A Straight Skeleton Based Connectivity Restoration Strategy in the Presence of Obstacles for WSNs

Xiaoding Wang 1,2, Li Xu 1,* and Shuming Zhou 1

1 School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China; wangdin1982@fjnu.edu.cn (X.W.); zhoushuming@fjnu.edu.cn (S.Z.)
2 School of Online and Continuing Education, Fujian University of Technology, Fuzhou 350118, China
* Correspondence: xuli@fjnu.edu.cn

Received: 2 September 2017; Accepted: 3 October 2017; Published: 10 October 2017

Abstract: Connectivity has significance in both of data collection and aggregation for Wireless Sensor Networks (WSNs). Once the connectivity is lost, relay nodes are deployed to build a Steiner Minimal Tree (SMT) such that the inter-component connection is reestablished. In recent years, there has been a growing interest in connectivity restoration problems. In previous works, the deployment area of a WSN is assumed to be flat without obstacles. However, such an assumption is not realistic. In addition, most of the existing strategies chose the representative of each component, which serves as the starting point of relay node deployment during the connectivity restoration, either in a random way or in the shortest-distance based manner. In fact, both ways of representative selection could potentially increase the length of the SMT such that more relay nodes are required. In this paper, a novel connectivity restoration strategy is proposed—Obstacle–Avoid connectivity restoration strategy based on Straight Skeletons (OASS), which employs both the polygon based representative selection with the presence of obstacles and the straight skeleton based SMT establishment. The OASS is proved to be a 3-opt approximation algorithm with the complexity of $O(n \log n)$, and the approximation ratio can reduce to $\frac{3\sqrt{3}}{2}$ while it satisfies a certain condition. The theoretical analysis and simulations show that the performance of the OASS is better than other strategies in terms of the relay count and the quality of the established topology (i.e., distances between components, delivery latency and balanced traffic load) as well.

Keywords: connectivity restoration; representatives; obstacles; straight skeleton; WSNs

1. Introduction

Wireless sensor networks (WSNs) are well-known by their significant advantages in monitoring, such as battlefield surveillance, environment monitoring and biological detection. To achieve events surveillance, a large amount of sensors are intentionally placed to monitor a certain geographic area. All data collected are forwarded toward a base station along at least a multi-hop wireless path. In this case, the reachability between any pair of sensors implies the importance of the network connectivity. Due to such special architecture, WSNs are more vulnerable to human sabotages and natural disasters that can partition a WSN into disjointed components. Once the connectivity is lost, the data will be no longer sent back to the center that will cause severe consequences.

Due to the significance of network connectivity as mentioned above, the problem of connectivity restoration has been receiving increasing attention in recent years. All known solutions consider how to reestablish the connectivity through building a Steiner Minimal Tree (SMT) among all components with the minimum cost, which includes relay device consumption and energy consumption [1]. According to the classification of the cost, there exist two major types of restoration strategies. One utilizes the least number of relay devices (i.e., Relay Nodes (RNs) and Mobile Data Collectors (MDCs)) to...
federate disjointed components [2–18]. However, most strategies of this type choose the representative either in a random way or in the shortest-distance based manner. In fact, both ways of representative selection could potentially increase the length of the SMT such that more RNs are required. In addition, they assume the deployment area of a WSN is flat without obstacles. Note that, if there are no obstacles between two nodes, then a signal originated from one node could be received by the other one even with a slight signal degradation caused by a long distance transmission. On the other hand, another type of connectivity restoration strategy pursues the minimum energy consumption influenced by a variety of realistic terrains (e.g., mountain, river, forest, swamp, etc.) [19–23]. In this paper, the connectivity restoration strategy with minimum RN consumption is developed. In addition, we consider the mountain as the only available terrain factor. If there exist buildings and other structures along with mountains that are big enough and hard to penetrate, then all of these objects are considered as impenetrable obstacles.

**Our Contribution**

This paper presents an Obstacle–Avoid connectivity restoration strategy based on Straight Skeletons, namely OASS, which consists of an adapted minimum spanning tree construction algorithm $Ad$-Prim, an obstacle–avoid algorithm OA and a straight skeleton based SMT construction algorithm SSIN. The details of our contributions are listed as follows:

1. We modify the well-known Minimum Spanning Tree (MST) construction algorithm Prim to a circle-tolerant one, namely $Ad$-Prim, by choosing multiple nodes on the perimeter of each component as corresponding representatives such that the MST is constructed with the rule that circles formed by nodes on the same components should be neglected. Compared with other strategies that choose only one node for each component as its representative, the spanning tree established by $Ad$-Prim is much shorter.

2. We develop an obstacle–avoid algorithm, OA, to deal with the case that there exist obstacles lying on the line-segment between each pair of nodes. Instead of choosing a detour going along the perimeter of an obstacle, the candidate point on the perimeter of the obstacle is carefully chosen such that a shortest path around the obstacle consists of at least two line-segments tangent at such point. In addition, compared with other strategies disregarding the presence of obstacles, OA can be applied to more realistic scenarios.

3. We devise a straight skeleton based algorithm, SSIN, to build an SMT such that the shortest inter-component connection is achieved. Each straight skeleton will be placed within the potential convex hull of MST as the deployment route for RNs. Furthermore, if multiple line-segments between pairs of nodes have obstacles, then the SSIN is an option to avoid obstacles.

4. We prove that OASS is a $3$-opt approximation algorithm with the complexity of $O(n \log n)$, and the approximation ratio can reduce to $\frac{3\sqrt{3}}{2}$ while the network topology can be decomposed into a series of convex hulls. The theoretical analysis and simulations show that the performance of OASS is better than other strategies, especially a series of which deploy RNs toward the center of the deployment area, not only in the number of RNs required but the quality of the established topology (i.e., distances between components, delivery latency and balanced traffic load) as well.

The rest of the paper is organized as follows. Related work is covered in Section 2. The notions, terminologies and the problem description are introduced in Section 3. The strategy is elaborated in Section 4. Section 5 gives the theoretical analysis on approximation radio and complexity of OASS. In addition, the validation results are presented in this section as well. We conclude this paper in Section 6.

2. Related Works

According to the classification of the cost, there exist two major types of restoration strategies [1]. One utilizes the least number of relay devices (i.e., RNs and MDCs) to federate disconnected
components [2–18], while the other one pursues the minimum energy consumption influenced by realistic terrains [19–23].

