



**HAL**  
open science

## Simplicial homology for future cellular networks

Anaïs Vergne, Laurent Decreasefond, Philippe Martins

► **To cite this version:**

Anaïs Vergne, Laurent Decreasefond, Philippe Martins. Simplicial homology for future cellular networks. 2013. hal-00914659v1

**HAL Id: hal-00914659**

**<https://hal.science/hal-00914659v1>**

Preprint submitted on 5 Dec 2013 (v1), last revised 11 Sep 2014 (v4)

**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.

# Simplicial homology for future cellular networks

A. Vergne, L. Decreusefond and P. Martins Institut Telecom, TELECOM ParisTech, LTCI  
Paris, France

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

**Abstract**—In this article, we apply the simplicial complex representation to cellular networks, then using a reduction algorithm for simplicial complexes we provide algorithms for the management of future cellular networks. First, we propose a frequency auto-planning algorithm for the self-configuration of future cellular networks. It aims at minimizing the number of planned frequencies while maximizing the usage of each one. Then, we present an energy conservation algorithm for the self-optimization of future cellular networks during off-peak hours. It takes into account not only coverage, but also user traffic. Finally, we present and discuss the performance of a disaster recovery algorithm using determinantal point processes to patch coverage holes.

## I. INTRODUCTION

Long Term Evolution (LTE) is the 3GPP standard specified in Releases 8 and 9. Its main goal is to increase both capacity and speed in cellular networks. Indeed, cellular network usage has changed over the years and bandwidth hungry applications, as video calls, are now common. Achieving this goal costs a lot of money to the operator. A solution to limit operation expenditures is the introduction of Self-Organizing Networks (SON). 3GPP standards have identified self-organization as a necessity for future cellular networks [1]. A full description of SON in LTE can be found in [18]. SON features include self-configuration, self-optimization, and self-healing functions.

First, self-configuration functions aims at the plug-and-play paradigm: new transmitting nodes should then be automatically configured and integrated to the existing network. Upon arrival of a new node, the neighboring nodes update their dynamic neighbor tables thanks to the Automatic Neighbor Relation (ANR) feature. This information is equivalent to the connectivity information in wireless sensor networks, needed to build the simplicial complex representation. Among self-configuration functions, we can find the dynamic frequency auto-planning problem. It is a known problem from spectrum-sensing cognitive radio where equipments are designed to use the best wireless channels in order to limit interference [17]. The different nodes of the secondary cognitive network have to choose the best frequency to use in order to maximize the coverage and minimize the interference with the base stations of the primary network. The hierarchy in networks makes this solution not directly practicable to future cellular networks. While in earlier releases, static frequency planning was preferred, it has become a critical point since the network has a dynamic behavior with arrivals and departures of base stations, and does not always follow a regular pattern with the introduction of Femtocells and Heterogeneous Networks (HetNet).

The second main SON feature is the category of the self-optimization functions, which defines the ability of the network to adapt its behavior to different traffic scenarii. Indeed, in LTE cellular networks, eNode-Bs (eNB) have multiple configurable parameters. An example is output power, so cells size can be configured when capacity is the limitation rather than coverage. Moreover, fast and reliable X2 communication interfaces connect eNBs. So the whole network has the capability to adapt to different traffic situations. Then, users traffic can be observed via eNBs and User Equipments (UEs) measurements. Therefore, the self-optimization functions aim at using this traffic observations to adapt the whole network, and not only each cell independently, to the traffic situation. One case where self-optimization is often needed is the adaptation to off-peak hours. Typically a cellular network is deployed to match daily peak hours traffic requirements. Therefore during off-peak hours, the network is daily under-used. This leads to a huge unneeded amount of energy consumption. An idea is thus to switch-off some of the eNBs during off-peak hours, while other eNBs adjust their configuration parameters to keep the entire area covered. In case of a growth in traffic, the switched-off eNBs could be woken up to satisfy the users demand.

The third and last of the SON main functions is self-healing. In future cellular networks, nodes would be able to appear and disappear at any time. Since the cellular network is not only constituted of operators base stations anymore, the operator does not control the arrivals or departures of nodes. But the disappearances of nodes can be more generalized: for example in case of a natural disaster (floods, earthquakes or tsunamis...). The self-healing functions aim at reducing the impacts from the failures of nodes must it be in isolated cases, like the turning off of a Femtocells, or more serious cases where the whole network is damaged. We are interested in this latter case. In case of a disaster, a cellular network can be seriously damaged: some of its nodes can be completely destroyed. However such networks are not necessarily built with redundancy and then can be sensitive to such damages. Coverage holes can appear resulting in no signal for communication or no monitoring at all of a whole area. Paradoxically, reliable and efficient communication and/or monitoring is especially needed in such situations. Therefore, solutions for damage recovery for the coverage of cellular networks are much needed.

In this article, we use the simplicial complex representation in order to represent a cellular network with ANR information. First, we first propose a frequency auto-planning algorithm which, for any given cellular network, provides a frequency planning minimizing the number of frequencies needed for a given accepted threshold of interference. The algorithm calls several instances of a reduction algorithm, introduced in [26],

for the allocation of each frequency. Using simplicial complex representation combined to the reduction algorithm allows us to obtain a homogeneous coverage between frequencies. In a second part, we enhance the reduction algorithm to satisfy any user traffic. The reduction algorithm, as it is presented in [26], only satisfies perfect connectivity and coverage. However, in cellular networks, especially in urban areas, coverage is not the limiting factor, capacity is. So the optimal solution is not optimal coverage anymore, but depends on the required traffic. We present an enhanced reduction algorithm to reach an optimally used network. The performance of our energy conservation algorithm is discussed and compared to the minimum configuration required by traffic. Finally, we present an algorithm for disaster recovery of wireless networks first introduced in [27]. Given a damaged cellular network, the algorithm first adds too many nodes then runs the reduction algorithm to reach an optimal result. For the addition of vertices we propose the use of a determinantal point process which has the inherent ability to locate areas with low density of vertices: namely coverage holes. We thoroughly evaluate the performance of our algorithm compared to the classic recovery algorithm: the greedy algorithm for the set cover problem, comparing their complexity, number of added nodes, and smoothed robustness.

The remainder of this article is organized as follows. After a section on related work on self-configuration, self-optimization and recovery of future cellular networks in Section II, we introduce some simplicial homology definitions as well as the reduction algorithm for simplicial complexes used all along the article in Section III. Then in Section IV, we introduce our frequency auto-planning algorithm. The energy conservation algorithm is presented in Section V. Finally, we provide the disaster recovery description and performance evaluation in Section VI.

## II. RELATED WORK

A complete survey on SON for future cellular networks is given in [4].

