Building a coverage hole-free communication tree

Anais Vergne, Laurent Decreusefond, Philippe Martins

Building a coverage hole-free communication tree

Anaïs Vergne, Laurent Decreusefond, and Philippe Martins

LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

***Abstract*—Wireless networks are present everywhere but their management can be tricky since their coverage may contain holes even if the network is fully connected. In this paper we propose an algorithm that can build a communication tree between nodes of a wireless network with guarantee that there is no coverage hole in the tree. We use simplicial homology to compute mathematically the coverage, and Prim's algorithm principle to build the communication tree. Some simulation results are given to study the performance of the algorithm and compare different metrics. In the end, we show that our algorithm can be used to create coverage hole-free communication groups with a limited number of hops.**

I. Introduction

Wireless networks are everyday more present in our lives: WiFi is the main internet access in our homes, cellular systems such as 4G and soon 5G provide its access everywhere else. Moreover with IoT, every object in our kitchen or in our bathroom will in the near future be connected as well. When managing a network, it is often useful to build a communication tree of the network nodes in order to transmit messages to every node efficiently. The spanning tree is the answer to that problem: the fact that it is a tree guarantees that there are no superficial links, and spanning means that all connected nodes are included. Several well-known algorithms allow to find the minimum spanning tree in a graph according to a given metric. We can cite Kruskal's algorithm [1], Prim's algorithm [2], or Borůvka's algorithm [3].

However, the quality of service of wireless network is primarily providing access to its users, in other terms provide coverage. Therefore, a communication tree with coverage holes could be pointless. Meanwhile, deciding whether a set of base stations does cover a whole domain is not that easy when the network is irregularly deployed, as it is the case for cellular networks see [4] or [5]. Based on the geometrical data of the network, we can build a combinatorial object to represent it: the simplicial complex. Basically a simplicial complex is the generalization of the concept of graph, it is made of k-simplices where 0-simplices are vertices, 1-simplices are edges, 2-simplices are triangles, 3-simplices are tetrahedron and so on. In particular, geometrical simplicial complexes such as the Čech complex and the Vietoris-Rips complex allows to represent exactly and approximately the coverage of the union of the coverage disks as stated in the Nerve lemma in [6]. Then algebraic topology, [7], is the mathematical tool used to compute the number of connected components, of coverage holes, and of 3D voids, that are the so-called Betti numbers of the simplicial complex representing the network, as detailed in [8].

In this article, we introduce an algorithm that can build a communication tree between the connected nodes of a wireless network with guarantee that there is no coverage hole in the tree. First, we use simplicial homology to represent the network, and algebraic topology to compute its coverage. Then we modify Prim's algorithm in order to only select vertices that do not create coverage holes. We provide simulation results to measure the performance of our algorithm in terms of number of rejected nodes, and surface of covered area. We then compare different metrics for the weight of edges, and find that the height metric, from the simplicial complex representation, provides results with the shortest branches both in terms of hops and length without losing any covered area. Finally, we extend our algorithm to build coverage hole-free communication trees in larger networks.

This is the first algorithm of this type that we know of. Finding a spanning tree in a graph is an old and classic problem [1]–[3]. But the use of simplicial homology for wireless networks is just about a decade old [6]. Since, the computational time to obtain the Betti numbers can explode with the size of the simplicial complex, many works focus on faster ways to compute them, for instance in a decentralized way [9], using persistent homology [10], thanks to chain complexes reduction [11] , or with witness complexes reduction [12]. Simplicial complexes reduction can also be used for coverage hole detection [13] and energy efficiency in cellular networks [14].

In Section II, we provide the mathematical background. Then in Section III, we give the algorithm for building a coverage hole-free communication tree along with simulation results in Section IV. Finally we propose an extension to the building of communication groups in Section V.

II. Coverage of a network

A. Simplicial homology and algebraic topology

Considering a set of points representing network nodes, the first idea to apprehend the topology of the network would be to look at the neighbors graph: if the distance between two points is less than a given parameter then an edge is drawn between them. However this representation is too limited to transpose the network's topology. First, only 2-by-2 relationships are represented in the graph, there is no way to grasp interactions between three or more nodes. Moreover, there is no concept of coverage in a graph. That is why we are interested in more complex objects.