Due to the variety of relay devices, strategies based on relay device consumption can be subdivided into two categories. One aims to deploy the minimum number of RNs to reestablish connectivity as follows, the performance of which are always measured by the approximation ratio [2–10]. In [2], Chen et al. first prove the MST based approach is 3-opt and then they propose a 4-star based 3-opt algorithm. Similarly, Cheng et al. [5] employ 3-star to connect disjointed components and then design a 3-hypergraph based algorithm, the approximation ratios of which are 3 and 2.5, respectively. It is worth mentioning that, although 2.5 is the minimum one among all approximation ratios, the complexity of the 3-hypergraph based algorithm is $O(n^3q_p^2)$, where $q_p = \max\{q_{a,b,c}|a, b, c \subset P\}$, $q_{a,b,c}$ denotes the number of RNs required to connect nodes $a, b, c$ and $P$ represents the set of disconnected nodes. In [4], Lloyd et al. give an improved MST based algorithm with a approximation ratio 7. Tang et al. [3] map the deployment area into a series of regular cells, by partition the set of components into a number of convex hulls and build an MST for inter-partition placing a cluster head node within each cell in charge with inter-cell communications the connectivity is restored and the approximation ratio of which is 4.5. Efrat et al. [10], Yang et al. [7] and Misra et al. [6,8] give weight values to different types of connections (i.e., node to node, node to relay and relay to relay) to build a weighted complete graph. Then, they employ the construction algorithm of the minimum weight Steiner tree [24] to restore the connectivity in different scenarios, the approximation ratio of this algorithm for each scenario is 3.11, 6.43, 6.2 and 12.4, respectively, and the complexities for all scenarios are at least $O(|S|^k|V − S|^{k−2} + k|S|^{2k+1} \log |S|) < O(n^{2k−2} + n^{2k+1} \log n)$, where $S \subset V$ denotes the set of nodes required to be connected, $|V| = n$ and $k \geq 3$. Recently, Wang et al. [9] proposes a 3-opt algorithm based on the center of mass and 3-star. There are some efficient algorithms without giving approximation ratios [11–18]. Lee et al. propose efficient algorithms in [13,15], respectively, to deploy RNs toward the center of the deployment area. In [14], Ranga et al. first construct a distance function, then based on the zero gradient point of which the relay node deployment route is planned. Joshi et al. [12] employ the straight skeleton as the relay node deployment route. In [11], Senel et al. design two efficient algorithms. One is an optimal triangle construction algorithm based on the MST, while the other one is based on Delaunay triangulation, both of which employ Fermat points to achieve the connectivity restoration. In [16], Uwitonze et al. employ space network coding, Delaunay triangulation, non-uniform partitioning techniques and linear programming to design a relay node deployment route. On the other hand, another type of strategy based on relay devices’ consumption considers the situation that MDCs serve as a channel to offer intermitted connection between disjoint components while only insufficient RNs are available [17,18]. However, these types of strategies always evaluate the performances through simulations. Both Joshi et al. [18] and Abbas et al. [17] partition the set of components into a number of convex hulls and build an MST for inter-partition connection. Then, they employ convex hulls as routes for MDCs, which are shortened in [18] through an angle bisector based optimization approach, in order to provide intermitted connections. At last, the connectivity is restored by deploying RNs along the MST. It is worth mentioning that OASS solves both of the representative selection problem and the obstacle–avoid problem; although neither of them are taken into consideration by all of these works above. Furthermore, although the optimal approximation ratio of OASS is $\frac{3\sqrt{3}}{2}$, which is a little higher than 2.5, the complexity of OASS is only $O(n \log n)$. The comparison of some contemporary heuristic algorithms for connectivity restoration in WSNs through RNs placement is shown in Table 1.

There are many terrain based connectivity restoration strategies. In [20], Senturk et al. quantifies the influence of realistic terrains through a grid based mapping. Then, they design the ReBAT that considers realistic terrains such that the least cost paths for connectivity restoration is built. Similarly, Wang et al. [23] devise a hybrid strategy to achieve the 2-connectivity restoration based on realistic terrains. Zhou et al. [19] propose an extended Rapidly exploring Random Tree (RRT) based algorithm to find a least cost path to avoid obstacles and federate components. Truong et al. [21] design a family of algorithms that consider the impact of obstacles on mobility and communication to restore
the connectivity with a minimum number of RNs and meanwhile minimizes the mobility cost of agents. In [22], Mi et al. investigate how to avoid convex obstacles and inter-sensor collisions during connectivity restoration. Unlike these works, only one terrain factor, the mountain, is considered as an impenetrable obstacle by OASS.

Table 1. Comparison of some contemporary heuristic algorithms for connectivity restoration in Wireless Sensor Networks (WSNs) through Relay Nodes (RNs) placement.

| Authors          | Approximation Ratio | Complexity                        |
|------------------|---------------------|-----------------------------------|
| Misra et al. [8] | 12.4                | $O(n^{2k-2} + n^{2k+1} \log n)$  |
| Lloyd et al. [4] | 7                   | $O(n \log n)$                     |
| Yang et al. [7]  | 6.43                | $O(n^{2k-2} + n^{2k+1} \log n)$  |
| Misra et al. [6] | 6.2                 | $O(n^{2k-2} + n^{2k+1} \log n)$  |
| Tang et al. [3]  | 4.5                 | Not available                      |
| Efrat et al. [10]| 3.11                | $O(n^{2k-2} + n^{2k+1} \log n)$  |
| Cheng et al. [5] | 3                   | $O(n^4)$                           |
| Chen et al. [2]  | 3                   | $O(n^3)$                           |
| Wang et al. [9]  | 3                   | $O(n^2)$                           |
| OASS (this paper)| 3                   | $O(n \log n)$                     |
| OASS satisfying a certain condition (this paper) | $\frac{3\sqrt{3}}{2}$ | $O(n \log n)$ |
| Cheng et al. [5] | 2.5                 | $O(n^3q^2p)$                       |

3. Preliminary

3.1. System Model

The disconnected network is mapped to a graph $G(V, E)$. Each node $s_i \in V$ represents a sensor with a communication rage $r$, while $s_is_j \in E$ denotes the communication link between a pair of sensors $s_i$ and $s_j$. Since the network is partitioned into a number of disjoint components $C_i$s, all $s_is \in C_i$ remain connected. Some notations used throughout this paper are given first, and the important symbols with their definitions are collected in Table 2.

Table 2. Notions.

| Symbols | Descriptions |
|---------|--------------|
| $P_{C_i}$ | the polygon consists of nodes on the perimeter of $C_i$ |
| $CH_O$ | the convex hull of an obstacle O |
| $p_{s_i,s_j}$ | a path from $s_i$ to $s_j$ |
| $P_{s_i,s_j}^O$ | a shortest path from $s_i$ to $s_j$ around the obstacle O |
| $d_{s_i,s_j}$ | the Euclidean distance of an edge $s_is_j$ |
| $L_G$ | the length of the graph $G$ in Euclidean space, that is $L_G = \sum_{e \in G} d_e$ |
| $N_A$ | RNs consumption function of the approach $A$ |

Definition 1. [25] A Euler Closed Trail, abbreviated as ect, is a closed trail that visits every edge of graph $G$ exactly once.
Definition 2. A potential Convex Hull, abbreviated as pch, is a path $s_1s_2 \ldots s_k$ such that $s_1s_k \bigcup_{i=1}^{k-1} s_is_{i+1}$ is a convex hull of the set of nodes $\{s_i|1 \leq i \leq k\}$.

Definition 3. [26] The straight skeleton of a polygon is defined by a continuous shrinking process in which the edges of the polygon are moved inwards parallel to themselves at a constant speed. As the edges move in this way, the vertices where pairs of edges meet also move, at speeds that depend on the angle of the vertex. If one of these moving vertices collides with a nonadjacent edge, the polygon is split in two by the collision, and the process continues in each part. The straight skeleton is the set of curves traced out by the moving vertices in this process.

The shortest SMT considered as the shortest inter-component connection has the following property [27]. Note that each terminal denotes to a representative of a $C_i$, while each Steiner point represents a rendezvous point at which three edges are adjacent to each other.

