Simplicial homology based energy saving algorithms for wireless networks

Ngoc-Khuyen Le, Philippe Martins, Laurent Decreusefond, Anaïs Vergne

Simplicial homology based energy saving algorithms for wireless networks

Ngoc-Khuyen Le, Philippe Martins, Laurent Decreusefond and Anaïs Vergne

Institut Telecom, TELECOM ParisTech, LTCI, Paris, France
Email: ngoc-khuyen.le, martins, decreuse, avergne@telecom-paristech.fr

Abstract. Energy saving is one of the most investigated problems in wireless networks. In this paper, we introduce two homology based algorithms: a simulated annealing one and a downhill one. These algorithms optimize the energy consumption at network level while maintaining the maximal coverage. By using simplicial homology, the complex geometrical calculation of the coverage is reduced to simple matrix computation. The simulated annealing algorithm gives a solution that approaches the global optimal one. The downhill algorithm gives a local optimal solution. The simulated annealing algorithm and downhill algorithm converge to the solution with polynomial and exponential rate, respectively. Our simulations show that this local optimal solution also approaches the global optimal one. Our algorithms can save at most 65% of system's maximal consumption power in polynomial time. The probability density function of the optimized radii of cells is also analyzed and discussed.

1 Introduction

Wireless networks provide more and more innovative and powerful services. Operators deploy a large number of base stations to satisfy user demands in terms of coverage and traffic. This evolution leads to higher energy consumption as well as higher human exposure to electromagnetic waves. These problems challenge the development of future wireless technologies. The next generation of wireless networks should take into account environmental and public health considerations while still complying with traffic and coverage requirements.

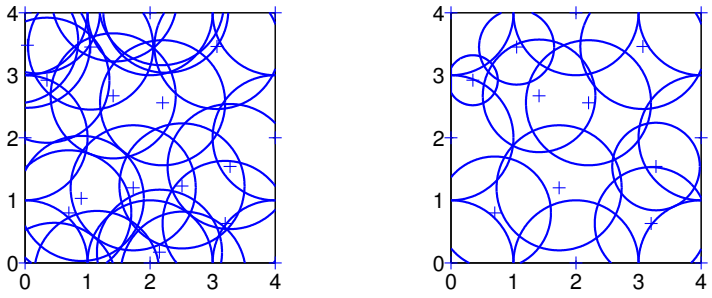
A typical wireless network is composed of many cells. Each cell ensures its own coverage, called local coverage. The coverage of the whole system, or global coverage, is a union of all local coverages. Clearly, some cells can overlap. The overlapping regions may cause transmission power wastage mainly due to interference. One promising approach to save energy at network level is to design new methods that minimize transmission power by adjusting cells' coverage while keeping unchanged global coverage.

Traditionally, to ensure the global coverage, geometrical methods are used. Cells are deployed according to some regular patterns. For example, in mobile phone networks, hexagon is the conventional configuration. In wireless sensor networks, configurations such as hexagon, square grid, etc. are widely used as introduced in [1]. All these patterns ensure the global coverage of system but

they require that cells are identical as well as uniformly distributed. In practice, it is very difficult to deploy cells following these conditions. Cells' sizes are not always identical. In mobile networks, cells' sizes should be different depending on density of users [2]. In high density region, a lot of small cells should be deployed to provide a high transmission capacity. But, in low density regions, a few macro cells are enough to satisfy coverage and traffic constraints. In wireless sensor networks, one might want to implement more than one type of sensors. In critical regions, for exact and frequent information, a collection of small and sensitive sensors are suitable. In non-critical regions, for low deployment costs, big and low-sensitive sensors are acceptable. Furthermore, with a large number of cells, it is difficult to achieve such a precise deployment. Coverage computation becomes an intractable problem. Another approach is to consider cells as a random deployment. In [3], the authors provided a probability method which gives a guaranteed size for cells. The guaranteed size is the smallest size of cells which allows full coverage of the system in probability. This method only considers full coverage problem. It does not take into account the power saving problem. It also assumes that all cells have the same size. In [4], Silva and Ghrist introduced the simplicial complex as a representation of coverage topology. Then, they use homology, a tool from computational algebra, to examine the topology of the network. Coverage properties are then determined by some topological invariants such as Euler characteristic or Betti numbers. Coverage computation via homology is reduced to simple matrix computation. This tool was used in [5] and [6] to detect coverage holes. Their distributed versions are also introduced in [7] and [8]. The accuracy of this method was discussed in [9]. However, all these contributions only consider the problem of coverage holes detection. In [10], Vergne introduced an algorithm which deletes some vertices from a given simplicial complex without modifying its homology. This algorithm can be applied to turn off redundant cells. But, the cells that can be turned off must be completely overlapped with their neighbors. So, this algorithm is only suitable for networks whose cells are deployed with high density. In addition, turning off cells can not optimize the coverage of networks because it does not minimize the overlapping region.

