IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 24, 2019, accepted May 21, 2019, date of publication June 7, 2019, date of current version July 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921617

ATCFS: Effective Connectivity Restoration Scheme for Underwater Acoustic Sensor Networks

LINGFENG LIU[1,2], **MAODE MA**[3], **(Senior Member, IEEE), CHUNFENG LIU**[2], **(Member, IEEE), WENYU QU**[2], **(Senior Member, IEEE), GANG ZHANG**[2], **AND YANTAI SHU**[2], **(Member, IEEE)**

[1]National Ocean Technology Center, Tianjin 300112, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin 300072, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

Corresponding author: Chunfeng Liu (cfliu@tju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China–Guangdong Joint Fund under Grant U1701263, in part by the National Natural Science foundation of China (NSFC) under Grant 61871286, Grant 61672131, and Grant 61402327, in part by the Research and Development Foundation of Tianjin Transportation Science and Technology under Grant 2018-30, and in part by the Tianjin Key Laboratory of Advanced Networking (TANK).

ABSTRACT Underwater acoustic sensor network (UASN) has become one of the enabling technologies for the development of future ocean observation systems (OOSs). However, the UASN could be severely damaged because of the harsh environment that causes the simultaneous failure of many sensor nodes, thus leading to the partition of the network into multiple disconnected segments. In this paper, we studied the crucial issue to reestablish the network connectivity with the least quantity of employed relay nodes. To achieve a clear understanding of the issue, we present its integer nonlinear programming formulation, which is generally NP-hard. So, with the aim of solving the problem efficiently, an original heuristic scheme is proposed in this paper. Two fundamental algorithms are integrated into the scheme, namely, alternating tree construction and Fermat-point selection (ATCFS) as a whole. The results of extensive simulation experiment have confirmed that the ATCFS can solve this problem simply and effectively.

INDEX TERMS Computer network performance, network reliability, relay control systems, underwater acoustic communication.

I. INTRODUCTION

Underwater Acoustic Sensor Networks (UASNs) are envisioned to enable a growing list of potential applications, such as environmental monitoring, ocean's resources exploration, disasters prediction, and so on. An UASN comprises of a set of underwater sensor nodes (SNs) that communicate via acoustic link and are spread throughout the body of water. The inter-connectivity of those SNs is crucial for the data collection as well as their collaborative operation. In our previous work [1], we have discussed the strategy of prolonging the network lifetime, i.e. the time duration that the UASN connectivity could maintain. However, due to the harsh underwater environment, limited on-board energy or hostile destruction by an enemy, the network may suffer from failure of sensor nodes and is then divided into several isolated partitions. Therefore, the UASNs should be able to deal with such damages and reestablish the network connectivity in an efficient manner.

In UASNs, there are two major categories of sensor node failures, single SN failures and multiple SN failures [2]. In this work, we focus on restoring connectivity in the case of multiple SN failures, which is much more complicated than the single SN failures. To restore the damaged UASN topology, two classes of methods can be adopted. One is the proactive strategy that can provide additional resources to the network which could be used to relieve the effects of sensor node failures. The other is the reactive strategy which attempts to recover the lost connectivity when node failures took place. For multiple SN failures, proactive strategies, such as forming a k-connected network [3] or setting backups for critical nodes [4], [5], are not effective since the network

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

topology established by these techniques cannot tolerate collocated failures of many nodes [2]. Therefore, researchers had to develop reactive connectivity restoration strategies which fall into the following three categories.

A. RESTORATION THROUGH NODE RELOCATION

Some previous works, such as [6]–[8], have attempted to recover the network connectivity by moving some healthy nodes to specific positions or directions. For instance, Autonomous Repair (AuR) is an algorithm for WSN which try to restore connectivity based on the idea of Coulomb's law between charges. The neighbor nodes of the failed SNs initiate the restoration procedure. They move toward the damaged area and then toward the center of the WSN's working area [8]. In [9], two algorithms have been designed for UASN connectivity restoration whose nodes model is similar to ours. The proposed two algorithms can reestablish UASN connectivity by adjusting the depth of the sensor nodes. Such kind of strategies support network self-healing. The main challenge for them is to identify the scope of the failure nodes and coordinate the movement of the healthy nodes. However, the damage area may be too large to be restored by only repositioning some alive nodes in the network. Furthermore, the node relocation may change the sensor's coverage area, which could make the network 'useless' even if the connectivity is successfully restored.

B. RESTORATION WITH MOBILE NODES

In a mobility-assisted strategy, a mobile node serves as one or more following roles: a mobile relay node (MRN) which forwards data among the disjoint partitions, a mobile base station (MBS) which acts as a mobile destination of the sensor data, or a mobile data carrier (MDC) which accesses the isolated partitions and carries the data they produced to the base station [2]. In [10], authors have tried to recover the connectivity of individual partitions by employing some resource-rich MRNs. They have studied the influence of the number of the MRNs and the travel route selection on the data delivery latency. Work [11] has proposed a strategy to find an appropriate movement pattern for the MDCs with the purpose of minimizing the average delay of the overall network. The work [12] has studied the optimized movement pattern of the mobile nodes which served as MDCs and also MBSs. Seah et al. have proposed a solution in [13] that uses multiple Underwater Unmanned Vehicles (UUVs) as MRNs in UASNs to enhance connectivity by carrying the sensor data from the disconnected partitions to the connected partitions. Mobile nodes can carry the collected data and make the network capable of transient communication. However, the network topology cannot get stably repaired. Moreover, underwater mobile entities, such as UUVs, are very expensive and cannot be widely applied in engineering.

C. RESTORATION WITH STATIONARY NODES

After the network getting separated into multiple isolated segments, stationary relay nodes (RNs) could be used to restore the inter-segment connectivity. Taking into account the cost of restoration, minimizing the use of RNs has been an instinctive objective. Therefore, the connectivity recovery problem could be mapped to a classic NP-hard problem, the construction of a Steiner Minimum Tree with Minimum number of Steiner Points and Bounded Edge Length (SMT-MSPBEL) [14]. As a result, heuristic algorithms have been studied in the previous literature to find approximate solutions to this problem. In work [14], authors also have presented a heuristic algorithm. Firstly, the algorithm constructs a minimum spanning tree of the disjoint partitions. Then, along each edge of the tree, relay nodes are placed at every communication radius R_c. This algorithm can restore the network connectivity, but considers little about reducing the employment of RNs. In work [15], authors have proposed a bio-inspired heuristic algorithm named SpiderWeb which tries to restore the network connectivity and considers the factor of coverage and load balance. However, saving the employed relay nodes is not the main goal of this algorithm. Some other algorithms, such as Incremental Optimization based on Delaunay Triangulation (IO-DT) [16] and Federating network Segments via Triangular steiner tree Approximation (FeSTA) [17], deploy RNs and recover the network connectivity by finding local suboptimal solutions to the problem. Compared to work [14], these algorithms provide mechanisms to reduce the total quantity of necessary RNs. However, none of them has considered the characteristic of the three dimensional (3D) underwater scenario.

Motivated by the goal to repair a multiple node failure and reestablish the network connectivity of the UASNs, this work investigated the problem of Constrained UASN Connectivity Recovery (CUCR). The 'constrained' not only means the deployment points of the RNs on the surface of the water are predefined, but also means the maximal distance between any two directly connected nodes is limited. The objective of this research is to reconnect the partitioned network with the minimum number of RNs. The main contributions of this article can be summarized as follows.

1) The CUCR problem for 3D UASNs is defined and formulated. To the best of our knowledge, this is the first research work that raises and tries to solve the CUCR problem. As a major contribution we make in this work, the problem formulation of CUCR is presented in details. It is an integer nonlinear programming (INLP) and is a known NP-hard problem.