Property 1.

1. Each terminal is a leave and each Steiner point connects to exactly three terminals.
2. The angle between each pair of adjacent edges at a Steiner point is exactly $120^\circ$.
3. There are totally $n-2$ Steiner points for $n$ terminals.

3.2. Problem Statement

This paper is dedicated to the connectivity restoration problem, which is formally stated as follows: Given a graph $G$ with $n$ disjoint components $C_i$s that consist of sensors with a transmission range $r$ and obstacles $O$s, the goal is to provide an efficient solution that ensures that $n$ components will be 1-connected by deploying RNs at an interval $r$ as few as possible. That is,

$$\min N_A \quad \text{st.} \quad \begin{align*}
1. & \exists O_s, \\
2. & \forall s_i \in C_i, s_j \in C_j, \\
& \exists P_{s_is_j} \neq \emptyset,
\end{align*} \quad (1)$$

where $N_A$ denotes the number of RNs required by employing approach $A$.

4. The OASS Approach

As a three-phase strategy, OASS is devised to cope with the connectivity restoration problem with the presence of obstacles. In phase one, an adapted minimum spanning tree construction algorithm $Ad$-$Prim$ is designed to deal with the improper representative selection problem. In phase two, an obstacle–avoid algorithm $OA$ is developed to build the shortest paths around obstacles. In the last phase, a straight skeleton based algorithm $SSIN$ is conceived to achieve the shortest inter-component connection and obstacles avoidace.

4.1. Adapted Minimum Spanning Tree Construction Algorithm (Ad-$Prim$)

Before we employ the straight skeleton based RNs deployment to achieve the establishment of inter-component connection, the candidate positions for the placement of straight skeletons, which are potential convex hulls, should be properly chosen along the MST of all disjointed components. The reason for finding the shortest MST is that, if a longer MST is built, then the area of each potential convex hull could be enlarged such that the length of each corresponding straight skeleton is increased. No double that a longer MST will eventually result in more RNs required. In fact, although the MST construction algorithm $Prim$ [24] helps to locate the tree with minimum length, without properly chosen representatives, the generated MST is not the shortest one. Therefore, the selection for eligible representatives are crucial during the MST construction.
In previous works, there exist two methods of representative selection. The first one [2,4–8,10–12,17,18] is randomly choosing a node within a component to represent it, while the second one [9,13,15] chooses a number of nodes, one from each component, closest to each other as corresponding representatives. If the Prim is used to find an MST, then it is obvious that the latter one will result in a shorter MST. However, both methods can hardly build the shortest MST, since choosing multiple nodes as representatives of a $C_i$ can potentially shorten inter-component distances. Due to the reasons explained above, we adapt the Prim to a circle-tolerant algorithm, namely Ad-Prim. The Ad-Prim will generate the shortest MST through the following steps.

1. The perimeter of each component $C_i$ is modelled to a polygon $P_{C_i}$. All nodes on the $P_i$ are the representatives of $C_i$.

2. All representatives and the shortest paths between them around obstacles found by OA are prepared for the MST construction in the next step.

3. The Prim is employed to build an MST among all representatives with one rule that each circle formed by nodes on the same component should be neglected.

According to the description above, the pseudo code of Ad-Prim is given as Algorithm 1.

**Algorithm 1** Adapted Minimum Spanning Tree Construction Algorithm Ad-Prim.

**Input:** all nodes on the perimeter of all components and a set of shortest paths $\{P_{s_is_j} | 1 \leq i,j \leq n\}$

**Output:** an MST

- choose all nodes on $P_{C_i}$ as representatives of $C_i$
- employ Prim to construct an MST among all representatives with one rule that each circle formed by nodes on the same component should be neglected
- return an MST

Although the proper selection of multiple representatives for each component helps to build the shortest MST, there is a chance that an edge $s_is_j \in$ MST may intersect with an obstacle. This requires an obstacle–avoid algorithm.

### 4.2. Obstacle–Avoid Algorithm (OA)

In a real scenario, there are always some obstacles on the deployment area. If an obstacle lies right on the middle of two disjointed nodes, then the line-segment between such pair of nodes for RNs deployment is not an option. In this case, although a detour going along the perimeter of the obstacle could be a solution, yet it is not the best choice. In fact, the shortest detour between a pair of nodes with an obstacle sitting right on the middle consists of a series of paths with at least two tangents involved. To be more specific, an obstacle is modelled to a polygon first. Then, the convex hull based on the polygon of the obstacle is constructed. At last, by carefully choosing the line-segments between two nodes and the points on the convex hull, the shortest path around the obstacle will be found. Note that there are two cases that should be considered. One is the shortest path composed of only two tangents between a specific point on the convex hull and two nodes, respectively (see Figure 1a). The other one not only includes all paths in the previous case but involves more paths between multiple points on the convex hull (see Figure 1b) as well. Furthermore, since an MST intersecting with obstacles is built, it is much easier to deal with each edge $s_is_j \in$ MST intersecting with an obstacle in order to construct the $P_{s_is_j}^O$. 
Figure 1. The shortest paths around the obstacle w.r.t (a) when there is only one point; (b) when there are more points sitting in the middle.

The OA will find shortest paths through the following steps.

1. For two nodes $s_1, s_2$ and an obstacle $O$, find two points $x, y$ on $CH_O$, such that there exist two tangents $s_1x$ and $s_2y$.

2. If $x = y$, then $s_1xs_2$ is the $P^O_{s_1s_2}$. Otherwise, let the set of points $S = \{x_i|0 \leq i \leq k - 1\}$ denote the least number of points from $x$ to $y$ on $CH_O$ where $x_0 = x$ and $x_k = y$. Then, $P^O_{s_1s_2} = s_1x \bigcup y_2 \bigcup_{i=0}^{k-1} x_ix_{i+1}$.

According to the description above, the pseudo code of OA is given as Algorithm 2.

Algorithm 2 Obstacle–Avoid Algorithm OA.

Input: the set of edges $\{s_is_j|i \neq j, 1 \leq i,j \leq n\}$

Output: each a set of shortest paths around obstacles $\{P^O_{s_is_j}|1 \leq i,j \leq n\}$

for each pair of nodes $s_i$ and $s_j$ do

if there is an obstacle $O$ on the line-segment $s_is_j$ then

find two points $x$ and $y$ on $CH_O$ such that there exist two tangents $s_1x$ and $s_2y$

using the least number of points $\{x_i|x_0 = x, x_k = y, 0 \leq i \leq k - 1\}$ on $CH_O$ to construct $P^O_{s_1s_2} = s_1x \bigcup y_2 \bigcup_{i=0}^{k-1} x_ix_{i+1}$

end if

end for

return a set of shortest paths around obstacles $\{P^O_{s_is_j}|1 \leq i,j \leq n\}$

4.3. Straight Skeleton Based SMT Construction Algorithm (SSIC)

Although deploying RNs along the MST can accomplish the connectivity restoration, yet it consumes a large number of RNs. In order to reduce the consumption of RNs, a better way is required that can offer a connection structure simpler and shorter than the MST. In fact, a straight skeleton is a tree such that all terminals are leaves and each pair of adjacent edges at a Steiner node has an angle of exactly $120^\circ$ [28]. Such characteristics satisfy property 1. Therefore, the straight skeleton can reduce the length of the inter-component connection, which implies the RNs required are less than that of the MST. Before we apply the straight skeleton to connectivity restoration, we need to know where a straight skeleton should be placed. More importantly, using a convex hull to generate a straight skeleton only costs $O(n \log n)$ [28]. There is no doubt that each convex hull should be found to place a straight skeleton. In fact, when the MST is built, every convex hull will be located along the MST. Furthermore, if there exist obstacles between pairs of nodes, then the straight skeleton is an option to avoid obstacles (see Figure 2).
Next, we will show how SSIC achieves the construction of the straight skeleton based inter-component connection through the following steps.