Indeed, graphs can be generalized to more generic combinatorial objects known as simplicial complexes. While graphs model binary relations, simplicial complexes can represent higher order relations. A simplicial complex is thus a combinatorial object made up of vertices, edges, triangles, tetrahedra, and their n-dimensional counterparts. Given a set of vertices X and an integer k, a k-simplex is an unordered subset of $k+1$ vertices $\{x_0, \dots, x_k\}$ where $x_i \in X, \forall i \in \{0, \dots, k\}$ and $x_i \neq x_j$ for all $i \neq j$. Thus, a 0-simplex is a vertex, a 1-simplex an edge, a 2-simplex a triangle, a 3-simplex a tetrahedron, etc. See Fig. 1 for instance.

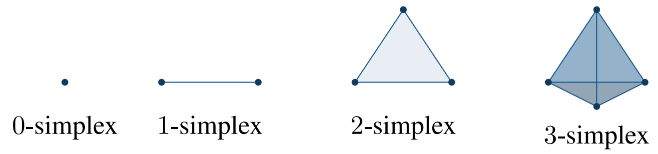

Fig. 1. Examples of k-simplices.

Any subset of vertices included in the set of the $k+1$ vertices of a k-simplex is a face of this k-simplex. A k-face is then a face that is a k-simplex. The inverse notion of face is coface. An abstract simplicial complex is a set of simplices such that all faces of these simplices are also in the set of simplices.

In this article, we are interested in representing the topology of a wireless network, we introduce the following abstract simplicial complex:

Definition 1 (Vietoris-Rips complex)**.** *Let (X, d) be a metric space, ω a finite set of points in X, and r a real positive number. The Vietoris-Rips complex of parameter r of ω, $\mathcal{R}_r(\omega)$, is the abstract simplicial complex whose k-simplices correspond to the unordered $(k+1)$-tuples of vertices in ω which are pairwise within distance less than r of each other.*

The Vietoris-Rips complex is easy to build since it is only based on the neighbors graph information. Moreover it provides an approximation of the exact topology of the network, which is given by the Čech complex (see the Nerve lemma in [6]). This approximation is quite good: in the case of a random uncorrelated deployment with network nodes deployed according to a Poisson point process the error is less than 0.06% in the computation of the covered area [15]. An example of a Vietoris-Rips complex representing a wireless network can be seen in Fig. 3.

Given an abstract simplicial complex, one can define an orientation on the simplices by defining an order on the vertices, where a change in the orientation, that is a swap between two vertices, corresponds to a change in the sign. Then let us define the vector spaces of the k-simplices of a simplicial complex, and the associated boundary maps:

Definition 2. *Let S be an abstract simplicial complex.*

For any integer k, $\mathscr{C}_k(S)$ is the vector space spanned by the set of oriented k-simplices of S.

Definition 3. *Let S be an abstract simplicial complex and $\mathscr{C}_k(S)$ the vector space of its k-simplices for any k integer.*

The boundary map ∂_k is defined as the linear transformation $\partial_k : \mathscr{C}_k(S) \to \mathscr{C}_{k-1}(S)$ which acts on the basis elements $[x_0, \dots, x_k]$ of $\mathscr{C}_k(S)$ via:

$$\partial_k[x_0, \dots, x_k] = \sum_{i=0}^{k} (-1)^i [x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_k].$$

For example, for a 2-simplex we have:

$$\partial_2([x_0, x_1, x_2]) = [x_1, x_2] - [x_0, x_2] + [x_0, x_1]$$

As its name indicates, the boundary map applied to a linear combination of simplices gives its boundary. The boundary of a boundary is the null application. Therefore the following theorem can be easily demonstrated (see [7] for instance):

Theorem 1. *For any k integer, $\partial_k \circ \partial_{k+1} = 0$.*

Let S be an abstract simplicial complex. Then we can denote the k-th boundary group of S as $B_k(S) = \operatorname{im} \partial_{k+1}$, and the k-th cycle group of S as $Z_k(S) = \ker \partial_k$. We have $B_k(S) \subset Z_k(S)$. We are now able to define the k-th homology group and its dimension:

Definition 4. *The k-th homology group of an abstract simplicial complex S is the quotient vector space:*

$$H_k(S) = \frac{Z_k(S)}{B_k(S)}.$$

The k-th Betti number of the abstract simplicial complex S is:

$$\beta_k(S) = \dim H_k(S).$$

According to its definition, the k-th Betti number counts the number of cycles of k-simplices that are not boundaries of $(k+1)$-simplices, that are the k-th dimensional holes. In small dimensions, they have a geometrical interpretation:

- β_0 is the number of connected components,
- β_1 is the number of coverage holes,
- β_2 is the number of 3D-voids.

