

Article

A Depth-Adjustment Deployment Algorithm Based on Two-Dimensional Convex Hull and Spanning Tree for Underwater Wireless Sensor Networks

Peng Jiang *, Shuai Liu, Jun Liu, Feng Wu and Le Zhang

College of Automation, Hangzhou Dianzi University, Hangzhou 310018, China; liushuaihdu@163.com (S.L.); liujunhdu@163.com (J.L.); fengwu@hdu.edu.cn (F.W.); lezhang@hdu.edu.cn (L.Z.)

* Correspondence: pjiang@hdu.edu.cn; Tel.: +86-571-8691-9131 (ext. 512); Fax: +86-571-8687-8566

Academic Editor: Leonhard M. Reindl

Received: 30 May 2016; Accepted: 9 July 2016; Published: 14 July 2016

Abstract: Most of the existing node depth-adjustment deployment algorithms for underwater wireless sensor networks (UWSNs) just consider how to optimize network coverage and connectivity rate. However, these literatures don't discuss full network connectivity, while optimization of network energy efficiency and network reliability are vital topics for UWSN deployment. Therefore, in this study, a depth-adjustment deployment algorithm based on two-dimensional (2D) convex hull and spanning tree (NDACS) for UWSNs is proposed. First, the proposed algorithm uses the geometric characteristics of a 2D convex hull and empty circle to find the optimal location of a sleep node and activate it, minimizes the network coverage overlaps of the 2D plane, and then increases the coverage rate until the first layer coverage threshold is reached. Second, the sink node acts as a root node of all active nodes on the 2D convex hull and then forms a small spanning tree gradually. Finally, the depth-adjustment strategy based on time marker is used to achieve the three-dimensional overall network deployment. Compared with existing depth-adjustment deployment algorithms, the simulation results show that the NDACS algorithm can maintain full network connectivity with high network coverage rate, as well as improved network average node degree, thus increasing network reliability.

Keywords: underwater wireless sensor networks (UWSNs); two-dimensional convex hull; spanning tree; time marker; network reliability; full network connectivity

1. Introduction

Underwater wireless sensor networks (UWSNs) are an aquatic environment monitoring network system that consists of many nodes with data acquisition, storage, processing, and wireless acoustic transmission features. UWSNs have extensive applications in water environmental pollution prediction, ocean exploitation, marine monitoring system, and other areas [1–4]. With the increasing depletion of land resources, coastal countries are paying increasing attention to maritime rights and resource exploitation, and the ocean economy plays a vital role in the development of the world economy. Therefore, UWSN technology has become a popular research topic in the field of wireless sensor networks. As a technology extension of terrestrial wireless sensor networks (TWSNs) [5,6] to aquatic environments, UWSNs inherit some characteristics of TWSNs, however due to the special application environment and underwater acoustic communication model of UWSNs, they ordinarily bring some new characteristics, namely, their network structure, which is distributed in three dimensions, high latency of acoustic signal, limited bandwidth of communication, high transmission bit-error-rates, constrained movement of nodes, very limited energy, and others [1,7,8]. In view of the abovementioned characteristics of UWSNs, the deployment algorithms and routing protocols of

TWSNs cannot be directly migrated to the UWSNs. These new characteristics must be combined to design more reasonably effective deployment algorithms and positioning and routing protocols for UWSNs [9–11]. The protocol and algorithm design of UWSNs mostly relate to node deployment, topology control, node localization, time synchronization, routing protocol, and others. Meanwhile, the node deployment is not only directly related to the quality of network monitoring but also dramatically affects subsequent processing on the design of all kinds of protocols and algorithms. Therefore, in this study, this basic and crucial problem is further investigated, namely, node deployment.

According to whether the target location of the node is determined, the node deployment algorithm of UWSNs can be divided into the following two major categories: stochastic deployment algorithm and deterministic deployment algorithm [12]. After all of the nodes are scattered on the monitoring water surface uniformly and randomly by boat or aircraft, if these nodes only execute random depth adjustment in the vertical direction, then the deployment strategy is referred to as stochastic deployment. If the node can be moved to a definite position calculated by the corresponding optimization algorithm, then the deployment algorithm is referred to as deterministic deployment. Compared to the stochastic deployment, the deterministic deployment can achieve better network performance and therefore is widely used. However, for the node deterministic deployment, based on the assumption of node mobility, the existing deterministic deployment algorithms can be classified into three categories, namely, vertical movement deployment [13–15], static deployment [16–18], and free movement deployment [19–21].

In this study, we mostly consider vertical movement deployment (depth-adjustment deployment). Depth-adjustment deployment refers to the deployment in which all of the nodes only execute depth-adjustment along the vertical direction in the underwater monitoring space. Different from free movement deployment, vertical movement deployment can only achieve node vertical depth adjustment by adjusting the length of the rope or some other economical method [13], thus the vertical movement deployment strategy or algorithm has strong feasibility and applicability. Among existing vertical movement deployments, the deployment strategy based on graph theory in the underwater node deployment is widely used. This algorithm adopts relevant concepts (graph coloring problem or connected dominating set) of graph theory to adjust the depth of nodes through the depth-adjustment to reduce redundant coverage, thus completing the overall 3D coverage. Akkaya et al. [14] proposed a self-deployment depth adjustment algorithm that mainly focuses on maximizing network coverage by constantly adjusting node depths to reduce coverage overlaps between two neighboring nodes. Meanwhile, this algorithm adopts graph coloring problem to solve the distribution of network group IDs, based on coverage overlaps of intra-cluster nodes for different node assigned with different group IDs, and the cluster head node applies the centralized approach to nodes executing depth adjustment, decreasing coverage overlaps and thus increasing overall network coverage for the 3D monitoring space. Senel et al. [15] proposed depth adjustment algorithms based on a connected dominating set, and this algorithm is derived from [14]. Compared with the latter algorithm, the former algorithm initially constitutes a connected backbone, and then the dominant node respectively optimizes its non-dominators located inside their connected dominating set in an iterative way. Moreover, the algorithm ensures the full network connectivity, maximizes network overall coverage rate, and reduces the deployment energy consumption.

Meanwhile, many scholars have proposed deployment algorithms based on computational geometry for traditional sensor networks and supported many applications in real industrial markets. Heo et al. [22] proposed a self-deployment algorithm based on Voronoi diagrams. The algorithm constant constructs Voronoi diagrams of all nodes to reflect the nodes' movement state and optimize the location of nodes, thus improving network operator performance. Wang et al. [23] used Voronoi diagrams to detect network coverage holes and to control the movement of nodes to repair coverage holes. The algorithm argues that calculating the correct Voronoi polygons can help to optimize its healing efficiency of coverage holes, if they exist. However, network connectivity is not considered. The construction of the Voronoi diagram needs to execute global computation, so the communication costs are large. In recent years, some scholars have transferred the concept of computational geometry

deployment strategies to UWSN applications. Wu et al. [24] proposed Voronoi-based depth-adjustment algorithm (VBDA) for UWSNs. First, all of the nodes are randomly deployed on the water surface through the formation of the Voronoi polygon of each node to calculate the network coverage redundancy, the Voronoi region area of each node in same layer to determine the depth-adjustment distance, and then the iterative operating process of depth adjustment until the last layer network deployment arrives at the water bottom threshold. The VBDA is first proposed based on the Voronoi deployment strategy for UWSNs. Through the vertical depth adjustment of the nodes to reduce the network coverage redundancy, the 3D network coverage is increased. However, the algorithm does not consider the network connectivity between two neighboring layers, which makes the network connectivity performance very poor with a small α (the ratio of node communication radius to sensing radius). Moreover, as the process of network deployment proceeds, fewer nodes are available to execute network deployment when the process of depth-adjustment reaches the water bottom threshold, then its network coverage performance is relatively inferior. Meantime, the VBDA is a centralized algorithm, and all nodes have more communication costs and then consume more communication energy, thereby affecting network operator performance.