### A. Self-configuration in future cellular networks

Configuration of the different nodes (eNBs, relays, Femtocells) of a cellular network has to be done during the deployment of the network, but also upon the arrival and departure of every node. The classic manual configuration done for previous generations of cellular networks can not be operated in future cellular networks: changes in the network, such as arrivals and departures, occur too often. Moreover, the commercialization of private Femtocells leads to the presence in the network of nodes with no access for manual support. So the future cellular networks are heterogeneous networks with no regular pattern for its nodes. They need to be able to self-configure themselves. The initial parameter that a node needs to configure are its IP address, its neighbor list and its radio access parameters. IP addresses are out of the scope of this work, but we will discuss the two other parameters.

The neighbor list is the connectivity information we use for the construction of the simplicial complex which represents the

cellular network. The selection of the nodes to put on one's neighbor list can be based on the geographical coordinates of the nodes and take into account the antenna pattern and transmission power [22]. However, this approach does not consider changing radio environment, and requires exact location information which can be easy to obtain for eNBs, but not for Femtocells. The authors of [21] propose a better criterion for the configuration of the neighbor list: each node scans in real time the Signal to Interference plus Noise Ratio (SINR) from other nodes.

Among radio access parameters, we can find frequency but also propagation parameters since the apparition of beamforming techniques via MIMO. Let us focus on the former which is the subject of Section IV. The frequency planning problem was first introduced for GSM networks. However the constraints were not the same: the frequency planning was static with periodic manual optimizations, and in simulations base stations were regularly deployed along an hexagonal pattern. With the commercialization of Femtocells and the deployments of outdoor relays and Picocells, cells do not follow a regular pattern anymore and can appear and disappear at any time. Therefore the frequency planning problem has to be rethought in an automatic way. A naive idea for frequency auto-planning would simply be applying the greedy coloring algorithm to the sparse interference graph [19]. However, even if it reaches an optimal solution for the number of needed frequencies, their utilization can be disparate: one frequency can be planned for a large number of nodes compared to another planned for only few of them. Then if the level of interferences increase (more users, or more powered antennas), this could lead to communication problems for the over-used frequency, and a whole new planning is needed. While a more homogeneous resource utilization can be more robust if interferences increase, since there are less nodes using the same frequency on average. We provide here a frequency auto-planning algorithm which aims at a more homogeneous utilization of the resources. Moreover, the planning of frequency channels for new nodes that do not interfere with existing nodes while still provide enough bandwidth is still an open problem. It has been addressed in the cognitive radio field, but these algorithms usually enable opportunistic spectrum access [25]. However, it is not possible to extend this type of algorithm to the frequency allocation of new nodes in cellular networks. Indeed, the new nodes would be part of the primary network, with a quality of service to achieve, so their frequency allocation needs to be guaranteed and not opportunistic. The algorithm we propose aims at allocating frequency channels to future cellular networks with non-regular deployment, so upon the arrival of a new node, the whole network is re-configured.

### B. Self-optimization in future cellular networks

In order to ensure that future cellular networks are still efficient in terms of both Quality of Service (QoS) and costs, the self-configuration is not sufficient. Indeed, future cellular networks have the ability to adjust their parameters to match different traffic situation. Periodic optimization based on log reports, and operated centrally is not a effective solution

in terms of speed and costs. That is why we need self-optimization. Self-optimization can be classified in three types depending on its goal.

First we can consider load balancing optimization. There is multiple ways to adapt a cellular network to different loads: it is for example possible to adapt the resources available in different nodes. These schemes were mainly introduced for GSM [10], and then CDMA [3], but the universal frequency reuse of LTE and LTE-Advanced diminishes their applicability. Then one can adapt the traffic strategy with admission controls on given cells and forced handovers [14]. However, as the previous solution, it is not very suitable for LTE and LTE-A which require hard handover. Finally it is possible to modify the coverage of a node by changing either its antennas radiation pattern [12] or the output power [11]. We use this latter approach to reach an optimal result for our algorithm: we adapt the coverage radius of each node to be the minimum required to cover a given area.

The second type self-optimization is the capacity and coverage adaptation via the use of relay nodes [20], while the third is interference optimization. Our energy conservation algorithm presented in Section V could lie in this third category as the simplest approach towards interference control is switching off idle nodes. It is done based on cell traffic for Femtocells in [7]: after a given period of time in idle mode, the node puts itself on stand-by. However, if one wants to take into account the whole network, it has to consider the coverage of the network before disconnecting, which is not the case of Femtocells, which are by definition redundant to the base stations network. Without considerations of traffic, we proposed in [26] an algorithm that reduce power consumption in wireless networks by putting on standby some of the nodes without impacting the coverage. We can also cite [8] that proposes a game-theoretic approach in which nodes are put on standby according to a coverage function, but unmodified coverage is not guaranteed. In both these works, only coverage is taken into account. This approach could eventually fit the requirements of cellular network in non-urban cells, if their deployment has coverage redundancy. But it is not valid for urban cells, where it is not coverage but capacity that delimits cells. Our algorithm goes a little bit further by adapting the switching-off of the nodes to the whole network situation, the traffic and the coverage.

### C. Recovery in future cellular networks

The first step of recovery in cellular networks is the detection of failures. The detection of the failure of a cell occurs when its performance is considerably and abnormally reduced. In [24], the authors distinguish three stages of cell outage: degraded, crippled and catatonic. This last stage matches with the event of a disaster when there is complete outage of the damaged cells. After detection, compensation from other nodes can occur through relay assisted handover for ongoing calls, adjustments of neighboring cell sizes via power compensation or antenna tilt. In [5], the authors not only propose a cell outage management description but also describe compensation schemes. These steps of monitoring and detection, then compensation of nodes failures are comprised

under the self-healing functions of future cellular networks described in [2].

In Section VI, we are interested in what happens when self-healing is not sufficient. In case of serious disasters, the compensation from remaining nodes and traffic rerouting might not be sufficient to provide service everywhere. In this case, the cellular network needs a manual intervention: the adding of new nodes to compensate the failures of former nodes. However a traditional restoration with brick-and-mortar base stations could take a long time, when efficient communication is particularly needed. In these cases, a recovery trailer fleet of base stations can be deployed by operators [23], it has been for example used by AT&T after 9/11 events. But a question remains: where to place the trailers carrying the recovery base stations. An ideal location would be adjacent to the failed node. However, these locations are not always available because of the disaster, and the recovery base stations may not have the same coverage radii than the former ones. Therefore a new deployment for the recovery base stations has to be decided, in which one of the main goal is complete coverage of damaged area. This can be viewed as a mathematical set cover problem. It suffices to define the universe as the area to be covered and the subsets as the balls of radii the coverage radii. Then the question is to find the optimal set of subsets that cover the universe, considering there are already balls centered on the existing vertices. It can be solved by a greedy algorithm [9],  $\epsilon$ -nets [16], or furthest point sampling [13], [6]. But these mathematical solutions provide an optimal mathematical result that do not consider any flexibility at all in the choosing of the new nodes positions, and that can be really sensitive to imprecisions in the nodes positions.

