Resilient greedy routing on GPS-free surface sensor networks

Buri Ban, Xuan Li and Miao Jin

Abstract
We design a greedy routing scheme specifically for GPS-free large-scale wireless sensor networks deployed on surfaces of complex-connected three-dimensional settings. Compared with other greedy embedding–based surface network routing scheme, the proposed one is cut free such that no pair of nodes suffers a long detour to reach each other. The routing scheme is designed to be resilient to node or link failures especially under random node or link failure model where each node in a network has an equal and independent probability of failure during some time interval. The proposed algorithm is fully distributed and scalable to both the size and the topological complexity of a network. Each sensor node requires only limited and constant storage. Simulation results show the proposed routing scheme with a higher successful delivery ratio, a lower average stretch factor, and a lower normalized communication cost compared with other resilient routing methods.

Keywords
Greedy routing, surface network, resilient, GPS-free

Date received: 4 April 2019; accepted: 17 February 2020

Handling Editor: Michele Amoretti

Introduction
A wireless sensor network may be deployed on a two-dimensional (2D) plane (e.g. for crop sensing in fields or wildlife tracking on plains), or in a three-dimensional (3D) volume (for underwater or space reconnaissance), or on a 3D surface (e.g. with sensors mounted on ocean floor, mountain surface, or the surface of various man-made structures). Routing has always been a fundamental challenge in large-scale wireless sensor networks. A diversity of sensor network routing algorithms have been developed, but they primarily target at 2D planar or 3D volumetric settings. This article focuses on designing a routing algorithm resilient to random node or link failures in a large-scale wireless sensor network deployed on the surface of a complex-connected 3D setting. Such 3D setting generally has multiple handles and forms a high genus surface. Applications for such wireless sensor networks on high genus surfaces have been demonstrated for structural health monitoring (SHM), for example, at the Golden Gate Bridge and National Stadium of China, where sensors are deployed on megastructures to gauge changes in materials or geometric properties that could hinder the system’s performance. Other applications can be found along the corridors of buildings for fire detection, and in water, sewer, or gas system for monitoring underground pipelines as introduced in Yu et al. We consider densely deployed sensors that operate on high radio frequency and extremely low transmission power with short communication range. Only nearby sensors along a 3D surface can communicate...
with each other, whereas the wireless links connecting remote sensors across space are negligible and can be removed via a simple preprocessing. As a result, the dense sensors form a 3D surface network.

An overview of resilient routing

Node and link failures are unavoidable in a large-scale distributed sensor system. For example, nodes may die out of battery and communication links can be temporarily or permanently disabled due to attacks or interference. We can model network failures with two failure models: the independent node failure model and the geographically correlated failure model. In the independent node failure model, each node in a network has an equal and independent probability of failure during some time interval. Independent node failure can represent failure due to energy dissipation or localized environmental effects. In the geographically correlated failure model, all nodes within a certain range or hop count fail. Geographically correlated failure may happen due to natural environmental effects (such as rain fades) or some human activity within a geographic region.

Resilient routing has been discussed in conventional 2D sensor networks. For example, Directed Diffusion is a well-known routing algorithm that establishes gradients and uses gradual reinforcement of better paths to allow path recovery, enabling a system to be robust to a local network and sink dynamics. Multipath routing represents an efficient routing strategy to increase network resilience to node and link failures. Algorithms have been proposed for Internet multipath routing and sensor network multipath routing. In Ganesan et al., the authors consider two different approaches to construct multipaths between two nodes. One is node-disjoint multipath, where the alternate paths do not intersect with each other. The other approach builds many braided paths—partially disjoint alternate paths. In Lou, a branch-aware flooding scheme is utilized to construct a spanning tree and discover a set of node-disjoint paths for each sensor node back to the base station. Later, a tree structure is applied for multipath routing. One recent advance in multipath routing is homotopy-based routing. The authors apply the concept of homotopy to effectively evaluate the diversity of alternative paths of a network deployed on 2D plane with uncovered holes. The simplest case to illustrate the concept of paths with different homotopy types is a network deployed on a planar region with a lake inside. Two paths connecting a pair of source and destination are homotopic to each other if they are on the same side of the lake; otherwise, they belong to different homotopy types because one path cannot continuously deform to the other without crossing the lake. The most recent work in multipath routing is introduced in Liu et al. with a security disjoint routing-based verified message (SDRVM) scheme proposed to improve the network performance with comprehensive consideration of universal resistance to attacks, energy efficiency, and transmission delay. The main strategy is to divide nodes into several sets based on their remaining energy and assign different duties to each set.

Multipath routing schemes are effective for geographically correlated failures. Alternative paths, especially those disjoint with the failed one, have a high probability to avoid a network failure region. Homotopy-based multipath routing schemes choose alternative paths topologically different from the failed one. These alternative different paths provide a high probability of routing success even under massive geographically correlated failures. However, under the independent node failure model, the performance of multipath routing schemes decrease dramatically with a slight increase in node failure probability. The reason is that an alternate path is always longer than the original one for any multipath method, so the failure probability of the alternate path is always higher than the original one considering each node has an equal probability of failure. It is obvious that multipath routing schemes, especially those choosing alternative paths based on disjoint or different homotopy types, have intrinsically a very low resilience to independent node failures. If the probability of node failure is not low, switching to an alternate path does not help.

