of each S_A node as the preference to choose a root node. Each S_A accepts some S_C nodes on the purpose of maximizing its group. Once an S_C node joins a group, it becomes S_U node. When there are no S_C nodes, this round of matching problem is solved.

The execution of distributed HAS^4 on S_C nodes is as shown in Algorithm 2. For each S_C node, in the self-elect inquiry stage, it temporarily changes itself to be an S_A node with probability 1/lifetime and broadcasts a self-elect-inquiry message to its neighbours. If it does not receive any self-elect-inquiry messages from its neighbours or its lifetime is less than all its temporarily S_A nodes, it will successfully be an S_A node and broadcast a self-elect-affirm message. Otherwise, it will change back to S_C. Once an S_C node receives a self-elect-affirm message, it will add this node to its root candidate set U. In the root select stage, an S_C node chooses the maximum lifetime node from U as root, and broadcasts a root-inquire message. If an S_C node receives a positive message in the root affirm stage, it will temporarily become an S_U node and broadcast a channel-occupation message based on the positive message. Once an S_C node receives a channel-occupation message, it will update its available channels. If a temporary S_U node does not receive any channel-occupation message or it has a longer lifetime or is closer to the root, it will successfully be an S_U node. Otherwise, it will change back to S_C and broadcast a channel-collision message.

Algorithm 2: Distributed HAS^4 S_C.

```
$U \leftarrow \emptyset$;
ChangeFlag $\leftarrow$ true;
while $\exists N_v \in S_C$ do
    Change $v$ to $S_A$ with probability 1/lifetime;
    if change to $S_A$ then
        broadcast a self-elect-inquiry message;
    end
    Receiving self-elect-affirm messages;
    On triggering timer$_l$;
    $U \leftarrow U \bigcup \{M_i\}$;
    Descending sort $U$ ;
    Choose the first node $u$ from $U$ as root;
    On triggering timer$_2$;
    Broadcast a root-inquiry message ;
    Remove $u$ from $U$;
    Receiving the decision broadcast;
    if choose by root then
        $S_U \leftarrow S_U \bigcup v$;
        On triggering timer$_{2l+2}$;
        Broadcast a channel-occupation message;
    end
    Update available channel;
    Receiving channel-occupation messages;
    if collision then
        Broadcast a channel-collision message;
    end
    else
        Update available channel;
        Change itself to $S_U$;
    end
end
```

Algorithm 3: Distributed HAS^4 S_A

```
while $\exists N_v \in S_C$ do
    Set timers;
    if $ChangeFlag = true$ then
        Receiving self-elect-inquiry messages $\{M_i\}$;
        On triggering timer$_1$;
        if $M_i.lifetime > v.lifetime \; \forall M_i \in \{M_i\}$ $OR$ $\{M_i\} = \emptyset$ then
            Broadcast a self-elect-affirm message;
            ChangeFlag $\leftarrow$ false;
        end
        else
            $U \leftarrow U \bigcup \{M_i\}$;
            Descending sort $U$ ;
            Choose the first nodes from $U$ $u$ as root;
            Broadcast the root-inquiry message ;
            Remove $u$ from $U$;
            Change itself to $S_C$;
        end
    end
    Receiving root-inquiry message;
    On triggering timer$_{l+2}$;
    Solve $\chi_{ij}^{m}(t)$ to maximum the leaf number;
    Broadcast the decision message;
    On triggering timer$_{2l+2}$;
    Broadcast the channel-occupation message;
    Receiving channel-occupation message;
    Update available channel;
    Receiving channel-collision messages;
    On triggering timer$_{3l+3}$;
    Update available channel and leaf nodes;
end
```

Algorithm 4: Distributed HAS^4 S_U.

```
denote $s$ to be the hops to root;
while $\exists N_v \in S_C$ do
    Receiving root-inquiry messages;
    On triggering timer$_{2+l-s}$;
    Transmit root-inquiry messages to its father node;
    Receiving the decision broadcast;
    On triggering timer$_{2+l+s}$;
    if $involved$ then
        Forward the decision to its child;
        update available channel;
        On triggering timer$_{2+2l}$;
        Broadcast a channel-occupation message;
    end
    Receiving channel-occupation messages;
    On triggering timer$_{3+2l}$;
    Update available channel;
    Receiving channel-collision messages;
    On triggering timer$_{3L+3-s}$;
    Transmit channel-collision messages to its father node;
end
```

The execution of distributed HAS^4 on S_A nodes is as shown in Algorithm 3. For each S_A node, it receives root-inquire messages, takes some appropriate S_C nodes into its group according to all the constraints in Section 3.2, and broadcasts this decision. Once an S_A node receives a channel-occupation message or channel-collision message, it will update the available channels. The execution of distributed HAS^4 on S_U nodes is as shown in Algorithm 4. In distributed HAS^4, S_U nodes are in charge of forwarding the messages between S_A and S_C.