1. A leave node $s_1$ on the MST is randomly chosen as the starting node to draw a Euler closed trail $ect = s_1s_2 \ldots s_ns_1$ of the MST, where $n$ denotes the number of representatives.

2. Starting from $s_1$, we try to locate every potential convex hull $pch_i$ along $ect$. If there exists an edge $s_is_{i+1} \in pch_i$ such that $s_is_{i+1} = s_ip_i \cup p_{i+1} \cup p_{i+k-1} \cup p_{i+k} \cup s_{i+1}$, then let $pch_i = p_ip_{i+1} \cup pch_i$. It is worth mentioning that, if there is a $pch_i = s_is_{i+1} \ldots s_{i+k}$ such that the set $S = \{s_{i+j} | 1 \leq j \leq k\}$ of nodes are on the $C_i$, then let $pch_i = s_is_{i+1} \ldots s_{i+j}$ and choose $s_{i+k}$ to proceed. Otherwise, we start from $s_{i+k+1}$ and repeat the $pch$ localization process until all representatives are looped over.

3. For each $pch_i$, if there are no obstacles intersecting with the straight skeleton within the $pch_i$, then this straight skeleton serves as the deployment route for RNs. Otherwise, according to the number
$k$ of obstacles, we split the $pch_i$ into $k + 1$ parts such that $pch_i = \bigcup_{t=1}^{k+1} pch^t_i$ and there exists at least one part, say $pch^t_i$, the straight skeleton of which is not intersecting with obstacles. Then, we place RNs along the straight skeleton inside the $pch^t_i$. If there exists a $k$ such that $k \neq j$, then each edge $e \in pch^t_i$ is a relay node deployment route. It is worth mentioning that all straight skeletons can be built at $pchs$ once and for all.

According to the description above, the pseudo code of SSIN is given as Algorithm 3.

**Algorithm 3** Straight Skeleton based SMT Construction Algorithm (SSIC).

**Input:** an MST that consists of $n$ representatives

**Output:** a straight skeleton based tree

randomly choose a leave node $s_1 \in MST$ to draw an euler closed trail $ect = s_1s_2\ldots s_n s_1$

for all $s_i \in ect$ do

if there exists an edge $s_is_{i+1} \in pchi$ such that $s_is_{i+1} = s_ip_1 p_i p_{i+1} \ldots p_{i+k-1} p_{i+k} \bigcup p_i s_{i+k}$

then $pchi = \bigcup p_i p_{i+1} \bigcup pchi_i$

end if

if there is a $pchi_i = s_is_{i+1}\ldots s_{i+k}$ such that $S = \{s_{i+j} | s_{i+i} \in C_i, 1 \leq j \leq k\}$ then $pchi_i = s_is_{i+1}\ldots s_{i+k}$ and $s_i = s_{i+k}$

else $s_i = s_{i+k+1}$

end if

end for

for all $pchi$s do

if there is no obstacles intersecting with the straight skeleton within the $pchi$ then

the straight skeleton of the $pchi$ is a relay node deployment route

else split the $pchi$ into $k + 1$ parts such that $pchi = \bigcup_{t=1}^{k+1} pchi^t_i$

if there exists a $pchi^t_i$, the straight skeleton of which is not intersecting with obstacles then

the straight skeleton will be the route for placing RNs

end if

for other $pchi^t_i$s, $k \neq j$, do each edge $e \in pchi^t_i$ is a relay node deployment route

end for

end if

end for

return a straight skeleton based tree

Next, we give an example of how to restore the connectivity of a disconnected network. As shown in step 1 of Figure 3, there are 19 sensor nodes that consist of 10 components. Since these 19 sensor nodes are on the perimeter of all 10 components, all sensor nodes serve as representatives. Obviously, there is an obstacle on the line-segment between $s_{16}$ and $s_{17}$. In step 2 of Figure 3, the MST of 12 representatives are built using $Ad$-$prim$ with three circles $s_{14}s_{15}s_{16}s_{14}, s_{25}s_{24}s_{22}$ and $s_{33}s_{32}s_{31}s_{33}$ neglected. Note that sensor nodes $s_3, s_4, s_5, s_9, s_{10}, s_{11}$ and $s_{12}$ are no longer useful in the following process; therefore, we ignore them for simplicity. As shown in step 3 of Figure 3, since there is only one path $s_{16}s_{17}$ intersecting with the obstacle, we use $OA$ to locate the $p_{316}^{s_{17}}$ which consists of two tangents $s_{16}x$ and $xs_{17}$. In step 4 of Figure 3, SSIC first randomly chooses a leave node, say $s_1$, as the beginning to draw the euler closed trail $ect = s_1s_2s_3s_4s_5s_{16}s_{17}s_{18}s_{19}s_{18}s_{17}s_{16}s_{15}s_{14}s_{13}s_{12}s_{11}s_{10}s_9s_8s_7s_6s_5s_4s_3s_2s_1$s. As shown in step 5 of Figure 3, three potential convex hulls $s_{15}s_{16}s_{17}s_{18}s_{19}$ and $s_{14}s_{13}s_{12}s_{11}s_{10}s_9s_8s_7s_6s_5s_4s_3s_2s_1$s are found along the $ect$. Note that, although $s_{15}s_{16}s_{17}s_{18}s_{19}$ and $s_{14}s_{13}s_{12}s_{11}s_{10}s_9s_8s_7s_6s_5s_4s_3s_2s_1$s are convex hulls respectively, edges $s_{15}s_{16}, s_{16}s_{17}, s_{11}s_{10}, s_{10}s_9$s and $s_{13}s_{12}s_{11}s_{10}s_9$s belong to a component, for which such three edges should be removed form three $pchs$. At last three straight skeletons are placed within three $pchs$ since there are no obstacles intersecting with straight skeletons as shown in step 6 of Figure 3. Eventually, RNs are deployed along the straight skeleton based tree to restore the connectivity.
5. Performance Analysis

5.1. Theoretical Analysis

In this section, we give the theoretical analysis on OASS about the approximation ratio, the advantage over a series of connectivity restoration strategies in terms of RNs required and the complexity.

We call all approaches $A_{rc}$, if they either randomly choose a node from each component as the corresponding representative or select a number of nodes, each from a component, closest to each other as representatives [2,4–13,15,17,18]. Let $L_{MST_{rc}}$ denote the length of an MST built by $A_{rc}$. For a graph $G$, we have the following theorem holding true.

Theorem 1. For a graph $G$, if $t$ is a tree constructed by Ad-Prim, then we have following two results:

1. $t$ is the MST of $G$.
2. $L_t \leq L_{MST_{rc}}$.