## III. PRELIMINARIES

### A. Simplicial homology

First we need to remind some definitions from simplicial homology for a better understanding of the simplicial complex representation of wireless networks.

When representing a wireless sensor network, one's first idea will be a geometric graph, where sensors are represented by vertices, and an edge is drawn whenever two sensors can communicate with each other. However, the graph representation has some limitations; first of all there is no notion of coverage. 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 3-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 [15].

We consider the nodes of a cellular network, and their coverage radii. The construction of the Čech abstract simplicial complex for a fixed radius  $r$  is given:

*Definition 1 (Čech complex):* Given  $(X, d)$  a metric space,  $\omega$  a finite set of  $N$  points in  $X$ , and  $r$  a real positive number. The Čech complex of  $\omega$ , denoted  $\mathcal{C}_r(\omega)$ , is the abstract simplicial complex whose  $k$ -simplices correspond to  $(k + 1)$ -tuples of vertices in  $\omega$  for which the intersection of the  $k + 1$  balls of radii  $r_i$  centered at the  $k + 1$  vertices is non-empty. It is possible to build the Čech complex of a cellular network with different coverage radii using the intersection of different size coverage balls. The Čech complex characterizes the coverage of the network. The  $k$ -th Betti numbers of an abstract simplicial complex  $X$  are defined as the number of  $k$ -th dimensional holes in  $X$  and are computed via linear algebra computations. For example,  $\beta_0$  counts the number of 0-dimensional holes, that is the number of connected components. And  $\beta_1$  counts the number of holes in the plane. Therefore the Betti number  $\beta_1$  of the Čech complex counts the number of coverage holes of the wireless network it represents.

## B. Reduction algorithm

In this section, we recall the steps of the reduction algorithm for simplicial complexes presented in [26]. The algorithm takes as input an abstract simplicial complex: here it is a Čech complex and a list of boundary vertices that can be given by the network operator or computed via the convex hull. Then it removes vertices without modifying the connectivity nor the coverage of the complex, and thus the ones of the cellular network.

The first step is to characterize the superfluousness of 2-simplices for the coverage with a degree. The degree of a 2-simplex is then defined to be the size of the largest simplex a 2-simplex is the face of. Next since we need to remove vertices and not 2-simplices, we need to transmit the superfluousness of its 2-simplices to a vertex via what is called an index. An index of a vertex is defined to be the minimum of the degrees of the 2-simplices it is a face of. Indeed, a vertex is as sensitive for the coverage as its most sensitive 2-simplex.

Finally, the indices give an optimal 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. The boundary vertices are given a negative index to mark them as unremovable by the algorithm: we do not want the covered area to be shrunk. Therefore, the vertices with the greatest index are candidates for removal: one is chosen randomly. If its removal does not change the homology, i.e. if it does not modify its Betti numbers  $\beta_0$  and  $\beta_1$ , then it is effectively removed. Otherwise it is flagged as unremovable the same way the boundary vertices are: with a negative index. The

algorithm goes on until every remaining vertex is unremovable, thus achieving optimal result. For more information on the reduction algorithm we refer to [26].

## IV. SELF-CONFIGURATION FREQUENCY AUTO-PLANNING ALGORITHM

### A. Main idea

We consider a cellular network that we represent by a geometric simplicial complex: the transmitting nodes (eNBs, Femtocells, relays...) are represented by vertices, then we build the Čech complex corresponding to the different communication radii. In this application, we are not interested in the characterization of the topology of the network, the main goal is to assign frequencies to the networks nodes, so the network's topology is not considered nor modified. The disks we consider to build the complex are therefore communication disks: when communication disks intersect, the corresponding nodes can communicate between each other. With the simplicial complex representation, we can characterize groups of nodes that are close to each other using simplices. Indeed, in the frequency planning problem, the goal is to assign to each node a frequency so that the interference between them is the smallest possible using a minimum number of frequencies. Here we will only consider co-channel interference: interference between two nodes using the same frequency. Moreover, we will only consider the one frequency per node case but the main idea of the algorithm can be extended to several frequencies per nodes.

We introduce what we call a rejection radius, this radius defines around every node a rejection disk. If one node is within the rejection disk of another node, then we consider that they shall not share the same frequency or the level of interference will be too high for reliable communication within each one of the two cells. This rejection radius defines the interference threshold that is acceptable in the cellular network. It is the interference criterion we will use for our auto-planning algorithm.

The algorithm begins by computing the degrees of 2-simplices and the indices of vertices as defined for the reduction algorithm presented in Section III. Then we apply a modified version of the reduction algorithm: the order in which the vertices are removed is still decided by the indices but the stopping condition is not the same anymore. Indeed we are not interested into achieving optimal coverage anymore. Therefore, instead of stopping when the maximum index among every remaining vertices is below a given number, the algorithm stops when there is no more vertices in any rejection disk of any other vertex. The vertices of the resulting simplicial complex are assigned the first frequency.

Then all the removed vertices are collected, and the corresponding simplicial complex recovered. This simplicial complex is a subset of the initial simplicial complex so there is no need to build another simplicial complex from scratch. The next step is then to reapply the modified coverage reduction algorithm to this recovered simplicial complex to obtain a second set of vertices to which we assign the second frequency. The algorithm goes on until every vertex has an assigned frequency.

At the end, we have a frequency assigned to every node. We ensured that no two nodes sharing the same frequency will be too close to each other: interference will be under a given threshold. Moreover, the use of our coverage reduction algorithm with the optimized order for vertices removal allows us to obtain a homogeneous usage of every frequency.

### B. Algorithm description

We give in Algorithm 1 the full frequency auto-planning algorithm: it requires the set of vertices  $\omega$ , and their communication and rejection radii, then returns the list of assigned frequencies for every vertex of  $\omega$ .

---

#### Algorithm 1 Frequency auto-planning algorithm

---

**Require:** Set  $\omega$  of  $N$  vertices, for each vertex  $v$  its communication radius  $r_v$ , and its rejection radius  $R_v$ .  
 Computation of the Čech complex  $X = \mathcal{C}_r(\omega)$   
 Computation of  $D_1(X), \dots, D_{s_2}(X)$   
 Computation of  $I[v_1(X)], \dots, I[v_{s_0}(X)]$   
 $I_{\max} = \max\{I[v_1(X)], \dots, I[v_{s_0}(X)]\}$   
 $N_{\text{planned}} = 0$   
 Interference = 1  
 $X' = X$   
 $i = 0$   
**while**  $N_{\text{planned}} < N$  **do**  
  **while** Interference == 1 **do**  
    Draw  $w$  a vertex of index  $I_{\max}$   
     $X' = X' \setminus \{w\}$   
    Computation of  $D_1(X'), \dots, D_{s'_2}(X')$   
    **for**  $i = 1 \rightarrow s'_0$  **do**  
      **if**  $I[v_i(X')] == I_{\max}$  **then**  
        Recomputation of  $I[v_i(X')]$   
      **end if**  
    **end for**  
     $I_{\max} = \max\{I[v_1(X'), \dots, I[v_{s'_0}(X')]\}$   
    Interference = 0  
    **for all**  $u, v \in X'$  **do**  
      **if**  $\|v - u\| < \max(R_v, R_u)$  **then**  
        Interference = 1  
      **end if**  
    **end for**  
  **end while**  
  **for all**  $v \in X'$  **do**  
    Frequency( $v$ ) =  $i$   
     $N_{\text{planned}} = N_{\text{planned}} + 1$   
  **end for**  
   $X' = X \setminus X'$   
   $i = i + 1$   
**end while**  
**return** List of assigned frequencies Frequency( $v$ ),  $\forall v \in \omega$ .