Since in root select, root affirm and conflict handler stages, messages from S_C (S_A) should be forwarded hop-by-hop to S_A (S_C) sensors, while, in the other stages, sensors just independently broadcast messages. Thus, the duration of root select stage is equal to the root affirm stage and conflict handler stage, and relates to the maximum hop in the network, which is l time slots. In addition, the duration of self-elect inquiry, self-elect affirm and channel-use broadcast is much shorter and equals one time slot. Thus, there are $3l + 3$ time slots in each round of distributed HAS^4.

Since the message that transmitted in a distributed algorithm is not determined, the message transmitting time of each time slot should be randomly chosen in distributed HAS^4 to reduce the collision probability. The execution of distributed HAS^4 will exit on a sensor, when itself and all its neighbors are not within S_C.

For the worst execution of the distributed HAS^4 that there is only one S_A node and all node are sequentially joined in the group, the S_A node should answer $n - 1$ messages, and the message number for each of the rest nodes would not exceed $2n$. Thus, the distributed HAS^4 is a $\Omega(n^2)$ message algorithm.

5. Simulation

5.1. Simulation Setting

In this simulation, sensors are randomly deployed in a 10 km × 10 km × 1 km area, and the results are from 1000 runs. The initial energy of each sensor is 40 kW·h. The transmission power of acoustical communication is 190 dB re 1 μPa (approximate 6.67 W). Since MI communication can be powered by AUV, we don't take MI communication energy into account. The frequency of data gathering is 10 h. In addition, the data sensing rate follows the normal distribution of $\mathcal{N}(10, 2)$ kbit/s. The benchmark in this simulation is statically using the result of first round HAS^4's execution.

5.2. Results and Analysis

As shown in Figure 5, network lifetime decreases with the increment of network size. The reason is that, in a large network, the data amount forwarded by relay nodes is much larger than a small network that consumes more energy on data forwarding. We can also observe that centralized algorithms achieve better performance than distributed algorithms. This is because centralized algorithms can leverage all sensors' information to make a sub-optimal solution.

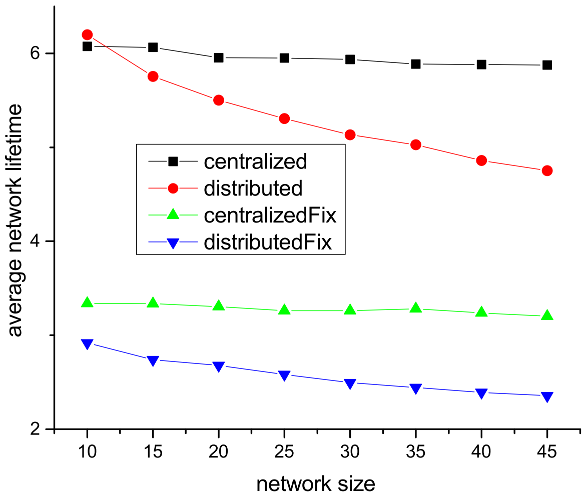

Figure 5. Network lifetime vs. network size with different sink selection algorithms.

Due to the channel capacity, more S_A sensors are needed in a larger network. Thus, as shown in Figure 6, AUV travels longer in a larger network than a small one.

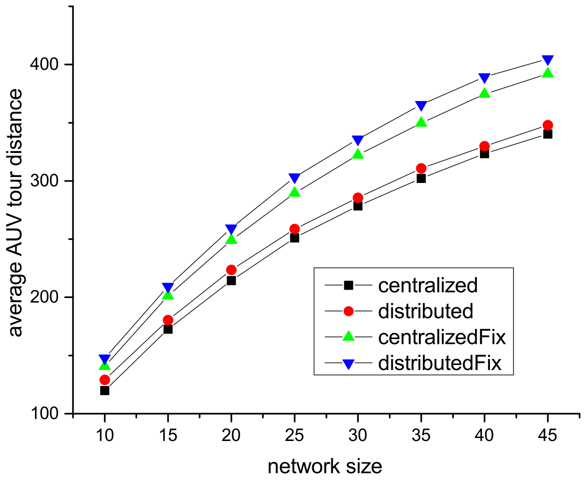

Figure 6. Tour vs. network size with different sink sensor selection algorithms.

