Topology Design to Increase Network Lifetime in WSN for Graph Filtering in Consensus Processes

Leila Ben Saad, Thilina Weerasinghe and Baltasar Beferull-Lozano
WISENET Lab., Dept. of Information & Communication Technology, University of Agder, Norway

***Abstract*—Graph filters, which are considered as the workhorses of graph signal analysis in the emerging field of signal processing on graphs, are useful for many applications such as distributed estimation in wireless sensor networks. Many of these tasks are based on basic distributed operators such as consensus, which are carried out by sensor devices under limited energy supply. To cope with the energy constraints, this paper focuses on designing the network topology in order to maximize the network lifetime when applying graph filters. None of the existing works in the literature have studied such problem when graph filters are used. The problem is a complex combinatorial problem and in this work, we propose two efficient heuristic algorithms for solving it. We show by simulations that they provide good performance and increase significantly the network lifetime.**

I. Introduction

Recently, significant efforts have been performed to extend classical signal processing concepts to the graph setting, allowing the emergence of Graph Signal Processing [1], where the main interest focuses on signals defined over the nodes of a graph. In this area, one of the key results is the analysis of graph signals in the graph frequency domain. The workhorses of graph signal analysis are graph filters, which represent the building blocks for processing the spectral content of graph signals. Graph filters are useful to process, analyze networked data and solve wide range of problems and ideal for many tasks and applications [2], [3] such as distributed estimation.

Average consensus is a key distributed task in network processes that allows nodes to compute global averages from local initial data by only exchanging information with neighbors. Over the past years, consensus has gained a lot of interest in wireless sensor networks (WSNs). Such networks are composed of a large number of spatially distributed autonomous devices, which usually have low capabilities in terms of storage, processing and energy and equipped with a variety of sensors to monitor physical quantities of the environment. In WSNs, when applying graph filters and performing distributed processing algorithms, the limited energy supply of sensors should be preserved as much as possible. In fact, it is crucial for WSNs to be autonomous and capable of executing different tasks for a long time without the replacement of sensors' batteries. The energy consumed by the sensor devices depends on the configuration of the nodes, the communication topology among the nodes and in case of using graph filters on the number of exchanges needed to reach convergence.

In the past few years, there have been some works dedicated to redesign the network topology in order to maximize the convergence speed of consensus processes [4], [5]. Some other works have looked on the optimal topology in consensus processes in order to both minimize the convergence time and the energy consumption (or extend the network lifetime) [6-8]. However, none of these works consider the case of applying graph filters in the network. In this paper, we formulate the problem of optimizing the topology in WSNs in order to maximize the network lifetime when graph filters are used. This problem is a complex combinatorial problem and can not be solved efficiently in polynomial time. Therefore, in order to obtain some insights about how to design an efficient polynomial-time algorithm that approaches a close-to-optimal topology, genetic algorithms and simulated annealing are used. Then, inspired from these heavy optimization methods, we propose two efficient heuristic algorithms: the first one is centralized and the second one is distributed. We show by extensive simulations that the centralized algorithm provides a slightly better performance than the distributed one. But, both of them approach well the solution given by simulated annealing and genetic algorithms, which are expected to provide a solution close to the optimal. To the best of our knowledge, this is the first paper that focuses on the topological design problem in WSNs to increase the network lifetime and reduce the energy consumption when applying graph filters.

The remainder of this paper is organized as follows. Section 2 presents the main background. In section 3, the energy consumption model is presented. In section 4, our problem of network topology design is formulated. Section 5 and 6 present respectively the proposed heuristic algorithms and their performance evaluation. Section 7 concludes the paper.

II. Background

In this section, we review the main background related to the concepts of graph theory and graph filters.

Let $G(V,\mathcal{E})$ denote an undirected graph where V is a set of N vertices or nodes and $\mathcal{E}$ is a set of links or edges such that if node i is connected to j, then $(i,j) \in \mathcal{E}$. For any given graph G, we define the $N \times N$ adjacency matrix $\mathbf{A}$ with nonzero elements A_{ij} if and only if $(i,j) \in \mathcal{E}$. The set of neighbors of node i is defined by $\Omega_i = \{j \in V : (i,j) \in \mathcal{E}\}$. The degree of node i is $d_i = \sum_{j \in \Omega_i} A_{ij}$ and $\mathbf{D}$ is the degree matrix.

