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 and reduce the energy consumption when applying graph filters. 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 in terms of the network lifetime and the total energy consumption of the filtering process.

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 [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 $G(V,E)$ denote an undirected graph where $V$ is a set of $N$ vertices or nodes and $E$ is a set of links or edges such that if node $i$ is connected to $j$, then $(i,j) \in E$. For any given graph $G$, we define the $N \times N$ adjacency matrix $A$ with nonzero elements $A_{ij}$ if and only if $(i,j) \in E$. The degree of node $i$ is defined by $\Omega_i = \{j \in V : (i,j) \in E\}$. The degree of node $i$ is $d_i = \sum_{j \in \Omega_i} A_{ij}$ and $D$ 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 $x : V \rightarrow \mathbb{R}$, and represented as a vector $x = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$. The $i$-th component $x_i$ represents the signal value at the $i$-th vertex in $V$. Any graph $G$ can be endowed with a graph-shift operator $S$, which can be represented as a matrix $S \in \mathbb{R}^{N \times N}$ satisfying $S_{ij} = 0$ for $i \neq j$ and $(i,j) \notin E$. There are several possible choices for the shift $S$ such as the adjacency matrix $A$, the Laplacian matrix $L = D - A$ and other generalizations defined on $L$, such as $W=I-\delta L$ where $\delta \in \mathbb{R}$. The shift $S$ is assumed to be
diagonalizable so that it can be decomposed as $S = \mathbf{V} \Lambda \mathbf{V}^{-1}$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N)$ 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 $x$ as an input, processes it, and produces another graph signal $y$ 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 impulse 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 $S$ of degree $L - 1$, with coefficients $h_i = [h_0, \ldots, h_{L-1}]^T$. The graph signal output $y$ that is generated when the node-invariant graph filter $\mathbf{H}_{\text{inv}}$ is applied, is given by:

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

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

2) **Node-variant graph filter**: In this case, each node applies different weights, collected in $N \times 1$ vector $h_i = [h_i^{(0)}, \ldots, h_i^{(L-1)}]^T$, to the shifted signals $S^l x$. 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 $y$ that is generated when the node-variant graph filter $\mathbf{H}_{\text{nv}}$ is applied, is given by [9]:

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

To implement any linear transformation $\mathbf{B}$ as a node-variant graph filter, the shift operator $S = \mathbf{V} \Lambda \mathbf{V}^{-1}$ 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 $\|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_F$. The optimal filter coefficients associated to node $i$, which are collected in an $L \times 1$ vector $h_i = [h_i^{(0)}, \ldots, h_i^{(L-1)}]^T$, are given by [9]:

$$h_i = \left( (\mathbf{V}^{-1})^T \text{diag}(\mathbf{V}^T e_i) \mathbf{\Psi} \right)^{\dagger} \mathbf{B}^T e_i$$

where $e_i$ 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 $\mathbf{\Psi} e_i = \lambda_i^{-1} e_i$. Notice that the $k$-th entry of a vector $e_i$ is denoted as $e_k = [e_i]_k$. A node runs out of energy, reaching the graph filter convergence can not be guaranteed and the application of the filter may fail due to sensitivity of FIR GFs to graph variation. 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:

\[
L_i = \frac{E_0}{n_{ex}(d_i(\alpha_R + \gamma) + \alpha_T)}
\]  

(9)

The network lifetime \(L\), 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:

\[
L = \min_{i \in V} L_i = \min_{i \in V} \left( \frac{E_0}{n_{ex}(d_i(\alpha_R + \gamma) + \alpha_T)} \right)
\]  

(10)

To extend the network lifetime \(L\), we need to maximize the lifetime of the node with the shortest lifetime \(i\) or minimize the consumption of the node consuming the highest energy \(i\). Therefore, our objective is to minimize the following function.

\[
\max_{i \in V} n_{ex}\left( \sum_{j \in \Omega_i} A_{ij} (\alpha_R + \gamma) + \alpha_T \right).
\]

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:

\[
\text{minimize}_{\{A\}} \max_{i \in V} n_{ex}\left( \sum_{j \in \Omega_i} A_{ij} (\alpha_R + \gamma) + \alpha_T \right)
\]  

(11)

\[
s.t. \quad \left\| \sum_{l=0}^{n_{ex}} \text{diag}(h^{(l)}) S^l x \right\|_2 \leq \epsilon
\]

where \(\epsilon\) is a small positive constant that ensures 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 \(S\) through the nodes’ degrees \(d_i\), where \(S = I - \delta L = I - \delta (D - A)\). Notice that given a shift \(S\), the filter coefficients \(h^{(l)}\) also change and their optimal values are determined by the second constraint. The third constraint represents the algebraic connectivity \(\lambda_2(L)\), which should be higher than a small positive constant \(\xi\) to ensure 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 distance between the two nodes involved and the transmission range at each node.