For any $k \geq d$ where d is the dimension, we have $\beta_k = 0$.

We can now define the Euler characteristic of an abstract simplicial complex:

Definition 5. *The Euler characteristic an abstract simplicial complex S is the alternated sum of its Betti numbers:*

$$\chi(S) = \sum_{k=0}^{d-1} (-1)^k \beta_k.$$

But it can also be defined as:

$$\chi(S) = \sum_{k=0}^{\infty} (-1)^k s_k,$$

where s_k is the number of k-simplices in S.

For further reading on algebraic topology, see [7].

B. Percolation and coverage holes

We are now considering coverage in light of percolation. Indeed when a network is regularly deployed, think about the hexagonal model for instance, if the network is connected then there is no coverage hole. However, in real-life deployments, network cells are not hexagons. When considering all the frequency bands owned by an operator, network nodes are actually more similar to a Poisson point process [5]. In this case, percolation does not guarantee coverage.

In [16], the authors studied the moments of the number of k-simplices for a Vietoris-Rips complex based on a set of points drawn according to a Poisson point process with the uniform norm on the d-dimensional torus. We are especially interested in the mean of the Euler characteristic:

$$\mathbf{E}[\chi(S)] = -\left(\frac{a}{r}\right)^d \sum_{k=0}^{\infty} \frac{(-\lambda r^d)^k k^d}{k!},$$

where a is the side of the torus, r is the Vietoris-Rips distance for which two points are in the same simplex, d is the dimension, and λ is the intensity of the Poisson point process.

In two dimensions, this formula can be simplified:

$$\mathbf{E}[\chi(S)] = a^2\lambda(1 - \lambda r^2)e^{-\lambda r^2}.$$

We plot $\mathbf{E}[\chi(S)]$ in function of λ for a=10 and r=1 in Fig. 2.

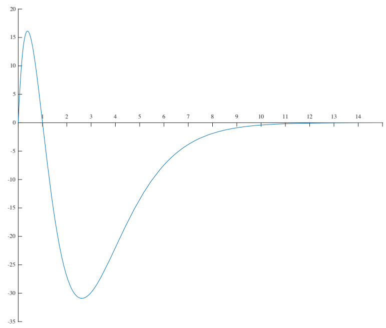

Fig. 2. $\mathbf{E}[\chi(S)]$ for $\lambda = 0...15$.

However, in two dimensions, $\chi = \beta_0 - \beta_1$. Since the Betti numbers are positive, we can interpret the previous plot:

- When λ is smaller than 0.5, there are multiple connected components of just some points each, that are not large enough to create coverage holes. Then β_0 grows with λ and β_1 is close to 0.
- Around $\lambda = 0.5$, χ attains a maximum: percolation occurs. The number of connected components β_0 starts decreasing. On the other hand, coverage holes appear: β_1 begins to increase.
- When $\lambda = 1$, χ becomes negative, that means that the number of coverage holes β_1 outnumbers the number of connected components β_0. β_1 goes on increasing while β_0 continues decreasing.
- When λ is greater than 3, percolation has occurred: there is enough points to have only one connected component, and new points begin to fill coverage holes. That is to say that β_0 is close to 1 and β_1 decreases.
- Finally when λ is large enough, there is one unique component and no coverage hole: $\beta_0 = 1$ and $\beta_1 = 0$.

From this, we can see that when network nodes are deployed randomly following a Poisson point process, percolation occurs before full coverage happens, and the network stays in this regime for many values of λ. That means that while the network is connected and every node can communicate with each other through a path of nodes, there still exists regions that are uncovered. Therefore when simply building a spanning tree, one is not sure not to include some coverage holes. That is why we propose an algorithm for the construction of a coverage hole-free communication tree.

III. Coverage hole-free tree

A. Principle

A spanning tree in a connected graph with n vertices is a connected subgraph of it which includes all of the n vertices and has exactly $n - 1$ edges. Finding a minimum or maximum spanning tree in a graph is a well-known problem in computer science that is resolved by well-known algorithms such as Kruskal's algorithm, Borůvka's algorithm, and Prim's algorithm. The minimum or maximum property is based on a weight associated with each edge. It is possible to use any interesting metric: minimum distance, maximum distance, or maximum redundancy for instance.