A. Graph signal and graph shift operator

A graph signal, defined on the set of nodes of the graph, is a mapping $\mathbf{x} : V \to \mathbb{R}$, and represented as a vector $\mathbf{x} = [x_1, ..., x_N]^T \in \mathbb{R}^N$. The i-th component x_i represents the signal value at the i-th vertex in V. Any graph G can

DOI: 10.23919/EUSIPCO.2017.8081227

be endowed with a graph-shift operator $\mathbf{S}$, which can be represented as a matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ satisfying $S_{ij} = 0$ for $i \neq j$ and $(i,j) \notin \mathcal{E}$. There are several possible choices for the shift $\mathbf{S}$ such as the adjacency matrix $\mathbf{A}$, the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and other generalizations defined on $\mathbf{L}$, such as $\mathbf{W}{=}\mathbf{I}{-}\delta\ \mathbf{L}$ where $\delta \in \mathbb{R}$. The shift $\mathbf{S}$ is assumed to be diagonalizable so that it can be decomposed as $\mathbf{S}{=}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_N)$ is the diagonal matrix of N eigenvalues and $\mathbf{V}$ is the corresponding eigenvector matrix.

B. Graph filters

A graph filter (GF) is a system $\mathbf{H}$ that takes a graph signal $\mathbf{x}$ as an input, processes it, and produces another graph signal $\mathbf{y}$ as an output. A graph filter $\mathbf{H} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a map between graph signals which is represented by an $N{\times}N$ matrix. In this paper, our focus will be restricted to *Finite Impulse Response* (*FIR*) GFs since they can be easily implemented in a distributed way. FIR GFs, which are designed such that their inpulse responses are finite in the vertex domain, can be classified as being node-invariant or node-variant [9]:

1) Node-invariant graph filter: It is a polynomial in $\mathbf{S}$ of degree $L-1$, with coefficients $\mathbf{h}{=}[h_0, ..., h_{L-1}]^T$. The graph signal output $\mathbf{y}$ that is generated when the node-invariant graph filter $\mathbf{H}_{inv}$ is applied, is given by:

$$\mathbf{y} = \mathbf{H}_{inv}\mathbf{x} = \sum_{l=0}^{L-1} h_l\ \mathbf{S}^l\mathbf{x} = \sum_{l=0}^{L-1} h_l\ \mathbf{x}^{(l)} \tag{1}$$

where $\mathbf{x}^{(l)}$ = $\mathbf{S}^l\ \mathbf{x}$ = $\mathbf{S}\ \mathbf{x}^{(l-1)}$.

2) Node-variant graph filter: In this case, each node applies different weights, collected in $N{\times}1$ vector $\mathbf{h}^{(l)} = [h_1^{(l)}, ..., h_N^{(l)}]^T$, to the shifted signals $\mathbf{S}^l\mathbf{x}$. In general, node-variant graph filters outperform node-invariant graph filters, since the number of degrees of freedom to design the coefficients is much larger. Thus, it can be viewed as a generalization of node-invariant graph filters. The graph signal output $\mathbf{y}$ that is generated when the node-variant graph filter $\mathbf{H}_{nv}$ is applied, is given by [9]:

$$\mathbf{y} = \mathbf{H}_{nv}\mathbf{x} = \sum_{l=0}^{L-1} diag(\mathbf{h}^{(l)})\ \mathbf{S}^l\mathbf{x} \tag{2}$$