2) To efficiently solve the CUCR problem, an original heuristic scheme has been designed in this work, which is named as Alternating Tree Construction and Fermat-point Selection (ATCFS). The scheme consists of two fundamental algorithms including Tree Construction Algorithm (TCA) and Fermat-point Selection Algorithm (FSA). The TCA is a constrained steinerized minimum spanning tree construction algorithm, which is to establish connected topologies for the UASN. The FSA is a mixed tetrahedron and triangle based Fermat point selection algorithm which tries to reduce the total number of employed RNs. The above two algorithms

could be executed alternately because adding RNs on the Fermat points may change the construction of the constrained steinerized minimum spanning tree. The results of the simulation experiments have demonstrated the effectiveness of the ATCFS scheme.

This article is structured as below. The system model and the CUCR problem definition is presented in the next section. The heuristic scheme, ATCFS, is described in details in section III. Section IV provides the convergence and complexity analysis of the proposed scheme. The performance of the ATCFS is evaluated through simulation and presented in Section V. The major contributions of this paper are summarized and the directions of future work are discussed in the last section.

II. SYSTEM MODEL AND PROBLEM DEFINITION

A. SYSTEM MODEL AND ASSUMPTIONS

The problem based on the 3D underwater architecture is investigated in this article. It is modeled that the shape of the water body is a cuboid with length, width and depth as L, W and D, respectively. A 3D coordinate system with the axis of coordinates illustrated in Fig.1a was used to describe it. Generally, the nodes in the UASN can be categorized as sensor nodes (SNs) and relay nodes (RNs). It is assumed that both two types of nodes have an identical communication radius R_c. The establishment of an acoustic link between each pair of nodes can be achieved on the condition that they are within the communication radius of each other.

We deploy the SNs in the water body based on a given sensing application such as intrusion detection. Their underwater locations are predetermined and not adjustable. Because the operation of UASN normally happens in harsh environments, it prompts the easy and frequent malfunction of SNs. Large scale malfunction may separate the network into many separated partitions. In the scenario of this paper, the UASN has been split into N disjoint partitions. It is assumed that each partition i is represented by one SN h_i, which is called a head node (HN). The set of all HNs is denoted as H. The distance between each pair of HNs must be larger than R_c due to their disconnected property. The basic objective of this paper is to restore the connectivity of these HNs and thus the connectivity of the UASN can be reestablished.

RNs, capable of communication only, are applied to restore the connectivity of the UASN. Considering the engineering factor, the RNs are carried by an Unmanned Surface Vehicle (USV) which patrols on the surface of the water [18]. It can deploy RNs along its route on demand at the predefined candidate deployment point as shown in Fig.1b. The RNs can adjust their depth in the vertical direction [19]. Let S be the set of all candidate deployment points on the surface of the water. The distance between the two adjacent candidate deployment points is d in the x axis direction and r in the y axis direction. The value of d and r must not be larger than R_c in order to guarantee the connectivity recovery function. For each candidate deployment point i, k_i RNs are actually deployed. R is the set of all actually deployed RNs. The size of R is denoted as $|R|$, $|R| = \sum_S k_i$.

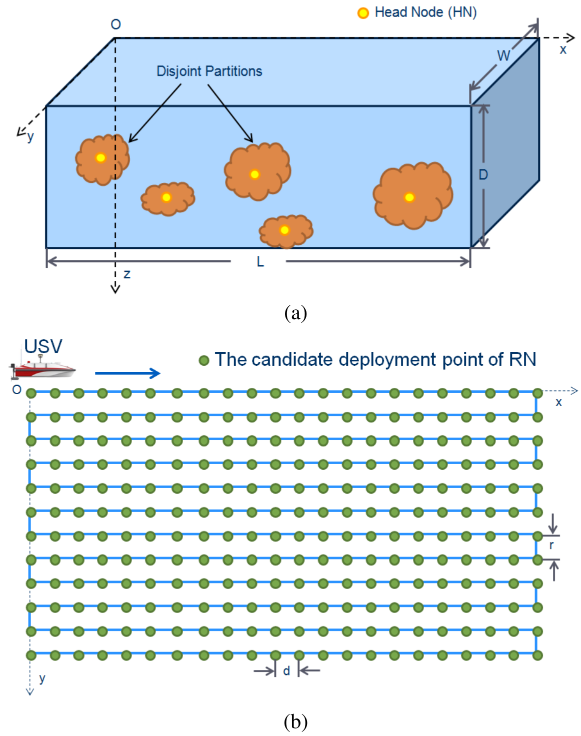


FIGURE 1. The system model.

Definition 1: If a set of RNs R is needed to reestablish the connectivity of a set of nodes A by mechanism T, the size of R, $|R|$, is called the cost to connect A by T and is denoted as $Cost_T(A) = |R|$.

B. PROBLEM DEFINITION AND ANALYSIS

Considering the cost of RNs, the following problem needs to be addressed: Given N disconnected underwater partitions, which is represented by the set of HNs H, determine the least number of needed RNs to reestablish the connectivity of these partitions, i.e. using the minimum amount of RNs to form a connected graph G spanning H, such that every edge in G has length at most R_c and the x-y coordinates of RNs can only be selected from a predefined set S. In the context of this paper, the above problem is denoted as a Constrained UASN Connectivity Recovery (CUCR) problem.

The CUCR problem could be defined more formal as follows. The objective of the CUCR problem is $min(|R|)$, i.e. to minimize the cost to connect H, and form a graph $G = (V, E)$ where $V = H \bigcup R$, i.e. V is the union of the set of all HNs H and deployed RNs R, and E is the set of links between two nodes of V. G should satisfy the following three properties.

1) G is a connected graph and can span all HNs, i.e. for $\forall a, b \in H$, a path $Path(a, b) = (a, p_1, p_2, \cdots, p_k, b)$

can be found in G where $p_1, p_2, \cdots, p_k \in V$ and $(a, p_1), (p_1, p_2), \cdots, (p_{k-1}, p_k), (p_k, b) \in E$.

2) The edge length of any e in E is smaller than R_c, i.e. for $\forall e = (i, j) \in E$, the 3D Euclidean distance $d(i, j) \leq R_c$.

3) The x-y coordinates of RNs can only be selected from S, i.e. for $\forall r = (x_r, y_r, z_r) \in R$, $\exists s \in S, s.t.\ s = (x_r, y_r)$.

The CUCR problem can be formulated as below:

$$\min(|R|) = \min(\sum_{S} k_i) \tag{1}$$

$$\text{subject to: } k_i \geq 0, \quad k_i \in I \tag{2}$$

$$(x_r, y_r) \in S, \quad \forall r = (x_r, y_r, z_r) \in R \tag{3}$$

$$0 \leq z_r \leq D, \quad z_r \in I \tag{4}$$

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \quad \forall i, j \in H \cup R \tag{5}$$

$$a_{ij} = \begin{cases} 1 & d(i, j) \leq R_c \\ 0 & d(i, j) > R_c, \end{cases} \quad \forall i, j \in H \cup R \tag{6}$$

$$A = (a_{ij})_{(N+|R|)\times(N+|R|)} \tag{7}$$

$$B = (b_{ij})_{(N+|R|)\times(N+|R|)} = \sum_{i=1}^{N+|R|-1} A^i \tag{8}$$

$$b_{ij} \neq 0, \quad \forall i, j \in \{1, 2 \ldots, N + |R|\} \tag{9}$$

The Equation (1) presents the optimization goal of CUCR which is to minimize the total RNs employed. The constraint expressed by (2) is the number of RNs at each candidate deployment point is a non-negative integer (I denotes integer numbers). Constraint (3) is the x-y coordinates' constraint of RNs. Constraint (4) point out that the z-coordinates of RNs must be a non-negative integer and cannot exceed the depth of the water body. The equation (5) is the three dimension Euclidean distance computation formula between two nodes of the UASN. The equations (6) and (7) construct an adjacent matrix A. The value of its elements is determined by the distance between the corresponding two nodes. If they are within the communication radius R_c, the value is 1, otherwise the value is 0. Based on the judgment criterion of a graph connectivity described in work [20], the equation (8) constructs a judgment matrix B, which is calculated by the matrix operation of A. If there is no zero element in B as constraint (9), the graph whose vertex is composed of all HNs and the deployed RNs is a connected graph.