In this paper, we introduce two algorithms: a simulated annealing one and a downhill one to minimize the total consumed power for wireless networks at network level. These algorithms consider a random deployment of cells. They use Čech complex to describe and analyze the coverage structure of cells. Then, these algorithms adjust the radius of each cell to obtain the minimal total consumed power while keeping unchanged the global coverage of the networks. The simulated annealing algorithm is a heuristic whose objective is to find the global optimal solution. The downhill algorithm offers a local optimal solution. Our simulations show that this local optimal solution also approaches the global optimal one. We can see an example of simulated cells before and after optimization in Figure 1. The complexity and the convergence rate of our algorithms are also analyzed. We also discuss about the probability density function (pdf) of opti-

mized cell's radius. The pdf gives ideas of the optimal size of cell and how many cells are enough to cover the space.



(a) Before optimization (b) After optimization

Fig. 1: A network before and after optimization.

The rest of this paper is organized as follows. Section 2 briefly introduces the theory of simplicial homology. The next section is devoted to the description of our algorithms. Our simulation and results are given in the Section 4. In the last section, some conclusions are drawn and future work directions are outlined.

2 Simplicial homology

In this section, notions of simplicial homology are briefly introduced. For further details, see documents [11] and [12]. Given a set of vertices V, a k-simplex is an unordered subset $\{v_0, v_1, \ldots, v_k\}$, where $v_i \in V$ and $v_i \neq v_j$ for all $i \neq j$. The number k is its dimension. Figure 2 presents some examples: a 0-simplex is a point, a 1-simplex is a segment of line, a 2-simplex is a filled triangle, a 3-simplex is a filled tetrahedron, etc.

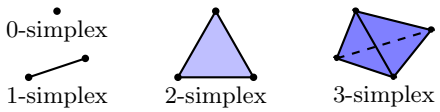


Fig. 2: Examples of simplices.

An oriented simplex is an ordered type of simplex, where swapping position of two vertices changes its orientation. The change of orientation is represented by a negative sign as:

$$[v_0, v_1, \ldots, v_i, v_j, \ldots, v_k] = -[v_0, v_1, \ldots, v_j, v_i, \ldots, v_k]$$

Removing a vertex from a k-simplex creates a $(k-1)$-simplex. This $(k-1)$-simplex is called a face of the k-simplex. Thus, each k-simplex has $(k+1)$ faces.

An abstract simplicial complex is a collection of simplices such that: every face of a simplex is also in the simplicial complex. Let X be a simplicial complex. For each $k \geq 0$, we define a vector space $C_k(X)$ whose basis is a set of oriented k-simplices of X. If k is bigger than the highest dimension of X, let $C_k(X) = 0$. We define the boundary operator to be a linear map $\partial : C_k \to C_{k-1}$ as follows:

$$\partial[v_0, v_1, \ldots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k]$$

This formula suggests that the boundary of a simplex is the collection of its faces, as illustrated in Figure 3. For example, the boundary of a segment is its two endpoints. A filled triangle is bounded by its three segments. A tetrahedron has its boundary comprised of its four faces which are four triangles.

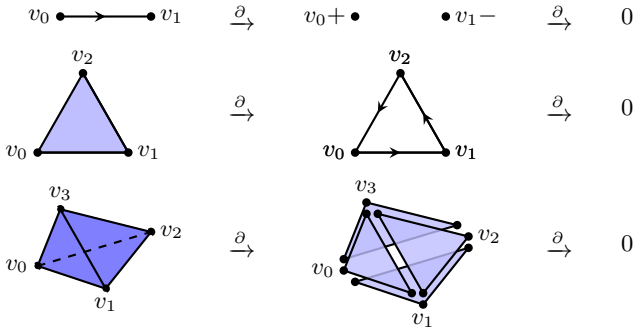


Fig. 3: Boundary operator.

The composition of boundary operators gives a chain of complexes:

$$\cdots \xrightarrow{\partial} C_{k+1} \xrightarrow{\partial} C_k \xrightarrow{\partial} C_{k-1} \cdots \xrightarrow{\partial} C_1 \xrightarrow{\partial} C_0 \xrightarrow{\partial} 0$$

Consider two subspaces of $C_k(X)$: cycle-subspace and boundary-subspace, denoted as $Z_k(X)$ and $B_k(X)$ respectively. Let ker be the kernel space and im be the image space. By definition, we have:

$$Z_k(X) = \ker(\partial : C_k \to C_{k-1})$$
$$B_k(X) = \mathrm{im}(\partial : C_{k+1} \to C_k)$$

$Z_k(X)$ includes cycles which are not boundaries while $B_k(X)$ only includes boundaries. A k-cycle u is said homologous with a k-cycle v if their difference is a k-boundary: $[u] \equiv [v] \iff u - v \in B_k(X)$. A simple computation shows that $\partial \circ \partial = 0$. This result means that a boundary has no boundary. Thus, the k-homology of X is the quotient vector space:

$$H_k(X) = Z_k(X) \backslash B_k(X)$$

The dimension of $H_k(X)$ is called the k-th Betti number:

$$\beta_k = \dim H_k = \dim Z_k - \dim B_k \tag{1}$$

This number has an important meaning for coverage problems. The k-dimensional Betti number counts the number of k-dimensional holes in a simplicial complex. For example, the 0-dimensional Betti number counts the connected components while 1-dimensional Betti number counts the coverage holes, etc. In our algorithm, we only consider these two first Betti numbers.