To implement any linear transformation $\mathbf{B}$ as a node-variant graph filter, the shift operator $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ must satisfy two properties: all the entries of $\mathbf{V}$ are non-zero and all the eigenvalues $\{\lambda_k\}_{k=1}^N$ are distinct [9]. If these two conditions can not be satisfied, an approximate implementation of the graph filter can still be designed to approach as much possible $\mathbf{B}$, allowing to determine the optimal filter coefficients that minimize the Frobenius error norm $\|\mathbf{H}_{nv} - \mathbf{B}\|_F$. The optimal filter coefficients associated to node i, which are collected in an $L{\times}1$ vector $\mathbf{h}_i = [h_i^{(0)}, ..., h_i^{(L-1)}]^T$, are given by [9]:

$$\mathbf{h}_i = \Big((\mathbf{V}^{-1})^T diag(\mathbf{V}^T\mathbf{e}_i)\ \mathbf{\Psi}\Big)^{\dagger}\ \mathbf{B}^T\ \mathbf{e}_i \tag{3}$$

where $\mathbf{e}_i$ is a vector with all entries zero except for the i-th entry which is one and $\mathbf{\Psi}$ is the $N{\times}L$ Vandermonde matrix such that $\Psi_{ij}{=}\lambda_i^{j-1}$. Notice that the k-th entry of a vector $\mathbf{c}$ is denoted as $c_k = [\mathbf{c}]_k$, the entry of a matrix $\mathbf{A}$ is denoted as $A_{ij} = [\mathbf{A}]_{ij}$ and † stands for Moore-Penrose pseudoinverse. This paper will be mainly based on node-variant graph filters since they offer more flexibility and better performance.

III. Energy consumption model

We consider a network composed of N sensor nodes, with distinct identifiers and omni-directional antennas, deployed uniformly and randomly. Each sensor is supposed to use a broadcast communication, which means that a single transmission of a node can be received by any neighbor within its transmission range, which reduces the communication needs to send information to other nodes. This assumes that there is an underlying MAC protocol that resolves collisions and maintains the topology. A node can ignore the packets received from certain neighbors for a topology design consideration. Since the communication in sensor devices is very costly in most applications, both energy costs of sending and receiving packets are considered. In the following, the adopted energy consumption model is presented. This model considers the fact that the application of a graph filter in WSNs can be divided in different exchange periods. At each exchange period, a sensor node receives messages (packets of a certain size z) from its neighbors, makes some local computation and broadcasts its processed messages to its neighbors. Let $n_{ex} \in \mathbb{N}$ define the number of exchanges needed for convergence when applying the node-variant graph filter. The particularity of any FIR GF is that it can be implemented in at most $L-1$ exchanges of information, which means that n_{ex} can not exceed $L-1$ with $L \leq N$. The energy consumed by a node i when applying the node-variant graph filter is determined as follows:

$$E_i = n_{ex}(d_i(\alpha_R + \gamma) + \alpha_T) \tag{4}$$

where α_R is the energy spent by a sensor node in the reception of one message, which depends on the size z of the packet in bits and the energy $E_{R_X elec}$ spent by the electronic circuitry at the receiver in J/bit [10].

$$\alpha_R = E_{R_X elec}\ z \tag{5}$$

The energy expended in joules to transmit a packet of z bits across a link with a distance r_{ij}, is given by [10]:

$$\alpha_T = (E_{T_X elec} + E_{amp} r_{ij}^{\beta})\ z \tag{6}$$

where $E_{T_X elec}$ is the energy spent by the electronic circuitry at the transceiver in J/bit, E_{amp} is the energy dissipated at the amplifier in J/bit/m^2, β is the path loss exponent which is about 2 for freespace and higher in-door, r_{ij} is the distance between the transmitting sensor i and the receiving sensor j. In this work, due to the use of broadcast communications, the distance r_{ij} is assumed to be equal to the maximum transmission radio range R of sensor nodes.

The energy spent by a sensor node, when making some computations in a single exchange period, is represented by γ and is calculated as follows:

$$\gamma = E_{cp}\ z \tag{7}$$

where E_{cp} is the energy dissipated in processing in J/bit.

