



HAL
open science

Homology based algorithm for disaster recovery in wireless networks

Anaïs Vergne, Ian Flint, Laurent Decreasefond, Philippe Martins

► **To cite this version:**

Anaïs Vergne, Ian Flint, Laurent Decreasefond, Philippe Martins. Homology based algorithm for disaster recovery in wireless networks. 2013. hal-00800520v1

HAL Id: hal-00800520

<https://hal.science/hal-00800520v1>

Preprint submitted on 13 Mar 2013 (v1), last revised 21 Mar 2014 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Homology based algorithm for disaster recovery in wireless networks

A. Vergne, I. Flint, L. Decreusefond and P. Martins

Institut Telecom, TELECOM ParisTech, LTCI

Paris, France

Email: avergne, flint, decrease, martins@telecom-paristech.fr

Abstract—In this paper, we provide a homology based algorithm for disaster recovery of wireless networks. We consider a damaged wireless network presenting coverage holes that we need to restore by patching the holes. We propose an algorithm which adds supernumerary vertices to cover the entire area, then run an improved reduction algorithm to reach an optimal result with a minimum number of added vertices. The algorithm is homology based, using simplicial homology representation to compute the coverage and abstract simplicial complex structure to know redundancy vertices information. We present various original methods for the addition of vertices of which we compare the performances between each other and with known methods.

I. INTRODUCTION

Wireless networks are present everywhere, must it be sensor networks or cellular networks. Fields where wireless sensor networks can be used range from battlefield surveillance to target enumeration in agriculture and include environmental monitoring. In most applications, the topology of the network, such as its connectivity and its coverage, is a critical factor. Cellular networks are used for radio communication, where coverage is also a critical factor. Indeed the covered area is often the main characteristic of a cellular network. However such networks are not necessary built with redundancy and can be sensitive to disasters.

In case of disaster, a wireless network can be damaged: coverage holes can appear, connectivity can be lost. Paradoxically, reliable and efficient communication is especially needed in such situations. Therefore solutions for damage recovery for the coverage of wireless networks are much needed. Extensive research on the coverage problem in wireless networks exists: we can cite location-based [6] and range-based [17] methods. However, connectivity based schemes seem of greater interest since they do not need exact location or distance information. In [8], the authors introduced the Vietoris-Rips complex, based on the proximity graph of a wireless network, as a tool to compute its topology. Coverage computation via simplicial homology comes down to linear algebra computations. It is for instance used in [4] and [13] as a tool for a network operator to evaluate the quality of its network. We can also mention the distributed algorithms of [16] that detects coverage holes.

In this paper, we present a homology based algorithm for disaster recovery of wireless networks. We represent wireless networks with Čech simplicial complexes characterizing its coverage. Given a set of vertices and their communication

radius, our algorithm first adds supernumerary vertices in order to patch every existing coverage hole, then runs an improved version of the reduction algorithm presented in [15] in order to reach an optimal result with a minimum number of added vertices. Various methods for the addition of vertices are proposed. We compare classic methods with new and original methods.

This is the first algorithm that we know of that adds too many vertices then remove them to reach an optimal result instead of adding the exact needed number of vertices. This allows flexibility in the choice of the new vertices positions, which can be useful when running the algorithm in a real life scenario. The disaster recovery problem can be viewed as a set cover problem: where the universe is the area to be covered and the subsets are the balls of communication range, then the question is to find the optimal set of subsets that cover the universe. A greedy algorithm can solve this problem [3]: a grid of potential vertices is unfold on the area, the potential vertex the furthest from the existing vertices is added, and so on until the furthest potential vertex is in communication range. ϵ -nets [10] provide also an algorithm for the set cover problem via a sampling of the universe. We can also cite landmark-based routing, seen in [7] and [1], which, using furthest point sampling, provides a set of communication nodes for routing that we can interpret as a minimal set of vertices to cover an area. On top of flexibility, the originality of our algorithm lies in the choice of the vertices addition methods we suggest. Indeed, we compare classic methods (grid, uniform positioning) with new methods never applied to wireless networks. Determinantal point processes are objects from stochastic geometry in which there is repulsion between points, therefore they have the ability to locate coverage holes where there are no close points. We suggest also the use of the Sobol sequence, a statistical tool built to provide uniform coverage of the unit square.