Definition 1 (Čech complex). *Given (M, d) a metric space, ω a finite set of points in M and $\epsilon(\omega)$ a sequence of real positive numbers, the Čech complex with parameter $\epsilon(\omega)$ of ω, denoted $C_{\epsilon(\omega)}(\omega)$, is the abstract simplicial complex whose k-simplices correspond to non-empty intersection of $(k+1)$ balls of radius $\epsilon(\omega)$ centered at the $(k+1)$ distinct points of ω.*

If we choose $\epsilon(\omega)$ to be the cell's coverage range R, the Čech complex verifies the exact coverage of the system. In the Čech complex, each cell is represented by a vertex. A covered space between cells corresponds to a filled triangle, tetrahedron, etc. In contrast, a coverage hole between cells corresponds to an empty (or non-filled) triangle, rectangle, etc.

Definition 2 (Index of a vertex). *The index of a vertex v is the biggest integer k such that for every $i \leq k$ each $(i-1)$-simplex of v is a face of at least one i-simplex of v.*

The index of a vertex tells us how many times the corresponding cell of this vertex overlaps with its neighbors. An index of zero indicates that corresponding cell separates from others (it's isolated). A cell whose index is one connects to others by edges. A cell whose index is higher than one connects with others by triangles, tetrahedra etc.

3 Energy saving algorithms

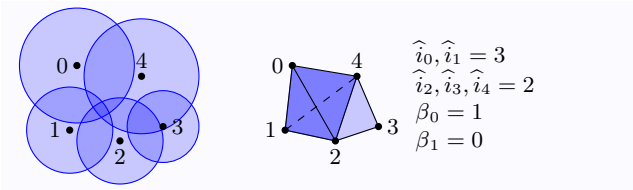
We consider a wireless network whose cells are randomly distributed on the plane. These cells can have different coverage radii. Let N be the number of cells. It is assumed that each cell i can operate with the coverage radius R_i which varies from $R_{\min,i}$ to $R_{\max,i}$, where $i = 0, 1, ..., N$. The consumed power for each cell is estimated by using the simplified path loss model: $P_{t,i} = K_0 R_i^{\gamma}$, where K_0 is a constant factor and γ is the path loss exponent. In this paper, we assume that $K_0 = 1$ for simplification. The total consumed power for all N cells is:

$$P_T = \sum_{i=1}^{N} P_{t,i} = \sum_{i=1}^{N} R_i^{\gamma}.$$

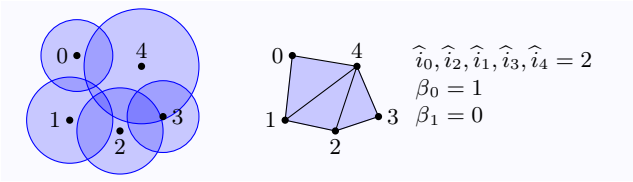
Our algorithms have the objective to minimize the total consumed power P_T while providing the maximal global coverage for the network.

At the beginning of our algorithms, all cells are turned on and each cell i works with its own maximal radius $R_{\max,i}$. At this state, the system has the maximal global coverage and also its cells strongly overlap. An initial Čech complex is constructed and its Betti numbers β_0^* and β_1^* are computed. The construction of the Čech complex is based on the verification of intersection between cells. For the detail of this construction, see the document [13]. The Betti numbers can be computed by equation (1). The Čech complex tells us about the coverage structure of the system. We then adjust the radius for each cell i to minimize the overlapping region without any modification to the global coverage. We are solving the power saving problem, so we do not locate the coverage holes, if any. We assume that, all fence cells and boundary cells are known before an execution of our algorithms. Our algorithms only modify the coverage radius for internal cells and do not make any change to fence cells or boundary cells. There are two strategies to adjust the radius of cells. The radius of a cell can be increased or decreased depending on each strategy. The simulated annealing algorithm allows both reduction and enlargement of cells but the downhill algorithm only accepts reduction. After an enlargement, there is no new coverage hole. However, after a reduction, a new coverage hole may appear. So, we need to verify the global coverage of the system. To do that, we recompute the Čech complex after a reduction. The Betti numbers β_0 and β_1 are also recomputed. If the updated Betti numbers β_0 and β_1 are the same as the initial ones β_0^* and β_1^*, no new hole has appeared. The reduction is accepted. Conversely, if the Betti numbers have changed, the reduction is refused. We can see an example of the global coverage verification in Figure 4. This figure represents the internal cells of a network. The external area of these cells are covered by fence cells. To emphasis the verification if a new coverage hole appears, we do not draw the fence cells here. In Figure 4a, all cells are at their initial stage. Each cell reaches its maximal range. These cells are highly overlapped. They are represented by one tetrahedron and one triangle. The Betti numbers are $\beta_0 = 1$ and $\beta_1 = 0$. They indicate that all cells are joined together in one component and there is no hole. Cell 0 and 1 have index three. In Figure 4b, after the reduction of radius for cell 0, the Čech complex now includes three filled triangles. The Betti numbers β_0 and β_1 are not changed but the indices of cell 0 and cell 1 are now reduced to two. This means that the overlapping level is reduced but no hole appears. The algorithm accepts the reduction of cell 0. But, in Figure 4c, cell 4 tries to reduce its radius and creates a new coverage hole. This hole is represented by an empty rectangle and the first Betti number β_1 is now changed to one. The indices of cell 0, cell 1, cell 2, cell 4 are all reduced to one because these cells are now on the border of an empty hole. Then, the reduction of cell 4 is refused. Its radius is returned to its previous value.