The deployment issues mentioned above, considering the above-proposed deployment algorithms of UWSNs based on computational geometry consisting of vertical movement deployment strategy, the VBDA cannot improve network coverage performance while ensuring full network connectivity. Furthermore, the communication energy consumption still has ample room for improvement. Therefore, we proposed a node depth adjustment deployment algorithm based on two-dimensional (2D) convex hull and spanning tree (NDACS). A sink node is fixed at the center of the monitored water surface, and all of the nodes are scattered on the monitored water surface uniformly and randomly and are in a sleep state. First, the sink node sends control information to activate three nodes according to the node ID from small to large and forms a very simple 2D convex hull (triangle). Thereafter, it uses the geometric characteristics of the 2D convex hull and empty circle to minimize the 2D plane network coverage overlaps as its optimization goal. The optimal sleep node that needs be activated is found, whereas the 2D coverage efficiency could be greatly improved. When the coverage threshold of the first layer is reached, the algorithm forms the best fully 2D node activation convex hull. At the same time, most of the nodes are in a sleep state to save overall network energy. Thereafter, the sink node acts as a root node of all active nodes on the 2D convex hull and gradually forms a small spanning tree. Eventually, a new spanning forest is formed through the adjustable communication radius model in the 2D water surface. The child nodes closer to the sink node have greater probability of data transmission and relay and have to consume more energy. Therefore, for the network to be able to continuously operate stably, more nodes must be deployed near the sink node, in which more subtrees are found around the sink node, and the size of surrounding subtrees should be smaller. We adopt the time marker to define the 2D network level and make the interlayer neighbor intersection be as close to the parent node of the spanning tree as possible, increasing the node deployment density around the sink node and balancing network energy consumption. Finally, the active node is used as the parent node of its neighbor sleep node, and the depth-adjustment strategy based on the time marker is used to achieve the 3D overall network coverage. The simulation results show that compared to the typical depth-adjustment deployment algorithm (VBDA), the NDACS algorithm can maintain full network connectivity with a high network coverage rate while reducing communication energy consumption with better network reliability during deployment. Compared with existing depth-adjustment deployment algorithms, our proposed algorithm contributes through the following aspects:

- (1) The geometric characteristics of the 2D convex hull and empty circle is used to minimize the 2D plane network coverage overlaps, with a minimum number of activated nodes to meet the requirement of the first layer of the network deployment.
- (2) The activation scheduling mechanism is adopted to reduce the energy consumption of communication between nodes.

- (3) The adjustable communication radius model is established to ensure that all active nodes have full network connectivity in the first layer. In addition, in the process of depth adjustment, the communication connection is guaranteed between the parent and its child nodes in real time.
- (4) Depth adjustment is based on the time marker, adjacent time marker neighbor nodes set is dominated by its parent node, and it is a perfect approach to balance the density of node deployment. When the set is empty, the neighbor nodes intersect to execute depth-adjustment assignment, ensuring that the network deployment performance of near water bottom is relatively good and that network operation is more reliable. Overall, our work optimizes network coverage rate under the premise of full network connectivity, reducing the energy consumption of communication.

The rest of this paper is organized as follows: in Section 2, the related works about the node deployment problem in three categories algorithm for UWSNs. In Section 3, the models of UWSNs and related definitions are described. In Section 4, the problem is analyzed, and the details of the NDACS algorithm are implemented. In Section 5, the communication complexity and time complexity of network deployment for NDACS algorithm are analyzed. In Section 6, the description of algorithm simulation and the detailed analysis of the simulation results are discussed. Finally, in Section 7, the conclusions and future work of our study are drawn.

2. Related Works

In addition to a number of different strategies for node deployment of UWSNs, many researchers have focused on static deployment [16–18] and free movement deployment [19–21]. The static deployment algorithm refers to the assumption that nodes do not have the ability to move. The nodes need artificial deployment to a fixed location determined by advance calculation, and when the network deployment is completed, besides, the nodes no longer move. Alam et al. [16] used different polyhedrons to fill three-dimensional (3D) deployment space and study the maximization of 3D coverage space with a minimum number of nodes, offering mathematical proof and derivation to explain why truncated octahedral cells result in the best strategy and achieve the greatest network coverage performance. Subsequently, basing on the study of [16], Alam et al. [17] used space-filling truncated polyhedrons to execute network coverage deployment by controlling the cell-vertex nodes' active or sleep states, and adjusting the space-filling polyhedron radius can be flexible to achieve k -coverage. Liu et al. [18] proposed a geometric hierarchical clustered deployment strategy where the cluster head nodes are alternated by using the geometric characteristics of the cluster shape and constraint condition that ensures network connectivity and full coverage. The optimal network lifetime mathematical model and cluster shape model are derived, and optimal node deployment is achieved. On the whole, the node goal location has been determined before the real deployment. Therefore, the static deployment algorithm usually can achieve good network coverage or connectivity performance. However, the node needs to be fixed to the definite position through artificial means, and obviously, the difficulty of deployment is excessively large and the deployment cost is excessively high for aquatic environments. In addition, when the network scale is expanded or topological changes occur because of the effect of water environment (water flow or animal collision), the applicability of the static deployment algorithm decreases rapidly.

Free movement deployment refers to the assumption that nodes have the ability to move freely in each direction and can be moved to any location in the 3D network space. Considering the covering of the uneven isolated incident in the monitoring area, Xiana et al. [19] first proposed fish swarm-inspired node deployment and subsequently [20] proposed a similar algorithm—the particle swarm-inspired node deployment. These two algorithms simulate the behavior of fish swarms or particle swarms and introduce the model of congestion degree control. The proposed algorithms can drive nodes to cover the events and match event distribution density in the water environment. The author only considers the coverage rate of the events and ignores the network connectivity rate during the deployment process. Li et al. [21] proposed the 3D virtual forces deployment algorithm. This algorithm uses virtual

force to adjust node location and improve network coverage rates and connectivity rates. However, this algorithm could not achieve full network connectivity, that is, the network connectivity rate is 1. For this kind of free movement, in all of the abovementioned proposed deployment algorithms, the node has the ability to move in each direction and can move to an optimized location where network coverage and connectivity performance are obviously improved. However, free movement deployment needs to equip these nodes with autonomous underwater vehicle (AUV) or other mobile devices under normal circumstances then its strategy induces relatively high production costs, which make the node free movement deployment algorithm difficult to spread in the practical UWSNs application.

As a research team which has remained focused on WSNs, our team also proposed some node deployment algorithms for UWSNs in recent years [25–28]. The node self-deployment algorithm based on an uneven cluster with radius adjusting (URSA) for UWSNs was proposed in [25], and the difference between the URSA and NDACS was the core concept of the URSA algorithm based on uneven clustering. However, our proposed algorithm main is based on computational geometry. The algorithms in [25,26] belong to depth-adjustment deployment, while the algorithms in [27,28] belong to free movement deployment.

3. Network Models and Related Definitions

3.1. Network Models

3.1.1. Network System Model

Referring to the 3D characteristic of UWSNs in this paper, we assume that n nodes are distributed randomly and uniformly on the water surface, that they are floating with the help of buoys, and that the area of 3D monitoring space is $L \times W \times H$. A sink node is fixed at the center of monitoring water surface. The typical UWSNs model is described in Figure 1, including the following preliminaries:

- (1) The node has no ability to perceive global coordinate information in 3D network space, and merely the node perceives own depth information.
- (2) Each node has a unique identity ID.
- (3) The boundary effect is negligible because the node communication radius R_c and sensing radius R_s are significantly smaller than the length L , width W , or height H of the 3D monitoring space.
- (4) All nodes use a uniform acoustic signal to transmit and receive information, and the sink node transmits the information to the base station by radio signal.
- (5) All nodes are identical and have the same initial energy E_{init} , sensing radius R_s , and the initial communication radius $R_{c_{init}}$. Moreover, the energy consumption of the sink node is negligible.

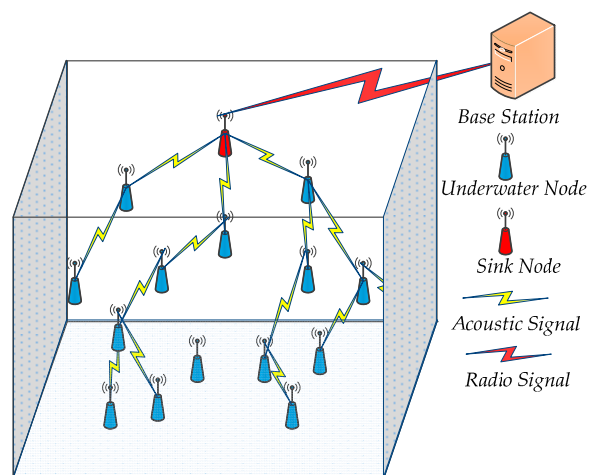


Figure 1. 3D UWSNs system model.