Instead of choosing an alternative path, a local detour is more efficient under the independent node or link failure model. We can consider dead nodes or regions with broken links as holes. To avoid torrents of triggered global updates of network topology, a practical strategy is to apply Face routing and its alternatives to locally detour packages around the boundary of each hole. Specifically, a node always forwards a packet to one of its neighbors closest to the destination of the packet. When greedy forwarding of a packet fails due to holes, the packet can be forwarded along the boundary of holes in either clockwise or counterclockwise direction until greedy forwarding is achievable.

However, Face routing and its alternatives require all sensor nodes to know their own positions. For networks deployed on indoor, underground or underwater surfaces, GPS signal is forbidden. They also require networks to be planar. For networks deployed on surfaces of complex-connected 3D settings, when greedy forwarding of a packet is stuck at a local minimum including a hole, as shown in Figure 1(a), there is no deterministic path to successfully forward the packet to the destination based on local information only. We propose a fully distributed algorithm to apply such local detour strategy to achieve efficient resilient.
Routing for GPS-free sensor networks deployed on surfaces of complex-connected 3D structures. Routing is based on local information only. Network dynamic information only needs to be updated locally.

**Our approach**

A few greedy routing methods\(^{19,22}\) have been developed recently to address the GPS-free surface sensor network routing problem. In Yu et al.\(^{19,22}\), the authors propose to cut a closed high genus surface network open and then map the cut-open one to a planar rectangle with two pairs of boundaries. Furthermore, the planar rectangle can be mapped to a planar annulus with one pair of boundaries aligned and glued together. Each sensor node stores a set of planar coordinates and uses it to enable greedy routing. However, these methods suffer a large stretch factor in average. The averaged stretch factor is defined as an average of the ratios of the hop counts of greedy forwarding paths to the shortest ones. When the original surface network is cut open and mapped to either a planar disk, rectangle, or annulus, nodes located at the two sides of the cut-open boundaries need to take a long detour to reach each other instead of directly crossing the boundary. Figure 1(b) shows an example of a pair of such nodes.

The problem of a large stretch factor is intrinsic to the methods proposed in Yu et al.\(^{19,22}\). They need to cut the originally closed surface network open and then map the network to plane. We propose a cut-free embedding of the original network in the plane. Locally, any region of the network is mapped to plane one-to-one and continuously. The key of the seamless embedding is to extract a triangular mesh structure from the connectivity graph of the original network and compute a set of special edge lengths for the mesh. Based on the computed edge lengths, the original surface network can be embedded in the plane seamlessly. A packet is greedily forwarded based on the planar coordinates stored at each node. Later, the network dynamically maintains the mesh structure. Each node periodically updates its neighboring nodes and local mesh structure due to a potential node or link failure. A face of the mesh is always an \(n\)-polygon with \(n \geq 3\). We consider all faces with \(n > 3\) holes. When a packet is routed to hit a hole, the packet is forwarded along the boundary of the hole in either counterclockwise or clockwise direction until greedy forwarding is achievable.

The simulation shows the proposed routing scheme with a higher successful delivery ratio, a lower average stretch factor, and a lower normalized communication cost compared with other resilient routing methods. Such results indicate the proposed routing scheme has a strong resilience to network failures, a low cost to detour, and long durability of the network system.

**Our contributions.** Our contributions can be summarized as follows:

- We design a greedy routing scheme specifically for GPS-free wireless sensor networks deployed on surfaces of complex-connected 3D settings. Compared with other greedy embedding-based surface network routing scheme, the proposed approach is cut free such that no pair of nodes suffers a long detour to reach each other.

![Figure 1](image.png)
The routing scheme is designed to be resilient to node or link failures especially under random node or link failure model. Simulation results show a high successful delivery ratio, a low average stretch factor, and a low normalized communication cost.

The proposed algorithm is fully distributed and scalable to both the size and the topological complexity of a network.

The rest of this article is organized as follows: section “Uniformization metric” gives a brief introduction of uniformization metric—the special set of edge lengths we compute for a surface network. Section “Algorithms” elaborates the proposed resilient routing algorithms. Section “Simulations” presents simulation and comparison results. Section “Conclusion” concludes the article.

**Uniformization metric**

In this section, we introduce the uniformization metric, which is essentially a special set of edge lengths we compute to support the proposed resilient routing algorithm. Denote \( S \) a closed surface embedded in \( \mathbb{R}^3 \) and \( L \) a loop on \( S \). \( L \) is surface separating if it can be expressed as the symmetric difference of boundaries of topological disks embedded in a surface as shown in Figure 2 with \( L_1 \) and \( L_2 \); otherwise, it is non-separating as shown in Figure 2 with \( L_3 \).

The genus of \( S \), equivalent to the number of handles of \( S \), is the maximum number of disjoint non-separating loops \( L_1, L_2, \ldots, L_g \) in \( M \); that is, any \( L_i \) and \( L_j \) have no topological intersection if \( i \neq j \), and \( M \setminus (L_1 \cup \cdots \cup L_g) \) is connected. The genus number is the most basic topology information of a surface. For example, a torus is a genus one surface, and a double torus as shown in Figure 2 is a genus two surface.