We are especially interested in Prim's algorithm since in this greedy algorithm the spanning tree grows one edge at a time while staying always connected [2]. Indeed, at the beginning of the algorithm, the tree is reduced to one vertex, chosen randomly. Then at each step, the minimum-weight (or maximum-weight) edge among all the edges that join a vertex of the tree to a vertex outside the tree is added to the tree. The algorithm stops when all vertices are in the tree.

To build a coverage hole-free tree, our idea is simply to modify the Prim's algorithm in order to check coverage at each step thanks to simplicial homology, and to reject the edge, and consequently its extremity vertex, if a coverage hole is created. Therefore, at the end, a tree free of coverage holes is obtained.

B. Algorithm

First, our algorithm computes the Vietoris-Rips complex based on the set of vertices and the distance parameter given in input. It is important to note that we only need to compute the complex up to the 2-simplices since we are only interested in the computation of β_0 and β_1. Then the weights of the edges are computed according to a given metric.

After that, the tree T is created with only the root, which is uniformly drawn, and no edge. A set of potential edges E_{test} with one extremity in the tree T and the other outside T is defined. Then, while there are vertices outside the tree and there are potential edges left, a potential edge of minimum weight is added if it does not create a coverage hole. If it does, the edge is removed from the set of potential edges. We give in Algorithm 1 the pseudo-code.

Algorithm 1 Coverage hole-free tree building algorithm.

Require: set V of n vertices, connection distance r.
  Computation of $S = \mathcal{R}_r(V)$
  $E := \{$1-simplices of $S\}$ *%Edges of S*
  Computation of the weights $\{w(e), e \in E\}$;
  Draw uniformly a vertex $r \in V$ to be the root
  $T := \{r\}$ *%Tree*
  $E_T := \emptyset$ *%Edges of the tree*
  $E_{\text{test}} := (T, V \setminus T)$ *%Potential edges*
  while $|T| < n$ and $|E_{\text{test}}| > 0$ **do**
    Take e such that $w(e) = \min\{w(f), f \in E_{\text{test}}\}$
    Let x be the extremity of e in $V \setminus T$
    Computation of $S_T = \mathcal{R}_r(T \cup \{x\})$ and $\beta_1(S_T)$
    if $\beta_1(S_T) \neq 0$ **then**
      $E_{\text{test}} = E_{\text{test}} \setminus \{e\}$
    else
      $T := T \cup \{x\}$
      $E_T := E_T \cup \{e\}$
    end if
  end while
    return T, E_T

We can see on the first two figures of Fig. 3 a wireless network with two coverage holes and its Vietoris-Rips complex. The result of our algorithm can be seen in the two following figures with the tree highlighted in red. There are 2 vertices (in blue) that are not in the tree in order to avoid coverage holes. We can verify that the coverage of the tree is hole-free.