The formulation of the CUCR problem is in the form of integer nonlinear programming (INLP), which is generally NP-hard [21]. Current algorithms for general INLPs include outer approximation, branch and bound, branch and reduce, and generalized benders decomposition, (just to name a few). Some software packages can solve INLPs to proven optimality, such as BARON, GloMIQO, LINDO-Global, α-BB, and Couenne. However, a reasonably good solution could be found by these tools only for a small number of variables [22]. To tackle the CUCR problem effectively, we propose the heuristic scheme in the following section.

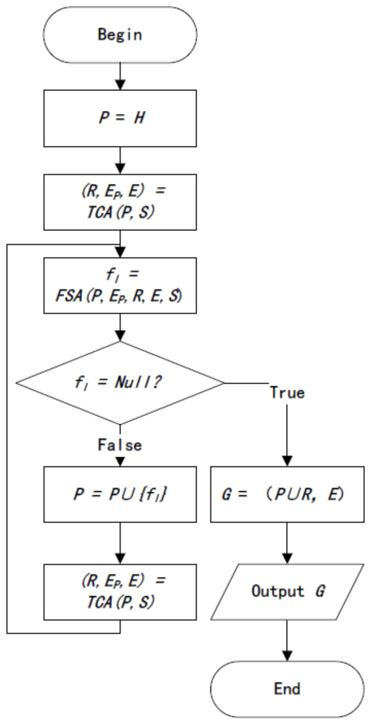


FIGURE 2. The procedure of ATCFS.

III. ATCFS: A HEURISTIC SOLUTION

Based on the definition and formulation of the CUCR problem, to recover the network connectivity is the basic objective that needs to be guaranteed. Under the condition of implementing the connectivity reestablishment, RNs should be used as few as possible, while the connectivity property of the UASN still remains. However, it is hard to achieve the two objectives simultaneously especially under the situation that the number of isolated partitions is large. So our basic idea to solve the problem is to optimize the two objectives alternately.

First, Tree Construction Algorithm (TCA), a constraint steinerized minimum spanning tree construction algorithm, is executed. The objective of TCA is to restore the UASN connectivity. Certainly, the cost to restore the connectivity may not be optimal. So a relay-node optimization algorithm FSA, a mixed tetrahedron and triangle based Fermat point selection algorithm, is then invoked trying to reduce the total number of employed RNs. The above two algorithms could be executed alternately because adding RNs on the Fermat points may change the construction of constraint steinerized minimum spanning tree. So the overall integrated heuristic scheme is named as Alternating Tree Construction and Fermat-point Selection (ATCFS). Fig. 2 illustrates the procedure of the ATCFS in a sequence flow diagram. The ATCFS ends if no new Fermat points could be selected to further reduce the cost and returns the connected graph G as its output.

A. TREE CONSTRUCTION ALGORITHM (TCA)

To reestablish the network connectivity, the most intuitive idea is forming a steinerized minimum spanning tree (MST) with the RNs deployed on its steiner vertexes. For WSN, Lin et al. [14] have proposed an algorithm called the Minimum Spanning Tree Heuristic (MSTH) which populates RNs at a distance of at most R_c apart on the edges of the MST. However, this algorithm cannot be directly applied in this paper, not only because the 3D underwater scenario is different from 2D WSN, but also the predefined x-y coordinates should be considered when RNs are used as steiner vertexes. So an algorithm named Tree Construction Algorithm (TCA) is proposed and have the following steps to work.

Step1: Using Kruskal's minimum spanning tree construction algorithm [23], a MST $T = (P, E_P)$ is constructed where P is the set of the nodes needed to be connected and E_P indicates the set of the edges of the MST. For the first time TCA runs in ATCFS, $P = H$, i.e. the set of all head nodes.

Step2: Then for each edge (u, v) in E_P whose length is longer than R_c, it should be subdivided into several short edges with the length not greater than R_c. Due to the x-y coordinates constraint of RN, this cannot be done as MSTH which directly inserts degree-2 Steiner points (RNs) to subdivide the edge (u, v) as is shown in Fig.3a. The idea of TCA is to select appropriate deploy points of RNs from the surface of the water and adjust the depth of them around the edge (u, v) to form the shortest path that can connect node u and v with a fold line. Intuitively, the fold line should be as close as possible to the straight line because it is the shortest distance between two points. Connecting the two points of T with the fold line is shown in Fig.3b.

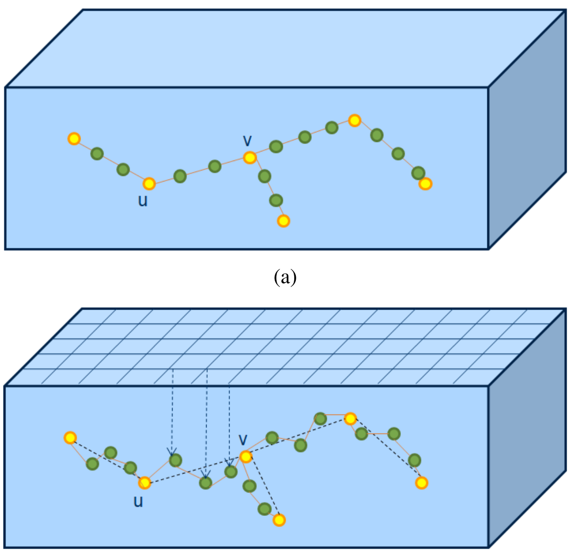


FIGURE 3. Illustration of connecting the points of T with RNs.

So starting from u and let $r_0 = u$, the algorithm determines the positions of RNs between u and v one by one. If the position before $r_i (i \geq 0)$ has been decided and v is not in the communication radius of r_i, the following two tasks should be accomplished to decide the position of r_{i+1}. Firstly, the set Q which is the RN's candidate deployment points on the surface of the water should be identified by ensuring that the 2D Euclidean distance between r_i and the candidate deployment point $q \in Q$ is smaller than R_c, i.e. $D(r_i, q) \leq R_c$, where $D(r_i, q)$ indicates the 2D Euclidean distance between r_i and q. Secondly, for each candidate deployment point $q \in Q$, $x_{r_{i+1}} = x_q$, $y_{r_{i+1}} = y_q$ and $z_{r_{i+1}}$ is a non-negative integer according to the coordinate system of this paper. Then find a position of r_{i+1} whose coordinates can minimize the angle θ between vector $\overrightarrow{uv}$ and $\overrightarrow{r_i, r_{i+1}}$. Note that the position of r_{i+1} must keep the constraint that it must be in the communication radius of r_i. The objective and the constraints of the selection of r_{i+1} can be formulated as follows.

$$\min(\theta) = \min\left(\angle\left(\overrightarrow{uv}, \overrightarrow{r_i r_{i+1}}\right)\right) = \min\left(\arccos\left(\frac{\left(\overrightarrow{uv} \cdot \overrightarrow{r_i r_{i+1}}\right)}{\left|\overrightarrow{uv}\right| \cdot \left|\overrightarrow{r_i r_{i+1}}\right|}\right)\right) \tag{10}$$

$$\text{subject to: } \left|\overrightarrow{r_i r_{i+1}}\right| = \sqrt{\left(x_{r_{i+1}} - x_{r_i}\right)^2 + \left(y_{r_{i+1}} - y_{r_i}\right)^2 + \left(z_{r_{i+1}} - z_{r_i}\right)^2} \leq R_c \tag{11}$$

$$D\left(r_i, r_{i+1}\right) = \sqrt{\left(x_{r_{i+1}} - x_{r_i}\right)^2 + \left(y_{r_{i+1}} - y_{r_i}\right)^2} \leq R_c \tag{12}$$

$$\left(x_{r_{i+1}}, y_{r_{i+1}}\right) \in S \tag{13}$$

$$0 \leq z_{r_{i+1}} \leq D, z_{r_{i+1}} \in I \tag{14}$$

$$\left|\overrightarrow{uv}\right| = \sqrt{(x_v - x_u)^2 + (y_v - y_u)^2 + (z_v - z_u)^2} \tag{15}$$

$$\overrightarrow{uv} \cdot \overrightarrow{r_i r_{i+1}} = (x_v - x_u)\left(x_{r_{i+1}} - x_{r_i}\right) + (y_v - y_u)\left(y_{r_{i+1}} - y_{r_i}\right) + (z_v - z_u)\left(z_{r_{i+1}} - z_{r_i}\right) \tag{16}$$