A special set of edge lengths to embed \( S \) to plane without cutting is called uniformization metric. Uniformization metric exists for general closed surfaces according to Uniformization Theorem.35

Informally speaking, let \( S \) with Riemannian metric induced from the Euclidean metric of \( S \). We denote the Riemannian metric of \( S \) as \( g \). There exists a unique metric represented as \( g = e^{2u}g_0 \), where \( u : S \to \mathbb{R} \) is a scalar function defined on \( S \). We call \( g \) a uniformization metric of \( S \). It can be verified that \( g \) is also a Riemannian metric on \( S \) and \( g \) induces constant Gaussian curvature on \( S \).

Specifically, based on surface uniformization metric, \( S \) with genus one can be periodically embedded into Euclidean plane and \( S \) with genus larger than one can be periodically embedded into the hyperbolic plane. If \( S \) is a discrete surface, for example, a triangular mesh, we have the following discrete definitions of Riemannian metric and Gaussian curvature.

Given a triangular mesh denoted as \( M \), we denote a vertex set as \( V \), an edge set as \( E \), and a face set as \( F \). \( c_{ij} \) represents the edge connecting vertices \( v_i \) and \( v_j \), and \( f_{ijk} \) denotes the face formed by \( v_i, v_j, \) and \( v_k \). The edge lengths of a triangular mesh are sufficient to define a discrete Riemannian metric on the discrete surface

\[
l : E \to \mathbb{R}^+
\]
as long as, for each face \( f_{ijk} \), the edge lengths satisfy the triangle inequality: \( l_{ij} + l_{jk} > l_{ki} \).

The discrete Gaussian curvature denoted as \( K_i \) on a vertex \( v_i \in V \) can be computed from the angle deficit

\[
K_i = \begin{cases} 
2\pi - \sum_{f_{ijk} \notin F} \theta_{ijk}^i, & v_i \notin \partial M \\
\pi - \sum_{f_{ijk} \in F} \theta_{ijk}^i, & v_i \in \partial M 
\end{cases}
\]

where \( \theta_{ijk}^i \) represents the corner angle attached to vertex \( v_i \) in the face \( f_{ijk} \), and \( \partial M \) represents the boundary of the mesh. Since \( \theta_{ijk}^i \) can be computed directly from edge lengths of \( f_{ijk} \), it is obvious that discrete Gaussian curvatures are determined by the discrete metrics.

Uniformization metric also exists on a discrete surface. It is a special set of edge lengths such that the discrete surface can be embedded in the plane based on the edge lengths with vertex Gaussian curvatures being zero everywhere. Discrete surface Ricci flow is an efficient tool to compute discrete surface uniformization metric. The algorithm of discrete surface Ricci flow is fully distributed, so it is well suited for wireless sensor network applications. We refer readers to Jin et al.36 for details of discrete surface Ricci flow.

**Algorithms**

This section describes the proposed routing scheme in three steps. Section “Preprocessing” gives the preprocessing step. Section “Computing virtual planar
coordinates” introduces the algorithm to compute virtual planar coordinates for each sensor node of a giving network. Based on the computed virtual planar coordinates, resilient greedy routing is introduced in section “Resilient greedy routing.” Later, we analyze the time complexity and communication cost in section “Time complexity and communication cost.”

Preprocessing

Given a wireless sensor network deployed on a high genus surface, we apply the algorithm proposed in Zhou et al.37 to extract a triangular mesh denoted as $M$ from the connectivity graph of the network based on locally measured distances between nodes within one-hop communication range. Vertices of the triangular mesh are the set of sensor nodes. An edge between two neighboring vertices indicates the communication link between the two sensors. The process is local and has no constraint on communication models.

For densely deployed sensor nodes, we assume each face of the extracted mesh an n-polygon with $n=3$. However, we allow the mesh with non-triangular faces, n-polygons with $n>3$, due to the randomness of sensor deployment. We consider these non-triangular faces as holes. We will show later that these holes won’t affect the computation of uniformization metric and resilient greedy routing.

The topology of the surface of a 3D setting, that is, the genus number, could be available before the deployment of sensor nodes. Otherwise, we can still have the following simple algorithm to automatically detect the surface network topology.

The algorithm starts from one randomly chosen vertex $v_1$ of $M$. $v_1$ can be the one with the smallest node ID. $v_1$ initiates a package with three counters $v_n$, $e_n$, and $f_n$ and sets them as 1, 0, 0, respectively. $v_1$ marks itself and sends the package to one of its direct neighbors, denoted as $v_j$. $v_j$ marks itself and adds $v_n$ by one. $v_j$ checks its neighboring edges. For one with both ending vertices marked (e.g. $e_{ij}$), $v_j$ adds $e_n$ by one. $v_j$ also checks its neighboring triangles. For one with three ending vertices marked, $v_j$ adds $f_n$ by one. $v_j$ then sends the package to one of its unmarked neighbors. Once every vertex has been marked, $v_n$, $e_n$, and $f_n$ in the package have counted the total numbers of vertices, edges, and faces of $M$, respectively. The number of holes of $M$, denoted as $b$, can be counted in a similar way. A non-triangular face initiates a package with a counter $b$ set to one. The package is forwarded to neighboring faces until all faces in $M$ have been visited. Once $v_n$, $e_n$, $f_n$, and $b$ are flooded through the network, each node can easily compute the genus number of $M$ based on Euler–Poincaré Theorem.