3.1.2. Node Perception Model

The Boolean perception model describes the node sensing in this study. The coordinate of node S_i is (x_i, y_i, z_i) , and the Euclidean distance between S_i and point $p(x, y, z)$ is $d(S_i, p) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$. Therefore, under this perception model, the probability of point p is perceived by node S_i and can be presented as follows:

$$c_p(S_i) = \begin{cases} 1, & d(S_i, p) < R_s \\ 0, & d(S_i, p) \geq R_s \end{cases} \quad (1)$$

where R_s denotes the sensing radius of node S_i .

3.1.3. Adjustable Communication Radius Model

The maximum communication radius of the node is R_c , and its initial value is $R_{c_{init}}$, the adjustable level of communication radius is defined as θ and has an initial value of 1. Therefore, we can obtain the relationship between R_c and θ through the following formula:

$$R_c = R_{c_{init}} + (1 + \theta) B_u \quad (2)$$

where B_u is the increasing communication range per communication level.

3.1.4. Node Energy Consumption Model

With the energy consumption of node sensing and data receiving significantly smaller than that of data transmission and node movement [29], so we only consider the latter as a primary component of energy consumption. In this study, the energy consumption model of the node based on the method is mentioned in [30,31], and it can be calculated through the following formula:

$$E_c(d, f) = P_0 T_p d^k 10^{\alpha(f)d/10} \quad (3)$$

where P_0 denotes the power threshold for a node to receive the information package. T_p denotes transmission delay of the information package, d denotes the transmitting distance of the information package, and k is the energy spreading factor. The absorption coefficient $\alpha(f)$ can be calculated through the following formula:

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

where f is the frequency of the carrier acoustic signal in HZ, and $\alpha(f)$ is in dB/m. In addition, the movement energy consumption M_e of node can be expressed as the product of the movement distance m_d and the energy consumption of per movement distance is m_u . The relationship is described as follows:

$$M_e = m_d * m_u \quad (5)$$

3.2. Related Definitions

3.2.1. Network Coverage Rate

As shown in Figure 2, the 3D UWSNs are divided into a plurality of small cube grid g , and all of small grids position can be presented by selecting the coordinates of the center of these grids. Through the Boolean perception model (Section 3.1.2), the network coverage rate C_c is defined by the ratio of the number N_S of grid centers covered in the monitored underwater environment space to the total number N_N of grids:

$$C_c = \frac{N_S}{N_N} \quad (6)$$

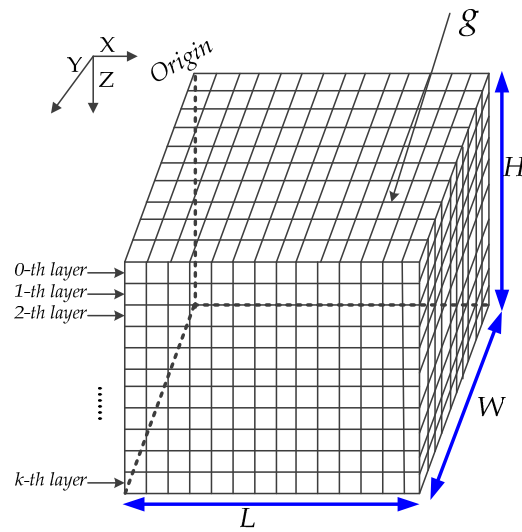


Figure 2. Grid division for 3D UWSNs.

3.2.2. Network Connectivity Rate

Network connectivity rate C_n is defined as the ratio of n_c to n , where n_c denotes the number of nodes that can communicate with the sink node through single-hop or multi-hop communication, and n is the total number of nodes in the monitored underwater environment space, therefore, C_n can be calculated as follows:

$$C_n = \frac{n_c}{n} \quad (7)$$

If C_n is equal 1, the network achieves full connectivity, namely, all nodes can communicate with the sink node through single-hop or multi-hop communication.

3.2.3. Network Reliability

As a crucial indicator for evaluating the network quality of service (QoS) [32], network reliability is the probability that the network can maintain full network connectivity at a high level. This study mainly investigates UWSNs deployment issues. Thus the network should still maintain a high network connectivity rate, which is a crucial optimization goal of our algorithm, as some nodes begin to die because of persistent network operation. Therefore, in this study, network reliability can be described as the average node degree (not in the same depth):

$$D_a = \frac{\sum_{i=1}^n N_i(nei)}{n} \quad (8)$$

where D_a denotes the average node degree, $N_i(nei)$ denotes the neighbor node number of node S_i .

3.2.4. Two-Dimensional Convex Hull

Given a points set $p = \{p_t, t = 1, 2, \dots, n\}$ on a 2D plane V , any finite points at set p constructs all of convex combination and then forms a set H , which is the convex hull of set p and is denoted by $H(p)$ [33]:

$$H(p) = \left\{ \sum_{i=1}^m \varepsilon_i x_i \mid x_i \in p, \varepsilon_i \geq 0, i = 1, 2, \dots, m, \sum_{i=1}^m \varepsilon_i = 1, m \in N \right\} \quad (9)$$

Concretely, h is the minimum convex polygon in the 2D plane V that can enclose all points of set p and is referred to as the 2D convex hull of the set p . As shown in Figure 3, the geometry is a convex hull of the point set surrounded by solid lines.

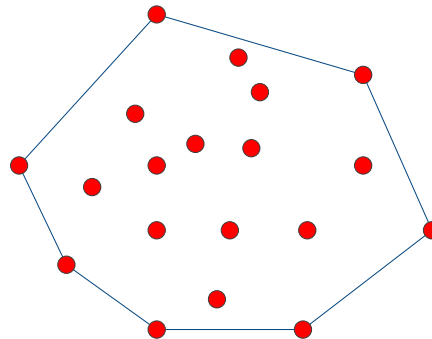


Figure 3. Schematic of a 2D convex hull.

3.2.5. Empty Circle

A point set $p = \{p_t, t = 1, 2, \dots, n\}$ is given on a 2D plane V . A circle $C_m(p_i, p_j, p_k)$ is constructed with three out-of-line points p_i, p_j, p_k , and this circular region does not contain any arbitrary point within set p . Moreover, the center of circle C_m is located in given plane V and is an empty circle.

4. Problem Analysis and Algorithm Description

4.1. Problem Analysis

In using energy-constrained UWSNs to monitor a target water area, we must ensure that the network has high coverage and full connectivity at the same time and effectively manage nodes sleep or active states to save communication energy consumption. Moreover, the network should maintain high network connectivity, which is a crucial optimization goal of our algorithm, as some nodes begin to die because of persistent network operation, that is, the network still has relatively good reliability. However, in most of the move-constrained deployment algorithms of [13–15], coverage overlaps are reduced by node depth adjustment, thus increasing the global network coverage rate. On the other hand, all nodes are in an active state during the deployment process, and this increases their network communication energy use. Meanwhile, these algorithms do not consider the average node degrees between different depth networks, subsequently affecting network reliability and connectivity rate under network operation conditions. Moreover, in view of the node move-restricted depth-adjustment algorithm based on computational geometry, through the vertical depth-adjustment of nodes in reducing network coverage redundancy, the 3D network coverage is increased, Wu et al. [24] proposed the VBDA algorithm for UWSNs. However, the VBDA algorithm does not consider the network connectivity between two neighboring layers, resulting in a very poor network connectivity performance with a small α (the ratio of node communication radius to sensing radius). As the network is deployed, fewer nodes are available to execute depth adjustment when the process of deployment reaches the water bottom threshold. Thereafter, network coverage performance becomes relatively inferior. In addition, the VBDA algorithm does not consider network connectivity in the deployment process. The nodes are always in the active state, and thus all nodes require more communication, thereby consuming more communication energy. The limited energy for the nodes and the difficulty of charging the batteries in the water monitoring space affect the network operator performance, including network coverage rate, network connectivity rate, and network reliability. Meanwhile, VBDA is a centralized algorithm, and its practical application is more difficult to achieve than that of our proposed system.

To better solve the problem of the abovementioned VBDA algorithm, we proposed NDACS in this study. A detailed description of the NDACS algorithm is given in the next section.

4.2. Algorithm Description

The main ideas of the NDACS algorithm can be described as the following steps: (1) optimal activation strategy based on 2D convex hull and empty circle on the water surface; (2) formation process of a 2D convex hull spanning tree; (3) and depth-adjustment strategy based on time markers in achieving 3D network coverage. The algorithm framework is shown in Figure 4.