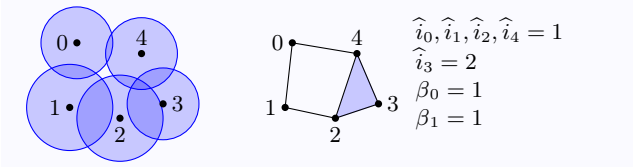
The adjustment process continues until the terminal condition is met. The terminal conditions in the simulated annealing algorithm and the downhill one are different. These conditions are mentioned in the detail description of these algorithms.



(a) Initial stage, each cell reach its own $R_{\max}$.



(b) First reduction, cell 0 is accepted to reduce radius.



(c) Second reduction, cell 4 is refused to reduce radius.

Fig. 4: Reduction of cell radius and Čech representation.

3.1 Simulated annealing algorithm

Simulated annealing (SA) is a method introduced to solve various difficult optimization problems. Starting with an initial solution, where all cells are set to their maximal radius, this algorithm approaches the final global optimized solution by the following process: it randomly chooses a cell and attempts to increase or decrease its radius by an amount of ΔR. Then, the difference of power consumption is calculated by $\Delta P = (R \pm \Delta R)^\gamma - R^\gamma$, where γ is the path loss exponent. If the radius is decreased, the transition is called downhill move. This downhill move is only accepted if no hole appears. The SA algorithm does not use the index system, so the Čech complex with dimension two is enough to verify coverage holes. If the radius is increased, the transition is called uphill move. In this case, $\Delta P > 0$, this uphill move is accepted with probability $\exp(-\Delta P/T)$ where T is the current temperature of the system (a control parameter). Thanks to uphill moves, the process can jump out from a local minimum to search for the global minimum. This process is repeated L times at this temperature to get the thermodynamic equilibrium state. After that, the temperature is gradually decreased by a cooling schedule $T_k = T_0\alpha^k$. In this function, T_0 is the initial temperature which is chosen large enough to make the probability of an uphill move at initial state to be close to one. The cooling factor α, a real positive number such that $0 < \alpha < 1$, guarantees a smooth cooling schedule. The number k is an index varying from 0 to K. At each temperature T_k, the process is

repeated L times until the final temperature T_K. The number K is chosen large enough to make the probability of accepting an uphill move to be near zero at the final temperature. The final configuration of radius is then close to the global minimum solution. For more details about the SA algorithm, see papers [14] and [15]. Our SA algorithm is summarized in Algorithm 1.

Algorithm 1 Simulated annealing algorithm

```
ℂ ← collection of cells;
set R(c) ← R_max(c) for each c ∈ ℂ
set T_0, K, L, α;
build initial Čech complex and compute initial β_0^*, β_1^*;
for k = 1 → K do
  T = α^k T_0;
  for l = 1 → L do
    c ← a random cell ∈ ℂ;
    sign ← a random value ∈ {−1; 1};
    if sign = −1 then
      R_c ← R_c − ΔR_c;
      build Čech complex and compute β_0 and β_1;
      if β_0 ≠ β_0^* or β_1 ≠ β_1^* then
        R_c ← R_c + ΔR_c;
      end if
    else
      compute ΔP ← (R_c + ΔR_c)^γ − R_c^γ;
      R_c ← R_c + ΔR_c with probability exp(−ΔP/T);
    end if
  end for
end for
return collection of optimal cells ℂ
```

3.2 Downhill algorithm

In the downhill algorithm, only the reductions of cells are accepted so it only gives the local optimal solution. This algorithm concerns the overlapping level of cells. The overlapping level of each cell can be known by computing the index of the corresponding vertex in the Čech complex.

From the definition of the index of a vertex, we verify if each k-simplex of v is a face of at least one $(k+1)$-simplex of v. Starting at $k = 0$, it increases k by one if all k-simplices are verified. The algorithm stops if there is a k-simplex which is not a face of any $(k+1)$-simplex. The highest value of k is the index of v. The index computation for a vertex v is described in Algorithm 2.

Let the index of a vertex be also the index of the corresponding cell. The reduction process begins at the cell whose index is the highest. If there are more than one cell whose indices are maximal, the larger cell is chosen. After each reduction of radius, the coverage of the system is verified. If no new hole appeared,

Algorithm 2 Index computation for a vertex v

$k \leftarrow 1$;
while exist k-simplex of v **do**
  $S_k \leftarrow$ collection of k-simplices of v;
  $S_{k+1} \leftarrow$ collection of $(k+1)$-simplices of v;
  for each k-simplex s in S_k **do**
    if s is not a face of any $(k+1)$-simplex $\in S_{k+1}$ **then**
      return k;
    end if
  end for
  $k \leftarrow k+1$;
end while
return k;

the reduction is accepted. Otherwise, the reduction is refused. This cell is flagged as not reducible and its index is set to -1 to avoid a repeating reduction. Its radius is also reversed to its previous value. The reduction process continues with another cell. This process terminates when every cell is irreducible. The details of the reduction process is introduced in Algorithm 3.