**Theorem 1. Euler–Poincaré Theorem.** Denote $v_n$ the number of vertices, $e_n$ the number of edges, $f_n$ the number of faces, $g$ the genus number, and $b$ the number of holes of a triangular mesh $M$. The following equation holds

$$v_n - e_n + f_n = 2 - 2g - b$$

Computing virtual planar coordinates

With the extracted triangular mesh and known topology of the surface, we apply discrete surface Ricci flow36 to compute the special set of edge lengths, the uniformization metric of $M$. As we discussed in section “Preprocessing,” we allow the mesh with non-triangular faces, n-polygons with $n>3$, due to the randomness of sensor deployment. For a face with $n>3$, the algorithm simply adds one virtual vertex and virtual edges to connect the virtual vertex with all other vertices of the face. The face with $n>3$ is then split into $n$ triangle faces.

We refer readers to Jin et al.36 for theoretical background introduction of discrete surface Ricci flow. Here, we only provide the algorithm of computing uniformization metric using discrete surface Ricci flow. The algorithm is fully distributed, and each node only needs to exchange information with its one-range neighbors.

Specifically, each edge $e_{ij}$ has its initial edge length set to unit one. Each vertex $v_i$ is associated with a unit circle with radius denoted as $\gamma_i = 1$. The intersection angle of the two unit circles centered at $v_i$ and $v_j$ is set as the edge weight $\Phi_{ij}$ at $e_{ij}$. In each iteration, each $v_i$ exchanges its current $\gamma_i$ with its direct neighbors and updates its adjacent edge lengths $\{l_{ij} | e_{ij} \in E\}$ according to the following Euclidean cosine law for genus one surface

$$\tilde{l}_{ij} = \gamma_i^2 + \gamma_j^2 + 2\gamma_i \gamma_j \cos \Phi_{ij}$$

or the hyperbolic cosine law for genus larger than one surface

$$\cosh l_{ij} = \cosh \gamma_i \cosh \gamma_j + \sinh \gamma_i \sinh \gamma_j \cos \Phi_{ij}$$

With the updated edge lengths, $v_i$ updates its corner angles $\{\theta_{i}^{lk}|e_{jk} \in F\}$ according to the following inverse Euclidean cos law for genus one surface

$$\theta_{i}^{lk} = \cos^{-1} \frac{l_{ik}^2 + l_{ij}^2 - l_{jk}^2}{2l_{ik}l_{ij}}$$

or the inverse hyperbolic cos law for genus larger than one surface

$$\theta_{i}^{lk} = \cos^{-1} \frac{\cosh l_{ik} \cosh l_{ij} - \cosh l_{jk}}{\sinh l_{ik} \sinh l_{ij}}$$
Then, $v_i$ computes its current discrete Gaussian curvature $K_i$ (equation (1)). If for every $v_i$, its current Gaussian curvature $K_i$ is less than a threshold, discrete surface Ricci flow converges. Otherwise, each $v_i$ updates its $u_i$: $u_i = u_i + \delta(K_i - K_0)$, where $u_i = \log\gamma_i$ and $\delta$ is the step length.

With the computed edge lengths when discrete surface Ricci flow converges, any node can initiate an embedding process. We assume the node with the smallest node ID, denoted as $v_j$. Its planar coordinates denoted as $uv(v_j)$ are set to $(0,0)$. Then, it chooses one of its direct neighbors, denoted as $v_k$. For genus one surface $M$, the planar coordinates of $v_j$ is set to $uv(v_j) = (0,l_{jk})$. Otherwise, the planar coordinates of $v_j$ is set to $uv(v_j) = (\tanh(l_{jk}/2),0)$. For vertex $v_k$, adjacent to both $v_j$ and $v_j$, it calculates the intersection points of two circles centered at $v_j$ and $v_j$ with radii $l_{jk}$ and $l_{jk}$, respectively. For genus one surface $M$, the two circles are Euclidean circles; otherwise, they are hyperbolic circles. A hyperbolic circle can be easily converted to a Euclidean circle according to equation (3) given in Appendix 1, so the computation of the intersection points of two hyperbolic circles is equivalent to the computation of the intersection points of two Euclidean circles. $v_j$ then chooses one intersection point as its planar coordinates such that $v_j$, $v_j$, and $v_j$ satisfy right-hand rule in plane. With $f_{ijk}$ embedded in plane, we continue to embed adjacent triangles with $f_{ijk}$ into plane. Such breadth-first embedding continues until each vertex of $M$ has at least two sets of planar virtual coordinates.

**Resilient greedy routing**

With the stored virtual planar coordinates at each node, the whole network is embedded in plane seamlessly. Locally, any region of the network is mapped to plane one-to-one and continuously. The algorithm then removes all the virtual vertices added in section “Computing virtual planar coordinates” and considers faces with $n\geq3$ holes. For edges shared by only one triangle face, they are labeled as boundary edges, otherwise, non-boundary edges. Vertices connecting with boundary edges are labeled as boundary vertices.