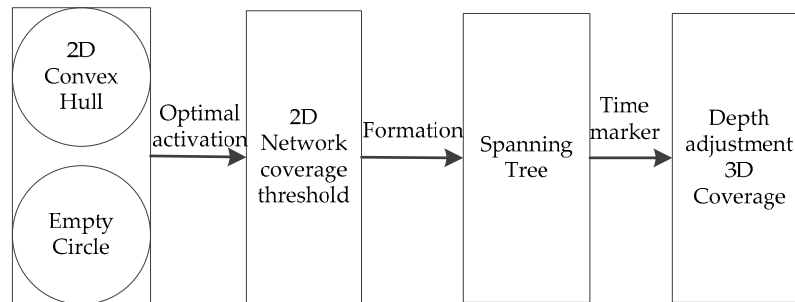


Figure 4. Algorithm framework.

4.2.1. Optimal Activation Strategy Based on 2D Convex Hull

In minimizing network coverage overlaps, nodes should be as far away from one another as possible, but they should not interrupt the communication links between source nodes and their neighboring nodes. The sink node broadcasts a message to all nodes, according to the received signal strength from node S_i , to calculate the distance between S_i and the sink node. In a similar manner, the 2D level coordinates of all nodes can be determined based on the sink node. The sink node sends a control signal to randomly activate three sleep nodes, which then form the simplest 2D convex hull, as shown in Figure 5. In the figure, the active nodes form a simplest 2D convex hull (a triangle) through a solid line. The set of currently active nodes is denoted by S_a , and the set of sleep nodes is denoted by S_s . Sleep nodes inside the 2D convex hull are referred to as $sleep_{in}$, and those outside are denoted by $sleep_{out}$, where $sleep_{in} \cap sleep_{out} = S_s$.

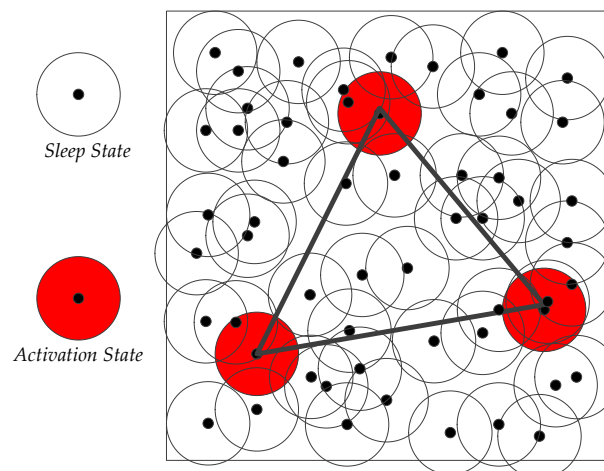


Figure 5. Schematic diagram of the 2D convex hull activation strategy.

The inside and outside of the 2D convex hull have many coverage holes and monitoring blanks. Obviously, the current regional network coverage area is determined by two parts: the coverage holes

of the convex hull interior and the monitoring blanks of the convex hull exterior. Therefore, we obtain the following formula:

$$S_c = S_{CH} - S_{Hole} \quad (10)$$

where S_c denotes the 2D network coverage area, S_{CH} denotes the area of the 2D convex hull, and S_{Hole} denotes the area of the coverage holes. Thus, we can improve the value of S_c by increasing S_{CH} or decreasing S_{Hole} .

Furthermore, the coverage area of the network can be expressed:

$$S_c = \bigcup_{i=1}^m C_i \quad (11)$$

where C_i denotes the coverage area of the active node S_i . m nodes are activated and $S_i \in Sa$.

A sleep node C_j is chosen from Ss to be activated, and then the area of the 2D network coverage is maximized. At the current moment, the coverage area is presented by S'_c , and our goal is to maximize S'_c .

$$\max S'_c = S_c \cup C_j = \left(\bigcup_{i=1}^m C_i \right) \cup C_j \quad (12)$$

where C_j denotes the coverage area of sleep node S_j , and $S_j \in Ss$.

To maximize S'_c , by comparing the different coverage gains from selecting the different activated sleep nodes derived from *sleep_in* or *sleep_out*, we determine the node that can obtain the greatest coverage gains. Therefore, the optimal activation strategy is transformed into the maximum coverage gains through the selection of activated sleep nodes in Ss . Before the optimal activation strategy is solved, the following theorem is proposed, and we will prove it subsequently.

Theorem 1. *If the radius Rh of an empty circle (Section 3.2.5) is greater than the sensing radius Rs , then the interior of the empty circle will certainly have coverage holes.*

Proof of Theorem 1. By the definition in Section 3.2.5, the interior of the empty circle must not have additional active nodes, which means that the interior of the empty circle maybe has some sleeping nodes. Obviously, when Rh is greater than Rs , the center of the empty circle must not be covered, so we can infer that coverage holes exist in the empty circle.

When Rh is larger, the active nodes are relatively more sparse, and the probability of occurrence of coverage holes is greater. Therefore, we choose to activate a sleeping node from the interior of the empty circle with a maximum Rh , and this activation strategy can effectively increase the network coverage area. At this moment, in selecting to activate the sleep node nearest to the center of this empty circle, the node becomes a candidate active node and is denoted by S_{Cand1} .

Moreover, the activation of a sleeping node from the exterior of the 2D convex hull can increase the area of the convex hull, thus forming a new 2D convex hull. In selecting to activate a sleeping node that can maximize the area of convex hull, the node becomes a candidate active node denoted by S_{Cand2} .

The next optimal active node must be either S_{Cand1} or S_{Cand2} , and the optimal activation strategy is shown in Formula (13):

$$Opt = \begin{cases} S_{Cand1}, & E_circle \geq H_add \\ S_{Cand2}, & H_add < E_circle \end{cases} \quad (13)$$

where E_circle denotes the maximum area of the empty circle when the active node is S_{Cand1} , H_add denotes the maximum area increase of the 2D convex hull when the active node is S_{Cand2} , and Opt denotes the node selected to be activated ultimately.

Node Opt will be added to Sa and deleted from Ss . Thereafter, in determining whether the current network coverage rate would satisfy the coverage requirement, if the result does not satisfy the

requirement, then the abovementioned strategy is adopted to continue activating a new sleeping node until the coverage threshold C_{th} is reached in this horizontal plane. Thus the whole optimal activation strategy process is completed, and the detailed flowchart of the activation strategy is as shown in Figure 6.

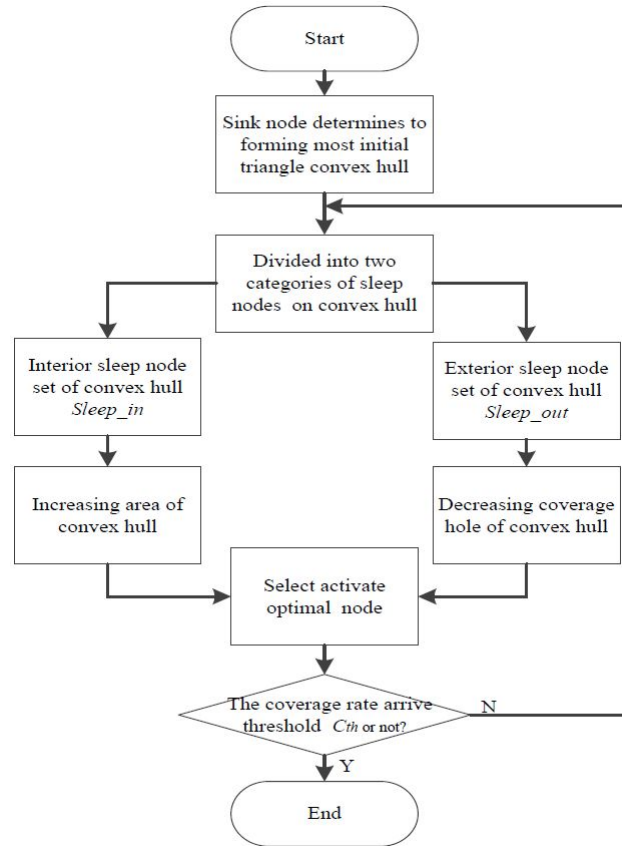


Figure 6. Flowchart of the optimal activation strategy.

4.2.2. Formation Process of 2D Convex Hull Spanning Tree

The sink node is the tree root node r found at the center of the water surface. It broadcasts information to other active nodes, according to the received signal strength from node S_i , to calculate distance ds_i ($1 \leq i \leq N_A$) between S_i and the sink node, and N_A denotes the number of active nodes. Moreover, the sink node assigns a unique activation identifier ID_A to all active nodes, where ID_A is recorded from 1 to N_A according to the size of ds_i . Furthermore, all active nodes in the convex hull are recorded as CH_nodes .