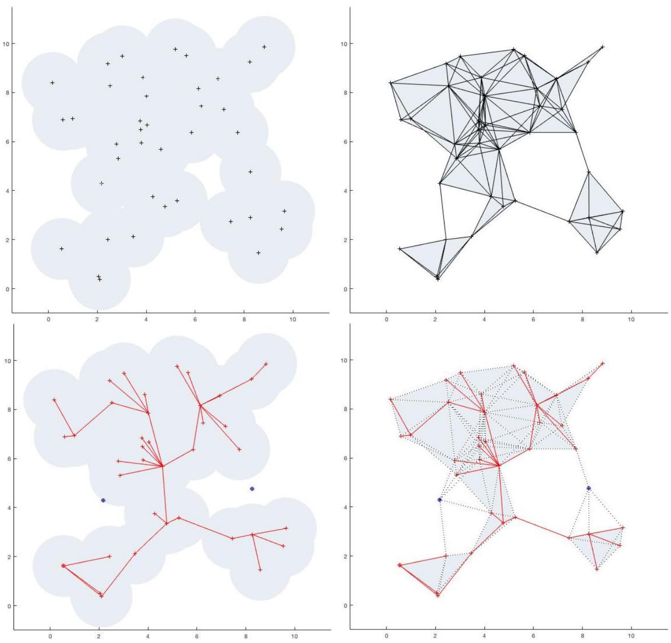

Fig. 3. A coverage hole-free communication tree in a wireless network

IV. Simulation results

A. Percentage of rejected vertices

For a start, we look at the percentage of vertices that are not in the final tree at the end of the algorithm. Vertices can be absent from the final tree for two reasons. First, if vertices are not in the same connected component as the root vertex, then they are unreachable. Second, if vertices are in the same connected component as the root vertex but create a coverage hole, they are then rejected by the test on β_1. We provide in Fig. 4 a bar chart on which are represented the percentage of unreachable, rejected, and tree vertices for different values for the number of initial vertices n, on average on 1000 simulations for each scenario. The chosen weight metric is the minimum distance, and the simulation is made on a square of side $a = 10$ with a connection distance of $r = a/4$.

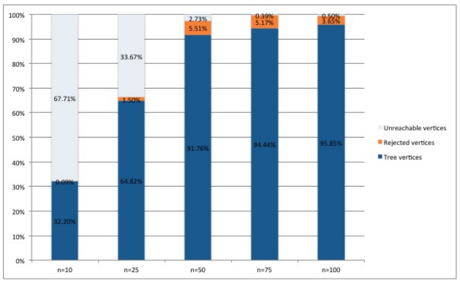

Fig. 4. Percentage of unreachable, rejected, and tree vertices.

We can see that when there is no percolation, very few vertices are rejected by the algorithm. But when percolation has occurred, that is when there are almost no unreachable vertices, the percentage of rejected vertices is below 6% and decreases when the number of initial vertices grows.

B. Percentage of covered area

Then we are interested in the loss of coverage that is induced by the reject of some vertices. To do that we compare the area covered before the algorithm runs with all vertices, and the area covered by only the tree vertices. The bar chart in Fig. 5 shows the results for $n = 75$ and $n = 100$ vertices when percolation has occurred there are almost no unreachable nodes. The configuration is the same as before otherwise.

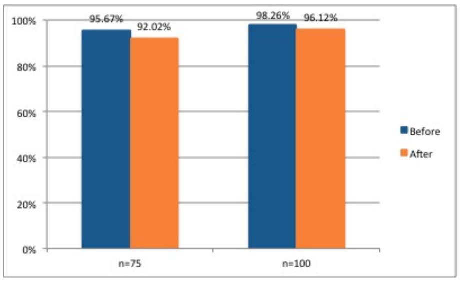

Fig. 5. Percentage of covered area before and after the algorithm.

We can see that the loss of coverage represents only between 2% and 3% of the covered area.

C. *Weight metric influence*

Finally, we look into the influence of the chosen weight metric on the branches on the tree. We compared three weight metric: minimum distance, maximum distance, and maximum height. The height of an edge is defined as the size of the largest simplex it is part of. It can be interpreted as a redundancy parameter. To evaluate the branches, we looked at the mean number of hops, the maximum number of hops, the mean length and the maximum length. The results in Fig. 6 are given for $n = 75$ vertices and the same parameters as before.

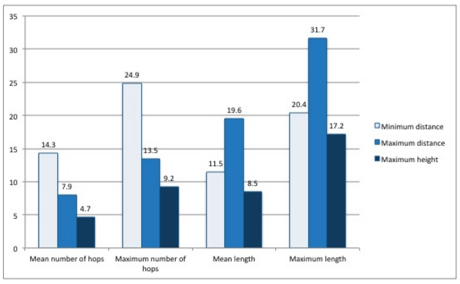

Fig. 6. Weight metric influence on the branches of the tree.

We can see that the maximum height minimizes the size of the branches both in number of hops and in total length. And logically, the minimum distance maximizes the number of hops, while the maximum distance maximizes the length of the branches.

Otherwise the weight metric does not change the size of the covered area of the final tree as we can see in Fig. 7 whatever the number n of initial vertices. Therefore, since the size of the covered area is not impacted, the height seems to be a good metric because long branches are synonyms of delays and a great number of hops increases the error probability.

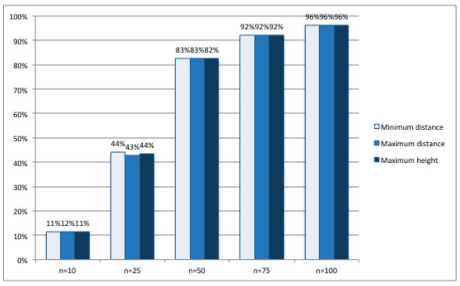

Fig. 7. Weight metric influence on the covered surface.