Given a pair of source and target nodes denoted as $S$ and $T$, respectively, considering both $S$ and $T$ having more than one set of planar coordinates, $S$ calculates the Euclidean/Hyperbolic distances of all combinations and chooses the closest pair as the source and target coordinates. $S$ attaches the target coordinates to a packet and then applies greedy perimeter stateless routing (GPRS)\textsuperscript{8} to forward it. Specifically, in greedy forwarding mode, a packet is forwarded to next hop with shortest Euclidean/Hyperbolic distance to $T$. When greedy forwarding encounters holes or local minimums, the packet is changed to perimeter mode. In perimeter model, if the current stuck node is a boundary node, the packet is forwarded along the boundary in counterclockwise direction until greedy forwarding is achievable. Otherwise, the packet is stuck at a local minimum. It is forwarded to the next node of the current triangle face following counterclockwise direction.

At the same time, each node periodically updates its neighboring nodes and local mesh structure due to potential node or link failures. However, the resulting mesh may exist some dangling vertices and edges as shown in Figure 3 that violate a triangular mesh structure. A dangling vertex is a vertex belonging to more than one boundaries. The existence of a dangling vertex confuses the orientation of boundaries. We remove dangling vertices by vertex split operation. As shown in Figure 3(a), $v_1$ is connected with boundary vertices $v_2$, $v_4$, $v_5$, and $v_7$ that belong to two disconnected boundaries with boundary edges marked with red color. We virtually split $v_1$ and denote the newly added node as $v'_1$. $v'_1$ shares the same planar coordinates as $v_1$. $v_1$ is then connected with $v_7$ and $v_2$, and $v'_1$ is connected with $v_4$ and $v_5$. Similarly, a dangling edge is an edge belonging to more than one boundary and not shared by any triangle face. We remove dangling edges by edge removal operation. As shown in Figure 3(b), $e_{01}$ belongs to two disconnected boundaries. We remove this edge such that the previously disconnected boundaries are merged into one. The graph now is a triangular mesh with each face a triangle. A boundary vertex or edge belongs to two disconnected boundaries with boundary edges marked red color. After vertex split, a boundary vertex belongs to one boundary only. (b) Dangling edge: edge $e_{01}$ with ending vertices $v_0$ and $v_1$ belongs to two disconnected boundaries with boundary edges marked with red color. After edge removal, the two boundaries are merged to one. Now a boundary edge belongs to one boundary and is adjacent with one triangle.
exactly one boundary. Therefore, boundary edges and vertices along newly generated holes can be easily detected and labeled.

**Time complexity and communication cost**

Given a network with \( n \) nodes, the time complexity to extract a triangular mesh \( M \) from the connectivity graph of the network is \( O(n) \). The time complexity to detect the network topology is \( O(n) \) too. The time complexity to compute uniformization metric of \( M \) using discrete surface Ricci flow is measured by the number of iterations. It is given by \(-C(\log \varepsilon/\lambda)\), where \( C \) is a constant, \( \varepsilon \) is the threshold of curvature error, and \( \lambda \) is the step length of each iteration.\(^{36}\) We set \( \varepsilon \) and \( \lambda \) to \( 1 \times 10^{-5} \) and 0.1, respectively. The total number of iterations of each network model is no more than hundreds in our simulations. The time complexity to seamlessly embed \( M \) in the plane is also linear to \( n \).

The majority of communication cost, measured by the number of messages, comes from computing uniformization metric using discrete surface Ricci flow. It is \( O(-C(\log \varepsilon/\lambda)nd) \), where \( d \) is the average number of vertex degree since each vertex needs to exchange messages with its direct neighbors in each iteration.

In section “Simulations,” we apply the normalized communication cost to measure the consumed energy of a network to conduct a routing. We calculate the average number of messages exchanged in the network during one communication, whether it succeeds or not. The results show that the communication complexity of our proposed resilient greedy routing algorithm is \( O(n) \) for any pair of source and target nodes based on the precomputed virtual planar coordinates. It is well scalable to the size of a network.

**Simulations**

The proposed resilient routing algorithm can be applied for a wide range of large-scale sensor networks densely deployed on surfaces of coal mine tunnels for disaster prevention and rescue, or corridors of buildings for fire detection, or sewer or gas systems for monitoring underground pipelines. These surfaces generally have complex shapes and multiple handles, that is, genus numbers. To this end, we create several representative network models as shown in Figure 4 to carry out simulations. These network models have various genus numbers, ranging from one to four. Their sizes also vary from 500 to 1100 nodes.

In our simulations, we check two different failure models of a network: independent node failure, where each node in network has an equal and independent probability of failure during some time interval; and geographically correlated failure, where all nodes within a fixed radius or hop count fail simultaneously and the center of failure region is along the primary path. We assume the source and target nodes can locate anywhere in a network. We use three metrics to evaluate the performance of a resilient routing algorithm. The first one is the successful delivery ratio. It measures the probability of an algorithm successfully finding a path when the nodes or links of a network are not reliable. The successful delivery ratio indicates the resilience of a routing algorithm. The second metric is the average stretch factor. We divide the end-to-end hop count from source to target including the cost of forwarding messages and local detours, by the hop count of a shortest path without failure triggered. The third metric is the normalized communication cost. We calculate the average number of messages exchanged in a network during one communication regardless of the communication succeeds or not. This metric reflects the consumed energy of the network to conduct one (successful/failed) communication and the durability of the system. For those broadcasting-based methods, broadcasting dominates the communication cost of a newly established pair of source–target nodes. However, different source–target pairs sharing the same source node need only one-time broadcasting to build various paths from one source to different target nodes. We define source-to-pairs ratio as the reciprocal of the number of source–target pairs sharing the same source. To be fair, we set two different source-to-pairs ratios: \( 1:100 \) and \( 1:400 \) in our simulations.