- (1) The tree root node r and set of active neighbor nodes Nei_A form an initial spanning tree $Tree_{init}$ as shown in Figure 7a. The iterative strategy is adopted to produce a next spanning tree generation according to the size of ID_A , until all active nodes of the 2D convex hull form a complete initial spanning forest $Forest_{init}$. Whether a parent node and its child nodes on spanning forest $Forest_{init}$ are able to communicate, if not, by Section 3.1.3, the communication radius is incremented until the parent node is connected to its child nodes so that the initial spanning forest $Forest_{init}$ can maintain full network connectivity.
- (2) The time marker T_{mark} ($0 \leq T_{mark} \leq T$) of root node r is initialized to 0, and the next generation of child nodes' time marker increments by 1, until all active nodes of the initial spanning forest $Forest_{init}$ are marked to the T level (i.e., all of CH_node are marked by the r node). The T_{mark} of the same subtree parent node set $\{f_j^i\}$ (j denotes ID_A ; i denotes T_{mark}) is recorded.

- (3) The active node inside the 2D convex hull $Sa_in(ID_A)$ acts as the parent of its neighboring sleeping nodes. Building a small spanning tree of individual nodes on $Forest_{init}$, we define a small spanning tree consisting of hybrid construction where the parent is the active node and the child is the sleeping node, as shown in Figure 7b.
- (4) To achieve the 3D global network coverage, the algorithm needs to deploy neighboring sleeping nodes $Nei(f_j^i)$ distributed at different depths and calculate the depth adjustment of the set of subtree parent nodes $\{f_j^i\}$ for their child sleeping nodes.

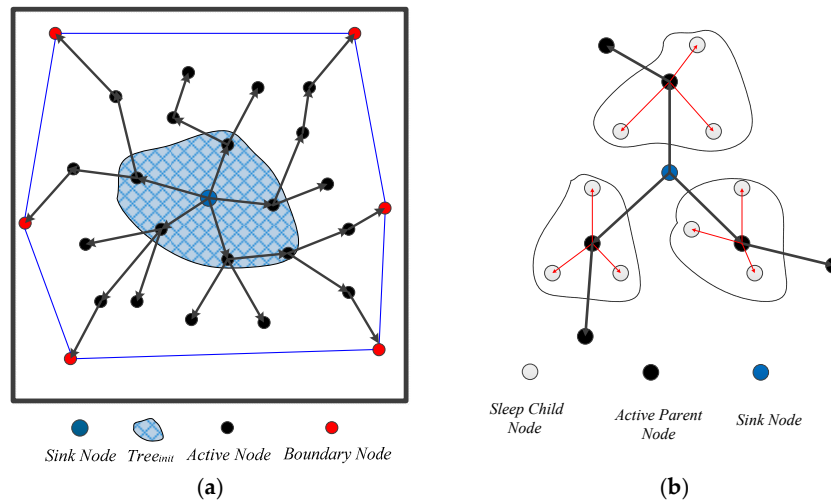


Figure 7. (a) Initial spanning forest and (b) Spanning tree consisting of hybrid construction.

4.2.3. Depth-Adjustment Strategy Based on Time Markers

For different parent nodes in $\{f_j^i\}$, we use a time marker T_{mark} to define network hierarchy. We need to consider the network energy consumption balance. A node near the sink node has low network hierarchy and high probability of data transmission and relay so it needs more energy. Therefore, for the network to have a sustained and stable operation, we should attempt to deploy more nodes in near the sink node and ensure that the surrounding sink node contains more subtrees. Moreover, subtree size should be as small as possible. Our goal is to reducing the redundancy degree of network coverage, increase the 3D space network coverage rate, and maintain constant full network connectivity between subtree parents and child nodes. This study proposes a strategy is based on the depth-adjustment strategy of time markers, and its detailed description is as follows:

- (1) The set of neighbor nodes $\{Nei(f_j^i)\}$ under the same network hierarchy (the same time marker i) are recorded, and then whether two sets of neighboring sleeping nodes intersect under the neighbor time marker and same path of tree is determined. If an intersection exists, then it is $NeigInter_i^{i+1}$ between time marker i and $i + 1$; otherwise, an empty set is recorded to $NeigInter_i^{i+1}$.
- (2) Time marker T_{mark} is initialized to 1, that is, $T_{mark} = 1$. The active node f_j^i inside the convex hull sends an awoken message $AWAKEN_MES$ to the set of non-intersecting neighboring nodes $Nei(f_j^i) - NeigInter_i^{i+1}$ and activates them. Node f_j^i builds a horizontal distance state table $Horizontal_Dis(f_j^i, Nei(f_j^i) - NeigInter_i^{i+1})$ according to the distance between the child elements of $Nei(f_j^i) - NeigInter_i^{i+1}$ and their parent nodes f_j^i from small to large. Thereafter, a node f_{min}^i is selected with a minimum value $min(Hor_Dis)$ on the state table as a depth-adjustment standard. Therefore, the drop-distance ver_dis of non-intersecting neighbor nodes set $Neig(f_j^i) - NeigInter_i^{i+1}$ is calculated by Equation (14):

$$ver_dis = \sqrt{Rc_{max}^2 - \{min(Hor_Dis)\}^2} \quad (14)$$

where $R_{c_{max}}$ denotes the maximum communication radius after adjusted and can maintain connectivity between parent node f_j^i and its child node.

- (3) Furthermore, node f_j^i submits a permission to its child node f_{min}^i and makes it as a parent node of the set of neighbor nodes $Nei(f_j^i) - NeigInter_i^{i+1}$. Thereafter, f_{min}^i is still used its time marker of parent node f_j^i . According to the size of the order of ID_A to all neighbor nodes on the same time marker adopting the depth-adjustment method of step (2) to adjust the depth of the other child nodes of $Nei(f_j^i) - NeigInter_i^{i+1}$, node f_j^i sends a sleep message $SLEEP_MES$ to its neighbor node and converts them to sleep state. Thereafter, $T_{mark} = T_{mark} + 1$, and the above depth-adjustment process is repeated for f_j^i until all CH_node is marked, achieving a layer depth adjustment.
- (4) The horizontal distance state table $Hor_Dis(f_j^i, Nei(f_j^i) - NeigInter_i^{i+1})$ is updated, and node f_{min}^i is adopted according to Steps (2) and (3) to achieve the next layer of the depth adjustment of its child node continually. If the node in this time marker drops down to a depth threshold d_{th} , then the depth adjustment on this time marker is completed. If the process finds no node to execute depth adjustment, that is, the set of neighbor nodes $Nei(f_j^i) - NeigInter_i^{i+1}$ is a null set, then node f_{min}^i applies to request other nodes in the set of neighbor intersections $NeigInter_i^{i+1}$ to continue the process of depth adjustment until all child nodes of the entire time marker complete the whole depth-adjustment process, finishing the global 3D network coverage deployment. The flowchart of the depth-adjustment strategy is shown in Figure 8.

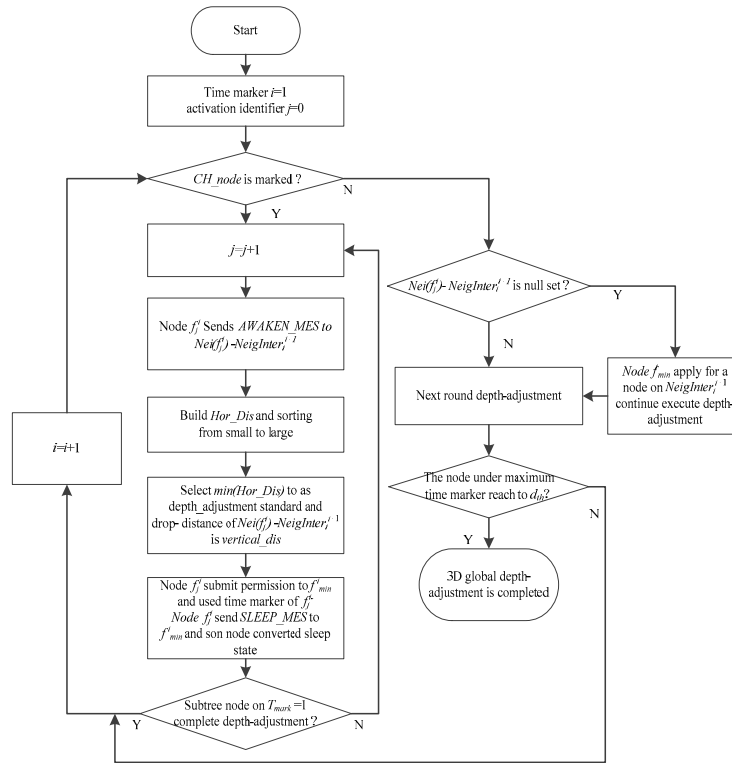


Figure 8. Flowchart of the depth-adjustment strategy.