In the above expression, the variables are the coordinates of r_{i+1}. Because the number of candidate deployment point that satisfied (12) is very limited, an analytic solution of r_{i+1} can be easily obtained and expressed by:

$$r_{i+1} = \left(x_{r_{i+1}}, y_{r_{i+1}}, z_{r_{i+1}}\right) = \arg\min(\theta) \tag{17}$$

The procedure to find the next RN's position ends until v is in the communication radius of the current RN. The fold line and the corresponding RNs that connect u and v are determined. These RNs are then added to R. According to Definition 1, the cost to connect u and v is the amount of RNs on the fold line. The edges, $(u, r_1), (r_1, r_2), \ldots, (r_k, v)$, are added to E. TCA does not end until all edges in E_P are subdivided into short edges with the length not greater than R_c. Then all edges that are used to connect the nodes in P are in set E. Algorithm 1 gives out the corresponding pseudo-code of TCA.

Algorithm 1 TCA (Tree Construction Algorithm)

Input: The set of RN's candidate deployment points on the water surface, S; The set of nodes that needs to be connected, P;

Output: The set of edges of the MST, E_P; The set of selected RNs with their locations, R; The set of edges that made up the connected graph, E;

Construct $T = (P, E_P)$ with Kruskal's minimum spanning tree algorithm;
for each uv in E_P **do**
  $r_0 \Leftarrow u, i \Leftarrow 0$;
  while $(d(r_i, v) > R_c)$ **do**
    Compute the coordinates of r_{i+1} using Equ.(10)-(17);
    Add r_{i+1} to R;
    Add $r_i r_{i+1}$ to E;
    $i \Leftarrow i + 1$;
  end while
  Add $r_i v$ to E;
end for
return R, E_P, E;

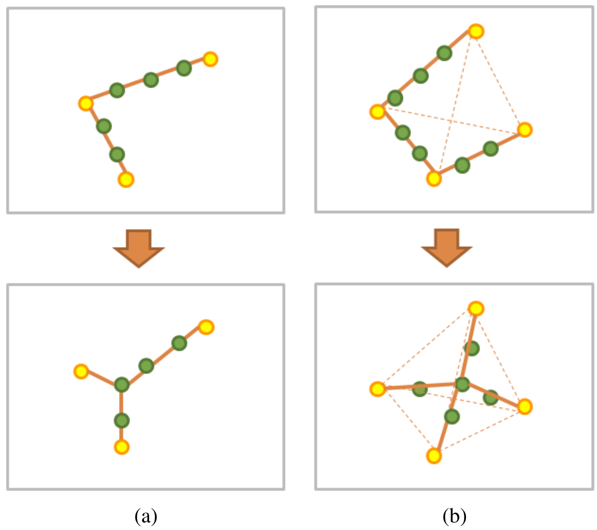


FIGURE 4. Two examples of how to use the Fermat point to reduce the employed RNs.

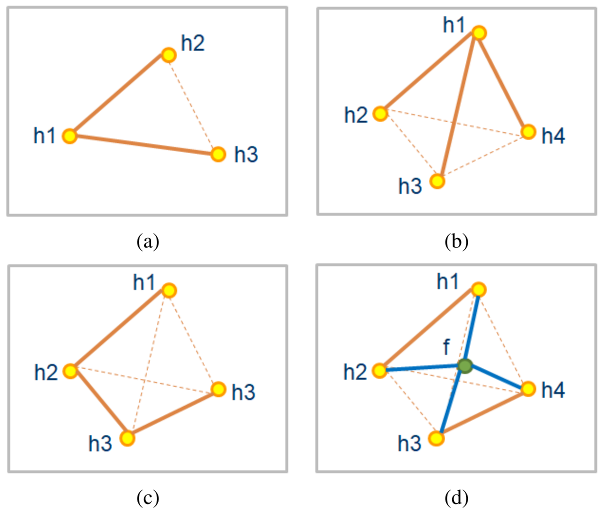


FIGURE 5. Illustration of identifying the triangular and tetrahedral subsets from T.

B. FERMAT-POINT SELECTION ALGORITHM (FSA)

TCA can guarantee the connectivity recovery of the UASN. However, the amount of RNs that are employed to fill the gap of the HNs may not be optimal. The main idea of this work to reduce the number of RNs is to introduce a Fermat point for each triangular or tetrahedral subset of T, the tree constructed in the first step of TCA. The Fermat point f is an inner point that can minimize the sum of Euclidean distance from itself to the vertexes of a triangle or a tetrahedron [24]. So with the help of the Fermat point, the count of RNs that need to connect the HNs could be reduced. The examples that use Fermat point to reduce the required number of RNs for triangle and tetrahedron are illustrated in Fig.4. For the convenience of expression, we directly plot the RNs on the edges that connecting two nodes instead of showing their actual positions on the fold lines around the edges. Certainly, if the suitable position of RNs are considered for more HNs at the same time, the cost to connect these HNs can be reduced more effectively. Nevertheless, the time complexity of the algorithm boosts rapidly with the number of vertexes increases [17]. So only three vertexes (triangular case) and four vertexes (tetrahedral case) are considered to find a sub-optimal solution to connect them. The algorithm includes the following three steps.

Step1: The triangular and tetrahedral subsets should be identified from T. It is relatively easy to find a triangular subset by selecting two edges that share a same vertex. For example, in Fig.5a, edge (h_1, h_2) and (h_1, h_3) share the same vertex h_1, so (h_1, h_2, h_3) is a triangular subset and is added to the triangle set Tr. To identify the tetrahedral subsets from T, two cases need to be considered. First, if three edges of T share a same vertex, the four vertexes of three edges form a tetrahedral subset. As is depicted in Fig.5b where edge (h_1, h_2), (h_1, h_3) and (h_1, h_4) share the same vertex h_1, so (h_1, h_2, h_3, h_4) is a tetrahedral subset. Second, if three edges are linked end-to-end as is illustrated in Fig.5c, they can also form a tetrahedral subset. For the case illustrated in Fig.5d, (h_1, h_2, h_3, h_4) is a tetrahedral subset, however Fermat point f cannot reduce the cost to connect $\{h_1, h_2, h_3, h_4\}$. Because the sum of any two sides of a triangle is greater than the third one, i.e. $d(f, h_1) + d(f, h_2) \geq d(h_1, h_2)$ and $d(f, h_3) + d(f, h_4) \geq d(h_3, h_4)$, then $d(f, h_1) + d(f, h_2) + d(f, h_3) + d(f, h_4) \geq d(h_1, h_2) + d(h_3, h_4)$ which means the cost to connect $\{h_1, h_2, h_3, h_4\}$ with f is not less than the cost without f. As a result, this case is not needed to be considered. All tetrahedral subsets are added to the tetrahedron set Te.