Therefore, the total energy E consumed in the network when applying the node-variant graph filter is given by:

$$E = \sum_{i=1}^{N} E_i \tag{8}$$

IV. Problem formulation

Our problem is to determine the optimal topology that maximizes the network lifetime when applying node-variant graph filters. In WSNs, sensor devices are usually powered by batteries which provide a limited energy capacity. If one sensor node runs out of energy, reaching the graph filter convergence can not be guaranteed and the application of the graph filter may fail. For these reasons, the lifetime of sensor nodes is very important when applying graph filters in WSNs. The lifetime of a sensor node i when applying a graph filter can be determined by dividing the initial energy E_0 available at the node (assuming here that all sensors have the same initial energy budget) and the energy consumed during the application of the graph filter:

$$\mathcal{L}_i = \frac{E_0}{E_i} = \frac{E_0}{n_{ex}(d_i(\alpha_R + \gamma) + \alpha_T)} \tag{9}$$

The network lifetime $\mathcal{L}$, which can be formulated as the number of graph filters that can be executed before the first sensor node runs out of energy, is given by:

$$\mathcal{L} = \min_{i \in V} \mathcal{L}_i = \min_{i \in V} \Big(\frac{E_0}{n_{ex}(d_i(\alpha_R + \gamma) + \alpha_T)} \Big) \tag{10}$$

To extend the network lifetime $\mathcal{L}$, we need to maximize the lifetime of the node with the shortest lifetime *i.e.* $\min\limits_{i \in V} \mathcal{L}_i$ or minimize the consumption of the node consuming the highest energy *i.e.* $\max\limits_{i \in V} E_i$. Therefore, our objective is to minimize the following function $\max\limits_{i \in V} n_{ex} \Big(\sum\limits_{j \in \Omega_i} A_{ij} \, (\alpha_R + \gamma) + \alpha_T \Big)$.

Our problem is to determine the optimal topology (*i.e.* adjacency matrix) that maximizes the network lifetime when applying the node-variant graph filter. The corresponding optimization problem can be formulated as follows:

$$\text{minimize}_{\{\mathbf{A}\}} \max_{i \in V} n_{ex} \Big(\sum_{j \in \Omega_i} A_{ij} \, (\alpha_R + \gamma) + \alpha_T \Big) \tag{11}$$

$$s.t. \quad \Big\| \Big(\sum_{l=0}^{n_{ex}} diag(\mathbf{h}^{(l)}) \, \mathbf{S}^l \mathbf{x} \Big) - \mathbf{B} \, \mathbf{x} \Big\|_2 \leq \epsilon$$

$$h_i^{(l)} = \Big[\Big((\mathbf{V}^{-1})^T diag(\mathbf{V}^T \mathbf{e}_i) \mathbf{\Psi} \Big)^{\dagger} \mathbf{B}^T \mathbf{e}_i \Big]_l \;\; \forall i \in V, l{=}0,..,n_{ex}$$

$$\lambda_2(\mathbf{L}) > 0$$

$$A_{ij} = A_{ji} \quad \forall i, j \in V$$

$$A_{ij} \in \{0,1\} \text{ if } r_{ij} \leq R \quad \forall i, j \in V$$

$$A_{ij} = 0 \qquad \text{if } r_{ij} > R \quad \forall i, j \in V$$

where ϵ is a small positive constant to ensure that the resulting normalized error at the graph filter convergence obtained after n_{ex} exchanges is very small. n_{ex} is affected by the topology and more specifically by the shift operator $\mathbf{S}$ through the nodes' degrees d_i, where $\mathbf{S} = \mathbf{I} - \delta \; \mathbf{L} = I - \delta \; (\mathbf{D} - \mathbf{A})$. Notice that given a shift $\mathbf{S}$, the filter coefficients $\mathbf{h}^{(l)}$ also change and their optimal values are determined by the second constraint. The third constraint represents the algebraic connectivity $\lambda_2(\mathbf{L})$, which ensures that the graph is connected. The fourth constraint guarantees that the adjacency is symmetric since undirected graphs are considered. The fifth and sixth constraints mean that the entries of the adjacency matrix are zeros or ones, depending on the existence of a link and by considering the distance between the two nodes and the transmission range at each node.

Our problem is a *combinatorial non convex* problem due to both binary variables and $h_i^{(l)}$ constraints and the fact that each of n_{ex}, $\mathbf{S}$ and $\mathbf{h}^{(l)}$ depend on the adjacency $\mathbf{A}$. This means that the problem can not be solved efficiently in polynomial time. Therefore in this paper, instead of attempting to solve this problem in an optimal way, our goal is to propose a feasible solution that is efficient in substantially increasing the network lifetime and has a polynomial time complexity.