5. Complexity Analysis

5.1. Communication Complexity Analysis

After the sink node completes the 2D convex hull activation process, the communication complexity of the process can be obtained, and it is $O(n \times h)$ according to the *GifWrapping* algorithm [34], where h denotes the required number of active nodes to achieve the 2D convex hull, and its asymptotic value is $L * W / \pi * R_s^2$. Given that the numbers of elements in

$Nei(f_j^i) - NeigInter_i^{i+1}$ and $NeigInter_i^{i+1}$ are N_{ij} and N_c^i respectively, node f_j^i broadcasts a message to its neighbor child nodes, and all parent nodes compare the distances of all neighbor child nodes and must send a total of $h * (n - h)$ messages for an overall situation significance. Node f_j^i submits the permission to its child node f_{min}^i for k times through an iterative method, sending a total of $k * N_A$ messages. The number of messages that must be sent by the parent node is twice the total number of nodes in the 2D convex hull activation process, namely, $2n$. After network deployment is completed, the value of N_A and n are approximately equal. Therefore, the complexity of network communication is $O(h * (n - h)) + O(k * N_A) + O(2n)$, then the big-O complexity is $O(cn)$, where $c = k + h + 2$.

5.2. Time Complexity Analysis of Network Deployment

Compared with TWSN deployment algorithms, UWSN deployment algorithms include the time cost of depth adjustment and the acoustic transmission delay. During network pre-deployment, the sink node calculates the optimal activation node position, and because the sink node's hardware configuration is high and has ability of fast calculation speed in general conditions, its time cost in the convex hull activation process can be ignored. The time cost is T_c of the node state changing from sleep to active or inversely, and acoustic transmission delay is T_d . Therefore, the total communication time cost is $2(Rc/p + T_d) + T_c$ between parent and its child nodes in the process of the spanning tree algorithm. Node f_j^i submits the permission to its child nodes f_{min}^i repeatedly, and its total time cost is $k * Rc/p + k * (2T_d + T_c)$. After the 2D spanning tree process is completed, approximately $(n - h)$ child nodes need to execute depth adjustment, and they will spend H/v until the water bottom in the worst case. Therefore, the average time cost of the global network deployment is around $2(Rc/p + T_d) + T_c + k * Rc/p + k * (2T_d + T_c) + H/v$, and the big-O time complexity of the NDACS algorithm is $O(H/v + k * Rc/p + k * (T_d + T_c))$. And the related symbolic descriptions are defined in Table 1.

Table 1. Symbolic description of complexity analysis.

H	Depth of Monitoring Space
v	Movement speed of node
K	Number of times of depth-adjustment
Rc	Communication radius of node
P	Speed of sound
T_d	Propagation delay of acoustic signal
T_c	Transition time from active to sleep or inversely

6. Simulation Evaluation

6.1. Parameter Settings and Evaluation Metrics

In analyzing and verifying the effectiveness of the NDACS algorithm, it is compared with VBDA in this study. Because VBDA is a node depth-adjustment deployment algorithm based on computational geometry, nodes can only move along the vertical direction, and VBDA's various hypotheses and preliminary conditions are similar to those of the NDACS algorithm. The algorithms are compared according to network connectivity rate and network coverage rate, energy consumption of communication and movement, and network reliability in this study.

MATLAB is used to simulate and analyze experiment results, and the simulation result is the mean of 30 experiments. In verifying the feasibility of the results, the 3D volume of water monitored is $500 \text{ m} \times 500 \text{ m} \times 400 \text{ m}$; the sensing radius R_s of the node is 50 m; the initial communication radius R_c is 60 m; and B_u is 5 m. When the network is deployed, the network coverage rate and connectivity rate are compared. Energy consumptions are compared based on the total energy consumed during the deployment process, and the rest of the parameters are set as shown in Table 2, with other settings derived from [30].

Table 2. Settings of Simulation Parameters.

Parameter	Value
Initial energy of node E_{init}	16,000 J
Coverage threshold C_{th}	0.9
Energy consumption on unit moving distance m_u	2 J/m
Power threshold P_0	0.05 W
Transmission delay T_p	0.2 s
Energy spreading factor k	2
Carrier frequency f	24 kHz
Sense radius of node R_s	50 m

6.2. Comparison and Analysis of Simulation Results

6.2.1. Network Coverage Rate

Figure 9 shows the comparison of the network coverage rate that varies with the number of nodes and the sensing radius on the NDACS and VBDA, respectively. Figure 9a shows the comparison of the network coverage rate varying with the number of nodes on the NDACS and VBDA, where the default sensing radius is 50 m in the simulation scenario. Figure 9b shows the comparison of the network coverage rate that varies with the sensing radius of the NDACS and VBDA, where the default number of nodes is 200 in the simulation scenario. As shown in Figure 9a, compared with the VBDA, the NDACS algorithm can maintain a higher network coverage rate consistently under the same sensing radius because VBDA can only guarantee the upper layer network with high network coverage but does not consider the coverage overlaps between its neighbor interlayer, resulting in the failure of deeper waters to achieve better coverage. Figure 9b shows that with the increase sensing radius, the NDACS algorithm for network coverage rate increases steadily and consistently outperforms VBDA. This is mainly because the NDACS algorithm, after completing the 2D convex hull activation process, executes depth-adjustment strategy of the time marker to greatly reduce the network coverage overlaps, and the set $NeigInter_i^{i+1}$ is established to provide more adjustment candidates on deeper location network deployment. Therefore, the NDACS algorithm can greatly improve the network coverage rate.

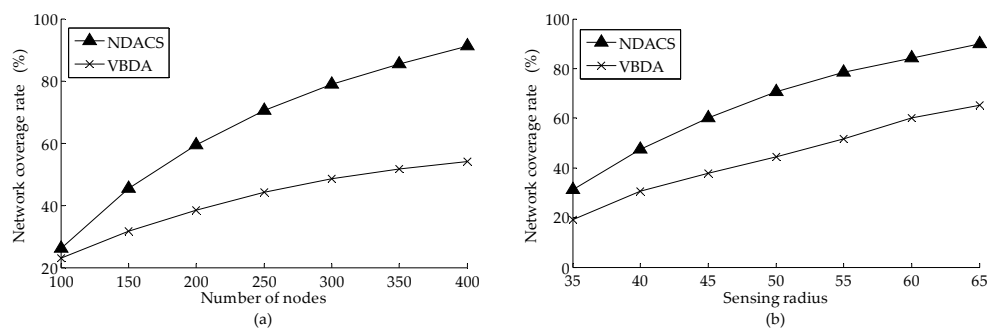


Figure 9. Comparison of network coverage rate between NDACS and VBDA: (a) comparison of network coverage rate between NDACS and VBDA varying with the number of nodes, and node sensing radius is 50 m; (b) comparison of network coverage rate between NDACS and VBDA varying with sensing radius, and the default number of nodes is 200.

6.2.2. Network Connectivity Rate

Figure 10 shows the comparison of the network connectivity rate that varies with number of nodes and the communication radius on NDACS and VBDA, respectively. Figure 10a shows the comparison of the network connectivity rate varying with the number of nodes on NDACS and VBDA,

where the default initial communication radius is 60 m in the simulation scenario. Figure 10b shows the comparison of the network connectivity rate varying with the communication radius of NDACS and VBDA, where the default number of nodes is 200 in the simulation scenario. Figure 10a,b show that compared with VBDA, the NDACS algorithm can guarantee the full network connectivity of the network (i.e., the network connectivity rate is equal 1). Figure 10c shows that VBDA can only maintain a relatively high network connectivity rate when $\alpha > 1.2$ (α is the ratio of node communication radius to sensing radius), and its connectivity rate will be very small in other situations. However, For the NDACS algorithm, the first layer network can guarantee full network connectivity on the spanning tree. In the depth-adjustment process, the parent node minimizes network coverage overlaps and maintains full network connectivity by using the adjustable communication radius model described in Section 3.1.3.