Algorithm 3 Downhill algorithm

$\mathbb{C} \leftarrow$ collection of cells
set $R(c) \leftarrow R_{\max}(c)$ for each $c \in \mathbb{C}$
$\mathbb{X} \leftarrow$ build initial Čech complex for $\mathbb{C}$;
compute initial Betti's numbers β_0^*, β_1^*;
compute index $\hat{i}_c$ for each $c \in \mathbb{C}$; {call to Algorithm 2}
flag fence, boundary cells as not reducible;
while exist a reducible cell **do**
  $\mathbb{C}^* \leftarrow$ a set of cells whose index $= \max\{\hat{i}_c | c \in \mathbb{C}\}$
  $c \leftarrow$ a cell $\in \mathbb{C}^*$ whose biggest radius;
  $\mathbb{X}^* \leftarrow \mathbb{X}$
  if $R_c - \Delta R_c \geq R_{c,\min}$ **then**
    $R_c \leftarrow R_c - \Delta R_c$;
  else
    turn off cell c;
  end if
  $\mathbb{X} \leftarrow$ rebuild Čech complex for $\mathbb{C}$; {call to Algorithm 4}
  compute β_0, β_1;
  if $\beta_0 \neq \beta_0^*$ or $\beta_1 \neq \beta_1^*$ **then**
    $R_c = R_c + \Delta R_c$;
    $\mathbb{X} \leftarrow \mathbb{X}^*$
    set cell c is not reducible and set index of c to -1;
  end if
  compute index for reducible cells; {call to Algorithm 2}
end while
return collection of optimal cells $\mathbb{C}$;

The reduction progress requires many recomputations of the Čech complex for the global coverage. In fact, a reduction of one cell's coverage can only make topology change for the region comprised of this cell and its neighbors. The recomputation of global Čech complex is reduced to the Čech computation for this region as follows:

Algorithm 4 Quick Čech complex re-build algorithm

$X \leftarrow$ the old Čech complex before radius changed;
$c \leftarrow$ the cell which changed radius;
$v \leftarrow$ the corresponding vertex of cell c in X;
$\mathbb{N} \leftarrow$ is neighbors collection of c;
$\mathbb{C}^* = \mathbb{N} \cup \{c\}$;
for each simplex u in X **do**
  remove u from X if $v \in u$;
end for
$Y \leftarrow$ build Čech complex for $\mathbb{C}^*$;
for each simplex u in Y **do**
  add u to X if $v \in u$;
end for
return X;

3.3 Complexity

Both simulated annealing algorithm and downhill algorithm require two main steps: the construction of the Čech complex and the computation of the Betti numbers. The downhill algorithm needs an additional step: computation of the indices of the vertices. We first compute the complexity for the construction of the Čech complex, the computation of the Betti numbers and the indices of vertices. Then, the complexity of our algorithms can be easily deduced.

At the beginning of these algorithms, the initial Čech complex is constructed. The construction of the Čech complex is to verify if a group of cells has a non-empty intersection. The 0-simplices are obviously a collection of vertices. Computing 1-simplices is to search neighbors for each cell. Its complexity is C_N^2, where N is the number of cells. To compute the 2-simplices, for each cell we take two of its neighbors and verify if this cell and the two neighbors have a non-empty intersection. Let n be the average number of neighbors of each cell, the complexity to compute the 2-simplices for each cell is C_n^2 on average. The complexity of the 2-simplices computation for all cells is NC_n^2. Similarly, to compute the k-simplices for each cell, we take k of its neighbors and verify if this cell and the neighbors have a non-empty intersection. The complexity of the k-simplices computation of one cell is C_n^k and for all cells is NC_n^k. The complexity to construct the initial Čech complex is: $\mathrm{CP}(\text{initial Čech}) = C_N^2 + N\sum_{k=2}^{d_{\max}} C_n^k$, where $d_{\max}$ is the highest dimension of the Čech complex. The complexity to construct the Čech complex up to dimension 2 is only $O(N^2 + Nn^2)$. If the Čech

complex is built up to its highest dimension, the sum $\sum_{k=2}^{d_{\max}} C_n^k$ can be upper bounded by 2^n if $d_{\max}$ is high enough. The complexity to construct the Čech complex up to the highest dimension is then as much $O(N^2 + N2^n)$.

After each reduction, we need to rebuild the Čech complex. This computation is only for the cell whose radius is reduced and its neighbors. Hence, the complexity to rebuild the Čech complex is CP(rebuilt Čech) $= C_{n+1}^2 + (n+1)\sum_{k=2}^{d_{\max}} C_n^k$. It is about $O(n^3)$ if the Čech is constructed up to dimension 2 and $O(n2^n)$ if the Čech complex is constructed up to its highest dimension.