V. Topological design with heuristic algorithms

This section focuses on the use of heuristic algorithms to design topologies that enhance the network lifetime and reduce the energy consumption when applying graph filters. First, in order to obtain an insight about how to design an efficient polynomial-time algorithm that approaches a close-to-optimal topology, we consider first metaheuristic algorithms based on Genetic Algorithms (GA) and Simulated Annealing (SA). Then, we propose efficient algorithms based on this insight.

A. Metaheuristic algorithms

SA and GA are metaheuristics [11], which can be defined as higher level heuristic algorithms that perform a robust search of a good solution and try to avoid local minimum. Both of them rely on randomness to generate good approximate solutions to combinatorial or NP-hard problems. GA is inspired by the process of natural selection, while SA is inspired by the physical process of heating a material and then slowly lowering the temperature to decrease defects. Starting from possible edges within the radio range R, Algorithm 1 and Algorithm 2 describe respectively how to apply GA and SA to our problem by considering all the constraints.

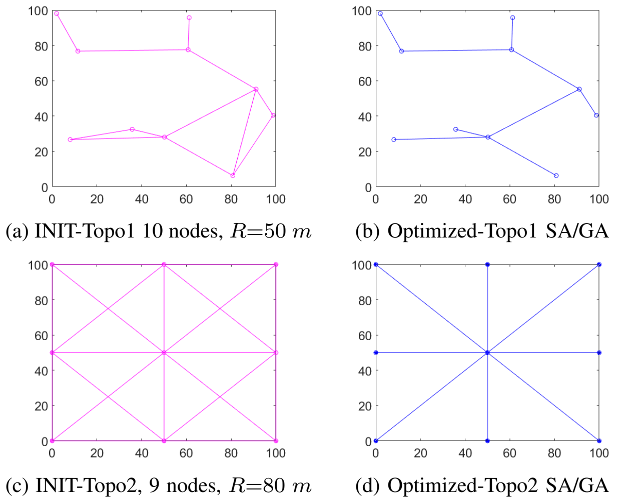
Algorithm 1 Genetic Algorithms (GA)

```
INPUT: Locations and transmission range $R$
Generate random initial population of 150 adjacency matrices
repeat
    Fitness function: Evaluate the cost of each adjacency matrix
    $cost(A_{ij}) = \max_{i \in V} n_{ex} \Big( \sum_{j \in \Omega_i} A_{ij} \, (\alpha_R + \gamma) + \alpha_T \Big)$
    Elite step: Select $0.01\%$ of adjacencies with best fitness function cost to go automatically to the new population
    Crossover step: Select two parent adjacency matrices from the population to be crossed over and to form new children
    Mutation step: Only for $0.01\%$ of the population of adjacency matrices, randomly change 2 binary variables in the child adjacency resulting from crossover while preserving the symmetry
    Acceptance step: Place new generated children adjacency matrices in the new population if they ensure graph connectivity.
until Maximum number of generations is reached
Return the best adjacency matrix in the current population
```

SA and GA offer practical approaches to solve complex problems of realistic scale. However, it is hard to solve instances with a large scale in reasonable computing times and it is difficult to implement them in a decentralized manner. On the other hand, they can be used to provide a good benchmark that is usually close to optimal, against which other polynomial-time heuristic algorithms can be compared. Since there is a need for computationally less time-consuming