Figures 7 and 8 illustrate how the network lifetime changes with different trade-off factors in different network sizes. As we can see, in centralized HAS^4, the network lifetime grows with the growth of trade-off factor. This is because the bigger the trade-off factor is, the less S_U nodes exist, which reduces the amount of acoustic data communication. In distributed HAS^4, since the trade-off factor is the longest source to end path. Thus, with a small trade-off factor, there are more S_A nodes, which reduce the acoustic communications.

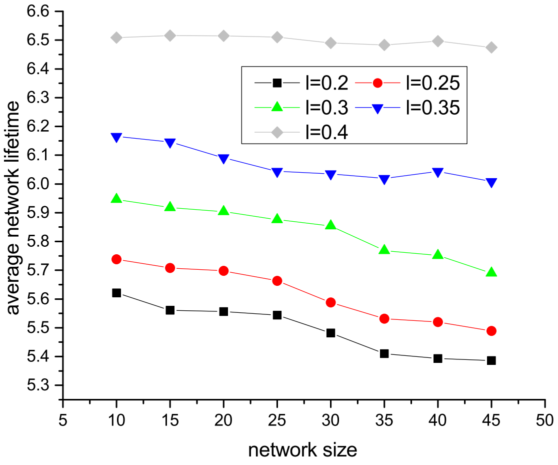

Figure 7. Effect of parameter *l* in centralized HAS^4 on network lifetime.

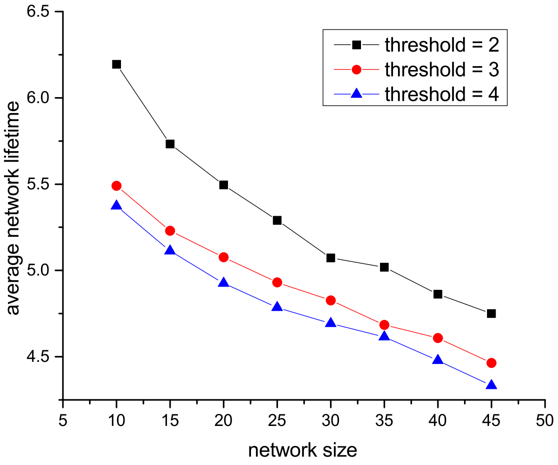

Figure 8. Effect of parameter *l* in distributed HAS^4 on network lifetime.

As we designed in centralized HAS^4, the trade-off factor is the maximum percentage of S_A nodes, the larger the factor is, the more S_A nodes will be. Thus, the reason that the AUV tour increases with the increment of trade-off factor in Figure 9 is with the trade-off factor grows, S_A nodes correspondingly increase, which leads to the increment of the average AUV tour. However, the threshold of distributed HAS^4 is designed as the maximum acoustic path from any S_U sensors to S_A sensors, which results in that as shown in Figure 10, and the number of S_A nodes will decrease with the increase of threshold.

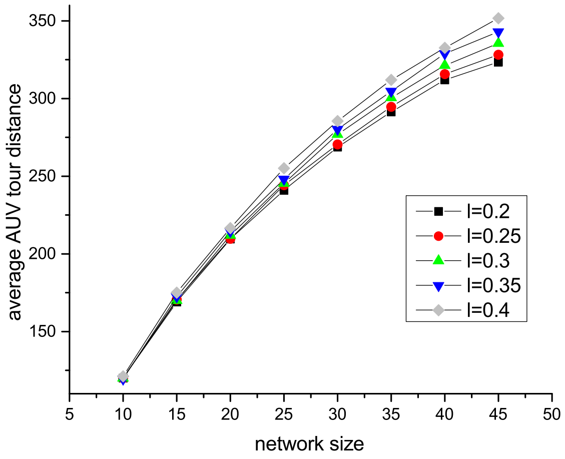

Figure 9. Effect of parameter *l* in centralized HAS^4 on tour.