---

It is interesting to note that we do not need to compute the Betti numbers of the Čech complex, since we are not effectively removing vertices so the topology of the network does not change. However we use the simplicial complex

representation information to locate the vertices too close to each other.

We introduce three parameters in the algorithm description. First of them is the number of planned vertices: it represents the number of vertices to which we assigned a frequency already. Then, the ‘Interference’ parameter is a binary number to represent if there are at least two vertices with potentially the same frequency within the rejection disk of each other. Finally we also introduce the ‘Frequency’ notation, for every vertex, this is where we store the frequency assigned to it.

### C. Performance

For simulation reasons, we only compute the Vietoris-Rips complex which is an approximation of the Čech complex based on its graph description only, in the 2-dimensional case.

To measure the performance of our algorithm we compare the number of frequencies our algorithm plans versus the number of frequencies the greedy coloring algorithm plans. Indeed, the frequency planning can be viewed as a graph coloring problem. We consider the geometric graph whose edges are added if at least one of its extremity vertex is within the rejection disk of the other one. Then the optimal number of frequencies to assign is the chromatic number of the graph. The greedy coloring algorithm provides a coloring for a given graph assigning the first new color available for each vertex. Therefore, the greedy coloring algorithm provides a frequency planning with a number of frequency equal to the maximum vertex degree plus one. The greedy coloring gives especially good results for the number of used colors for sparse graphs as the interference graph is. However, the greedy coloring algorithm leads to a disparate utilization of frequencies. Indeed, if there is only one clique of maximum size, one frequency will be only used for one vertex of this clique, and for no other vertex in the whole configuration. Therefore, this algorithm could give good results in a homogeneous network, but not for a cluster network for example. Our algorithm aims at a more homogeneous utilization of each resource.

We simulate the set of vertices with a Poisson point process of intensity  $\lambda = 12$  on a square of side  $a = 2$ . The communication radii are sampled uniformly between  $a/10$  and  $a^2/\lambda$ , each rejection radius was equal to half its corresponding communication radius. The results are obtained in mean over 1000 configurations. For each realization of the Poisson process, we compute the number of frequencies planned by the greedy coloring algorithm that we denote  $N_g$ . Then on all realizations with a given  $N_g$ , we compute the mean number of frequencies, denoted  $N_f$ , planned by our algorithm.

$N_g$	2	3	4	5	6	7
$\mathbf{E} [N_f   N_g]$	3.73	4.66	5.53	6.21	6.95	6.50
Occurrence	1.1%	29.9%	45.6%	19.0%	4.2%	0.2%

TABLE I  
MEAN NUMBER OF PLANNED FREQUENCIES  $\mathbf{E} [N_f | N_g]$  FOR EACH GIVEN  $N_g$ .

In Table I, we can see the mean number of planned frequencies given the number of frequencies planned by the

greedy coloring algorithm. We also indicate which percentage of the 1000 simulations these situations represent. We can see that there is a difference between the two solutions: it is not negligible in the beginning, but it decreases with the number of frequencies. Thus, our algorithm reaches its optimal performance when the number of frequencies grows for the same mean number of nodes, that is to say when there are clusters of nodes or when the rejection radii are quite large compared to the communication radii.

We are also interested in highlighting the strong point of our frequency auto-planning algorithm: the homogeneous utilization of frequencies. So we measure, for the same configuration parameters, the percentage of area covered by each frequency compared to the total covered area for our frequency auto-planning algorithm and for the greedy coloring algorithm. The percentages are given in mean over 1000 simulations. For an optimal utilization of frequencies, each frequency should cover the whole area, but it is not always achievable if there are not enough nodes to cover several times the whole area.

# frequencies	3	4	5	6	7	8
Frequency 1	77.8%	73.2%	67.6%	63.4%	59.0%	48.9%
Frequency 2	65.8%	58.6%	58.4%	54.4%	49.2%	45.9%
Frequency 3	37.7%	46.4%	46.7%	46.1%	47.2%	46.8%
Frequency 4		26.4%	34.7%	36.4%	37.2%	43.5%
Frequency 5			18.6%	28.0%	30.6%	36.6%
Frequency 6				16.2%	24.9%	28.6%
Frequency 7					13.2%	17.9%
Frequency 8						11.4%
Occurrence	6.3%	18.4%	30.0%	24.9%	13.6%	4.9%

TABLE II

MEAN PERCENTAGE OF COVERED AREA BY EACH FREQUENCY WITH OUR ALGORITHM

# frequencies	3	4	5	6	7
Frequency 1	98.1%	97.9%	97.7%	97.9%	98.2%
Frequency 2	57.7%	54.3%	55.7%	55.0%	52.3%
Frequency 3	17.7%	17.3%	19.1%	16.5%	18.0%
Frequency 4		4.1%	4.8%	4.2%	3.8%
Frequency 5			1.7%	1.2%	1.0%
Frequency 6				0.5%	1.0%
Frequency 7					1.0%
Occurrence	12.3%	44.7%	30.0%	11.3%	1.7%

TABLE III

MEAN PERCENTAGE OF COVERED AREA BY EACH FREQUENCY WITH THE GREEDY COLORING ALGORITHM

We can see in Table II and Table III the percentage of area covered by each frequency planned by our algorithm and the greedy coloring algorithm. The results are presented depending on the number of planned frequencies, we also indicate the number of simulations these results concern for statistical relevance. For our algorithm, even if the percentage decreases with the order in which the frequencies are planned, which is logical, we can see that a rather homogeneous coverage is provided. Doing that, our algorithm maximizes the usage of each resource. We can see that for the greedy coloring algorithm, the frequencies are not all used equally, the first two frequencies are always a lot more planned than the following, the other frequencies are under-used.

## D. Figures

We propose in this subsection figures illustrating the execution of our frequency auto-planning algorithm. In the first figure, Figure 1, we can see the initial cellular network and its coverage simplicial complex representation. In the cellular network figure, the black circles are the communication radii, while the pink ones are the rejection radii.