Step2: In the second step, the coordinates of the Fermat point should be determined for each triangle or tetrahedron in $Tr \cup Te$. Here we only analyze the tetrahedron case, the triangle case is relatively simple and can be done in a similar way. According to the definition of Fermat point, it should be an inner point that minimizes the distance to the four vertexes of a tetrahedron. However, due to the x-y coordinates of RNs can only be selected from predefined locations, only an Approximate Fermat Point (AFP) could be determined. For a tetrahedron $t = (h_1, h_2, h_3, h_4) \in Te$, the objective and the constraints of the selection of its AFP f_t can be formulated as follows.

$$\min \ (d(f_t, h_1) + d(f_t, h_2) + d(f_t, h_3) + d(f_t, h_4)) \tag{18}$$

$$\text{subject to: } (x_{f_t}, y_{f_t}) \in S \tag{19}$$

$$0 \leq z_{f_t} \leq D, z_{f_t} \in I \tag{20}$$

$$d(f_t, u) = \sqrt{(x_{f_t} - x_u)^2 + (y_{f_t} - y_u)^2 + (z_{f_t} - z_u)^2}, \quad u \in \{h_1, h_2, h_3, h_4\} \tag{21}$$

As a result, the coordinates of AFP is $f_t = (x_{f_t}, y_{f_t}, z_{f_t}) = argmin(d(f_t, h_1) + d(f_t, h_2) + d(f_t, h_3) + d(f_t, h_4))$. However, from the formulation of the AFP selection, we can see that it is also an integer nonlinear programming (INLP) problem. In order to tackle it effectively, this work follows a Simulated Annealing (SA) style heuristic algorithm. The algorithm is connected with a geometric progression $T = \{T_0, T_1, \ldots T_M\}$ which is also called a temperature series in the classical SA algorithm. For any $T_i \in T$, $T_i = T_0\alpha^i$, $T_0 > 0$, $0 < \alpha < 1$. $T_M \leq \varepsilon$ where ε is the termination parameter of the algorithm. Starting from T_0, $M + 1$ groups of searching is performed. The initial position of AFP L_0 is defined by the following equation.

$$L_0 = (x_0, y_0, z_0) = \left(\left\lfloor \frac{x_{h_1} + x_{h_2} + x_{h_3} + x_{h_4}}{4d} \right\rfloor \times d, \times \left\lfloor \frac{y_{h_1} + y_{h_2} + y_{h_3} + y_{h_4}}{4r} \right\rfloor \times r, \times \left\lfloor \frac{z_{h_1} + z_{h_2} + z_{h_3} + z_{h_4}}{4} \right\rfloor \right) \tag{22}$$

Aided by L_0, the cost to connect $\{h_1, h_2, h_3, h_4\}$ is $C(L_0)$. For each T_i (each group of searching), K possible positions of AFP are computed and compared. At any step j ($j \geq 1$), the algorithm randomly selects a new adjacent position of L_{j-1}, i.e. $L_j = (x_{j-1} \pm d, y_{j-1} \pm r, z_{j-1} \pm 1)$ which satisfies that the x, y and z coordinates of L_j are within the water body. Then we calculates its cost $C(L_j)$. The algorithm should decide whether to move to the new position or to stay with the current position. The decision is made based on whether the cost of new solution is smaller than the current one. So, $\Delta Cost = C(L_j) - C(L_{j-1})$ is calculated. If $\Delta Cost < 0$, the new position L_j is accepted. Otherwise, $\Delta Cost > 0$, the new position L_j is accepted with a probability of $P_A = e^{-\frac{\Delta Cost}{T_i}}$ to avoid the local optimum.

The SA algorithm ends until K possible AFP positions of T_M are searched. At the end of this step, each triangle or tetrahedron $t \in Tr \cup Te$ has determined the position of AFP f_t and got its corresponding cost $Cost_{AFP}(t)$.

Step3: For each triangle or tetrahedron $t \in Tr \cup Te$, two approaches could be used to connect the vertexes of t. $Cost_{TCA}(t)$ denote the cost to connect t by the algorithm TCA. $Cost_{AFP}(t)$ is the cost to connect the vertexes of t by adding AFP to the triangular or tetrahedral subset. Then the cost difference between $Cost_{AFP}(t)$ and $Cost_{TCA}(t)$ can be calculated for each t. $Diff(t) = Cost_{TCA}(t) - Cost_{AFP}(t)$. The algorithm iterates over $Tr \cup Te$. The triangle or tetrahedron l which has the highest positive cost difference, i.e. $l = argmax(Diff(t))$, $t \in Tr \cup Te$, is selected and its corresponding AFP f_l is the output of FSA. Algorithm 2 presents the corresponding pseudo-code of FSA.

Algorithm 2 FSA (Fermat-Point Selection Algorithm)

Input: The set of nodes that needs to be connected, P; The set of the edges of the MST, E_P; The set of selected RNs with their locations, R; The set of the edges that made up the connected graph, E; The set of RN's candidate deployment points on the water surface, S;

Output: The coordinates of AFP which corresponding triangle or tetrahedron has the largest cost difference, f_l;

```
Identify the triangle set Tr and tetrahedron set Te from T = (P, E_P);
MaxDiff = 0, f_l = NULL;
for each t in Tr ⋃ Te do
    Determine the AFP f_t for t;
    Diff(t) = Cost_TCA(t) − Cost_AFP(t);
    if Diff(t) > MaxDiff then
        MaxDiff ⇐ Diff(t);
        f_l = f_t;
    end if
end for
return f_l;
```

IV. ANALYSIS OF THE ATCFS HEURISTIC

A. CONVERGENCE ANALYSIS

First, we need to prove the correctness of the proposed ATCFS scheme, i.e. the connectivity of the disjoint network can be reestablished by ATCFS and the alternation between the two algorithms (TCA and FSA) used in the solution finally ends.

Theorem 1: ATCFS can connect all disjoint partitions and the scheme is convergent.

Proof: The conclusion that ATCFS can guarantee the connectivity restoration is easy to prove. In ATCFS, TCA construct a steinerized minimum spanning tree of HNs with finite number of RNs. Based on the definition of a minimum spanning tree [14], it can connect all HNs and the HNs can connect with all other sensor nodes in their corresponding disjoint partitions. So, ATCFS can reestablish the connectivity for a partitioned UASN.

To prove the convergence of ATCFS, we need to analyze the execution procedure of the proposed heuristic. As is shown in Fig. 2, TCA first runs to connect all HNs and

M RNs are employed. Then FSA runs. For this algorithm, 0 or 1 Fermat point could be selected as its output. If no Fermat point is selected, the ATCFS ends. If one Fermat point is selected, at least one RN can be saved according to the procedure of FSA. Then for the first round of ATCFS, the total cost to reestablish the connectivity is $M - s_1 \leq M - 1$ where s_1 denotes the number of RNs saved in this round. If X rounds are executed, the number of employed RNs is $M - \sum_{i=1}^{X} s_i \leq M - X$. One important fact is that the initial network is disconnected. As a result, the number of RNs that needed to restore the connectivity must be more than 0, i.e. $M - \sum_{i=1}^{X} s_i > 0$. So, $M - X \geq M - \sum_{i=1}^{X} s_i > 0 \Rightarrow X < M$. This means the total number of rounds that TCA and FSA runs cannot be larger than the finite number of RNs which is calculated from the first time that TCA runs. So the proposed ATCFS scheme is convergent. ■

B. COMPLEXITY ANALYSIS

The worst-case time complexity of ATCFS is analyzed in this subsection and FeSTA-C is used for comparison. The FeSTA is a typical algorithm for 2D WSN topology reparation, which iteratively finds the best subsets of nodes of size 3 and establishes their connectivity with the minimum number of RNs [17]. The FeSTA-C is a variant of FeSTA, that extends the FeSTA to a 3D UASN scenario with the x-y coordinates' constraint of the RNs' deployment point.

To analyze the time complexity of the ATCFS, the two algorithms in it are first analyzed respectively. For TCA, the worst case time complexity of Kruskal algorithm that constructs the minimum spanning tree is $O(|P| \log |P|)$ [23] where P is the set of nodes that needs to be connected and $|P|$ is the size of P. So the time complexity of the first step of TCA is $O(|P| \log |P|)$. For each edge in E_P, the time complexity that determines the positions of RNs that connect its two endpoints is $O(\left\lfloor \frac{R_C}{r} \right\rfloor \times \left\lfloor \frac{R_C}{d} \right\rfloor)$. The number of edges in E_P is $|P| - 1$. So the time complexity of the second step of TCA is $O(|P|)$. As a result, the worst case time complexity of TCA is $O(|P| \log |P|)$.