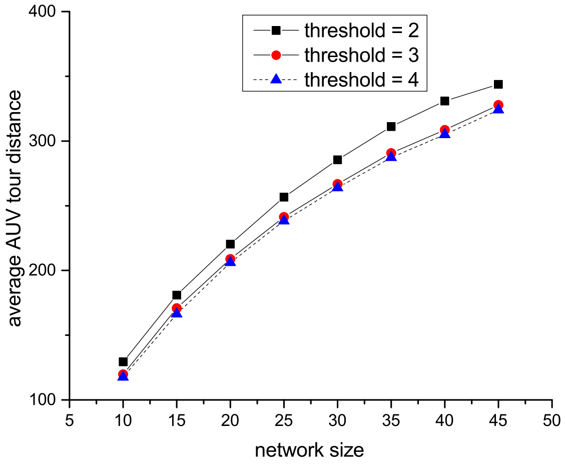

Figure 10. Effect of parameter *l* in distributed HAS^4 on tour.

6. Conclusions

In this paper, we solve the underwater data gathering problem in an efficient scenario, in which an AUV is utilized to visit part of the sensors and collect data via high speed MI communication, while the remaining nodes acoustically route their data to the ones that will be visited by the AUV. We propose both distributed and centralized algorithms HAS^4 to select sink sensor sets in this AUV-aid data gathering applications. By setting the trade-off factor, HAS^4 can achieve both energy and AUV efficiency.

For data gathering scenarios, the cooperation among multiple AUVs or the partition of the gathering task with high delay, unreliability, and low bandwidth acoustic communication in large scale UWSNs is much more challenging. Thus, in the future, we will further examine underwater data collection, including multi-AUV, task partition and online path planning.