The remainder of this paper is structured as follows: first we present the main idea of our algorithm, using some definitions from simplicial homology in the second section. Then in the third section, we propose different vertices addition method. The reduction algorithm is presented in section 4. Finally we provide performance comparison between our methods and known methods in the last section. We conclude in the last section.

II. MAIN IDEA

A. Simplicial homology

First we need to remind some definitions from simplicial homology for better understanding of our representation.

Graphs can be generalized to more generic combinatorial objects known as simplicial complexes. While graphs model binary relations, simplicial complexes represent higher order relations. A simplicial complex is a combinatorial object made up of vertices, edges, triangles, tetrahedra, and their n -dimensional counterparts. Given a set of vertices V and an integer k , a k -simplex is an unordered subset of $k+1$ vertices $[v_0, v_1, \dots, v_k]$ where $v_i \in V$ and $v_i \neq v_j$ 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.

Any subset of vertices included in the set of the $k+1$ vertices of a k -simplex is a face of this k -simplex. Thus, a k -simplex has exactly $k+1$ $(k-1)$ -faces, which are $(k-1)$ -simplices. For example, a tetrahedron has four 2-faces which are triangles. A simplicial complex is a collection of simplices which is closed with respect to the inclusion of faces, i.e. all faces of a simplex are in the set of simplices, and whenever two simplices intersect, they do so on a common face. An abstract simplicial complex is a purely combinatorial description of the geometric simplicial complex and therefore does not need the property of intersection of faces. For details about algebraic topology, we refer to [9].

The k -th Betti numbers of an abstract simplicial complex X are defined as the number of k -th dimensional holes in X . For example, β_0 counts the number of 0-dimensional holes, that is the number of connected components. And β_1 counts the number of holes in the plane. If we are in dimension d , the k -th Betti number for $k \geq d$ has no geometric meaning.

B. Algorithm

We consider as inputs the set of existing vertices: the antennas of a damaged wireless network, and their coverage radius $r = \frac{\epsilon}{2}$. In this paper, we only consider the 2-dimensional case for simulation reasons, although it is possible to extend our method to greater dimensions. Then we construct the Čech abstract simplicial complex:

Definition 1 (Čech complex): Given (X, d) a metric space, ω a finite set of points in X , and r a real positive number. The Čech complex of parameter r of ω , denoted $\mathcal{C}_r(\omega)$, is the abstract simplicial complex whose k -simplices correspond to $(k+1)$ -tuples of vertices in ω for which the intersection of the $k+1$ balls of radius r centered at the $k+1$ vertices is non-empty.

The Čech complex characterizes the coverage of a domain. Therefore the Betti number β_1 of the Čech complex counts the number of coverage holes of the wireless network it represents.

The algorithm begins by adding vertices in addition to the set of existing vertices presenting coverage holes. We experimented various vertices addition methods in order to provide comparison as flexibility is the greatest advantage of our algorithm. It is however possible to consider new

methods, especially methods where some vertices position are pre-defined in real life scenarios. All experimented methods are relatively simple, to not increment the total complexity of the algorithm. They are meant to not be precise and add too many vertices to patch all existing coverage holes.

We present four methods in the next section, three of them being based on random representations. In these last methods the number of added vertices N_a is calculated in the same manner. We choose to first add only the minimum number of vertices needed to cover the whole area considering there are already existing vertices. Then, the Betti numbers β_0 and β_1 are computed via linear algebra thanks to the simplicial complex representation. If the network is still disconnected or if there are still coverage holes, the number of added vertices is incremented with an exponential growth. If we denote N_i the number of existing vertices:

- N_a is set to: $N_a = \lceil \frac{a^2}{\pi(\epsilon/2)^2} \rceil - N_i$.
- After adding the N_a vertices, β_0 and β_1 are computed. If $\beta_0 \neq 1$ or $\beta_1 \neq 0$, one more vertex is added.
- Then if still $\beta_0 \neq 1$ or $\beta_1 \neq 0$, two vertices are added, then four, and eight, doubling the number of added vertices each time.

The next step of our approach is to run a reduction algorithm which maintains the topology of our network: the algorithm removes vertices from the abstract simplicial complex without modifying its Betti numbers. At this step, we remove some of the supernumerary vertices we just added in order to achieve an optimal result with a minimum number of added vertices. A full description of the reduction algorithm is given in section 4.