Since the primary target of the proposed method is network resilience under independent node failure model, we compare our method with other well-known

![Figure 4. Network models with extracted triangular structure: (a) Network I, (b) Network II, (c) Network III, and (d) Network IV.](image)
resilient routing methods that choose local detour instead of disjoint alternative paths, including Directed Diffusion, Braided Multipath Routing, and Robust Cooperative Routing. Besides, SINUS method is the most recent greedy routing algorithm specifically designed for networks on high genus surfaces. We implement it with GPSR module to enhance its resilience against node failures and consider it as another comparison method.

We randomly choose 20,000 source–target pairs on each network model. Simulation results given in sections "Independent node failure" and "Geographically correlated failure" show that our method achieves the best performance under both independent node and geographically correlated failure models.

**Independent node failure**

The first row of Figure 5 compares the successful delivery ratio of different methods as a function of the probability of independent node failure for networks shown in Figure 4. As the probability increases, the successful delivery ratios of Directed Diffusion, Braided, and Robust Cooperative Routing methods decrease dramatically. When the probability of independent node failure is 15%, more than half of the sampled source–target pairs of these methods fail. However, SINUS with GPSR and our method perform much better. As the probability of independent node failure continuously increases, our method shows a bigger advantage. The reason is that SINUS generates a planar embedding with uneven density. Failure nodes can easily cause isolated regions around the source or target nodes such that it is impossible to find a successful detour with GPSR in this situation.

For the same set of network models, the second row of Figure 5 shows the average stretch factors of successful deliveries under different probabilities of independent node failure. As we can see, our method has a very low stretch factor, while SINUS with GPSR generates much longer paths in the network. We have discussed the reason in section "Introduction." SINUS cut the original surface network open along computed virtual boundaries to an annulus and then embed the annulus to plane. A pair of nodes with the shortest path crossing a virtual boundary in the original network needs to take a much longer detour to reach each other based on the computed virtual planar coordinates. We set the maximum probability of individual node failure for comparison of the average stretch factor as 10% – 20% only. If the failure probability is higher, the successfully delivered paths of comparison methods are quite short since longer paths are more likely to fail. To avoid such biased influence on the comparison of the average stretch factor, we compare different methods when there is still a significant number of successfully delivered paths.

We set the probability of independent node failure as 10% when measuring the communication cost. Figure 6(a) and (b) shows the normalized total/successful/failed communication costs under independent node failure model when the source-to-pairs ratio is 1 : 400.
and 1 : 100, respectively. As we can see, even with a smaller source-to-pairs ratio, Directed Diffusion, Braided, and Robust Cooperative Routing spend much more energy than our method. When the ratio is larger, the energy consumption of these methods is more dramatic. Compared with SINUS with GPSR, our method consumes around 1/3 less energy for successful delivery in Network I and II. The difference is bigger when a network has more nodes and higher genus number. Besides, our method wastes much less for unsuccessful communications. Therefore, our method benefits the system a lot, especially for those time-sensitive applications that involve frequent communications in a network.

**Geographically correlated failure**

The first row of Figure 7 compares the successful delivery ratios of different methods as a function of the radius of a failure region. As the radius of a failure region increases, the delivery ratios of Directed Diffusion, Braided, and Robust Cooperative Routing decrease dramatically. However, our method consistently performs the best with the highest successful delivery ratio, even when the failure region is huge. The only exception is the Network II model. When the radius of a failure region is larger than 3 in Network II model, both SINUS with GPSR and our method drop dramatically. The reason is that Network II model has narrow handles with fewer nodes covered compared with other network models. When the primary path of a pair of source and target nodes passes along a handle, a large failure region can easily break the handle such that a local detour cannot find a successful alternative. It is obvious that when the radius of a failure region becomes huge in geographically correlated failure model, a routing method with high resilience needs to take the global topology information of a network into consideration and choose a path with different homotopy type, that is, a path along different handles.

The second row of Figure 7 shows the average stretch factors of successful deliveries under different radii of a failure region. Since the successful delivery ratio of Robust Cooperative Routing method drops to zero when the radius of a failure region is 1, we remove it from the comparison. As the radius of a failure region becomes larger, the successful delivery ratio of Directed Diffusion and Braided methods drops dramatically, and the successfully delivered paths are short ones since the longer paths all fail. To avoid a biased influence on the computation of average stretch factor, we set the maximum radius of a failure region as 2 only. With the increase in the radius of a failure region, the stretch factor of a successful routing path gets...
longer for all methods. However, our method still has a consistently lower stretch factor.

We set the radius of a failure region as 1 when measuring the communication cost. Figure 8(a) and (b)

**Figure 7.** Geographically correlated failure: the first row gives the successful delivery ratios of different methods as a function of the radius of a failure region. The second row shows the average stretch factors under different failure radii. (a) Network I, (b) Network II, (c) Network III, and (d) Network IV.