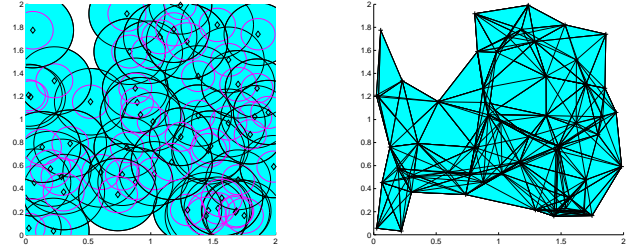


Fig. 1. The cellular network and its coverage representation.

Then the interference graph is represented in Figure 2 next to the frequency planning scheme obtained by our algorithm for the configuration of Figure 1. In the left figure, vertices that can induce interference to each other are linked by an edge. In the figure on the right, a different color represent a different frequency. We can see that our algorithm has planned four frequencies (black, red, green and blue). Finally in Figure 3, we represent the covered area for each frequency of the previously obtained planning.

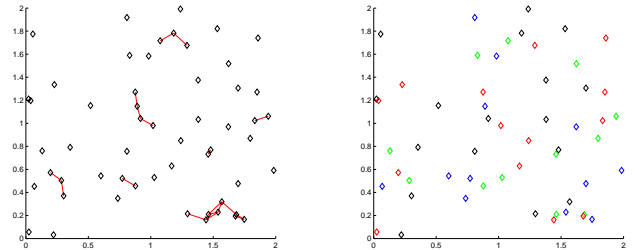


Fig. 2. Interference relations and frequency planning scheme.

## V. SELF-OPTIMIZATION ENERGY CONSERVATION ALGORITHM

### A. Main idea

We consider a cellular network and represent it by its associated coverage simplicial complex. Since we want to use a minimum number of nodes, we consider for each one its maximum coverage radius. Then we construct the Čech complex corresponding to the set of nodes with their maximum covered cells. In this application, we are not only interested in the topology of the network but also by the characterization of the clusters of nodes by simplices. Indeed, we intend to optimize the number of switched-on nodes for the user traffic requested in a given area. This area will be defined by the nodes that are serving it.

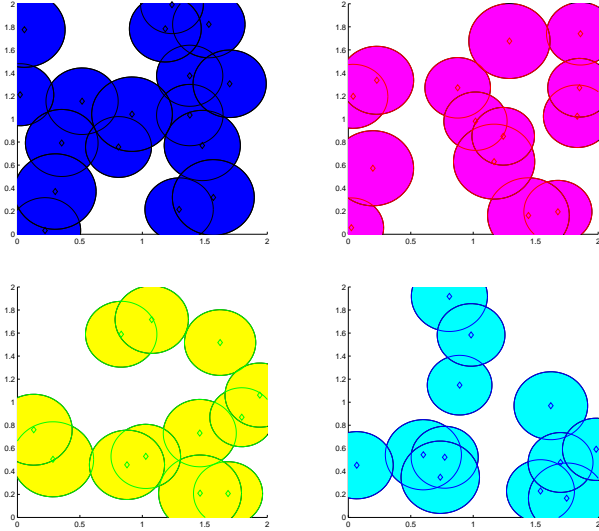


Fig. 3. Coverage for each frequency.

Before beginning the algorithm, we have to define how we will represent user demand. To do this we will create groups of vertices. These groups have to make sense geometrically: they need to represent clusters of vertices. So, every group will be defined by a simplex. Then these groups have to be defined such that each vertex is part of one group exactly. To consider user traffic, we will assign a traffic to each group: for every group of  $k$  vertices/nodes, we will draw uniformly an integer number between zero and  $k$  that is the number of required nodes to keep switched-on. This is what will represent the required QoS for the cellular network. This QoS metric is quite artificial, but the algorithm can take into account any QoS in terms of number of resources required from a given pool of resources available in a group of nodes. This one is the easier to implement if we want to consider that a same pool of resources is provided by nodes geographically close to each other with no location information.

The algorithm begins by the computation of the Betti numbers, since we do not want to modify the network's topology by turning off nodes. Then we compute the 2-simplices degrees and the vertices indices of the coverage reduction algorithm. As in the reduction algorithm of Section III, the order for the removal of vertices follows the principle of the coverage reduction algorithm. But the breaking point is different. Instead of stopping when the area is covered by a minimum number of vertices, the algorithm stops when each group has been reduced to its required QoS.

Then given this configuration of switched-on nodes, i.e. kept vertices, the algorithm tries to reduce as much as possible the coverage radius of each node without creating a coverage hole. The order in which the coverage radii are examined is random.

Finally, we obtain a configuration of vertices that defines the nodes to keep switched-on that is optimal. From the first part of the algorithm, we ensure that enough nodes are kept on to satisfy the user demand. Then, from the second part, we ensure that no energy is spend uselessly by optimizing the size of the serving cells.

## B. Algorithm description

We give in Algorithm 2 the full energy conservation algorithm. It requires the set of vertices  $\omega$  and their coverage radii, then returns the list of kept vertices with their new coverage radii.

---

### Algorithm 2 Energy conservation algorithm

---

**Require:** Set  $\omega$  of  $N$  vertices, for each vertex  $v$  its coverage radius  $r_v$ .

Computation of the Čech complex  $X = C_r(\omega)$   
 Creation of the list of boundary vertices  $L_C$   
 Computation of  $\beta_0(X)$  and  $\beta_1(X)$   
 Computation of  $D_1(X), \dots, D_{s_2}(X)$   
 Computation of  $I[v_1(X)], \dots, I[v_{s_0}(X)]$

**for all**  $v \in L_C$  **do**  
    $I[v] = -1$   
**end for**

$I_{\max} = \max\{I[v_1(X)], \dots, I[v_{s_0}(X)]\}$   
 $N_{\text{group}} = 0$

**for all** Simplex  $S_k \in X$  from largest to smallest **do**  
   **if**  $\forall v \in S_k, \text{Group}(v) == 0$  **then**  
      $N_{\text{group}} = N_{\text{group}} + 1$   
      $\forall v \in S_k \text{Group}(v) = N_{\text{group}}$   
      $\text{Size}(N_{\text{group}}) = k + 1$   
     Draw  $\text{QoS}(N_{\text{group}})$  among  $\{0, \dots, k + 1\}$   
   **end if**  
**end for**

**while**  $I_{\max} > 2$  **and**  $\text{Size} \geq \text{QoS}$  **do**  
   Draw  $w$  a vertex of index  $I_{\max}$   
    $X' = X \setminus \{w\}$   
   Computation of  $\beta_0(X'), \beta_1(X')$   
   **if**  $\beta_0(X') \neq \beta_0(X)$  **and**  $\beta_1(X') \neq \beta_1(X)$  **and**  
    $\text{Size}(\text{Group}(w)) \leq \text{QoS}(\text{Group}(w))$  **then**  
      $I[w] = -1$   
   **else**  
      $\text{Size}(\text{Group}(w)) = \text{Size}(\text{Group}(w)) - 1$   
     Computation of  $D_1(X'), \dots, D_{s'_{k_0}}(X')$   
     **for**  $i = 1 \rightarrow s'_0$  **do**  
       **if**  $I[v_i(X')] == I_{\max}$  **then**  
         Recomputation 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**