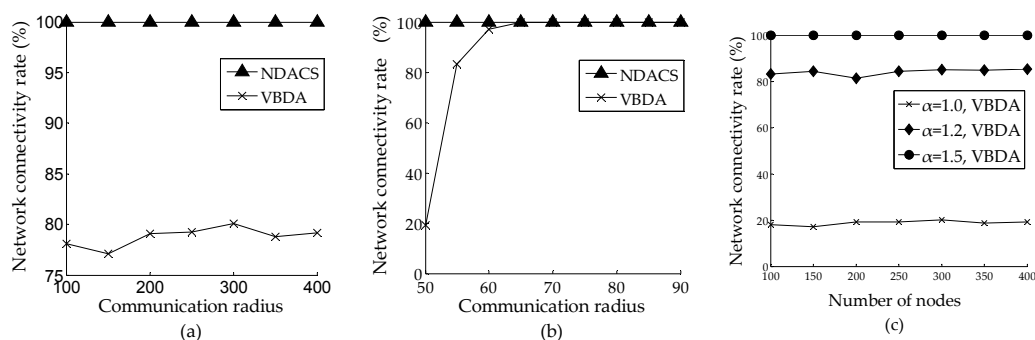


Figure 10. Comparison of network connectivity rate between NDACS and VBDA: (a) comparison of network connectivity rate between NDACS and VBDA varying with number of nodes, and the initial communication radius is 60 m; (b) comparison of network connectivity rate between NDACS and VBDA varying with communication radius, and the default number of nodes is 200; (c) network connectivity rate of VBDA varying with number of nodes on different values of α .

6.2.3. Energy Consumption of Communication and Movement

Figure 11a,b show the comparison of the each node's average energy consumption of communication and movement varying with the number of nodes on NDACS and VBDA, respectively, where the value of initial communication radius is 60 m in the simulation scenario (default settings for other parameters). As shown in Figure 11a, for VBDA, its rate of increase of average communication energy consumption is relatively fast as the number of nodes increases, and its value is always greater than that of the NDACS algorithm. This is because VBDA needs more communication energy, and all nodes are in an active state and must broadcast messages to their surrounding neighbor nodes. However, since the NDACS algorithm only needs some nodes in the active state, their states can be transformed flexibly through the spanning tree strategy. Therefore, VBDA needs more communication energy when the number of nodes increases. As shown in Figure 11b, the average movement energy consumption of VBDA presents a decreasing trend with the increase of the number of nodes. However, the average movement energy consumption of the NDACS algorithm surpasses that of VBDA, and it initially presents an increasing then decreasing trend. With this finding, the NDACS only executes depth-adjustment strategy on the set of non-intersecting neighbor nodes $Nei(f_j^i) - NeigInter_i^{i+1}$ for a long period of time. However, when $Nei(f_j^i) - NeigInter_i^{i+1}$ is null set, the NDACS algorithm may require the set $NeigInter_i^{i+1}$ to execute depth adjustment and deploy them to a deeper location such that they will consume more movement energy. Overall, as shown in Figure 10a,b, the total average energy consumption of VBDA is greater than that of the NDACS.

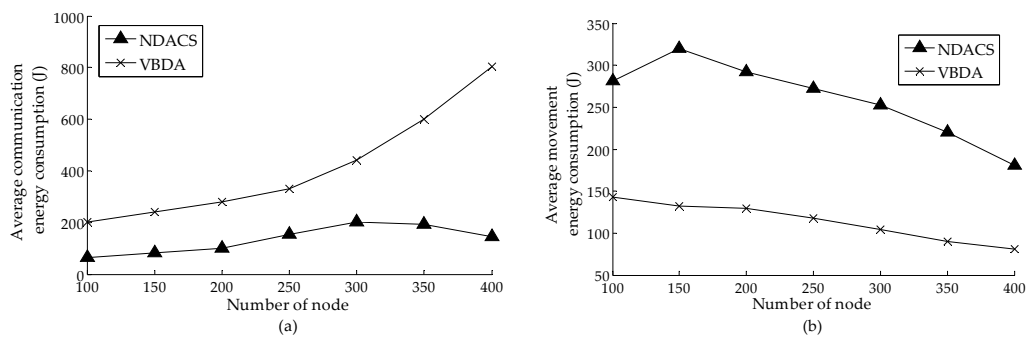


Figure 11. Comparison of energy consumption between NDACS and VBDA: (a) comparison of node average energy consumption of communication between NDACS and VBDA varying with the number of nodes, and the initial communication radius is 60 m; (b) comparison of node average energy consumption of movement between NDACS and VBDA varying with the number of nodes.

6.2.4. Network Reliability

Network reliability is defined in Section 3.2.3 as a criterion of QoS for UWSNs. In this section, Figure 12 shows the comparison of the average node degree varying with the number of nodes on NDACS and VBDA, and the influences of the simulation results on the different values of parameter α to average node degree are analyzed. As shown in Figure 12, compared with VBDA, the NDACS algorithm can maintain a higher average node degree, but the growth of the average node degree of VBDA is gentle, with an increment of the total nodes' number. With this finding, the NDACS algorithm can adjust the communication radius by using the adjustable communication radius model in Section 3.1.3 and considering real time connectivity between the parent node and its child nodes. However, VBDA cannot completely guarantee interlayer connectivity under different values of parameter α (as shown in Figure 10c). Therefore, the NDACS algorithm has a higher average node degree, which can maintain stronger communication connectivity between the node and the base station on the water surface, thus network operation is more reliable.

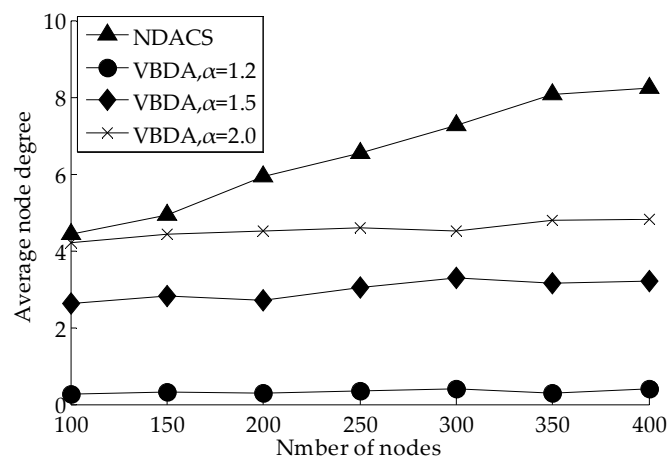


Figure 12. Comparison of average node degree between NDACS and VBDA.

7. Conclusions

This study addressed the problems of node depth-adjustment deployment (vertical movement deployment) for UWSNs. We have proposed NDACS, that forms a connected 2D convex hull spanning tree first and then connects the nodes of a spanning tree to calculate the depth-adjustment distances of their neighbor nodes based on time marker while maintaining the connectivity between the parent node

and their neighbor child nodes through the introduction of the mechanism of adjustable communication radius model, until the node deploys to the depth threshold and achieves the overall network coverage deployment. Through simulation experiments, we have evaluated the different indicators including network connectivity rate and network coverage rate, energy consumption of communication and movement, and network stability. The comparison of these simulation results shows that the various performances of the NDACS algorithm are better than those of VBDA. The symbolic notation of the NDACS algorithm is described in Table 3.

In the future, we plan to study the efficiency of network data acquisition when achieving network deployment by designing a more proper routing strategy for this deployment algorithm scenario. In real-world network systems, usually the model of communication and sensing is based on probability. The probability model of communication and sensing is worthwhile to adopt on simulation setup, then executes comparative analysis on experiment results. Meanwhile, the redeployment of the set of nodes $NeigInter_i^{i+1}$ is also worthy of further study. Furthermore, adding the obstacle model or water mobility model to increase the practical applicability of the algorithm is also included in our future research work.

Table 3. Main Symbolic Notation of the NDACS Algorithm.