We give in Algorithm 1 an outline of the whole algorithm. We take as input the list of initial vertices and their positions, the Čech complex parameter, and a list of boundary vertices from the initial vertices that let us to know the area to cover. It is important to note that we only need vertices positions in order to compute the connections between vertices: only connectivity information is needed to build the Čech complex.

Algorithm 1 Disaster recovery algorithm

Require: Set ω_i of N_i initial vertices, parameter ϵ , list L_b of boundary vertices.

Computation of the Čech complex $X = \mathcal{C}_{\epsilon/2}(\omega_i)$

$N_a = \lceil \frac{a^2}{\pi(\epsilon/2)^2} \rceil - N_i$

Addition of N_a vertices to X following chosen method

Computation of $\beta_0(X)$ and $\beta_1(X)$

$i = 1$

while $\beta_0 \neq 1$ or $\beta_1 \neq 0$ **do**

$N_a = N_a + i$

 Addition of N_a vertices to X following chosen method

 Computation of $\beta_0(X)$ and $\beta_1(X)$

$i = 2 * i$

end while

Reduction algorithm on X

return List L_a of kept added vertices.

III. NEW VERTICES ADDITION METHOD

We consider a damaged wireless network presenting coverage holes of which we can see an example in Figure 1.

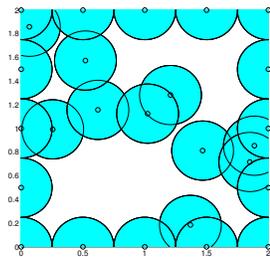


Fig. 1. A damaged wireless network with a fixed boundary.

We present here four methods for new vertices positioning. The first one is deterministic: the number of added vertices and their positions are set. The three other methods use stochastic geometry or random distributions. In these last methods, the number of vertices is computed as presented in the previous section. In this section, we denote (x_i, y_i) the position of the i -th vertex.

A. Grid

The first method we suggest to add vertices ensures perfect coverage: we place new vertices along a square grid in a lattice graph where the distance between two neighbouring vertices is $\epsilon/\sqrt{2}$. The number of vertices is set. Therefore this method is completely independent from the initial configuration. We give in Algorithm 2 the grid method on the square a^2 .

Algorithm 2 Grid method

```

 $N_a = \left( \lfloor \frac{a}{\epsilon/\sqrt{2}} \rfloor + 1 \right)^2$ 
for  $i = 1 \rightarrow m$  do
  for  $j = 1 \rightarrow m$  do
     $x_{N_i+i+(j-1)m} = \frac{(i-1)\epsilon}{\sqrt{2}}$ 
     $y_{N_i+i+(j-1)m} = \frac{(j-1)\epsilon}{\sqrt{2}}$ 
  end for
end for

```

We can see an example of the grid vertices addition method in Figure 2 on the damaged network of Figure 1. Existing vertices are black circles while added vertices are red plusses.

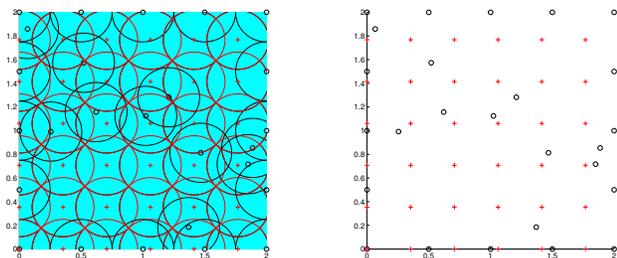


Fig. 2. With the grid addition method.

B. Uniform

Here, the number of added vertices N_a is computed accordingly to the method presented in section 2, taking into account the number of existing vertices. Then the N_a vertices are sampled following a uniform law on the entire area. The uniform addition method on the square a^2 is given in Algorithm 3, and an example is shown in Figure 3.

Algorithm 3 Uniform method

```

for  $i = 1 \rightarrow N_a$  do
   $x_{N_i+i} = \text{rand} * a$ 
   $y_{N_i+i} = \text{rand} * a$ 
end for

```

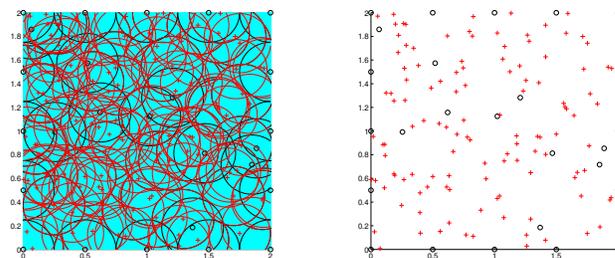


Fig. 3. With the uniform addition method.