**for all**  $v \in X$  taken in random order **do**  
    $X' = X$   
   **while**  $\beta_0(X') == \beta_0(X)$  **and**  $\beta_1(X') == \beta_1(X)$  **do**  
     Reduce  $r_v$   
   **end while**  
    $X = X'$   
**end for**

**return** List of kept vertices  $v$  and their coverage radii  $r_v$ .

---

For our simulations, we choose to give advantage to the larger simplices for the constitution of the groups. Thus, the



first group will consist of the largest simplex, or one randomly chosen among the largest ones, then simplices of smaller size will become groups until every vertex is part of a group. It is possible to consider other rules for the constitution of the groups, but it has to follow two conditions: every vertex must pertain to at least a group, and must not pertain to more than a group. The groups are represented by the variable ‘Group’, that for each vertex gives its group number. Then ‘QoS’ represents the minimum number of vertices required for a given group, while ‘Size’ represents its number of vertices.

We can see that the breaking point of the ‘while’ loop of the algorithm takes the ‘QoS’ parameter into account. Then a vertex can be removed if and only if it does not modify the number of connected component and the number of coverage holes, and it is not needed for the QoS requirements. Coverage radii reduction for each vertex is done in the last loop.

### C. Performance

For simulation reasons, we only consider the Vietoris-Rips complex in the 2-dimensional case. We choose the groups to be simplices from the largest to the smallest as in the algorithm description.

We compare the performance of our algorithm to an optimal, not always achievable solution. Indeed we do not know of a energy conservation algorithm that switch-off vertices during off-peak hours while maintaining coverage. We compare the number of switched-on vertices after the execution of our energy conservation algorithm, to the number of vertices needed for the QoS, given by the ‘QoS’ parameter. It is important to note that this optimal solution is not always achievable since it does not take into account that the area is to stay covered. Some vertices have to be kept for traffic reasons, while other are kept to maintain connectivity and/or coverage.

Our simulation results are computed on 1000 configurations of Poisson point processes of intensity  $\lambda = 6$  on a square of size  $a = 2$  with a fixed boundary of vertices. The coverage radii are sampled uniformly between  $a/10$  and  $a^2/\lambda$ , except for the boundary vertices for which it was set to  $1/\sqrt{\lambda}$ . For each group the ‘QoS’ number is a sampled integer between zero and the size of the group. We denote by  $N_o$  the optimal number of vertices, and by  $N_k$  the number of kept vertices with our energy conservation algorithm. First we compute the percentage of simulations for which we have a given difference between the obtained number and the optimal number of kept vertices.

$N_k - N_o$	0	1	2	3	4
Occurrence	8.3%	18.7%	24.5%	21.7%	13.6%
$N_k - N_o$	5	6	7	8 – 10	
Occurrence	8.6%	3.3%	1.0%	0.3%	

TABLE IV  
OCCURRENCES OF GIVEN DIFFERENCES BETWEEN  $N_k$  AND  $N_o$ .

We can see in Table IV the percentage of simulations in which the number of kept vertices is different from the optimal number of vertices. For 8.6% of the simulations the optimal

number is reached. In 87.1% of our simulations the difference between the optimal and the effective number of kept vertices is smaller than 4, and it never exceeds 10. The number of boundary vertices is 12, these vertices are not removable (in order to never shrink the covered area). So less than 10 vertices needed to cover the whole area on top of the needed vertices for the traffic seems plausible.

To have more advanced comparison, for the 1000 configurations, we compute the optimal number of vertices. Then for each optimal number of vertices  $N_o$  that occurred the most, we compute the mean number of kept vertices  $\mathbf{E}[N_k|N_o]$  over the simulations which have  $N_o$  for optimal number. The results are given in Table V. For comparison, we also compute the difference between  $N_o$  and  $N_k$  in percent. We finally indicate which percent of our 1000 simulations these cases occur to show the relevance of these statistical results.

$N_o$	20	21	22	23	24
$\mathbf{E}[N_k N_o]$	23.40	24.14	24.91	25.06	26.43
Difference	17.0%	14.9%	13.2%	8.9%	10.1%
Occurrence	7.0%	7.3%	7.9%	6.9%	6.1%
$N_o$	25	26	27	28	
$\mathbf{E}[N_k N_o]$	27.37	27.91	28.84	29.56	
Difference	9.5%	7.3%	6.8%	5.6%	
Occurrence	6.8%	7.9%	6.8%	5.4%	

TABLE V  
MEAN NUMBER OF KEPT VERTICES  $\mathbf{E}[N_k|N_o]$  FOR A GIVEN OPTIMAL NUMBER  $N_o$ .

We can see in Table V that the more vertices are needed, the less difference there is between our result and the optimal one. Indeed, if more vertices are needed, there is a great chance that these vertices can cover the whole area.

### D. Figures

We propose in this subsection figures illustrating the execution of our energy conservation algorithm. In the first figure, Figure 4, we can see the initial cellular network and its coverage simplicial complex representation. The vertices have different coverage radii, and there is no fixed boundary. The boundary vertices are in red.

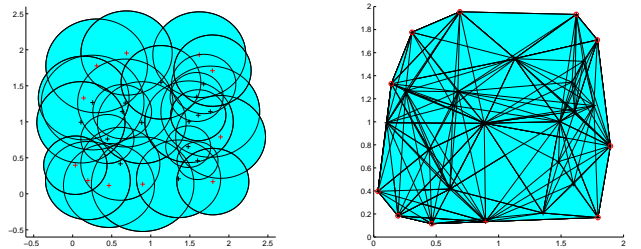
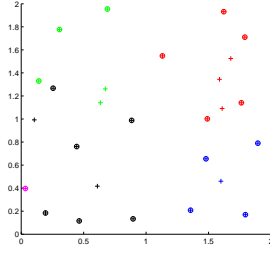


Fig. 4. The cellular network and its coverage representation.

Then for the configuration of Figure 4, we represent he groups of nodes in different colors, and give a table with their corresponding QoS and size in Figure 5. The kept vertices are circled.



Color	Black	Red	Blue	Green	Magenta
QoS	6	5	4	2	0
Size	8	8	5	5	1

Fig. 5. Groups of QoS.

Finally in Figure 6, we can see the final configuration of the cellular network with the optimized coverage radii and its corresponding simplicial complex for the configuration of Figure 4.