For algorithm FSA, the worst-case time complexity of its first step that identifies the triangle set and tetrahedron set from the minimum spanning tree is $O(|P|^2 + |P|^3) = O(|P|^3)$. For each t in $Tr \bigcup Te$, the time complexity of the Simulated Annealing algorithm that determines the AFP of t is $K \left\lceil \log_\alpha \frac{\varepsilon}{T_0} \right\rceil$. So the time complexity of the second step of FSA is $O(|P|^3)$. Taken together, the time complexity of FSA is $O(|P|^3)$.

As a result, the worst-case time complexity of ATCFS is $O(N \log N) + O(N^3) + O((N+1) \log (N+1)) + \cdots + O((N+X-1)^3) + O((N+X) \log (N+X)) + O((N+X)^3) = O(XN^3)$. In the above expression, the number of disjoint partitions is denoted by N. X denotes the total number of rounds that the loop consisting of FSA and TCA are executed. As is analyzed in IV.A, $X < M$ where M is the number of needed RNs that TCA runs for the first time. According to the system model of this work, the number of RNs needed to connect two HNs cannot be larger than $\left\lceil \frac{L}{d} \right\rceil + \left\lceil \frac{W}{r} \right\rceil + \left\lceil \frac{D}{R_C} \right\rceil$. So, $X < M \leq (N-1)(\left\lceil \frac{L}{d} \right\rceil + \left\lceil \frac{W}{r} \right\rceil + \left\lceil \frac{D}{R_C} \right\rceil)$. Therefore, the worst-case time complexity of ATCFS must be lower than $O(N^4)$.

According to [17], the time complexity of FeSTA-C is $O(N^4)$ in the worst case. Therefore, the proposed scheme ATCFS has lower time complexity than FeSTA-C. Moreover, ATCFS can give better performance in compare with FeSTA-C. The performance advantage of ATCFS can be shown by the simulation results of the next section.

V. SIMULATION EXPERIMENTS

In this section, simulation experiments are performed to investigate the performance of ATCFS. These experiments are completed with MatLab 2012b software. For each data in the result, 50 instances are simulated and their average is calculated and output. Section II has described the 3D UASN model of this work. The water body is a cuboid whose length $L = 5000$m, width $W = 5000$m and depth $D = 5000$m, respectively. Two key parameters are considered for each set of experiments. The first one is N, the number of disjoint partitions, which is set from 5 to 50. The other one is the communication radius of each node R_c, which is selected between 100m to 1000m. r and d, the distance between two adjacent rows and two adjacent columns of RN's candidate deployment point, are both set to $R_c/2$. Table 1 lists the detailed simulation parameters. Other two approaches are compared to demonstrate the performance advantage of the ATCFS scheme. Steinerized Minimum Spanning Tree (SMST) is a baseline algorithm that only uses TCA described in III.A which calculates the minimum spanning tree of the HNs and then steinerizes the edges. The other one is FeSTA-C which has been described in IV.B. In this work, the following three metrics are used for performance evaluation [17]:

Number of RNs: First of all, the total amount of RNs that are necessary to reestablish the network connectivity is reported. Clearly, the primary objective of our work is to repair the disjoint UASN with the least number of RNs.

Average hop count: Secondly, the average number of hops between each pair of HNs of the UASN is compared. Obviously, the smaller average hop count is superior, because it can reduce the network's data delivery delay.

Average node degree: Thirdly, the average amount of the adjacent nodes of each node is studied. The higher node degree is preferred, since it yields a stronger network connectivity and makes it easier to achieve the traffic load balancing.

Firstly, the number of RNs needed for the connectivity reestablishment is compared. We first consider the influence of the number of partitions (HNs). Fig.6a illustrates the simulation results. The x-coordinate represents the number of partitions (HNs). It is set from 5 to 50. The y-coordinate is the number of needed RNs. The communication radius is set to 500m for all nodes. As the figure shows, the employed RNs increases with the number of HNs for all three schemes that

TABLE 1. Parameter settings.

Symbol	Description	Value
L	length of the water body	5000m
W	Width of the water body	5000m
D	Depth of the water body	5000m
N	Number of the disjoint partitions (HNs)	$5 \sim 50$
R_c	Communication radius	100m $\sim$ 1000m
r	Distance between two adjacent rows of RN's candidate deployment point	$R_c/2$
d	Distance between two adjacent columns of RN's candidate deployment point	$R_c/2$
T_0	Original temperature of the SA (Simulated Annealing)	500m
K	Times iterated for each temperature of the SA	10000
α	Cooling rate of the SA	0.9
ε	Termination condition of the SA	0.01

indicates more RNs are necessary to reconnect more partitions. Obviously, the performance of the SMST is the lowest because it only use steinerized MST to connect the HNs with few mechanisms to reduce the usage of RNs. The FeSTA-C has better performance than that of SMST since it deploys relay nodes and connect partitions by finding local suboptimal solutions for 3 HNs (triangles). The ATCFS holds the best performance because it identifies the tetrahedrons and triangles from the network topology and then tries to utilize their Fermat points to reduce the usage of RNs for connectivity restoration. Compared with SMST and FeSTA-C, the ATCFS can save about 24.9% and 17.6% RNs on average.

Fig. 6b shows the relationship between the communication radius R_c and the number of RNs. The x-coordinate is the communication radius R_c. It is set from 100m to 1000m. The y-coordinate is the number of RNs needed for connectivity restoration. The number of HNs is set to 20. It is easy to understand that the number of RNs for establishing connected topology decreases along with the increase of the communication radius in all three schemes. The advantage of the ATCFS is significant. It can save about 29.4% RNs on average than SMST and about 24.1% than FeSTA-C.

In ATCFS, TCA reestablishes the network connectivity for the UASN and FSA tries to reduce the number of employed RNs. To demonstrate the effectiveness of FSA, we present the statistical result of the number of RN that can be saved by the proposed scheme compared with SMST (only the TCA is executed once). It is defined as the Optimization Gain (OG) of the ATCFS, i.e. $OG = M - |R|$. The results are shown in Fig. 6c and Fig. 6d. In Fig. 6c, the x-coordinate is the number of disjoint partitions which is set from 5 to 50. The y-coordinate is the percentage of the 4 stages of the Optimization Gain. The communication radius is set to 500m. The OG increases along with the increase of the number of HNs, because more HNs give ATCFS more choices in its FSA to select a more helpful Fermat point that can reduce the total number of needed RNs.

In Fig.6d, The x-coordinate is the communication radius R_c. It is set from 100m to 1000m. The y-coordinate