Proof. First, we prove result 1 by contradiction.

We are going to take Figure 4 as an example to complete the proof. Supposing that $t$ is not the MST of graph $G$ with $s_1s_2, s_1s_3 \in t$, let $t^*$ denote the MST of graph $G$ with $s_1s_2, s_2s_3 \in t^*$. It is obvious that $t \cup t^*$ has at least a circle $s_1s_2s_3$. Next, we try to distinguish three cases to prove the first result.

Case 1: $L_{s_2s_3} < L_{s_1s_3}$.
If $L_{s_2s_3} < L_{s_1s_3}$, then $s_2s_3$ should be chosen first, we can get a shorter $t = t \setminus s_1s_3 \cup s_2s_3$, this contradicts the structure of $t$.

Case 2: $L_{s_2s_3} = L_{s_1s_3}$.
If $L_{s_2s_3} = L_{s_1s_3}$, then we have $t = t^*$.

Case 3: $L_{s_2s_3} > L_{s_1s_3}$.
If $L_{s_2s_3} > L_{s_1s_3}$, then we can get a shorter $t^* = t^* \setminus s_2s_3 \cup s_1s_3$. This contradicts the assumption that $t^*$ is the MST of $G$.

Then, we prove result 2.

We are going to take Figure 5 as an example to complete the proof. Let three components $C_1, C_2$ and $C_3$ consist of $s_1s_2 \ldots s_6, s_7s_8s_9$ and $s_{10}s_{11} \ldots s_{14}$, respectively, and then the closest nodes be chosen as representative based on $A_{rc}$ are $s_5, s_8$ and $s_{14}$ for $C_1, C_2$ and $C_3$, respectively. The line-segment $s_4s_5$...
extends to the left and right until it reaches the \( s_7 \) and \( s_{10} \), respectively. By adjusting the positions of \( C_2 \) and \( C_3 \), we have \( L_{s_5s_7} = L_{s_5s_9} \) and \( L_{s_4s_{10}} = L_{s_4s_{11}} \) and both of \( \angle s_8s_5s_4 \) and \( \angle s_5s_4s_{14} \) are obtuse. This implies that \( L_{s_5s_{14}} > L_{s_4s_{14}} = L_{s_4s_{10}} \). Since \( Ad-Prim \) could choose multiple nodes, say \( s_4 \) and \( s_5 \), as representatives of \( C_1 \), the tree \( t \) built by \( Ad-Prim \) could consist of \( s_5s_7 \) and \( s_4s_{10} \). This implies \( L_t \leq L_{\text{MST}} \).

\[ N_{\text{A}_\text{MST}} = \frac{L_t - 2r(n - 2) - 2r}{r} + n - 2, \]

\[ N_{\text{OASS}} = \frac{L_t - 3r(n - 2) - nr}{r} + n - 2, \]

\[ N_{\text{A}_\text{tc}} = \frac{L_t - 2rn}{r} + 1. \]
Case 1: Since each pch_i is on the perimeter of a convex hull and RNs are placed on each edge s_is_j ∈ pch_i starting from s_i and s_j at the interval r, relay node consumption function N_{AMST} is given as follows:

\[ N_{AMST} = \frac{L_t - 2r(n - 2) - 2r}{r}. \] (2)

Case 2: If there exist two potential convex hulls pch_i and pch_j such that pch_i \cap pch_j = s_k, then both of pch_i and pch_j remain the same. The tree t built by OASS consists of a number of straight skeletons. Since a straight skeleton has property 1 described in Section 3, each Steiner point requires totally n - 2 RNs needed. In addition, these Steiner nodes serve as the starting point to deploy RNs toward all n leave nodes. Therefore, relay node consumption function N is given as follows:

\[ N_{OASS} = \frac{L_t - 3r(n - 2) - nr}{r} + n - 2. \] (3)

Case 3: For a center-toward tree t, if there exist two potential convex hulls pch_i, pch_j ∈ t such that pch_i \cap pch_j = s_k, then let pch_j = pch_i \setminus s_k. By doing so, all pch_i_s can be separated from each other such that each leave node s_i ∈ pch_i can deploy RNs toward the center. Since A_{tc} always have a center, at least a RN is required at the position of the center as the beginning in order to deploy RNs towards all n leave nodes. Therefore, relay node consumption function N_{A_{tc}} is given as follows:

\[ N_{A_{tc}} = \frac{L_t - 2rn}{r} + 1. \] (4)

Then, we prove the relationships between three relay node consumption functions N_{OASS}, N_{A_{tc}} and N_{AMST}.

**Theorem 3.** N_{OASS} ≤ N_{A_{tc}} ≤ N_{AMST}.

**Proof.** For simplicity, we let r = 1. According to Equations (2)–(4), we have the following results:

\[ L_t - 3n + 4 = N_{OASS}, \] (5)
\[ L_t - 2n + 2 = N_{AMST}, \] (6)
\[ L_t - 2n + 1 = N_{A_{tc}}. \] (7)

Since each pch consists of at least three nodes, we have n ≥ 3. Therefore, the following inequations hold true:

\[ N_{OASS} = L_t - 3n + 4, \]
\[ < L_t - 2n + 1 = N_{A_{tc}}, \] (8)
\[ < L_t - 2n + 2 = N_{AMST}. \]

In addition, for all r, r > 1, it is easy to verify the correctness of this theorem.

**Theorem 4.** [2] The approximation ratio of A_{MST} is 3.

According to Theorem 3, it is easy to deduce that, if RNs are equally employed, then we have L_{ss} < L_{tc} < L_{MST}, where L_{ss}, L_{tc} and L_{MST} denote the tree constructed by OASS, A_{tc} and A_{MST}, respectively.
Theorem 5. The approximation ratio of OASS is 3.

Proof. The OASS builds an obstacle–avoid MST first, then subdivides the MST into a number of subtrees. If a subtree \( t \in \text{MST} \) is a \( \text{pch} \), then a straight skeleton is placed within as a relay node deployment route. Otherwise, RNs are deployed along \( t \). We distinguish the following two cases to prove this theorem.

Case 1: the topology of graph \( G \) can be decomposed to a series of convex hulls.

In this case, we have \( \text{MST} = \bigcup \text{pch}_i \). If the \( \text{MST}_i \) is the MST of all nodes \( s_i \in \text{pch}_i \), then \( \text{MST}_i = \text{pch}_i \) due to the characteristics of a convex hull. It is obvious that \( L_{\text{MST}} \geq \sum L_{\text{MST}_i} \). Let \( ss_i \) denote the straight skeleton built within \( \text{pch}_i \). Theorem 3 ensures \( L_{ss_i} < L_{\text{MST}_i} \), which implies \( \sum L_{ss_i} < \sum L_{\text{MST}_i} \leq L_{\text{MST}} \) and \( N_{\text{OASS}} < N_{\text{MST}} \). According to Theorem 4, \( A_{\text{MST}} \) is a 3-opt; therefore, the approximation ratio of OASS is 3.

Case 2: the topology of graph \( G \) can’t be decomposed to a series of convex hulls.