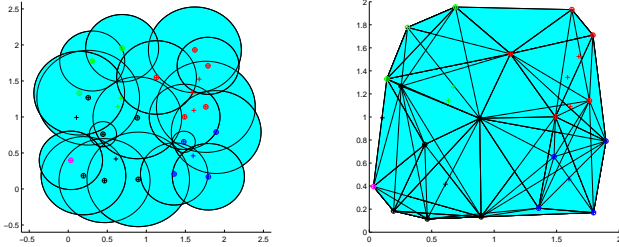


Fig. 6. Final configuration.

## VI. DISASTER RECOVERY ALGORITHM

In this section we present a disaster recovery algorithm introduced in [27] of which we remind the description and investigate more thoroughly the performance, especially its smoothed robustness.

### A. Main idea

We consider a damaged cellular network presenting coverage holes with a fixed boundary, in order to know the domain to cover. We consider as inputs the set of existing vertices: the nodes of a damaged cellular network, and their coverage radii. We also need a list of boundary nodes, these nodes can be fictional, but we need to know the whole area that is to be covered. Then we build the Čech complex characterizing the coverage of the network, the Betti number  $\beta_1$  of the Čech complex counting the number of coverage holes of the wireless network. In this section, we restrict ourselves to wireless networks with a fixed communication radius  $r$ .

The algorithm begins by adding new vertices in addition to the set of existing vertices presenting coverage holes. We suggest here the use of the determinantal addition method. It is possible to consider any vertices addition methods must they be deterministic or random based: flexibility is one of the greatest advantage of our algorithm. In particular, it is possible

to consider a method with pre-defined positions for some of the vertices in real-life scenarii.

We choose that the number of added vertices, that we denote by  $N_a$ , is determined as follows. First, it is set to be the minimum number of vertices needed to cover the whole area minus the number of existing vertices. This way, we take into account the number of existing vertices, that we denote by  $N_i$ . Then the Betti numbers  $\beta_0$  and  $\beta_1$  are computed via linear algebra thanks to the simplicial complex representation. If there is still more than one connected component, and coverage holes, then the number of added vertices  $N_a$  is incremented with a random variable  $u$  following an exponential growth:

- $N_a := \lceil \frac{a^2}{\pi r^2} \rceil - N_i$ .
- After adding the  $N_a$  vertices, if  $\beta_0 \neq 1$  or  $\beta_1 \neq 0$ , Then,  $N_a = N_a + u$ , and  $u = 2 * u$ .

Using determinantal point processes allows us to not only take into account the number of existing vertices, via the computation of  $N_a$ , but we also take into account the existing vertices positions, then every new vertex position as it is added. It suffices to consider the  $N_i$  existing vertices as the  $N_i$  first vertices sampled in the process, then each vertex is taken into account as it is drawn.

The next step of our approach is to run the coverage reduction algorithm presented in Section III which maintains the topology of the cellular network: the algorithm removes vertices from the 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.

### B. Algorithm description

We give in Algorithm 3 the outline of the algorithm. The algorithm requires the set of initial vertices  $\omega_i$ , the fixed coverage radius  $r$ , as well as the list of boundary vertices  $L_b$ . It is important to note that only connectivity information is needed to build the Čech complex.

---

#### Algorithm 3 Disaster recovery algorithm

---

**Require:** Set of vertices  $\omega_i$ , radius  $r$ , boundary vertices  $L_b$   
 Computation of the Čech complex  $X = \mathcal{C}_r(\omega_i)$   
 $N_a = \lceil \frac{a^2}{\pi r^2} \rceil - N_i$   
 Addition of  $N_a$  vertices to  $X$  following a determinantal point process  
 Computation of  $\beta_0(X)$  and  $\beta_1(X)$   
 $u = 1$   
**while**  $\beta_0 \neq 1$  or  $\beta_1 \neq 0$  **do**  
    $N_a = N_a + u$   
    $u = 2 * u$   
   Addition of  $N_a$  vertices to  $X$  following chosen method  
   Computation of  $\beta_0(X)$  and  $\beta_1(X)$   
**end while**  
 Coverage reduction algorithm on  $X$   
**return** List  $L_a$  of kept added vertices.

---

### C. Performance

We now compare the performance results of our disaster recovery algorithm to the most known coverage recovery algorithm: the greedy algorithm for the set cover problem. First, we compare the mean number of added vertices.

Results presented in Table VI are computed in mean over 1000 simulations for each algorithm in different scenarii on a square of side  $a = 1$  with coverage radius  $r = 0.25$ , and a Vietoris-Rips complex. Scenarii 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. They 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, number of added vertices and their positions being exactly the same.

% of area initially covered	20%	40%	60%	80%
Greedy algorithm	3.69	3.30	2.84	1.83
Homology algorithm	4.42	3.87	2.97	1.78

TABLE VI  
MEAN FINAL NUMBER OF ADDED VERTICES  $\mathbf{E}[N_f]$

The numbers of vertices added in the final state both with our recovery algorithm and the greedy algorithm are roughly the same. They both tend to the minimum number of vertices required to cover the uncovered area depending on the initial configuration. Nonetheless, we can see that our algorithm performs a little bit worse than the greedy algorithm in the less covered area scenarii because the vertices are not optimally positioned and it can be seen when just a small percentage of area is covered, and whole parts of the grid from the greedy algorithm are used, instead of isolated vertices. In compensation, our homology algorithm performs better in more covered scenarii.

To show the advantages of our disaster recovery algorithm we choose to evaluate the robustness of the algorithm when the added vertices positions are slightly moved, i.e. when the nodes positioning does not strictly follow the theoretical positioning. In order to do this, we apply a Gaussian perturbation to each the added vertices position. The covariance matrix of the perturbation is given by  $\Sigma = \sigma^2 \text{Id}$  with  $\sigma^2 = 0.01$ , which means that the standard deviation for each vertex is of  $\sigma = 0.1$ . Other simulations parameters are unchanged, results in Table VII and VIII are given in mean over 1000 simulations. First, we compute the average number of holes  $\beta_1$  created by the Gaussian perturbation in Table VII. Then in Table VIII, we counted the percentage of simulations in which the number of holes is still zero after the Gaussian perturbation on the new vertices positions.

We can see that the perturbation on the number of holes decreases with the percentage of area initially covered, since the initial vertices are not perturbed. Our homology algorithm clearly performs better, even in the least covered scenarii, there are less than 50% of simulations that create coverage holes, which is not the case for the greedy algorithm. The greedy

% of area initially covered	20%	40%	60%	80%
Greedy algorithm	0.68	0.65	0.45	0.35
Homology algorithm	0.62	0.53	0.37	0.26

TABLE VII  
MEAN NUMBER OF HOLES  $\mathbf{E}[\beta_1]$  AFTER THE GAUSSIAN PERTURBATION

% of area initially covered	20%	40%	60%	80%
Greedy algorithm	40.8%	47.7%	61.0%	69.3%
Homology algorithm	50.9%	58.1%	67.9%	75.3%

TABLE VIII  
PROBABILITY THAT THERE IS NO HOLE  $\mathbf{P}(\beta_1 = 0)$  AFTER THE GAUSSIAN PERTURBATION

algorithm also always create more coverage holes in mean than our disaster recovery algorithm for the same vertices positions perturbation. Therefore our algorithm seems more fitted to the disaster recovery case when a recovery network is deployed in emergency both indoor, via Femtocells, and outdoor, via a trailer fleet, where exact GPS locations are not always available, and exact theoretical positioning is not always followed.