**Figure 8.** Geographically correlated failure: the failure radius is set as 1. (a) The normalized total/successful/failed communication costs when the source-to-pairs ratio is 1:400. (b) The costs when the source-to-pairs ratio is 1:100.
shows the normalized total/successful/failed communication cost under geographically correlated failure model when the source-to-pairs ratio is 1 : 400 and 1 : 100, respectively. Since the successful delivery ratio of Robust Cooperative Routing method drops to zero when the radius of a failure region is 1, we remove it from the comparison. The results are similar to those under the independent node failure model. Our method consistently performs the best among all.

Conclusion
We have proposed a cut-free resilient greedy routing scheme for GPS-free large-scale wireless sensor networks deployed on surfaces of complex-connected 3D settings. It is designed to be resilient to node or link failures especially under random node or link failure model. Simulation results have shown that the proposed scheme achieves a higher successful delivery ratio, a lower average stretch factor, and a lower normalized communication cost compared with other resilient routing methods.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors are partially supported by NSF CNS-1320931.

ORCID iD
Miao Jin https://orcid.org/0000-0003-0610-6351

References
1. Akyildiz IF, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks. *IEEE Commun Mag* 2002; 40(8): 102–114.
2. Allred J, Hasan AB, Panichsakul S, et al. SensorFlock: an airborne wireless sensor network of micro-air vehicles. In: *Proceedings of the 5th international conference on embedded networked sensor systems (SenSys)* (ed S Jha), Sydney, NSW, Australia, 6–9 November 2007, pp.117–129. New York: ACM. http://dblp.uni-trier.de/db/conf/sensys/sensys2007.html#AllredHPGHHLM07
3. Xia S, Ding N, Jin M, et al. Medial axis construction and applications in 3D wireless sensor networks. In: *Proceedings of the 2013 IEEE INFOCOM*, Turin, 14–19 April 2013, pp.305–309. New York: IEEE.
4. Cui JH, Kong J, Gerla M, et al. The challenges of building mobile underwater wireless networks for aquatic applications. *IEEE Network* 2006; 20(3): 12–18.
5. Zhao MC, Lei J, Wu MY, et al. Surface coverage in wireless sensor networks. In: *Proceedings of the 2009 IEEE INFOCOM*, Rio de Janeiro, Brazil, 19–25 April 2009, pp.109–117. New York: IEEE.
6. Zhao Y, Wu H, Jin M, et al. Localization in 3D surface sensor networks: challenges and solutions. In: *Proceedings of the 2012 IEEE INFOCOM*, Orlando, FL, 25–30 March 2012, pp.55–63. New York: IEEE.
7. Bose P, Morin P, Stojmenovic I, et al. Routing with guaranteed delivery in ad hoc wireless networks. In: *Proceedings of the third workshop discrete algorithms and methods for mobile computing and communications*, Seattle, WA, 20 August 1999, pp.48–55. New York: ACM.
8. Karp B and Kung H. GPR: greedy perimeter stateless routing for wireless networks. In: *Proceedings of the MobiCom 2000*, Boston, MA, 6–11 August 2000, pp.243–254. New York: ACM.
9. Kranakis E, Singh H and Urrutia J. Compass routing on geometric networks. In: *Proceedings of the Canadian conference on computational geometry (CCCG)*, Vancouver, BC, Canada, 15–18 August 1999, pp.51–54.
10. Kuhn F, Wattenhofer R, Zhang Y, et al. Geometric ad-hoc routing: theory and practice. In: *Proceedings of the 22nd ACM symposium on the principles of distributed computing*, Boston, MA, 13–16 July 2003, pp.63–72. New York: ACM.
11. Kuhn F, Wattenhofer R and Zollinger A. Worst-case optimal and average-case efficient geometric ad-hoc routing. In: *Proceedings of the MobiHoc*, Annapolis, MD, 1–3 June 2003, pp.267–278. New York: ACM.
12. Mitra BLS and Liskov B. Path vector face routing: geographic routing with local face information. In: *Proceedings of the 13th IEEE international conference on network protocols (ICNP’05)*, Boston, MA, 6–9 November 2005, pp.147–158. New York: IEEE.
13. Frey H and Stojmenovic I. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In: *Proceedings of the MobiCom 2006*, Los Angeles, CA, 25 September 2006, pp.390–401. New York: ACM.
14. Tan G, Bertier M and Kermarrec AM. Visibility-graph-based shortest-path geographic routing in sensor networks. In: *Proceedings of the 2009 IEEE INFOCOM*, Rio de Janeiro, Brazil, 19–25 April 2009, pp.1719–2728. New York: IEEE.
15. Durocher S, Kirkpatrick D and Narayanan L. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. Berlin; Heidelberg: Springer, 2008.
16. Wang Y, Yi CW and Li F. Efficient geometric routing in three dimensional wireless networks. Berlin; Heidelberg: Springer, 2008.
17. Liu C and Wu J. Efficient geometric routing in three dimensional ad hoc networks. In: *Proceedings of the 2009 IEEE INFOCOM*, Rio de Janeiro, Brazil, 19–25 April 2009, pp.2751–2755. New York: IEEE.
18. Xia S, Wu H and Jin M. Trace-routing in 3D wireless sensor networks: a deterministic approach with constant overhead. In: *Proceedings of the 15th ACM international symposium on mobile ad hoc networking and computing (MobiHoc ’14)*, Philadelphia, PA, 11–14 August 2014, pp.357–366. New York: ACM.
19. Yu X, Yin X, Han W, et al. Scalable routing in 3D high genus sensor networks using graph embedding. In: Proceedings of the IEEE INFOCOM 2012, Orlando, FL, 25–30 March 2012, pp.2681–2685. New York: IEEE.