In this case, there exists at least a subtree \( t \in \text{MST} \), which is not a \( \text{pch} \), such that \( \text{MST} = \bigcup \text{pch}_i \bigcup t \). Similar to the previous case, we can get \( \sum L_{ss_i} + L_t < \sum L_{\text{MST}_i} + L_t \leq L_{\text{MST}} \) and \( N_{\text{OASS}} < N_{\text{MST}} \). According to Theorem 4, OASS is a 3-opt approximation algorithm.

We call an approach \( A_t \), if \( A_t \) deploys RNs along a tree \( t \) for connectivity restoration.

Theorem 6. If a tree \( t \) has property 1, then \( N_{A_t} < \sqrt{3} N_{A_{\text{MST}}} \).

Proof. According to literature [27] and Theorem 3, if a shortest tree \( t \) has property 1, then we have \( \frac{\sqrt{3}}{2} L_{\text{MST}} \leq L_t \leq L_{\text{MST}} \).

According to Equations (5) and (6), if \( \frac{\sqrt{3}}{2} L_{\text{MST}} = L_t \), then we have

\[
N_{A_{\text{MST}}} \times \frac{\sqrt{3}}{2} = (L_{\text{MST}} - 2n + 2) \times \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{2} L_{\text{MST}} - \sqrt{3}n + \sqrt{3} > \frac{\sqrt{3}}{2} L_{\text{MST}} - 3n + 4 = N_{A_t}.
\]

Therefore, this theorem holds.

It is apparent that a straight skeleton has property 1 as described in Section 3. More importantly, we give the comparison between the length of the SMT and that of a straight skeleton in Table 3.

The complexity of OASS is \( O(n \log n) \).

Proof. The OASS consists of three-phase processes, which are \( \text{Ad-Prim} \), OA and SSIC, respectively. We give individual complexity analysis to prove that the complexity of OASS is \( O(n \log n) \).
In phase one, since Ad-Prim is an adapted construction algorithm for the MST with one rule that circles formed by nodes on the same \( C_i \) should be ignored, and the complexity for Ad-Prim should be no more than that of Prim, which is \( O(n \log n) \) [25].

In phase two, OA discovers all obstacle–avoid paths in \( O(\epsilon) \) while there are obstacles on MST, where \( \epsilon \) is a constant.

In phase three, SSIC employs straight skeletons within \( \text{pchs} \) as relay node deployment routes. Suppose the number of \( \text{pchs} \in \text{MST} \) is \( k \) such that \( \text{pch}_i \cap \text{pch}_j = s_i \), where \( j = i + 1 \). Therefore, we have \( \sum |\text{pch}_i| = n + k - 1 \) that implies there are \( \frac{n + k - 1}{k} \) nodes for each \( \text{pch} \) on average. The construction of a convex hull, which is a \( \text{pch} \) in our case, costs no more than \( O(\frac{n + k - 1}{k} \log \frac{n + k - 1}{k}) \) [29]. Thus, \( k \) convex hulls will cost no more than \( O(k \times \frac{n + k - 1}{k} \log \frac{n + k - 1}{k}) = O(n \log n) \), where \( 2 \leq k < n \). In addition, each straight skeleton is built with a cost no more than \( O(n \log n) \) [28]. Since each straight skeleton can be built when a \( \text{pch}_i \) is established, all straight skeletons will be established with the complexity of \( O(n \log n) \).

To sum up, each phase of OASS costs no more than \( O(n \log n) \); therefore, the complexity of OASS is \( O(n \log n) \).

5.2. Validation Experiment

We first give the comparison between the SMT and the straight skeleton in terms of the Euclidean distance. Then, performance metrics and baseline approaches are introduced. At last, simulation results and comparison of the generated topology quality are presented.

5.2.1. Comparison between the SMT and the Straight Skeleton

Our tests were built in Euclidean space on some of Soukup examples [26]. In Table 3, we compare straight skeletons with Steiner Minimal Trees (SMTs). As illustrated in Table 3, the length of a straight skeleton is infinitely approximating to that of an SMT in all cases.

| Number of Soukup Example | Number of Nodes | SMT   | Straight Skeleton |
|---------------------------|-----------------|-------|-------------------|
| EX.1                      | 5               | 165.43| 166.57            |
| EX.2                      | 6               | 155.06| 156.04            |
| EX.3                      | 6               | 158.60| 159.05            |
| EX.4                      | 6               | 130.15| 131.03            |
| EX.5                      | 9               | 163.20| 163.85            |
| EX.6                      | 9               | 128.50| 129.10            |
| EX.7                      | 12              | 221.20| 222.06            |
| EX.8                      | 14              | 122.02| 122.64            |
| EX.9                      | 3               | 115.54| 116.32            |
| EX.10                     | 10              | 164.26| 165.20            |
| EX.11                     | 62              | 382.56| 383.62            |
| EX.12                     | 14              | 172.30| 173.23            |
| EX.13                     | 3               | 104.15| 105.10            |
| EX.14                     | 5               | 182.92| 183.05            |
| EX.15                     | 4               | 50.30 | 51.22             |
5.2.2. Experiment Setup, Performance Metrics and Baseline Approaches

A simulation environment is implemented and validated in C++ environment on NS2.34. The experiments are conducted in 2000 m \( \times \) 2000 m square area. Over 200 sensor nodes are deployed in this area. Overall, 30 random topologies in the given area are taken. For each topology, obstacles are randomly scattered over the deployment area. It is observed that, with 95% confidence level, the results are within 5–10% of the sample mean. The parameters and baseline approaches are introduced to evaluate the performance of the proposed solution are listed as follows.

**Communication range of relays** \( (r) \): The performance of OASS is affected by \( r \), basically longer distances between components require more RNs.

**Number of components** \( (N_c) \): Intuitively, a large number of components require a larger \( R_N \) count.

**Number of RNs** \( (N_{rn}) \): A better approximation algorithm for the optimization of \( R_N \) placement require less RNs than other strategies.

We compare the performance of OASS with the following four baseline approaches. The first one is ORNP [9] that employs the center of mass and 3-star to establish a relay node deployment route. The second one is RLC-GBP [14], which is short for Restore Relay Lost Connectivity using zero Gradient Based Point solution, deploys RNs toward a certain point of the deployment area. The third one is a straight skeleton based approach, named GSR [12]. IO-DT [11], the fourth one, is based on Delaunay triangulation.

**ORNP**: It is a two-phase algorithm. In phase one, 3-stars are constructed as many as possible. In phase two, the center of mass (CoM) of all remaining components is calculated. Then, each 3-star deploys RNs toward the CoM such that the connectivity is restored.

**RRLC-GBP**: It is one of A. Similar to OASS, RRLC-GBP uses an MST based algorithm to find all representatives. Then, a distance function \( F \) with respect to the coordinates of each representative is constructed in order to find a specific point zero gradient point \( p \) such that the total distance from all representatives to \( p \) is minimized. At last, each representative deploys RNs toward the point \( p \).

**GSR**: It is a straight skeleton based algorithm. According to the convex hull of all representatives, only one straight skeleton is constructed. Then, based on the position of each representative, RNs are populated along the straight skeleton until the connectivity is restored.

**IO-DT**: It is a three-phase algorithm. In phase one, the MST is built among all disjointed components. In phase two, along the MST, the Delaunay triangulation is employed based on which the Fermat weight of each triangle is calculated. In the last phase, according to the Fermat weight of each triangle the corresponding edges of which are used for inter-triangle connections or inner-triangle connections. Once the tree \( t \) that consists of triangles is built, RNs will deployed along \( t \).