Algorithm 2 Simulated Annealing (SA)

```
INPUT: Locations and transmission range $R$
Generate a random initial adjacency $\mathbf{A}_k$
Initialize system temperature $T$, cooling rate $\theta$ and $M_q$
repeat
  for $q = 1$ to $M_q$ do
    $cost(\mathbf{A}_k)= \max_{i \in V} n_{ex}\Big( \sum_{j \in \Omega_i} [\mathbf{A}_k]_{ij}\ (\alpha_R + \gamma) + \alpha_T \Big)$
    Slightly perturb $\mathbf{A}_k$ to generate a random symmetric adjacency matrix $\mathbf{A}_{k+1}$ whose graph is connected
    Calculate the $cost(\mathbf{A}_{k+1})$ of the new adjacency matrix
    if $cost(\mathbf{A}_{k+1}) < cost(\mathbf{A}_k))$ then
      $\mathbf{A}_{k+1}$ will be the new current solution
    end if
    if $cost(\mathbf{A}_{k+1}) > cost(\mathbf{A}_k)$ then
      accept $\mathbf{A}_{k+1}$ as the new current solution with probability $\exp\left(\dfrac{cost(\mathbf{A}_k) - cost(\mathbf{A}_{k+1})}{T}\right)$
    end if
  end for
  T= $\theta$ T
until Maximum number of iterations is reached
```



(a) INIT-Topo1 10 nodes, R=50 m (b) Optimized-Topo1 SA/GA

(c) INIT-Topo2, 9 nodes, R=80 m (d) Optimized-Topo2 SA/GA

Fig. 1: Example of optimized topologies with SA/GA.

Algorithm 3 Degree heuristic based on MST (D-MST)

```
INPUT: Locations and transmission range $R$
$cost$= $\max_{i \in V} n_{ex}\Big( \sum_{j \in \Omega_i} A_{ij}\ (\alpha_R + \gamma) + \alpha_T \Big)$
$k = -1$; $\Delta_m = \max_{i \in V} d_i$; $\bar{d} = \frac{1}{N} \sum_{i \in V} d_i$; $\mathbf{A}_{out} = \mathbf{A}$
while $k < \bar{d}$ do
  for each node $i$ = 1 to $N$ do
    for each node $j \in \Omega_i$ do
      if $d_i == (\Delta_m - k)$ or $d_j == (\Delta_m - k)$ then
        $[\mathbf{A}_w]_{ij}$ = $-1$
      else
        $[\mathbf{A}_w]_{ij}$ = $-(d_i + d_j)$
      end if
    end for
  end for
  $\mathbf{A}_{MST}$ = build Minimum-Spanning-Tree from $\mathbf{A}_w$
  $newCost$= $\max_{i \in V} n_{ex}\Big( \sum_{j \in \Omega_i} [\mathbf{A}_{MST}]_{ij}\ \alpha_R + \alpha_T + \gamma \Big)$
  if $newCost \leq cost$ then
    $cost = newCost$; $\mathbf{A}_{out} = \mathbf{A}_{MST}$
  end if
  $k = k + 1$
end while
```

algorithms with faster convergence, we propose in the following sections heuristic algorithms inspired from the solutions obtained by using SA or GA in consensus processes. In fact, from the optimal topologies provided by the best of both SA and GA, it can be seen that the nodes often keep links with neighbors that have the highest degree as shown in Figure 1. On the other hand, if we increase the transmission range such that all nodes can reach each other, resulting in a global coverage, the optimal topology is to keep all links because the number n_{ex} of exchanges needed to reach the consensus will be equal to one. However, ensuring a global coverage is usually not possible in WSNs, due to power constraints.