The Betti numbers computation can be done following equation (1). Its complexity has been discussed in [16]. Let m_k be the average number of k-simplices of the Čech complex. The computation of the Betti number β_0 has complexity of $O(m_1)$. The computation of the Betti number β_1 has complexity equivalent to the complexity of rank computation of a matrix with m_2 rows and m_2 columns. Then, the complexity to compute β_1 is $O(m_2^3)$. So the complexity to compute these two Betti numbers is CP(Betti numbers) $= O(m_1 + m_2^3)$. The number of k-simplices is upper bounded by NC_n^k, so the complexity to compute both these numbers is $O(N^3 n^6)$.

To compute the indices for each vertex i, we firstly find all k-simplices for this vertex, for all $k = 1, 2, \ldots, d_{\max}$. This step has the complexity of $\sum_{k=0}^{d_{\max}} m_k$. For each k-simplex of this vertex i, we need to check if this k-simplex is a part of every $(k+1)$-simplices. This step has the complexity of $\sum_{k=0}^{d_{\max}-1} m_k m_{k+1}$. The complexity of index computation for one cell is $\sum_{k=0}^{d_{\max}} m_k + \sum_{k=0}^{d_{\max}-1} m_k m_{k+1}$. Then, the complexity to compute indices for all cells is CP(indices) $= N(\sum_{k=0}^{d_{\max}} m_k + \sum_{k=0}^{d_{\max}-1} m_k m_{k+1})$. Because we have $m_k \leq NC_n^k$, then we have CP(indices) $\leq N^2 \sum_{k=0}^{d_{\max}} C_n^k + N^3 \sum_{k=0}^{d_{\max}} C_n^k C_n^{k+1}$. If the Čech complex is only constructed up to dimension 2, the complexity to compute indices for all cells is $O(N^2 n^3)$. If the Čech complex is constructed up to its highest dimension, the sum $\sum_{k=0}^{d_{\max}} C_n^k$ can be upper bounded by 2^n. Using the identity of Vandermonde, we can approximate the sum $\sum_{k=0}^{d_{\max}} C_n^k C_n^{k+1}$ by C_{2n}^{n-1}. It's shown that $C_{2n}^{n-1} < 2^{2n}$. The complexity to compute indices for all cells is then as much $O(N^3 2^{2n})$.

In the simulated annealing algorithm, an initial Čech complex is constructed and then the simulated annealing process is repeated for a huge number of inner and outer loops. Let w be the number of loops, we have $w = KL$. The complexity of the simulated annealing algorithm is:

$$\text{CP(SA)} = \text{CP(initial Čech)} + w\big(\text{CP(rebuilt Čech)} + \text{CP(Betti numbers)}\big)$$

In downhill algorithm, after constructing the initial Čech complex, the reduction process is repeated until there is no redundant space. Let S be the deployment space. The reducible space is $S_{\text{reducible}} = N\pi R_{\max}^2 - S$. The average space reduced after each reduction of radius for a cell is $\triangle S = \mathbb{E}\big(\pi R^2 - \pi(R - \triangle R)^2\big)$which equals $\pi R_{\max} \triangle R$. The number of loops w is $S_{\text{reducible}}/\triangle S = NR_{\max}/\triangle R - S/(\pi R_{\max} \triangle R)$. The complexity of the downhill algorithm is:

$$\begin{aligned}\text{CP(downhill)} = \text{CP(initial Čech)} + w\big(&\text{CP(rebuilt Čech)} \\ &+ \text{CP(Betti numbers)} + \text{CP(indices)}\big)\end{aligned}$$

The Table 1 list the complexity of our algorithms in the worst case in two probabilities: the Čech complex is constructed up to dimension 2 and up to its highest dimension.

Table 1: The complexity in the worst case

Complexity	$d_{\max} = 2$	$d_{\max} = \infty$
initial Čech complex	$O(N^2 + Nn^2)$	$O(N^2 + N2^n)$
rebuilt Čech complex	$O(n^3)$	$O(n2^n)$
Betti's numbers	$O(N^3n^3)$	$O(N^3n^6)$
indices	$O(N^2n^3)$	$O(N^32^{2n})$
SA algorithm	$O(KLN^3n^6)$	-
Downhill algorithm	$O(N^4n^6)$	$O(N^42^{2n})$

Note that, in the SA algorithm, the Čech complex is only built up to dimension 2. The SA algorithm and the downhill algorithm with the Čech complex built up to dimension 2 have polynomial time complexity. However, the number of inner loops K and outer loops L in SA algorithm are much greater than the number of cells N. So, the SA algorithm has higher complexity than the downhill one with dimension 2. With the Čech complex built up to dimension 10, the downhill algorithm has the highest complexity. It gives the solution in polynomial of exponential time.