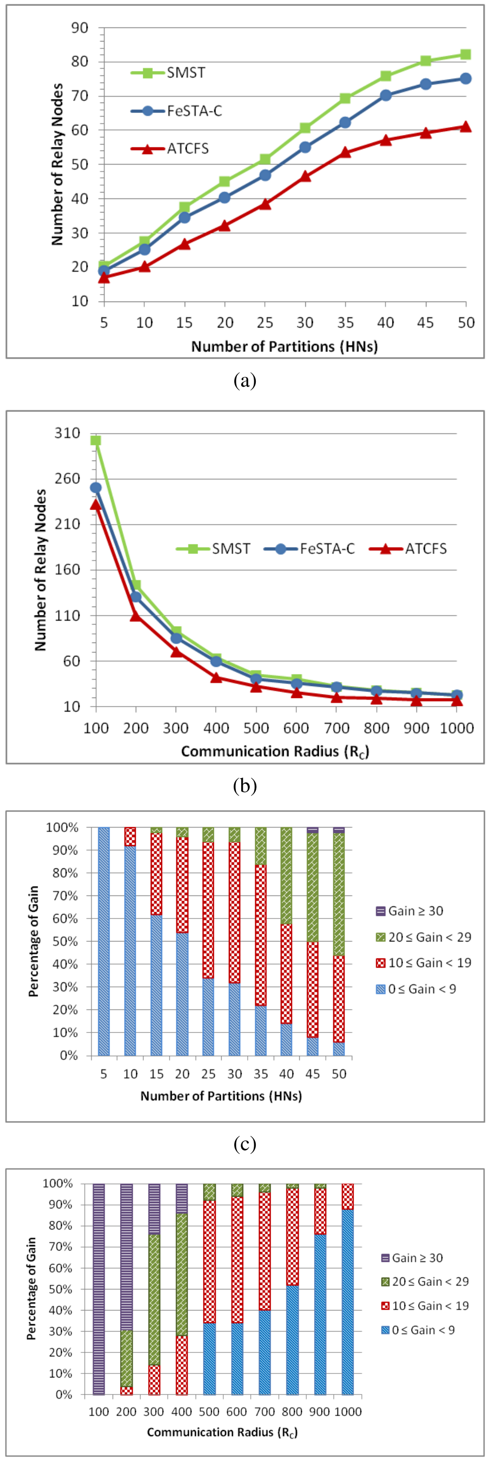
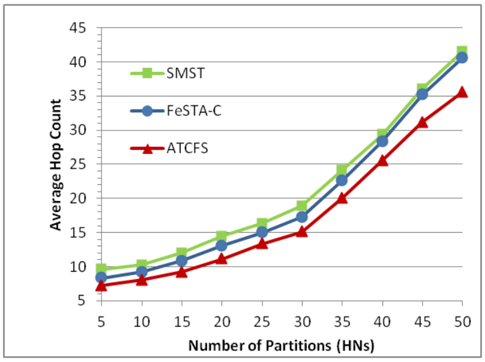
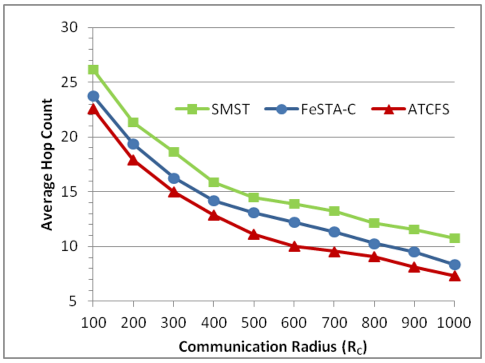


FIGURE 6. The performance comparison on the number of RNs.

is also the percentage of the 4 stages of the OG. The number of HNs is set to 20. As the communication radius increases, the Optimization Gain decreases. This is because the number of RNs that are employed to repair the network connectivity reduces along with the increase of the R_c. So the effect of FSA

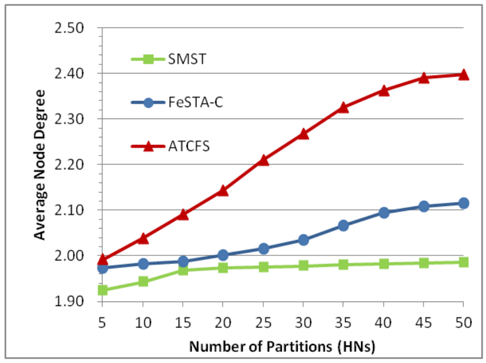


(a)

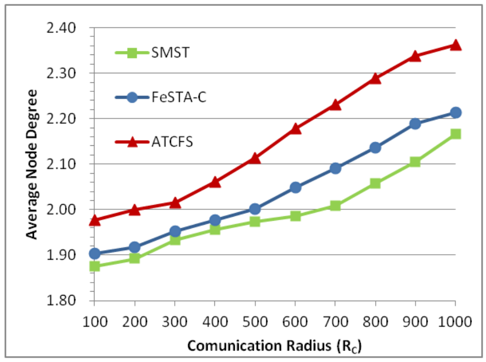


(b)

FIGURE 7. The performance comparison on the average hop count.



(a)



(b)

FIGURE 8. The performance comparison on the average node degree.

that tries to optimize the number of RNs is not significant for the long communication radius scenario.

Secondly, we consider another metric, average hop count. To get its value, the shortest path between every possible pair of HNs should be calculated first. In this work, it is calculated by Floyd–Warshall algorithm [25]. Then, the hop count of each path is computed and the average is reported. Fig. 7a illustrates the influence of the number of HNs on the average hop count. The x-coordinate represents the number of HNs that changes from 5 to 50. The y-coordinate shows the average hop count between each two HNs. The communication radius is set to 500m. The average hop count increases with the number of HNs for all three schemes. This is because the larger number of HNs is set, the more RNs are needed to reestablish the connectivity as is shown in Fig. 6a. So the average number of nodes between two HNs becomes larger, which naturally leads to more hops between them. The ATCFS has a better performance because it uses less relay nodes than the other two schemes in a tetrahedron or a triangle subset. So the local average hop count is smaller than the other two approaches. As a result, its overall average hop count is smaller accordingly. In our simulation, the ATCFS can reduce about 19.1% and 12.5% hops on average than SMST and FeSTA-C.

Fig. 7b shows the impact of the communication radius R_c on the average hop count. The x-coordinate denotes the communication radius R_c that is set from 100m to 1000m. The y-coordinate is the average hop count. Twenty disjoint partitions (HNs) are set in this group of experiments. As is illustrated in the graph, the average hop count decreases with the increase of R_c. The reason behind this phenomenon is that the larger R_c, the lesser number of RNs are needed to restore the network connectivity. Thus, the average hops between two HNs of the network are lesser accordingly. The other two schemes have lower performance than ATCFS. The average hop count of the ATCFS is about 76.6% of the SMST's and 88.4% of the FeSTA-C's.

Finally, the average node degree is compared for three schemes. Fig. 8a shows the relationship between the average node degree and the number of HNs. The communication radius R_c is set to 500m. The x-coordinate denotes the number of HNs. The y-coordinate represents the average node degree. For all three schemes, the average node degree is enhanced along with the increase of the number of HNs. Basically, the less RNs employed, the more other nodes that each node must connect with, i.e. have a larger node degree. SMST populates RNs at a distance of at most R_c apart based on the edges of the MST. As a result, most nodes have a degree of 2. FeSTA-C trys to identify the triangles and use their Fermat points to reduce the number of RNs and thus more nodes have a degree more than 2. So the FeSTA-C has a larger average node degree than the SMST. The proposed ATCFS considers the 2D and 3D case simultaneously. It identifies triangles and tetrahedrons from topology and then tries to further reduce the usage of RNs. As a result, the average node degree of

ATCFS is 12.8% higher than SMST and 8.9% higher than FeSTA-C.

Fig. 8b also demonstrates the performance advantage of ACTFS over SMST and FeSTA-C. In this figure, the x-coordinate represents the communication radius, which is set from 100m to 1000m, and the y-coordinate is the average node degree. 20 HNs are set in this group of experiments. It can be clearly observed that the average node degree of ACTFS is larger than that of SMST and FeSTA-C. It is about 8.1% and 5.5% higher than the above two schemes, respectively.

VI. CONCLUSION

In this work, we have investigated the Constrained UASN Connectivity Recovery (CUCR) problem that attempts to reestablish the connectivity for a partitioned 3D UASN with the least number of RNs. Based on the problem formulation, it is in the form of integer nonlinear programming which is NP-hard in general. To solve the problem effectively, we have presented a heuristic scheme, ATCFS, in this paper. ATCFS integrates two algorithms, Tree Construction Algorithm (TCA) and Fermat-point Selection Algorithm (FSA). It tries to reach the optimization goal by iteratively constructing the minimum spanning tree to guarantee the connectivity and selecting helpful Fermat point to reduce the usage of RNs. Three groups of simulation experiments have demonstrated the performance advantage of ATCFS. Compared with traditional schemes, it can save over 17% RNs. The average hop count can be reduced over 11% and the average node degree can be increased over 5%. To the best of our knowledge, it is a pioneering work to solve the CUCR problem. Possible future research directions include investigating the feasibility to use a hybrid mechanism, i.e. employing both mobile and stationary RNs, to repair the connectivity of the 3D UASNs more efficiently. Moreover, in this work, we employ a single head node to represent the whole partition in order to simplify the problem. In the future work, partitions could be represented by the comprised nodes. So, the connectivity recovery algorithm could be redesigned and the performance could be evaluated and compared with ATCFS.