B. Degree heuristic algorithm based on MST (D-MST)

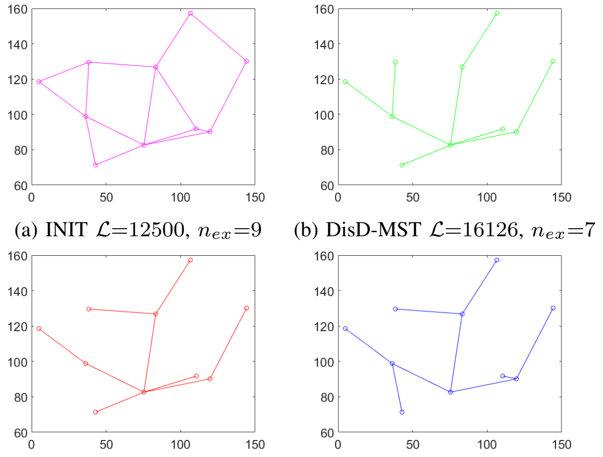
The solution provided by the D-MST algorithm consists of a topology based on a Maximum Spanning Tree, where the edges connected to neighbors that have the highest degree have higher probability to be kept as inspired from the results of SA and GA in Figure 1. For our topological design problem, based on the initial locations of the nodes and the transmission range R, we initialize the connectivity with all possible neighbors given R. Then, we build a Maximum Spanning Tree by using the sum of nodes' degrees as weights of the edges and compute the energy cost of this solution. The Maximum Spanning Tree can be computed by multiplying the weights for each edge by -1 and applying an algorithm to find a Minimum Spanning Tree (MST) such as Kruskal algorithm [12]. Since some nodes with certain degrees could be overloaded, we decrease, in a second step, the chances of selecting the edges connected to these nodes, by assigning -1 as weights to their corresponding edges when building the MST. Then, we recompute the new energy cost. The MST achieving less cost will be selected, as presented in Algorithm 3. The time complexity of the Kruskal algorithm is $\mathcal{O}(|\mathcal{E}| \log N)$ [12]. Since in the worst case Kruskal algorithm runs at most N times, the total computational complexity of D-MST algorithm is $\mathcal{O}(N|\mathcal{E}| \log N)$ in time.

C. Distributed Degree heuristic based on MST (DisD-MST)

The proposed D-MST algorithm can be modified in order to be implemented distributedly by using a distributed MST algorithm such as GHS [13]. The latter has a time complexity $\mathcal{O}(N \log N)$ and is optimal with respect to the required number of message transfers compared to other algorithms [14]. To implement our modified distributed algorithm DisD-MST, only a single MST is computed by taking as weights the sum of nodes' degrees. The time complexity of DisD-MST is the same as the complexity of GHS. However, the nodes initially only know the weights of the adjacent edges (by making each node send its degree to its 1-hop neighbors). Thus, they have to exchange additional messages with neighbors until the MST is constructed. The total number of messages required with GHS is $5N \log_2 N + 2\ |\mathcal{E}|$ [13]. The overhead due to exchanging more messages at network setup is slightly compensated in our algorithm when the graph filter is applied many times.

VI. Numerical Results

Extensive simulations were conducted in MATLAB to evaluate the performance of the proposed heuristic algorithms.



(a) INIT $\mathcal{L}$=12500, n_{ex}=9 (b) DisD-MST $\mathcal{L}$=16126, n_{ex}=7

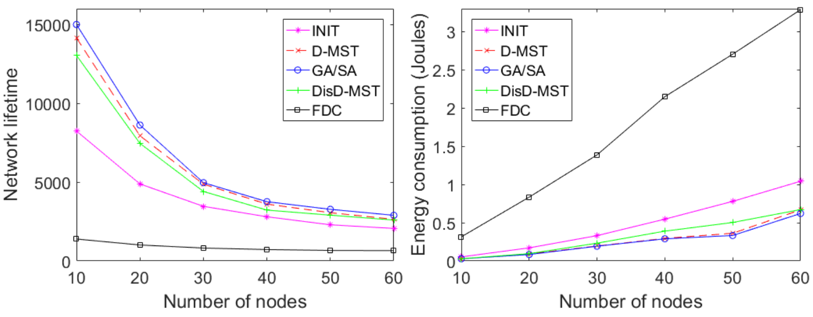
(c) D-MST $\mathcal{L}$=16130, n_{ex}=7 (d) SA/GA $\mathcal{L}$=27777, n_{ex}=5

Fig. 2: Example of topologies resulting from different heuristics over an initial network with N=10 and R=50 m (INIT).

A setup of N sensor nodes randomly and uniformly distributed over a certain area is considered in our simulations, where random input signal values x_i are injected in the network. We assume the following typical parameter values: z=100 Bytes, ϵ=0.01, E_0=50 J, β=2, E_{cp}=5 nJ/bit, $E_{Rx\,elec}$=$E_{Tx\,elec}$=50 nJ/bit, E_{amp}=100 $pJ/bit/m^2$ [10]. To implement graph filtering for consensus, the linear transformation $\mathbf{B}$ used is $\mathbf{B}$=$\mathbf{B}_{con}$=$\frac{\mathbf{1}\mathbf{1}^T}{N}$ and its application to the input signal $\mathbf{x}$ yields to the average. The shift $\mathbf{S}$ used is $\mathbf{S}$=$\mathbf{I}$$-\delta$ $\mathbf{L}$ where δ=$\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}$. In fact, such shift ensures faster convergence for consensus for a given topology [15].