V. Communication groups in a network

Our coverage hole-free communication tree building algorithm can be extended to create communication groups in a wireless network. Indeed a wireless network operator would rather choose several small communication trees rather than one giant communication tree. In order to do that with our algorithm, it suffices to limit the number of hops a branch of the tree can have. Then as long as there are still nodes in the network not in a communication tree, a new root is randomly chosen among them and a new tree is created.

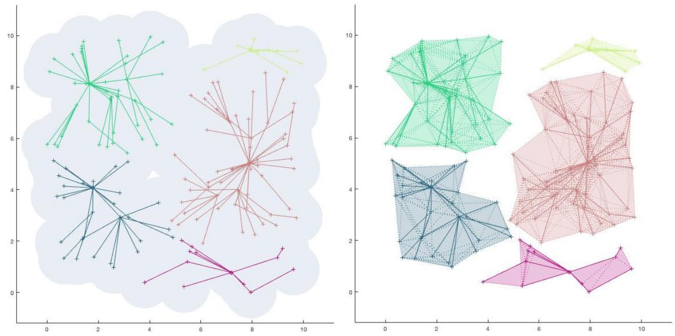

Fig. 8. Forest of coverage hole-free communication trees.

At the end, we obtain a forest of small coverage hole-free communication trees with branches no longer than a given number of hops as we can see in Fig. 8. The limit number of hops is set to 3. Each tree has a different color, their roots, which serve as communication hubs, are circled.

References

[1] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[2] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[3] O. Borůvka, "O jistém problému minimálním (About a certain minimal problem)," *Práce mor. přírodověd. spol. v Brně III*, vol. 3, pp. 37–58, 1926.

[4] N. Deng, W. Zhou, and M. Haenggi, "The Ginibre Point Process as a Model for Wireless Networks With Repulsion," *IEEE Trans. Wireless Communications*, vol. 14, no. 1, pp. 107–121, 2015.

[5] J. Gomez, A. Vasseur, A. Vergne, P. Martins, L. Decreusefond, and W. Chen, "A Case Study on Regularity in Cellular Network Deployment," *Wireless Communications Letters, IEEE*, vol. 4, no. 4, pp. 421–424, Aug. 2015.

[6] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005.

[7] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.

[8] V. de Silva and R. Ghrist, "Coordinate-free Coverage in Sensor Networks with Controlled Boundaries Via Homology," *International Journal of Robotics Research*, vol. 25, Dec. 2006.

[9] A. Muhammad and A. Jadbabaie, "Decentralized Computation of Homology Groups in Networks by Gossip," in *American Control Conference, 2007. ACC '07*, Jul. 2007, pp. 3438 –3443.

[10] A. Zomorodian and G. Carlsson, "Computing Persistent Homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005, 10.1007/s00454-004-1146-y.

[11] T. Kaczyński, M. Mrozek, and M. Ślusarek, "Homology computation by reduction of chain complexes," *Computers & Mathematics with Applications. An International Journal*, vol. 35, no. 4, pp. 59–70, 1998.

[12] V. de Silva and G. Carlsson, "Topological estimation using witness complexes," *IEEE Symposium on Point-based Graphic*, pp. 157–166, 2004.

[13] F. Yan, A. Vergne, P. Martins, and L. Decreusefond, "Homology-Based Distributed Coverage Hole Detection in Wireless Sensor Networks," *Networking, IEEE/ACM Transactions on*, vol. 23, no. 6, pp. 1705–1718, Dec. 2015.

[14] A. Vergne, L. Decreusefond, and P. Martins, "Simplicial Homology for Future Cellular Networks," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 8, pp. 1712–1725, Aug. 2015.

[15] F. Yan, P. Martins, and L. Decreusefond, "Accuracy of Homology based Approaches for Coverage Hole Detection in Wireless Sensor Networks," in *Proc. IEEE ICC*, Ottawa, Canada, Jun. 2012.

[16] L. Decreusefond, E. Ferraz, H. Randriambololona, and A. Vergne, "Simplicial homology of random configurations," *Advances in Applied Probability*, vol. 46, no. 2, pp. 325–347, 2014.