C. Determinantal

In this method, the number of vertices is still computed in the same way. However, this method allows us to not only take the number of existing vertices into account but also their positions: added vertices positions are sampled following a determinantal point process taking the N_i existing vertices as the first vertices of the process. These process have the particularity of creating repulsion between vertices, therefore offer a more uniform coverage of a given domain.

The principle behind determinantal point process repulsion is that the probability density used to draw the vertices positions, is computed following a Gram-Schmidt orthonormalisation process in a given basis [11]. So that the probability to draw a vertex at the exact same position of an already existing vertex is zero. And the probability increases with increasing distance from other vertices. Therefore the probability density is the greater in the location the farthest from every vertices, so this method has the ability to locate coverage holes and patch them with new vertices. This construction allows us to consider the first N_i vertices as vertices already drawn in the determinantal point process. Then each vertex position is taken into account as it is drawn.

The determinantal method is given in Algorithm 4. We denote v the vector of the chosen basis that takes as argument the vertices positions, and for more readability we write x_i for the position vector (x_i, y_i) . We simulated this method with a Ginibre basis, of which we can see a realisation in Figure 4.

Algorithm 4 Determinantal method

Computation of $v(x_1), \dots, v(x_{N_i})$
 $e_1 = \frac{v(x_1)}{\|v(x_1)\|}$
for $i = 2 \rightarrow N_i$ **do**
 $w = v(x_i) - \sum_{j=1}^{i-1} (e_j \cdot v(x_i))e_j$
 $e_i = \frac{w}{\|w\|}$
end for
for $i = 1 \rightarrow N_a$ **do**
 $p(x) = \frac{1}{N_a - i + 1} \left(\|v(x)\|^2 - \sum_{j=1}^{N_i + i - 1} |e_j \cdot v(x)|^2 \right)$
 Draw $x_{N_i + i}$ following $p(x)$
 $w = v(x_{N_i + i}) - \sum_{j=1}^{N_i + i - 1} (e_j \cdot v(x_{N_i + i}))e_j$
 $e_{N_i + i} = \frac{w}{\|w\|}$
end for

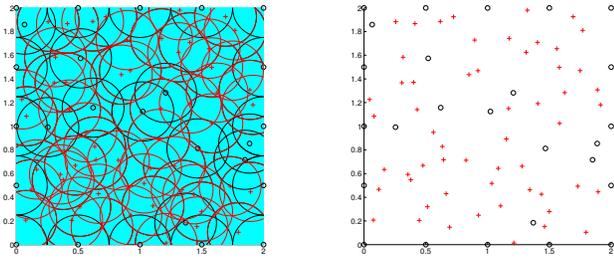


Fig. 4. With the determinantal addition method with a Ginibre basis.

D. Sobol sequence

In this last presented method, we do not take into account the positions of existing vertices at the initial state anymore, but we do take into account the positions of the new added vertices. The Sobol sequence [14] is a statistical tool used to form a uniform partition on the domain, and thus fills space in a highly uniform manner. The Sobol initialisation set is known, for instance on a square the first position is the middle of the square, then come the middles of the four squares included in the big square, etc. To randomize the positions drawn, the points are scrambled.

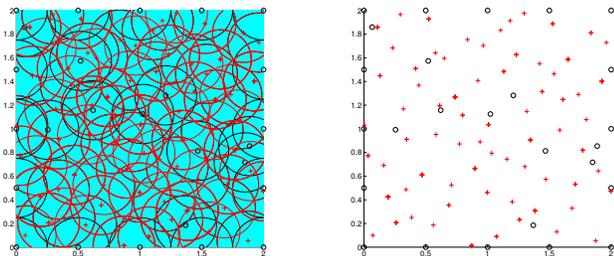


Fig. 5. With the Sobol sequence addition method.

For our simulation, we used the set of initialisation numbers provided by Bratley and Fox in [2]. Then we scrambled the points produced with the random method described in [12].