20. Kim S, Pakzad S, Culler D, et al. Health monitoring of civil infrastructures using wireless sensor networks. In: Proceedings of the 2007 6th international symposium on information processing in sensor networks, Cambridge, MA, 25–27 April 2007, pp.254–263. New York: ACM.

21. Shen Y, Yang P, Zhang P, et al. Development of a multi-type wireless sensor network for the large-scale structure of the national stadium in China. Int J Distrib Sens N 2013; 9: 709724.

22. Yu T, Jiang H, Tang G, et al. Sinus: a scalable and distributed routing algorithm with guaranteed delivery for WSNS on high genus 3D surfaces. In: Proceedings of the 2013 IEEE INFOCOM, Turin, 14–19 April 2013, pp.2175–2183. New York: IEEE.

23. Ganesan D, Govindan R, Shenker S, et al. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM SIGMOBILE Mob Comput Commun Rev 2001; 5(4): 11–25.

24. Intanagonwiwat C, Govindan R and Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceeding of the MobiCom 2000, Boston, MA, 6–11 August 2000, pp.56–67. New York: ACM.

25. Shand M and Bryant S. IP fast reroute framework, RFC, vol. 5714, 2010.

26. Atlas A and Zinin A. Basic specification for IP fast reroute: loop-free alternates, 2008, http://www.hjp.at/doc/rfc/rfc5286.html

27. Reichert C, Glickmann Y and Magedanz T. Two routing algorithms for failure protection in IP networks. In: Proceedings of the 10th IEEE symposium on computers and communications (ISCC’05), Murcia, 27–30 June 2005, pp.97–102. New York: IEEE.

28. Ray S, Guérin R, Kwong KW, et al. Always acyclic distributed path computation. IEEE ACM T Network 2010; 18(1): 307–319.

29. Ohara Y, Imahori S and Meter RV. MARA: maximum alternative routing algorithm. In: Proceedings of the 2009 IEEE INFOCOM, Rio de Janeiro, Brazil, 19–25 April 2009, pp.298–306. New York: IEEE.

30. Lou W. An efficient N-to-1 multipath routing protocol in wireless sensor networks. In: Proceedings of the IEEE international conference on mobile ad-hoc and sensor systems conference, Washington, DC, 7 November 2005, pp.664–672. New York: IEEE.

31. Motiwala M, Elmore M, Feamster N, et al. Path splicing. ACM SIGMOBILE Mob Comput Commun Rev 2008; 38(4): 27–38.

32. Gao J and Zhou D. Resilient and low stretch routing through embedding into tree metrics. In: Proceedings of the WADS, New York, 15–17 August 2011, pp.438–450. Berlin: Springer.

33. Zeng W, Sarkar R, Luo F, et al. Resilient routing for sensor networks using hyperbolic embedding of universal covering space. In: Proceeding of the 2010 IEEE INFOCOM, San Diego, CA, 14–19 March 2010, pp.1694–1702. New York: IEEE.

34. Liu X, Liu A, Wang T, et al. Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks. J Parallel Distri Com 2020; 135: 140–155.

35. Schlag W. A course in complex analysis and Riemann surfaces. Providence, RI: American Mathematical Society, 2014.

36. Jin M, Kim J, Luo F, et al. Discrete surface Ricci flow. IEEE T Vis Comput Gr 2008; 14(5): 1030–1043.

37. Zhou H, Wu H, Xia S, et al. A distributed triangulation algorithm for wireless sensor networks on 2D and 3D surface. In: Proceedings of the 2011 IEEE INFOCOM, Shanghai, China, 10–15 April 2011, pp.1053–1061. New York: IEEE.

38. Huang X, Zhai H and Fang Y. Robust cooperative routing protocol in mobile wireless sensor networks. IEEE T Wirel Commun 2008; 7(12): 5278–5285.

Appendix I

Hyperbolic geometry

The hyperbolic model we used in this article is the Poincaré disk. It is a unit disk on the complex plane, with Riemannian metric

\[ ds^2 = \frac{4dwdw}{1 - w^2} \]

in the Poincaré disk, the geodesics are circle arcs which are orthogonal to the unit circle.

A hyperbolic circle can be easily converted to a Euclidean circle because a hyperbolic circle \((c, r)\) coincides with a Euclidean circle \((C, R)\) with

\[ C = \frac{2 - 2\mu^2}{1 - \mu^2 |c|^2} c, R^2 = |C|^2 - \frac{|c|^2 - \mu^2}{1 - \mu^2 |c|^2} \]

(3)

where \(\mu = ((c^2 - 1)/(c^2 + 1))\).