3.4 Convergence rate.

The convergence rate measures how fast an algorithm converges. In [17], author modeled SA algorithm as an inhomogeneous Markov chain and studied its convergence. Let $P_{T,l}$ and $P_{T,*}$ denote the total consumed power at lth iteration and the optimal consumed power, respectively. By using Theorem 6.3 in [17], the convergence rate of our SA algorithm is estimated as $\sup \Pr(P_{T,l} - P_{T,*} > \epsilon) < K/l^{\theta}$, where ϵ and K are real positive numbers. The parameter θ is calculated by $\theta = \epsilon/(N \triangle P_*)$ where $\triangle P_*$ is the maximal increased or decreased amount of power after each iteration. The SA algorithm converges to the global optimum with polynomial rate. The downhill algorithm, which can be considered as a special case of the SA algorithm when the beginning temperature is set to 0, can be modeled as a homogenous Markov chain. As the result, it converges exponentially to the local optimum.

4 Simulation and results

The proposed algorithms were evaluated on a space 10×10 where cells were deployed randomly according to a Poisson point process. Each cell has a radius varying from $R_{\min} = 0.1$ to $R_{\max} = 1$. The intensity λ of the Poisson point

process was set to different values from 0.2 to 1. All simulations were repeated 1000 times.

The simulated annealing algorithm simulations were executed with initial temperature $T_0 = 1.95$. The initial accepting probability of an uphill move is 0.95. At each temperature, process is repeated $L = 1000$ times to approach the equilibrium solution. The temperature cooling factor was set to $\alpha = 0.95$ for $K = 100$ times. The final accepting probability of an uphill move is set to 0.05.

The downhill algorithm was tested with simplicial Čech complex built for two different maximal dimensions 2 and 10, i.e. simplices are computed up to 2-simplex and 10-simplex respectively.

4.1 Average consumption power per cell with optimized radius.

The average consumed power per cell with the optimized radius is shown in Figure 5. The higher density of cells is, the more power is saved. The SA algorithm which gives an approximation of the global optimal solution saves most power. At the highest density of cell, each cell operates with 35% of its maximal power in average, thus saving 65% power. The downhill algorithm saves 62% power with the Čech complex built up to dimension 2 and it saves 60% power with the Čech complex built up to dimension 10. These results show that the solution of the downhill algorithm also approaches the global optimal one. It suggests one should choose the downhill algorithm which gives the solution in polynomial time to optimize the networks.

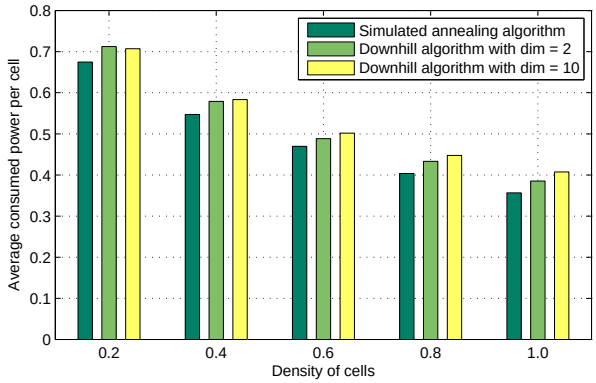


Fig. 5: Consumption power per cell minimized by different algorithms.

4.2 Probability density function of optimized radius.

The pdf of the optimized radius tells us how many cells are needed for the operation of the system. In addition, it also gives the optimal operating radii for cells. Although the average power consumed per cell by different algorithms are almost equal, the pdf of the optimized radius obtained by these algorithms are

quite different as shown in Figure 6, 7 and 8 for three densities of cells: 1 (high), 0.6 (medium) and 0.2 (low). The number of cells that can be turned off by using simulated annealing algorithm is always less than 10% for all values of density. Conversely, the number of turned off cells obtained by using downhill algorithm with dimension 2 is higher than 35% and with dimension 10 is higher than 40% for the high density. This result shows that the downhill algorithm can turn off more cells. In the pdf of the optimized radius by downhill algorithm, the number of cells whose radius is smaller than 0.3 is almost zero for all values of density. It suggests that such a tiny cell should be turned off. This behavior is similar with different cell density values.