IV. REDUCTION ALGORITHM

In this section, we present the reduction algorithm first introduced in [15], and explain how we adapt it to achieve an optimal result in patching holes of our damaged network. The algorithm takes as input an abstract simplicial complex: here it is the Čech complex of the wireless network plus the added vertices, and a list of boundary vertices in order to know the area to cover. At this step we have ensured that we have one connected component $\beta_0 = 1$, and no coverage hole $\beta_1 = 0$, via the addition of a sufficient number of vertices.

As seen in Section 2, we use the Čech simplicial complex to characterize the coverage of a wireless network. In simplicial homology, a given area is said to be covered if it is paved by 2-simplices, that are triangles. Therefore, we only need 2-simplices to ensure coverage, higher simplices are superfluous. To characterize the superfluosness of a 2-simplex, we define a degree for every 2-simplex:

Definition 2: The degree of a 2-simplex $[v_0, v_1, v_2]$ is the size of the greatest simplex it is a face of:

$$D[v_0, v_1, v_2] = \max\{d \mid [v_0, v_1, v_2] \subset d\text{-simplex}\}.$$

We immediately have $D[v_0, v_1, v_2] \geq 2$ for any 2-simplex.

Then, we are not interested in removing 2-simplices but 0-simplices: vertices. To transmit the superfluosness of its 2-simplices to a vertex, we define indices for every vertex:

Definition 3: The index of a vertex v is the minimum of the degrees of the 2-simplices it is a vertex of:

$$I[v] = \min\{D[v_0, v_1, v_2] \mid v \in [v_0, v_1, v_2]\},$$

If a vertex v is not a vertex of a 2-simplex then $I[v] = 0$. If a vertex v is a boundary vertex then $I[v] = -1$.

The indices give us an order for the removal of vertices: the greater the index of a vertex, the more likely it is superfluous for the coverage of its Čech simplicial complex. In our disaster recovery case, we do not want to remove the remaining vertices of our damaged network. So these remaining vertices are given a negative index to flag them as unremovable, and only the newly added vertices are considered for removing. Finally, the vertices with the greatest index are candidates for removal: one is chosen randomly. The removal of a vertex leads to the degradation of all the k -simplices it was a vertex of, to $(k - 1)$ -simplices for every integer k .

At every vertex removal, we need to ensure that the homology is unchanged: we compute the two first Betti numbers β_0 and β_1 every time a vertex is removed. If they change, the vertex is put back into the abstract simplicial complex with a negative index to flag it as unremovable. Otherwise, the removal of the vertex is confirmed. The modified degrees of the 2-simplices and the indices of the vertices are recalculated. We can note that only the vertices of maximum index can have their index changed. The algorithm goes on removing vertices until every remaining vertex is unremovable, thus achieving optimal result. For more information on the reduction algorithm we refer to [15].

We give in Algorithm 5 the reduction algorithm. We denote s_k the number of k -simplices, D_1, \dots, D_{s_2} the degrees of 2-simplices, and v_1, \dots, v_{s_0} the vertices.

Algorithm 5 Reduction algorithm

Require: Simplicial complex X , list L_b of boundary vertices.
 Computation of $D_1(X), \dots, D_{s_2}(X)$
 Computation of $I[v_1(X)], \dots, I[v_{s_0}(X)]$
for all $v \in L_i \cup L_b$ **do**
 $I[v] = -1$
end for
 $I_{\max} = \max\{I[v_1(X)], \dots, I[v_{s_0}(X)]\}$
while $I_{\max} > 0$ **do**
 Draw w a vertex of index I_{\max}
 $X' = X \setminus \{w\}$
 Computation of $\beta_0(X')$ and $\beta_1(X')$
 if $\beta_0 \neq 1$ or $\beta_1 \neq 0$ **then**
 $I[w] = -1$
 else
 Computation of $D_1(X'), \dots, D_{s'_2}(X')$
 for $i \rightarrow 1 : s'_0$ **do**
 if $I[v_i(X')] = I_{\max}$ **then**
 Computation of $I[v_i(X')]$
 end if
 end for
 $I_{\max} = \max\{I[v_1(X'), \dots, I[v_{s'_0}(X')]\}$
 $X = X'$
 end if
end while
return X

We can see in Figure 6 an execution of the reduction algorithm on the configuration of Figure 4. Removed vertices are represented by blue diamonds.