REFERENCES

[1] L. Liu, M. Ma, C. Liu, and Y. Shu, "Optimal relay node placement and flow allocation in underwater acoustic sensor networks," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2141–2152, May 2017.

[2] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Comput. Netw.*, vol. 58, no. 1, pp. 254–283, 2014.

[3] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 643–656, May 2010.

[4] K. Vaidya and M. Younis, "Efficient failure recovery in wireless sensor networks through active, spare designation," in *Proc. 6th IEEE Int. Conf. Distrib. Comput. Sensor Syst. Workshops*, 2010, pp. 1–6.

[5] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1028–1042, Jun. 2019.

[6] F. Senel, K. Akkaya, and M. Younis, "An efficient mechanism for establishing connectivity in wireless sensor and actor networks," in *Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM)*, Nov. 2007, pp. 1129–1133.

[7] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," *IEEE Trans. Comput.*, vol. 59, no. 2, pp. 258–271, Feb. 2010.

[8] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in wireless sensor network," in *Proc. IEEE Global Commun. Conf.*, Dec. 2013, pp. 652–657.

[9] E. Uzun, F. Senel, K. Akkaya, and A. Yazici, "Distributed connectivity restoration in underwater acoustic sensor networks via depth adjustment," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 6357–6362.

[10] H. M. Almasaeid and A. E. Kamal, "Data delivery in fragmented wireless sensor networks using mobile agents," in *Proc. 10th ACM Symp. Modeling, Anal., Simulation Wireless Mobile Syst. (MSWIM)*, Chania, Greece, Oct. 2007, pp. 86–94.

[11] J. L. V. M. Stanislaus and M. Younis, "Delay-conscious federation of multiple wireless sensor network segments using mobile relays," in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2012, pp. 1–5.

[12] H. M. Almasaeid and A. E. Kamal, "Modeling mobility-assisted data collection in wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM)*, Nov./Dec. 2008, pp. 1–5.

[13] W. K. G. Seah, H.-X. Tan, Z. Liu, and M. H. Ang, "Multiple-uuv approach for enhancing connectivity in underwater ad-hoc sensor networks," in *Proc. OCEANS*, vol. 3, Sep. 2005, pp. 2263–2268.

[14] G.-H. Lin and G. Xue, "Steiner tree problem with minimum number of Steiner points and bounded edge-length," *Inf. Process. Lett.*, vol. 69, no. 2, pp. 53–57, 1999.

[15] F. Senel, M. F. Younis, and K. Akkaya, "Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1835–1848, May 2011.

[16] F. Senel and M. Younis, "Optimized relay node placement for establishing connectivity in sensor networks," in *Proc. Global Commun. Conf.*, 2012, pp. 512–517.

[17] F. Senel and N. Younis, "Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation," *Comput. Commun.*, vol. 34, no. 16, pp. 1932–1941, 2011.

[18] J. Jin, J. Zhang, F. Shao, and T. Cui, "An unmanned surface vehicle for ocean environment monitoring and its oceanic application," *Coastal Eng.*, vol. 34, no. 3, pp. 87–92, Sep. 2015.

[19] C. Detweiler, M. Doniec, I. Vasilescu, and D. Rus, "Autonomous depth adjustment for underwater sensor networks: Design and applications," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 16–24, Feb. 2012.

[20] J. Z. Jia, J. Liu, and S. S. Song, *Judgement Criterion of Graph Connectivity Based on Adjacent Matrix*. Liaoning Sheng, China: Liaoning Technical Univ., 2003.

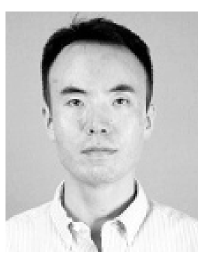
[21] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: W. H. Freeman, 1979.

[22] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2579–2590, Sep. 2005.

[23] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, Mar. 1985.

[24] P. Bicudo and M. Cardoso, "Iterative method to compute the Fermat points and Fermat distances of multiquarks," *Phys. Lett. B*, vol. 674, no. 2, pp. 98–102, 2009.

[25] H.-S. Kim, J.-H. Lee, and Y.-S. Jeong, "Method for finding shortest path to destination in traffic network using dijkstra algorithm or floyd-warshall algorithm," U.S. Patent 6 564 145 B2, May 13, 2002.



LINGFENG LIU received the B.S., M.S., and Ph.D. degrees from the College of Intelligence and Computing, Tianjin University, Tianjin, China, in 2000, 2005, and 2019, respectively. He is currently a Senior Engineer with the National Ocean Technology Center, Tianjin. His research interests include underwater acoustic sensor networks, network security, and long distance wireless communication.

MAODE MA received the Ph.D. degree from the Department of Computer Science, The Hong Kong University of Science and Technology, in 1999. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include network security and wireless networking. He has over 300 international academic publications including more than 140 journal papers and over 170 conference papers. He is a Fellow of the IET, and a Senior Member of the IEEE Communication Society and the IEEE Education Society. He is the Chair of the IEEE Education Society, Singapore Chapter. He was an IEEE Communication Society Distinguished Lecturer, from 2013 to 2016. He currently serves as the Editor-in-Chief for the *International Journal of Computer and Communication Engineering* and the *International Journal of Electronic Transport*. He also serves as a Senior Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and an Associate Editor for the *International Journal of Security and Communication Networks*, the *International Journal of Wireless Communications and Mobile Computing*, and the *International Journal of Communication Systems*.

CHUNFENG LIU received the B.S. degree in mathematics from Shandong Normal University, China, in 2000, and the M.S. and Ph.D. degrees in mathematics and computer science from Tianjin University, China, in 2003 and 2009, respectively. From 2003, she was an Assistant Lecturer, a Lecturer, and an Associate Professor with the School of Computer Science, Tianjin University. She was a Visiting Scholar with the School of Information Technology and Engineering, University of Ottawa, Canada, from 2007 to 2008. Her research interests include wireless networks, traffic modeling, protocol design, performance evaluation and QoS, and truthful computing in non-cooperative wireless networks.

WENYU QU received the B.S. and M.S. degrees from the Dalian University of Technology, China, in 1994 and 1997, respectively, and the Ph.D. degree from the Japan Advanced Institute of Science and Technology, Japan, in 2006. She was an Assistant Professor with the Dalian University of Technology, China, from 1997 to 2003. She is currently a Professor with the School of Software, Tianjin University. She was a Professor with Dalian Maritime University, China, from 2007 to 2015. She has published more than 80 technical papers in international journals and conferences. Her research interests include cloud computing, computer networks, and information retrieval. She is on the committee board for a couple of international conferences.

GANG ZHANG received the B.S., M.S., and Ph.D. degrees in computer science and technology from Tianjin University. Since 1983, he has been an Engineer, an Associate Professor, and a Professor with the School of Computer Science, Tianjin University. He has over 100 international academic publications. His current interests include computer communication networks, wireless sensor networks, and real-time systems.

YANTAI SHU received the B.S., M.S. degrees in electronics engineering from Tianjin University. He is currently a Professor of computer science with Tianjin University, China, where he was the Vice President, from 1993 to 1997. He has published over 120 papers and contributed to one book. His current research interest include computer communication networks, wireless networks, real-time systems, modeling, and simulation. From 1974 to 1991, he was employed in research positions with the Institute of Plasma Physics, Academia Sinica. He is a member of the ACM.