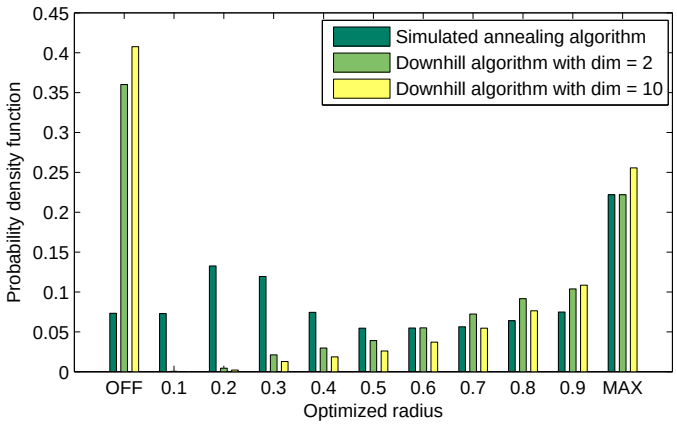


Fig. 6: Pdf of optimized radius at cell density = 1.0

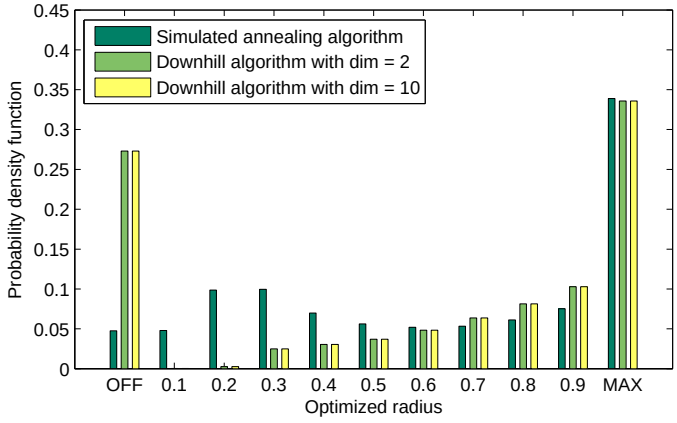


Fig. 7: Pdf of optimized radius at cell density = 0.6

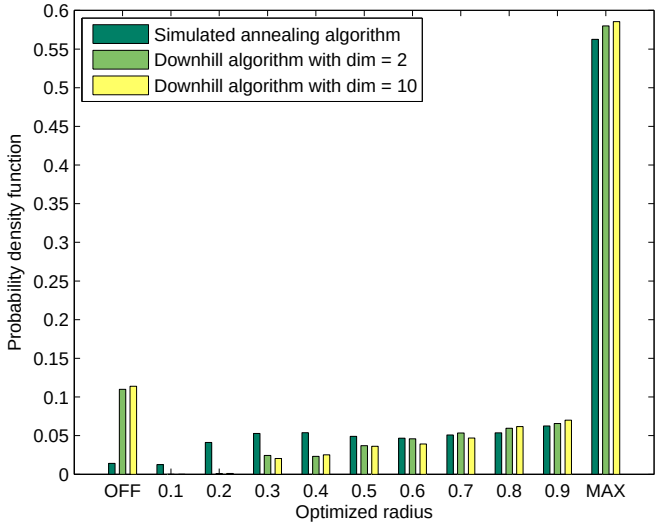


Fig. 8: Pdf of optimized radius at cell density = 0.2

5 Discussion and Conclusion

This paper introduced two homology based algorithms for saving energy in wireless networks: simulated annealing one and downhill one. The simulated annealing algorithm, which neglects priority between cells, saves the most power. But, it can only turn off a small number of cells. On the other hand, by considering priority between cells, the downhill algorithm firstly reduces power for cells in dense region and can turn off a larger number of cells. The higher the intensity of cells is, the higher the number of cells that can be turned off is. In addition, the difference between the total consumed power minimized by the two variants of algorithm is less than 5%. The downhill algorithm with dimension two has the highest convergence rate and the lowest complexity. It converges with exponential rate to the solution and achieves it in polynomial time. It suggests that, one should use the downhill algorithm with dimension two to optimize wireless systems transmission powers.

In future, we will develop a distributed version of these algorithms for autonomous systems and Self Organizing Networks.

References

1. X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 131–142. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132921

2. A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005, ch. 1.
3. H. Koskinen, "On the coverage of a random sensor network in a bounded domain," in *in Proceedings of 16th ITC Specialist Seminar*, 2004, pp. 11–18.
4. V. D. Silva and R. Ghrist, "Homological sensor networks," *Notices of the American Mathematical Society*, vol. 54, p. 2007, 2007.
5. R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 254–260.
6. V. De Silva and R. Ghrist, "Coordinate-free coverage in sensor networks with controlled boundaries via homology," *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1205–1222, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1177/0278364906072252
7. 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*, 2011, pp. 1–6.
8. P. Dlotko, R. Ghrist, M. Juda, and M. Mrozek, "Distributed computation of coverage in sensor networks by homological methods," *Appl. Algebra Eng., Commun. Comput.*, vol. 23, no. 1, pp. 29–58, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1007/s00200-012-0167-7
9. F. Yan, P. Martins, and L. Decreusefond, "Accuracy of homology based approaches for coverage hole detection in wireless sensor networks," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 497–502.
10. A. Vergne, L. Decreusefond, and P. Martins, "Reduction algorithm for simplicial complexes," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 475–479.
11. J. Munkres, *Elements of Algebraic Topology*. Addison Wesley, 1984.
12. A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.
13. N. K. Le, P. Martins, L. Decreusefond, and A. Vergne, "Construction of the generalized Cech complex," Tech. Rep., Sep. 2014. [Online]. Available: http://hal.archives-ouvertes.fr/hal-01069775
14. D. Bertsimas, J. Tsitsiklis *et al.*, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.
15. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
16. H. Edelsbrunner and S. Parsa, "On the computational complexity of betti numbers: Reductions from matrix rank," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '14. SIAM, 2014, pp. 152–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=2634074.2634085
17. A. Trouve, "Rough large deviation estimates for the optimal convergence speed exponent of generalized simulated annealing algorithms," *ANN. INST. H. POINCARE PROBAB. STATIST*, vol. 32, pp. 299–348, 1994.