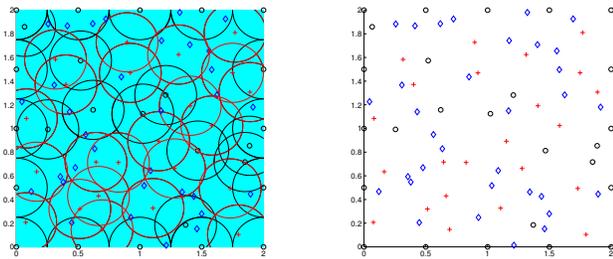


Fig. 6. The reduction algorithm run on the determinantal method example.

V. PERFORMANCE COMPARISON

A. Addition method comparison

First, we compare the complexities of our four methods. As before, N_i is the number of existing vertices and N_a the number of added vertices. Three of the presented methods are of complexity $O(N_a)$. The grid and the uniform methods only compute N_a positions, and the Sobol method scramble N_a positions already known by most simulation tools. The

determinantal method is a little bit more complex: first taking into account the existing vertices in the Gram-Schmidt orthonormalisation is of complexity $O(N_i^2)$, then the position drawing with the rejection sampling via orthonormalisation is of complexity $O(N_a(N_a + N_i))$ at most. Thus we have a final complexity of $O(N_i^2 + N_a^2 + N_a N_i)$. The determinantal method is the more complex because it takes into account the position of existing vertices. To these complexities we have to add the complexity of computing the coverage via the Betti numbers which is of the order of the number of triangles times the number of edges that is $O((N_a + N_i)^5 (\frac{\epsilon}{a})^6)$ according to [5].

Then, we can compare the mean number of added vertices. The first method being determinist, the number of added vertices is constant: $N_a = (\lfloor \frac{a}{\epsilon/\sqrt{2}} \rfloor + 1)^2$. This method is also mathematically optimal for the number of added vertices to cover the whole area, however it is not optimal in a real life scenario where positions can not be defined with such precision. We can see in Table I the mean number of added vertices on 500 simulations for each random method in different scenarios on a square of side $a = 1$ with communication range $\epsilon = 0.5$ and a Vietoris-Rips simplicial complex, which is the clique complex of the geometric graph. The Vietoris-Rips complex is an approximation of the Čech complex, much easier to simulate. Scenarios are defined by the mean percentage of area covered before running the recovery algorithm: if there are many or few existing vertices, and thus few or many vertices to add. We need to note that number of added vertices is computed following our incrementation method presented in section 2 and the presented results are for N_a the number of added vertices before the reduction algorithm ran.

% of area initially covered	20%	40%	60%	80%
Uniform method	32.51	29.34	24.64	15.63
Determinantal method	16.00	14.62	12.36	7.79
Sobol sequence method	29.00	29.40	24.09	16.96

TABLE I
MEAN NUMBER OF ADDED VERTICES

The determinantal method gives the best results, by far, thanks to its ability to locate coverage holes in the existing vertices distributions. We can also observe that the Sobol sequence method gives better results than the uniform method for a large percent of uncovered area. Since the Sobol sequence partitions uniformly the area, it takes advantages that there are not too many existing vertices.

Finally the grid method is good both in complexity and in number of added vertices. However it does not take advantages of our recovery algorithm which allows flexibility in the vertices positions sample method. The determinantal method provides the best results in all scenarios among the random methods. Nevertheless, if complexity is a key factor the Sobol sequence or the uniform method give good results depending on the percentage of covered area.

B. Comparison with the greedy algorithm

We now compare our performance results to the main known method: the greedy algorithm for the set cover problem. The greedy algorithm method lays a square grid of parameter $\epsilon/\sqrt{2}$ of potential new vertices. Then the first added vertex is the furthest from all existing vertices. The algorithm goes on adding the furthest potential vertex of the grid from all vertices. We can see that this algorithm is resource consuming: first the algorithm computes the distances of the N_i existing vertices to the $m = (\lfloor \frac{a}{\epsilon/\sqrt{2}} \rfloor + 1)^2$ potential vertices, then after adding the i -th vertex it computes the distances from this vertex to the $m - i$ remaining potential vertices. Therefore the complexity of the greedy algorithm is of $O((N_i + N_a)(\lfloor \frac{a}{\epsilon/\sqrt{2}} \rfloor + 1)^2))$ at most.