It is worth mentioning that ORNP, GSR and IO-DT don’t find the optimal solution to the representative selection problem. According to Theorem 1, the tree \( t \) built by OASS is shorter than that of ORNP, GSR and IO-DT. Furthermore, ORNP, RRLC-GBP, GSR and IO-DT assume that the deployment area is flat without obstacles, which is not realistic. In summary, OASS is developed in real scenarios with the presence of obstacles, the RNs required by which is less than ORNP, RRLC-GBP, GSR and IO-DT due to the tree built by OASS is shorter and the proper selection on representatives as well. Furthermore, compared with baseline approaches, OASS populates RNs around the obstacles in the shortest way. In order to show the advantage in the number of RNs consumed of OASS, we use OA to help all baseline approaches to avoid obstacles in our simulations.

5.2.3. Simulation Results and Comparison of the Generated Topology Quality

Several configurations with different combinations of \( N_c \) and \( r \) are simulated. We change the value of \( N_c \) from 4 to 8, while \( r \) varies from 50 to 190 with increment of 20. The results of individual experiments for 30 topologies are then averaged.

Figures 6 and 7 give the performance comparison between OASS and all baseline approaches, and OA is employed by all approaches to avoid obstacles.
Figure 6. The effect of varying \( r \) on the performance of OASS compared with all baseline approaches w.r.t. (a) when \( N_c = 4 \); (b) when \( N_c = 6 \).

Figure 7. The effect of varying \( N_c \) on the performance of OASS compared with all baseline approaches w.r.t. (a) when \( r = 50 \) m; (b) when \( r = 190 \) m.

Figure 6a,b show the required number of RNs while varying node communication range with \( N_c = 4 \) and \( N_c = 6 \), respectively. It is obvious that, as the communication range increases, RNs required for each approach decrease. It is clear that RRLC-GBP consumes less RNs than ORNP due to the fact that, compared with CoM, the zero gradient point is the point such that sum distance from all representatives to which is minimized. This is also confirmed in the simulation results, as shown in Figure 6a,b. Furthermore, the GSR shows better performance than ORNP, RRLC-GBP and IO-DT. The reason for that is: (1) ORNP and RRLC-GBP, as \( A_{tc}s \), will repair the connectivity with a tree longer than the straight skeleton built by GSR as proved earlier; (2) the structure established by IO-DT that consists of 3-stars is similar to a straight skeleton; however, the RNs required are more than that of a straight skeleton. The OASS shows an excellent result compared with ORNP, RRLC-GBP, GSR and IO-DT as the communication range intensively increases. The reason is that a small count of RNs is deployed from all representatives along the tree that consists of straight skeletons along with the proper representative selection, as proved earlier.

Figure 7a,b show the required number of RNs while varying the number of representatives range with \( r = 100 \) and \( r = 200 \), respectively. It is obvious that, as the number of representative increases, the number of RNs required for each approach grows when \( r = 100 \) as shown in Figure 7a. However, when \( r \) reach 200 m and \( N_c = 7 \), the RN count drops for all approaches as the \( N_c \) grows as shown in Figure 7b. The reason for that is the dense population of components shortens the inter-component distances, which implies that the RNs required are less. Again, Figure 7a,b confirm the advantage of OASS over ORNP, RRLC-GBP, GSR and IO-DT in RN count as the number of components increases.

It is obvious that, in the best case (e.g., the topology consists of convex hulls, all straight skeletons are not intersecting with obstacles), the OASS requires the minimum RN count. However, in the worst case (i.e., the topology not only consists of convex hulls, some straight skeletons are inevitable
intersecting with obstacles), Theorem 5 guarantees that the RN count required by OASS is only three times that of the optimum.

We then give a comparison to show the topological advantages of OASS over RRLC-GBP in terms of distance (i.e., hops between every pair of representatives), delivery latency and balanced traffic load of the repaired network.

**Distance (hops):** Figure 8 shows the comparison on the resulting topologies obtained by OASS and RRLC-GBP with the same setup. Accordingly, Table 4 presents the number of hops between every two segments in both topologies. We assume bidirectional data delivery between two representatives use the same path, where \( A \) denotes OASS as \( B \) denotes RRLC-GBP. Obviously, OASS deploys less RNs to forward data than RRLC-GBP. Overall, OASS requires 152 hops for delivery between all pairs of representatives, which is 22 hops less than that of RRLC-GBP. Therefore, OASS results in a shorter topology, which implies that data delivery requires less overall latency than that of RRLC-GBP.

![Figure 8](image_url)

**Figure 8.** The comparison of the resulting topologies obtained by (a) OASS and (b) Restore Relay Lost Connectivity using zero Gradient Based Point solution (RRLC-GBP).

**Table 4.** OASS vs. Restore Relay Lost Connectivity using zero Gradient Based Point solution (RRLC-GBP) in distance (hops).

| Hops | \( s_1 \) | \( s_2 \) | \( s_3 \) | \( s_4 \) | \( s_5 \) |
|------|-----|-----|-----|-----|-----|
|      | A   | B   | A   | B   | A   | B   | A   | B   | A   | B   |
| \( s_1 \) | 0   | 0   | 7   | 9   | 8   | 9   | 8   | 8   | 7   | 8   |
| \( s_2 \) | 7   | 9   | 0   | 0   | 5   | 10  | 9   | 9   | 8   | 9   |
| \( s_3 \) | 8   | 9   | 5   | 10  | 0   | 0   | 10  | 9   | 9   | 9   |
| \( s_4 \) | 8   | 8   | 9   | 9   | 10  | 9   | 0   | 0   | 5   | 7   |
| \( s_5 \) | 7   | 8   | 8   | 9   | 9   | 9   | 5   | 7   | 0   | 0   |
| Sum   | 30  | 34  | 29  | 37  | 32  | 37  | 32  | 33  | 29  | 33  |
| Ave.  | 6   | 6.8 | 5.8 | 7.4 | 6.4 | 7.4 | 6.4 | 6.6 | 5.8 | 6.6 |

**Delivery latency:** Since the delivery latency mostly results from data delivery via multi-hop links, the more hops a link processes, the more delivery latency will be generated. According to Table 4, it is obvious that the delivery latency of OASS is much less than that of RRLC-GBP.
**Balanced Traffic Load:** As shown in Figure 8a, it is clear that each pair of representatives should communicate with each other via relay nodes \(r_1\), \(r_2\) and \(r_3\), which serve as rendezvous points of each communication path, while relay node \(r_4\) is responsible for all communications between each pair of representatives as shown in Figure 8b. Statistically, about 32% of total traffic load are transferred via relay nodes \(r_1\) and \(r_3\), respectively, while relay node \(r_2\) transfers 36% of which. However, 100% traffic load are transferred via relay node \(r_4\). It is no doubt that \(r_1\), \(r_2\) and \(r_3\) scatter the traffic load of \(r_4\). In addition, the advantage of OASS over RRLC-GBP is quite obvious because the relay node as \(r_4\) will die much earlier than other newly deployed ones that require another time of relay node deployment at the position where \(r_4\) is located sooner or later.