### D. Figures

We propose in this subsection some figures to illustrate the disaster recovery algorithm. In Figure 7, we can see the initial damaged cellular network with a boundary of fictional nodes presenting coverage holes.

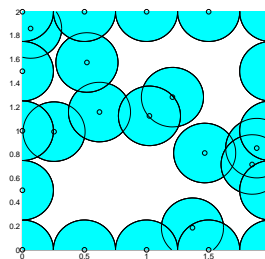


Fig. 7. A damaged cellular network with a fixed boundary.

We can see the first step of the disaster recovery algorithm illustrated in Figure 8 for the cellular network of Figure 7. Existing vertices are black circles while added vertices are red plusses.

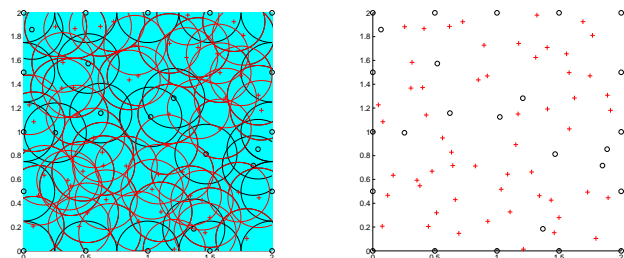


Fig. 8. With the determinantal addition method with a Ginibre basis.

Finally, we can see in Figure 9 an execution of the reduction algorithm on the intermediate configuration of Figure 8 which constitutes of the second and final step of the disaster recovery algorithm. Removed vertices are represented by blue diamonds.

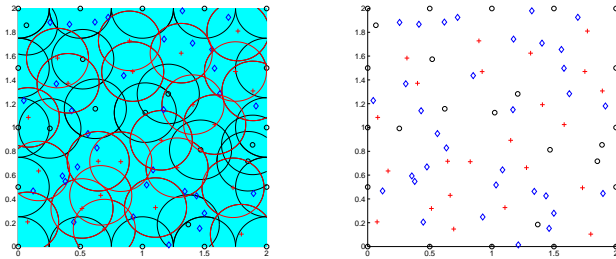


Fig. 9. The coverage reduction algorithm run on the determinantal method example.

## REFERENCES

- [1] 3GPP. TR 36.913: Requirements for further advancements for evolved universal terrestrial radio access (E-UTRA). *Tech. Rep.*, Mar. 2009.
- [2] 3GPP. Telecommunications management; self-healing OAM; concepts and requirements. *Tech. Rep.*, 3GPP TS 32.541 v1.6.1, 2010.
- [3] T. Al-Meshhadany and K. Al-Agha. VCB by means of soft2hard handover in WCDMA. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pages 487–491, 2002.
- [4] O. Aliu, A. Imran, M. Imran, and B. Evans. A survey of self organisation in future cellular networks. *Communications Surveys Tutorials, IEEE*, 15(1):336–361, 2013.
- [5] M. Amirijoo, L. Jorguseski, T. Kurner, R. Litjens, M. Neuland, L. Schmelz, and U. Turke. Cell outage management in LTE networks. In *Wireless Communication Systems, 2009. ISWCS 2009. 6th International Symposium on*, pages 600–604, 2009.
- [6] 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.
- [7] I. Ashraf, L. T. W. Ho, and H. Claussen. Improving energy efficiency of femtocell base stations via user activity detection. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–5, 2010.
- [8] E. Campos-Nañez, A. Garcia, and C. Li. A game-theoretic approach to efficient power management in sensor networks. *Oper. Res.*, 56(3):552–561, 2008.
- [9] V. Chvatal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.
- [10] S. Das, S. Sen, and R. Jayaram. A structured channel borrowing scheme for dynamic load balancing in cellular networks. In *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, pages 116–123, 1997.
- [11] S. Das, H. Viswanathan, and G. Rittenhouse. Dynamic load balancing through coordinated scheduling in packet data systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 786–796 vol.1, 2003.
- [12] L. Du, J. Bigham, and L. Cuthbert. An intelligent geographic load balance scheme for mobile cellular networks. In *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*, pages 348–353, 2002.
- [13] 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.
- [14] T. Fujii and S. Nishioka. Selective handover for traffic balance in mobile radio communications. In *Communications, 1992. ICC '92, Conference record, SUPERCOMM/ICC '92, Discovering a New World of Communications., IEEE International Conference on*, pages 1840–1846 vol.4, 1992.
- [15] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [16] 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.
- [17] S. Haykin. Cognitive radio: brain-empowered wireless communications. *Selected Areas in Communications, IEEE Journal on*, 23(2):201–220, 2005.
- [18] S. Hmlinen, H. Sanneck, and C. Sartori. *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley Publishing, 1st edition, 2012.
- [19] S. Irani. Coloring inductive graphs on-line. *Algorithmica*, 11(1):53–72, 1994.
- [20] P. Jiang, J. Bigham, and J. Wu. Self-organizing relay stations in relay based cellular networks. *Comput. Commun.*, 31(13):2937–2945, Aug. 2008.
- [21] D. Kim, B. Shin, D. Hong, and J. Lim. Self-configuration of neighbor cell list utilizing E-UTRAN nodeB scanning in LTE systems. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5, 2010.
- [22] J. Li and R. Jantti. On the study of self-configuration neighbour cell list for mobile WiMAX. *Next Generation Mobile Applications, Services and Technologies, International Conference on*, 0:199–204, 2007.
- [23] K. Morrison. Rapidly recovering from the catastrophic loss of a major telecommunications office. *Communications Magazine, IEEE*, 49(1):28–35, 2011.
- [24] C. Mueller, M. Kaschub, C. Blankenhorn, and S. Wanke. A cell outage detection algorithm using neighbor cell list reports. In K. Hummel and J. Sterbenz, editors, *Self-Organizing Systems*, volume 5343 of *Lecture Notes in Computer Science*, pages 218–229. Springer Berlin Heidelberg, 2008.
- [25] A. Nasif and B. Mark. Opportunistic spectrum sharing with multiple cochannel primary transmitters. *Wireless Communications, IEEE Transactions on*, 8(11):5702–5710, 2009.
- [26] A. Vergne, L. Decreusefond, and P. Martins. Reduction algorithm for simplicial complexes. In *INFOCOM, 2013 Proceedings IEEE*, pages 475–479, 2013.
- [27] A. Vergne, I. Flint, L. Decreusefond, and P. Martins. Homology based algorithm for disaster recovery in wireless networks. hal-00800520, Mar. 2013.