For the complexity of our algorithm, we consider first the complexity of building the simplicial complex associated with the network which is in $O((N_i + N_a)^C)$, where C is the clique number. This complexity seems really high since C can only be upper bounded by $N_i + N_a$ in the general case but it is the only way to compute the coverage when vertices position are not defined along a grid. Then the complexity of the reduction algorithm is of the order of $O((1 + (\frac{\epsilon}{a})^2)^{N_i + N_a})$ (see [15]). So the greedy algorithm appears less complex than ours in the general case. However when ϵ is small before a or when the dimension is greater than 2, then the power factor becomes $d > 2$ and C is a finite integer, so the trend is reversed.

The number of vertices added in the final state both with our recovery algorithm and the greedy algorithm tends to approximatively the same number: the minimum number of vertices required to cover the uncovered area depending on the initial configuration, hence the slight differences. Results presented in Table II were simulated in the same conditions as for Table I and concern the final number of added vertices: the number of added vertices kept after the reduction algorithm, or added with the greedy algorithm. It is important to note that our algorithm with the grid method gives the exact same result as the greedy algorithm, added vertices positions being the same.

% of area initially covered	20%	40%	60%	80%
Homology algorithm	4.04	3.63	2.85	1.78
Greedy algorithm	3.67	3.37	2.82	1.85

TABLE II
MEAN FINAL NUMBER OF ADDED VERTICES

The greedy algorithm is optimal, as ours, since it adds the minimum number of vertices. But it is not flexible at all, and its success depends highly on the adherence of the chosen positions: a coverage hole appears as soon as a vertex is slightly moved.

VI. CONCLUSION AND FUTURE WORK

In this paper, we adopt the simplicial homology representation for wireless networks which characterizes not only the connectivity but also the coverage of a given network.

Based on that representation, we write an algorithm which patches coverage holes of wireless networks. First, we suggest various methods for adding supernumerary vertices: classic methods that can reproduce the results of known algorithms or have minimum complexity, and original methods, from other fields, never applied to wireless networks. Among these methods, the determinantal method, which creates repulsion between vertices, provides the best results for patching holes with a minimum number of vertices. Then, we improve a simplicial complex reduction algorithm in order to remove any superfluously added vertex, and achieve an optimal result.

As for future work, we intend to explore new applications for the advantages we draw from the simplicial homology representation in Self Organizing Networks or frequency reuse in LTE cellular network, as well as in green networking.

REFERENCES

- [1] A.Nguyen, N.Milosavljević, Q.Fang, J.Gao, and L.J.Guibas. Landmark selection and greedy landmark-descent routing for sensor networks. In *Proceedings of IEEE INFOCOM 2007*, 2007.
- [2] P. Bratley and B. L. Fox. Algorithm 659: Implementing sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.*, 14(1):88–100, Mar. 1988.
- [3] V. Chvatal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.
- [4] V. de Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *International Journal of Robotics Research*, 25, december 2006.
- [5] L. Decreusefond, E. Ferraz, H. Randriambololona, and A. Vergne. Simplicial homology of random configurations. *Advances in Applied Probability*, 2013.
- [6] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2458 – 2468 vol.4, march 2004.
- [7] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang. GLIDER: Gradient landmark-based distributed routing for sensor networks. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2005.
- [8] 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*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.
- [9] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [10] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the second annual symposium on Computational geometry*, SCG '86, pages 61–71, New York, NY, USA, 1986. ACM.
- [11] J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probab. Surv.*, page 2006.
- [12] J. Matoušek. On the l2-discrepancy for anchored boxes. *J. Complex.*, 14(4):527–556, Dec. 1998.
- [13] A. Muhammad and A. Jadbabaie. Decentralized computation of homology groups in networks by gossip. In *American Control Conference, 2007. ACC '07*, pages 3438 –3443, july 2007.
- [14] I. M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *Computational Mathematics and mathematical physics*, 7(4):86+, 1967.
- [15] A. Vergne, L. Decreusefond, and P. Martins. Reduction algorithm for simplicial complexes. In *The 32nd Annual IEEE International Conference on Computer Communications (INFOCOM'2013)*, 2013.
- [16] F. Yan, P. Martins, and L. Decreusefond. Connectivity-based distributed coverage hole detection in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2011)*, 2011 IEEE, dec. 2011.
- [17] C. Zhang, Y. Zhang, and Y. Fang. Detecting coverage boundary nodes in wireless sensor networks. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 868 –873, 0-0 2006.