**6. Conclusions**

The connectivity has significance in WSNs. Once the connectivity is lost, relay nodes are deployed to build an SMT such that the inter-component connection is reestablished. Most of the previous works assumed that the deployment area is flat without obstacles. However, it is not realistic. A lot of existing strategies chose the representative of each component in a random way or in the shortest-distance based manner. Both ways of representative selection could potentially increase the length of the SMT such that more RNs are required. In this paper, we propose a novel connectivity restoration strategy OASS, which employs both the polygon based representative selection with the presence of obstacles and the straight skeleton based SMT establishment. The OASS is proved to be a \(3\)-approximation algorithm with the complexity of \(O(n \log n)\), and the approximation ratio can reduce to \(\frac{3\sqrt{3}}{2}\) while it satisfies a certain condition. The theoretical analysis and simulations show that the performance of the OASS is better than other strategies in terms of the relay count and the quality of the established topology (i.e., distances between components, delivery latency and balanced traffic load) as well.

Although the OASS is a highly efficient algorithm, it is designed only for connectivity restoration problems in two-dimensional scenarios. In addition, if the deployment area is full of obstacles, then the OASS may not place the least number of relay nodes to restore the connectivity. In the future, we should address the three-dimensional connectivity restoration problem with the presence of massive obstacles.

**Acknowledgments:** The authors wish to thank the National Natural Science Foundation of China (Nos. 61572010, 61771140, 61702103, 61702100), the Fujian Normal University Innovative Research Team (No. IRTL1207), the Natural Science Foundation of Fujian Province (No. 2016J01289), the Fujian Province University Industry Cooperation of Major Science and Technology Project (No. 2017H6005), and the China Postdoctoral Science Foundation (No. 2017M612107).

**Author Contributions:** S.Z. conceived, designed and performed the experiments; X.W. proposed and developed the algorithm; L.X. and X.W. analyzed the data and wrote the paper.

**Conflicts of Interest:** We declare that we have no conflicts of interest.

**References**

1. Younis, M.; Senturk, I.F.; Akkaya, K.; Lee, S.; Senel, F. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. *Comput. Netw.* **2014**, *58*, 254–283.
2. Chen, D.; Du, D.Z.; Hu, X.D.; Lin, G.H.; Wang, L.; Xue, G. Approximations for Steiner trees with minimum number of Steiner points. *J. Glob. Optim.* **2000**, *18*, 17–33.
3. Tang, J.; Hao, B.; Sen, A. Relay node placement in large scale wireless sensor networks. *Comput. Commun.* **2006**, *29*, 490–501.
4. Lloyd, E.L.; Xue, G. Relay Node Placement in Wireless Sensor Networks. *IEEE Trans. Comput.* **2007**, *56*, 134–138.
5. Cheng, X.; Du, D.Z.; Wang, L.; Xu, B. Relay sensor placement in wireless sensor networks. *Wirel. Netw.* **2008**, *14*, 347–355.
6. Misra, S.; Hong, S.D.; Xue, G.; Tang, J. Constrained relay node placement in wireless sensor networks: Formulation and approximations. *IEEE/ACM Trans. Netw.* **2010**, *18*, 434–447.
7. Yang, D.; Misra, S.; Fang, X.; Xue, G.; Zhang, J. Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations. *IEEE Trans. Mob. Comput.* 2012, 11, 1399–1411.

8. Misra, S.; Majd, N.E.; Huang, H. Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks. *IEEE Trans. Comput.* 2014, 63, 2933–2947.

9. Wang, X.; Xu, L.; Zhou, S. Restoration Strategy Based on Optimal Relay Node Placement in Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* 2015, 11, 409085.

10. Efrat, A.; Fekete, S.P.; Mitchell, J.S.B.; Polishchuk, V.; Suomela, J. Improved approximation algorithms for relay placement. *ACM Trans. Algorithms* 2016, 12, 20.

11. Senel, F.; Younis, M. Novel relay node placement algorithms for establishing connected topologies. *J. Netw. Comput. Appl.* 2016, 70, 114–130.

12. Joshi, Y.K.; Younis, M. Exploiting skeletonization to restore connectivity in a wireless sensor network. *Comput. Commun.* 2016, 75, 97–107.

13. Lee, S.; Younis, M. Optimized relay node placement for connecting disjoint wireless sensor networks. *Comput. Netw.* 2012, 56, 2788–2804.

14. Ranga, V.; Dave, M.; Verma, A.K. Relay node placement to heal partitioned wireless sensor networks. *Comput. Electr. Eng.* 2015, 48, 371–388.

15. Lee, S.; Younis, M.; Lee, M. Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance. *Ad Hoc Netw.* 2015, 24, 1–19.

16. Uwitonze, A.; Huang, J.; Ye, Y.; Cheng, W. Connectivity Restoration in Wireless Sensor Networks via Space Network Coding. *Sensors* 2017, 17, 902.

17. Abbas, A.; Younis, M. Establishing connectivity among disjoint terminals using a mix of stationary and mobile relays. *Comput. Commun.* 2013, 36, 1411–1421.

18. Joshi, Y.K.; Younis, M. Restoring connectivity in a resource constrained WSN. *J. Netw. Comput. Appl.* 2016, 66, 151–165.

19. Zhou, S.; Wu, M.; Shu, W. Terrain-constrained mobile sensor networks. In Proceedings of the C.GLOBECOM ’05, IEEE Global Telecommunications Conference, St. Louis, MO, USA, 28 November–2 December 2005; p. 5.

20. Senturk, I.F.; Akkaya, K.; Janansefat, S. Towards realistic connectivity restoration in partitioned mobile sensor networks. *Int. J. Commun. Syst.* 2014, 29, 230–250.

21. Truong, T.T.; Brown, K.N.; Sreenan, C.J. Multi-objective hierarchical algorithms for restoring wireless sensor network connectivity in known environments. *Ad Hoc Netw.* 2015, 33, 190–208.

22. Mi, Z.; Yang, Y.; Yang, J. Restoring connectivity of mobile robotic sensor networks while avoiding obstacles. *Sens. J. IEEE* 2015, 15, 4640–4650.

23. Wang, X.; Xu, L.; Zhou, S.; Wu, W. Hybrid Recovery Strategy Based on Random Terrain in Wireless Sensor Networks. *Sci. Program.* 2017, 2017, 1–19.

24. Robins, G.; Zelikovsky, A. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discret. Math.* 2005, 19, 122–134.

25. Bondy, J.A.; Murty, U.S.R. *Graph Theory with Applications*; Macmillan: London, UK, 1976; pp. 237–238.

26. Soukup, J.; Chow, W.F. Set of test problems for the minimum length connection networks. *ACM SIGMAP Bull.* 1973, 15, 48–51.

27. Du, D.Z.; Hwang, F.K. The state of art on Steiner ratio problems. *Comput. Euclidean Geom.* 1992, 1, 163–191.

28. Therese, B.; Martin, H.; Stefan, H.; Dominik, K.; Peter, P. A simple algorithm for computing positively weighted straight skeletons of monotone polygons. *Inf. Process. Lett.* 2015, 115, 243–247.

29. Barber, C.B.; Dobkin, D.P.; Huhdanpaa, H. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 1996, 22, 469–483.

© 2017 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/).