Author Contributions: Conceptualization, M.P.; Methodology, M.P. and X.W.; Writing—Original Draft Preparation, X.W. and D.W.; Writing—Review and Editing, X.W. and M.P.; Supervision, X.W. and J.C.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) (Grants No. 61772228), National Key Research and Development Program of China (Grant No. 2016YFB0201503).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garcia, M.; Sendra, S.; Atenas, M.; Lloret, J. Underwater wireless ad-hoc networks: A survey. *Mobile Ad Hoc Networks: Current Status and Future Trends*; CRC Press: Boca Raton, FL, USA, 2011; pp. 379–411.
2. Akyildiz, I.F.; Pompili, D.; Melodia, T. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Netw.* **2005**, *3*, 257–279. [CrossRef]
3. Luo, Y.; Pu, L.; Zuba, M.; Peng, Z.; Cui, J.H. Challenges and opportunities of underwater cognitive acoustic networks. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 198–211. [CrossRef]
4. Lloret, J. Underwater sensor nodes and networks. *Sensors* **2013**, *13*, 11782–11796. [CrossRef] [PubMed]
5. Climent, S.; Sanchez, A.; Capella, J.V.; Meratnia, N.; Serrano, J.J. Underwater acoustic wireless sensor networks: advances and future trends in physical, MAC and routing layers. *Sensors* **2014**, *14*, 795–833. [CrossRef] [PubMed]
6. Guo, H.; Sun, Z.; Wang, P. Channel modeling of mi underwater communication using tri-directional coil antenna. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
7. Akyildiz, I.F.; Wang, P.; Sun, Z. Realizing underwater communication through magnetic induction. *IEEE Commun. Mag.* **2015**, *53*, 42–48. [CrossRef]
8. Zhang, J.; Shi, C.; Sun, D.; Han, Y. High-precision, limited-beacon-aided AUV localization algorithm. *Ocean Eng.* **2018**, *149*, 106–112. [CrossRef]
9. Wei, X.; Wang, X.; Bai, X.; Bai, S.; Liu, J. Autonomous underwater vehicles localisation in mobile underwater networks. *Int. J. Sens. Netw.* **2017**, *23*, 61–71. [CrossRef]
10. Vasilescu, I.; Kotay, K.; Rus, D.; Dunbabin, M.; Corke, P. Data collection, storage, and retrieval with an underwater sensor network. In Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, San Diego, CA, USA, 2–4 November 2005; pp. 154–165.
11. Zhao, M.; Yang, Y. Bounded relay hop mobile data gathering in wireless sensor networks. *IEEE Trans. Comput.* **2012**, *61*, 265–277. [CrossRef]
12. Cheng, C.F.; Li, L.H. Data gathering problem with the data importance consideration in Underwater Wireless Sensor Networks. *J. Netw. Comput. Appl.* **2017**, *78*, 300–312. [CrossRef]
13. Jiang, J.; Han, G.; Guo, H.; Shu, L.; Rodrigues, J.J. Geographic multipath routing based on geospatial division in duty-cycled underwater wireless sensor networks. *J. Netw. Comput. Appl.* **2016**, *59*, 4–13. [CrossRef]
14. Li, N.; Martínez, J.F.; Meneses Chaus, J.M.; Eckert, M. A survey on underwater acoustic sensor network routing protocols. *Sensors* **2016**, *16*, 414. [CrossRef] [PubMed]
15. Liu, J.; Wang, Z.; Cui, J.H.; Zhou, S.; Yang, B. A joint time synchronization and localization design for mobile underwater sensor networks. *IEEE Trans. Mob. Comput.* **2016**, *15*, 530–543. [CrossRef]
16. Wei, L.; Wang, Z.; Liu, J.; Peng, Z.; Cui, J.H. Power Efficient Deployment Planning for Wireless Oceanographic Systems. *IEEE Syst. J.* **2018**, *12*, 516–526. [CrossRef]
17. Xie, P.; Cui, J.H.; Lao, L. VBF: Vector-based forwarding protocol for underwater sensor networks. In Proceedings of the International Conference on Research in Networking, Coimbra, Portugal, 15–19 May 2006; pp. 1216–1221.
18. Nicolaou, N.; See, A.; Xie, P.; Cui, J.H.; Maggiorini, D. Improving the robustness of location-based routing for underwater sensor networks. In Proceedings of the Oceans 2007-Europe, Aberdeen, UK, 18–21 June 2007; pp. 1–6.
19. Coutinho, R.W.; Vieira, L.F.; Loureiro, A.A. DCR: Depth-Controlled routing protocol for underwater sensor networks. In Proceedings of the 2013 IEEE Symposium on Computers and Communications (ISCC), Split, Croatia, 7–10 July 2013; pp. 000453–000458.
20. Yan, H.; Shi, Z.J.; Cui, J.H. DBR: depth-based routing for underwater sensor networks. In Proceedings of the International Conference on Research in Networking, Singapore, 5–9 May 2008; pp. 72–86.
21. Diao, B.; Xu, Y.; An, Z.; Wang, F.; Li, C. Improving both energy and time efficiency of depth-based routing for underwater sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 781932. [CrossRef]
22. Guangzhong, L.; Zhibin, L. Depth-based multi-hop routing protocol for underwater sensor network. In Proceedings of the 2010 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, 30–31 May 2010; Volume 2, pp. 268–270.
23. Tekdas, O.; Isler, V.; Lim, J.H.; Terzis, A. Using mobile robots to harvest data from sensor fields. *IEEE Wirel. Commun.* **2009**, *16*, 22–28. [CrossRef]

24. Hollinger, G.A.; Choudhary, S.; Qarabaqi, P.; Murphy, C.; Mitra, U.; Sukhatme, G.S.; Stojanovic, M.; Singh, H.; Hover, F. Underwater Data Collection Using Robotic Sensor Networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 899–911. [CrossRef]
25. Basagni, S.; Boloni, L.; Gjanci, P.; Petrioli, C.; Phillips, C.A.; Turgut, D. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In Proceedings of the 2014 IEEE Conference on Computer Communications (INFOCOM 2014), Toronto, ON, Canada, 27 April–2 May 2014; pp. 988–996.
26. Akbar, M.; Javaid, N.; Khan, A.H.; Imran, M.; Shoaib, M.; Vasilakos, A. Efficient data gathering in 3D linear underwater wireless sensor networks using sink mobility. *Sensors* **2016**, *16*, 404. [CrossRef] [PubMed]
27. Wei, D.; Yan, L.; Li, X.; Sun, Y.; Yuan, D.; Chen, J.; Pan, M. Exploiting Magnetic Field Analysis to Characterize MI Wireless Communications in Subsea Environments. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 805–809.
28. Hoffman, K.L.; Padberg, M.; Rinaldi, G. Traveling salesman problem. In *Encyclopedia of Operations Research and Management Science*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 1573–1578.