Symbol	Definition
N	Number of nodes
ID	Node unique identifier
R_s	Node sensing radius
R_c	Node communication radius
$R_{c_{init}}$	Initial communication radius
θ	Adjustable level of communication radius
B_u	Accumulation of communication radius
S_i	ID is i of node
D	Transmitting distance of the information package
P_0	Power threshold of packets can be received
F	Carrier frequency
T_p	Transmission delay of data transmission
K	Energy spreading factor
M_e	Movement energy consumption of node
m_d	Node movement distance
m_u	Energy consumption on unit moving distance
S_s	Set of sleep nodes
S_a	Set of active nodes
$sleep_in$	Set of sleep nodes inside the convex hull
$sleep_out$	Set of sleep nodes outside the convex hull
S_c	Network coverage area on process of convex hull activation
S_{CH}	Area of 2d convex hull
S_{Hole}	Area of coverage holes
R_h	Empty circle radius
N_A	Total number of active nodes
Nei_A	Active neighbor nodes set
$Nei(f_j^i)$	Neighbor nodes set of node f_j^i
$NeigInter_i^{i+1}$	Insertion of neighbor sleep nodes
$\{f_j^i\}$	Parent node set of subtree
T_{mark}	Time marker of active node
d_{th}	Depth-adjustment threshold of network
C_{th}	Coverage threshold on 2d network surface
ID_A	Activation identifier of node
C_c	Network coverage rate
C_n	Network connectivity rate
A	Ratio of node communication radius to sensing radius

Acknowledgments: This paper was supported by the Special Project of National Key R&D Program of the Ministry of Science and Technology of China (2016YFC0201400), the National Natural Science Foundation of China (NSFC61273072), the National Natural Science Foundation of China and Zhejiang Joint Fund for Integrating of Informatization and Industrialization (U1509217), the National Natural Science Major Foundation of Research Instrumentation of China (61427808), the Key Foundation of China (61333009), the National Key Basic Research Program of China (2012CB821204).

Author Contributions: Peng Jiang and Shuai Liu conceived and designed the research; Peng Jiang and Shuai Liu performed the research; Peng Jiang, Shuai Liu, Jun Liu, Feng Wu and Le Zhang wrote the paper.

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

References

1. Heidemann, J.; Stojanovic, M.; Zorzi, M. Underwater sensor networks: Applications, advances and challenges. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2012**, *370*, 158–175. [[CrossRef](#)] [[PubMed](#)]
2. Gkikopouli, A.; Nikolakopoulos, G.; Manesis, S. A survey on underwater wireless sensor networks and applications. In Proceedings of the IEEE Conference on Control and Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 1147–1154.
3. Heidemann, J.; Ye, W.; Wills, J. Research challenges and applications for underwater sensor networking. In Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC), Las Vegas, NV, USA, 4–5 April 2006.
4. Bhambri, H.; Swaroop, A. Underwater sensor network: Architectures, challenges and applications. In Proceedings of the IEEE International Conference on Computing for Sustainable Global Development, New Delhi, India, 5–7 March 2014; pp. 915–920.
5. Huang, X.; Zhang, X. A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*. [[CrossRef](#)]
6. Temel, S.; Unaldi, N.; Kaynak, O. On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 111–120. [[CrossRef](#)]
7. Manvi, S.; Manjula, B. Issues in underwater acoustic sensor networks. *Int. J. Comput. Electr. Eng.* **2011**, *3*, 101–111.
8. Guerriero, F.; Violi, A.; Natalizio, E. Modelling and solving optimal placement problems in wireless sensor networks. *Appl. Math. Model.* **2011**, *35*, 230–241. [[CrossRef](#)]
9. Cai, S.B.; Zhang, G.Z.; Liu, S.L. Weighted Localization for Underwater Sensor Networks. *Lect. Notes Electr. Eng.* **2014**, *295*, 325–334.
10. Huang, C.J.; Wang, Y.W.; Lin, C.F. A self-healing clustering algorithm for underwater sensor networks. *Clust. Comput.* **2011**, *14*, 91–99. [[CrossRef](#)]
11. Cheng, W.; Teymorian, A.Y.; Ma, L. Underwater Localization in Sparse 3D Acoustic Sensor Networks. In Proceedings of the 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008.
12. Han, G.; Zhang, C.; Shu, L. A survey on deployment algorithms in underwater acoustic sensor networks. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*. [[CrossRef](#)]
13. Pompili, D.; Melodia, T.; Akyildiz, I.F. Deployment analysis in underwater acoustic wireless sensor networks. In Proceedings of the 1st ACM International Workshop on Underwater Networks, Los Angeles, CA, USA, 25 September 2006; pp. 48–55.
14. Akkaya, K.; Newell, A. Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks. *Comput. Commun.* **2009**, *32*, 1233–1244. [[CrossRef](#)]
15. Senel, F.; Akkaya, K.; Erol-Kantarci, M.; Yilmaz, T. Self-deployment of mobile underwater acoustic sensor networks for maximized coverage and guaranteed connectivity. *Ad Hoc Netw.* **2014**, *34*, 170–183. [[CrossRef](#)]
16. Alam, S.M.; Haas, Z.J. Coverage and connectivity in three-dimensional networks. In Proceedings of the 12th annual international conference on Mobile computing and networking, Los Angeles, CA, USA, 24–29 September 2006; pp. 346–357.
17. Alam, S.M.N.; Haas, Z.J. Coverage and connectivity in three-dimensional networks with random node deployment. *Ad Hoc Netw.* **2015**, *34*, 157–169. [[CrossRef](#)]
18. Liu, L. A deployment algorithm for underwater sensor networks in ocean environment. *J. Circuits Syst. Comput.* **2011**, *20*, 1051–1066. [[CrossRef](#)]

19. Xia, N.; Wang, C.S.; Zheng, R.; Jiang, J.G. Fish Swarm Inspired Underwater Sensor Deployment. *Acta Autom. Sin.* **2012**, *38*, 295–302. [[CrossRef](#)]
20. Du, H.; Xia, N.; Zheng, R. Particle Swarm Inspired Underwater Sensor Self-Deployment. *Sensors* **2014**, *14*, 15262–15281. [[CrossRef](#)] [[PubMed](#)]
21. Li, X.Y.; Ci, L.L.; Yang, M.H.; Tian, C.P.; Li, X. Deploying three-dimensional mobile sensor networks based on virtual forces algorithm. *Commun. Comput. Inf. Sci.* **2013**, *334*, 204–216.
22. Heo, N.; Varshney, P.K. Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2005**, *35*, 78–92. [[CrossRef](#)]
23. Wang, G.; Cao, G.; La, P.T. Movement-assisted sensor deployment. *IEEE Trans. Mob. Comput.* **2006**, *5*, 640–652. [[CrossRef](#)]
24. Wu, J.; Wang, Y.; Liu, L. A Voronoi-Based Depth-Adjustment Scheme for Underwater Wireless Sensor Networks. *Int. J. Smart Sens. Intell. Syst.* **2013**, *6*, 244–258.
25. Jiang, P.; Xu, Y.; Wu, F. Node Self-Deployment Algorithm Based on an Uneven Cluster with Radius Adjusting for Underwater Sensor Networks. *Sensors* **2016**, *16*. [[CrossRef](#)] [[PubMed](#)]
26. Jiang, P.; Wang, X.; Jiang, L. Node deployment algorithm based on connected tree for underwater sensor networks. *Sensors* **2015**, *15*, 16763–16785. [[CrossRef](#)] [[PubMed](#)]
27. Jiang, P.; Liu, J.; Wu, F.; Wang, J.; Xue, A. Node Deployment Algorithm for Underwater Sensor Networks Based on Connected Dominating Set. *Sensors* **2016**, *16*. [[CrossRef](#)] [[PubMed](#)]
28. Jiang, P.; Liu, J.; Wu, F. Node non-uniform deployment based on clustering algorithm for underwater sensor networks. *Sensors* **2015**, *15*, 29997–30010. [[CrossRef](#)] [[PubMed](#)]
29. Partan, J.; Kurose, J.; Levine, B.N. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2006**, *11*, 23–33. [[CrossRef](#)]
30. Ahmed, S.; Javaid, N.; Khan, F.A. Co-UWSN: Cooperative Energy Efficient Protocol for Underwater WSNs. *Int. J. Distrib. Sens. Netw.* **2015**, *75*. [[CrossRef](#)] [[PubMed](#)]
31. Wang, P.; Li, C.; Zheng, J. A Dependable Clustering Protocol for Survivable Underwater Sensor Networks, Communications. In Proceedings of the IEEE International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 3263–3268.
32. Saxena, S.; Mishra, S.; Singh, M. Clustering Based on Node Density in Heterogeneous Under-Water Sensor Network. *Int. J. Inf. Technol. Comput. Sci.* **2013**, *5*, 49–55. [[CrossRef](#)]
33. Preparata, F.P.; Hong, S.J. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM* **1977**, *20*, 87–93. [[CrossRef](#)]
34. Rachman, R.; Laksana, E.P.; Putra, D.S. Energy Consumption at the Node in Underwater Wireless Sensor Network (UWSNs). In Proceedings of the 6th UKSim/AMSS European Symposium on Computer Modeling and Simulation IEEE Computer Society, Valetta, Malta, 14–16 November 2012; pp. 418–423.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).