Our problem is a combinatorial non convex problem due to both binary variables and \(h^{(l)}\) constraints and the fact that each of \(n_{ex}\), \(S\) and \(h^{(l)}\) depend on the adjacency \(A\). 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 topologies 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)

\[\text{INPUT:} \text{ Locations and transmission range } R\]

Generate random initial population of 150 adjacency matrices

\[\text{repeat}\]

\[\text{Fitness function: Evaluate the cost of each adjacency matrix }\]

\[\text{cost}(A_{ij}) = \max_{i \in V} n_{ex}\left( \sum_{j \in \Omega_i} A_{ij} (\alpha_R + \gamma) + \alpha_T \right)\]

\[\text{Elite step: Select 0.01\% of adjacency matrices with best fitness function cost to go automatically to the new population}\]

\[\text{Crossover step: Select two parent adjacency matrices from the population to be crossed over and to form new children}\]

\[\text{Mutation step: Only for 0.01\% of the population of adjacency matrices, randomly change 2 binary variables in the child adjacency resulting from crossover while preserving the symmetry}\]

\[\text{Acceptance step: Place new generated children adjacency matrices in the new population if they ensure graph connectivity.}\]

\[\text{until Maximum number of generations is reached}\]

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 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.
Algorithm 2 Simulated Annealing (SA)

**INPUT:** Locations and transmission range \( R \)

Generate a random initial adjacency matrix \( A_k \)

Initialize the system temperature \( T \), cooling rate \( \theta \) and \( M_q \)

repeat

for \( q = 1 \) to \( M_q \) do

\[
\text{cost}(A_k) = \max_{i \in V} n_{ex} \left( \sum_{j \in \Omega_i} [A_k]_{ij} (\alpha_R + \gamma) + \alpha_T \right)
\]

Slightly perturb \( A_k \) to generate a random symmetric adjacency matrix \( A_{k+1} \) whose graph is connected

Calculate the cost of \( A_{k+1} \) of the new adjacency matrix

if \( \text{cost}(A_{k+1}) < \text{cost}(A_k) \) then

\( A_{k+1} \) will be the new current solution

end if

if \( \text{cost}(A_{k+1}) > \text{cost}(A_k) \) then

accept \( A_{k+1} \) as the new current solution with probability

\[
\exp \left( \frac{\text{cost}(A_k) - \text{cost}(A_{k+1})}{T} \right)
\]

end if

end for

\( T = \theta T \)

until Maximum number of iterations is reached

---

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

**INPUT:** Locations and transmission range \( R \)

\[
\text{cost} = \max_{i \in V} n_{ex} \left( \sum_{j \in \Omega_i} [A]_{ij} (\alpha_R + \gamma) + \alpha_T \right)
\]

\( k = -1; \Delta_m = \max_{i \in V} d_i; \bar{d} = \frac{1}{N} \sum_{i \in V} d_i; A_{out} = A \)

while \( k < d \) 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

\( [A_w]_{ij} = -1 \)

else

\( [A_w]_{ij} = -(d_i + d_j) \)

end if

end for

end for

\( A_{MST} = \) build Minimum-Spanning-Tree from \( A_w \)

\( \text{newCost} = \max_{i \in V} n_{ex} \left( \sum_{j \in \Omega_i} [A_{MST}]_{ij} (\alpha_R + \gamma) + \alpha_T \right) \)

if \( \text{newCost} \leq \text{cost} \) then

\( \text{cost} = \text{newCost}; A_{out} = A_{MST} \)

end if

\( k = k + 1 \)

end while

---

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 \( O(|E| \log N) \) [12]. Since in the worst case Kruskal algorithm runs at most \( N \) times, the total computational complexity of D-MST algorithm is \( O(N|E| \log N) \) 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 \( O(N \log N) \) 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 applying 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 \( 5N \log_2 N + 2|E| \) [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 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, \( \epsilon=0.01 \), \( E_0=50 \) J, \( \beta=2 \), \( E_{cp}=5 \) nJ/bit/packet, \( E_{R_{X_{elec}}} = E_{T_{X_{elec}}} = 50 \) nJ/bit, \( E_{amp}=100 \) pJ/bit/m² [10]. To implement graph filtering for consensus, the linear transformation \( B=11/7/N \) is used and its application to the input signal \( x \) yields to the average. The shift operator used is \( S=I-\delta L \) where \( \delta = \frac{2}{\lambda_{max}(L)+\lambda_{min}(L)} \). In fact, such shift ensures faster convergence for consensus for a given topology [15].
In this paper, we formulate the problem of optimizing the network topology in order to maximize the network lifetime and reduce the energy consumption in WSNs when graph filters are used. Since this problem is very 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 finite-time 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 Offshore Mechatronics grant 237896/O30 and TOPPFRSK grant 250910/F20 from Research Council of Norway.

REFERENCES