Figure 2 shows typical topology instances, resulting from applying different heuristic algorithms, used to redesign a network of 10 nodes randomly deployed in a square area of 150 m side with R=50 m. It is interesting to see that the topologies obtained with D-MST and DisD-MST present similarities with that obtained with the best of both SA and GA. The results show also that with the proposed algorithms, the number n_{ex} of exchanges as well as the nodes' degrees are decreased, which improves the network lifetime.

Figure 3 shows the averaged network lifetime and energy consumption achieved by different heuristic algorithms, applied to random network topologies, deployed in a square area of 200 m side with R=80 m. As expected, applying graph filters for consensus processes outperforms the classical Fastest Distributed Consensus [15] (FDC), in terms of network lifetime and energy consumption, because FIR graph filters limit the number n_{ex} of exchanges. The results show also that the network lifetime is significantly improved when initial random topologies INIT with variable number of sensors are redesigned with D-MST and DisD-MST (see Figure 3a). It can also be seen that the network lifetimes, obtained with both algorithms, are very close to that obtained with SA/GA. In fact, the average improvement ratios of SA/GA over D-MST and DisD-MST are very low and are respectively about 5 % and 13 %. Figure 3b shows that the total energy consumption is significantly reduced with D-MST, even when topologies are redesigned for a main goal to increase the network lifetime.



(a) Network lifetime (b) Total energy consumption

Fig. 3: Performance of different heuristics applied to random topologies in square area of 200 m side with R=80 m.

The energy consumed by the sensors with D-MST algorithm is very close to that obtained with SA/GA, which clearly shows its efficiency. DisD-MST also reduces considerably the energy consumption, but it uses up slightly more energy than D-MST, due to the existence of some overloaded nodes and the exchange of more control messages at network setup.

VII. Conclusion

In this paper, we formulate the problem of optimizing the network topology in order to maximize the network lifetime in WSNs when graph filters are used. Since this problem is complex and combinatorial, we propose two efficient polynomial-time heuristic algorithms inspired from the results of SA and GA. The simulation results show that both proposed algorithms provide good performance for consensus processes, approaching well that obtained by SA and GA, which are expected to offer a solution close to the optimal.

Acknowledgment

This work was supported by PETROMAKS Smart-Rig grant 244205/E30, SFI grant 237896/O30 of Research Council of Norway.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Proc. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sept 2014.

[3] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 1080–1084.

[4] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *American Control Conference*, June 2005, pp. 2371–2378 vol. 4.

[5] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *45th IEEE Conference on Decision and Control*, Dec 2006, pp. 6605–6611.

[6] S. Sardellitti, S. Barbarossa, and A. Swami, "Optimal topology control and power allocation for minimum energy consumption in consensus networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 383–399, Jan 2012.

[7] B. Bjar and M. Vetterli, "Topology optimization for energy-efficient communications in consensus wireless networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 5085–5089.

[8] C. Asensio-Marco and B. Beferull-Lozano, "Energy efficient consensus over complex networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 2, pp. 292–303, March 2015.

[9] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed linear network operators using graph filters," *CoRR*, vol. abs/1510.03947, 2015.

[10] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Comm.*, vol. 1, no. 4, pp. 660–670, 2002.

[11] F. Glover and G. A. Kochenberger, *Handbook of metaheuristics*. Springer, 2003.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[13] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. Program. Lang. Syst.*, vol. 5, no. 1, pp. 66–77, 1983.

[14] S. Kutten and D. Peleg, "Fast distributed construction of smallk-dominating sets and applications," *J. Algorithms*, vol. 28, no. 1, pp. 40–66, 1998.

[15] L. Xiao and S. P. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.