ENERGY EFFICIENT CONSENSUS OVER DIRECTED GRAPHS

César Asensio-Marco and Baltasar Beferull-Lozano

Intelligent Signal Processing and Wireless Networks (WISENET)
Department of ICT, University of Agder, Grimstad, Norway
Email: {cesar.asensio, baltasar.beferull}@uia.no

ABSTRACT

Consensus algorithms are iterative methods that represent a basic building block to achieve superior functionalities in increasingly complex sensor networks by facilitating the implementation of many signal-processing tasks in a distributed manner. Due to the heterogeneity of the devices, which may present very different capabilities (e.g. energy supply, transmission range), the energy often becomes a scarce resource and the communications turn into directed. To maximize the network lifetime, a magnitude that in this work measures the number of consensus processes that can be executed before the first node in the network runs out of battery, we propose a topology optimization methodology for directed networks. Numerical results corroborate the merits of this work.

Index Terms**—** Energy efficiency, consensus algorithm, directed graph, wireless sensor networks

1. INTRODUCTION

The recent evolution of Wireless Sensor Networks (WSNs) towards The Internet of Things (IoT) [1] is resulting in the appearance of heterogeneous sensor networks whose nodes generally present very different capabilities. In this context, the collaboration between nodes is crucial to accomplish increasingly complex tasks, since most of these nodes present very limited energy and computational resources. Distributed and cooperative techniques avoid the need of performing all the computations at a central entity, thus, reducing congestion and increasing the robustness and scalability of the network.

One of the most important algorithms that allows to implement in a distributed manner a significant number of signal-processing tasks (e.g. signal detection [2] and estimation [3]), is the average consensus algorithm [4]. The goal of an average consensus algorithm is to obtain, in a distributed manner, the average of the sensor data by iteratively processing the measurements collected by sensor nodes. This iterative philosophy implies that a local exchange of data with an associated power consumption is repeated along the iterations, which may result in a prohibitive energy consumption if the process is not properly optimized. Another important limitation, which is present in most of the existing schemes for consensus, is the implicit assumption of homogeneous networks. In such schemes, nodes are always assumed to be able to bidirectionally communicate and to have the same energy capabilities. Oppositely, one typically encounters heterogeneous networks, as the ones considered in this work, where it may happen that some nodes are able to directly communicate with other nodes that cannot communicate back. In addition, since the energy available vary significantly among nodes, their lifetime may be very different. Nodes stopping their operation due to battery depletion is always an undesired situation, which may compromise the task performance (e.g. accuracy, time-response). Regardless of the task being solved, the concept of network lifetime is associated to the time until one or several nodes run out of battery [5][6][7]. Thus, it is not only important the energy consumed by the whole network, but also the energy consumed by individual nodes. In an average consensus process, every node must participate to obtain the global average, which is no longer possible as soon as the first node runs out of battery. In this case, a single node may determine the network lifetime.

While abundant methodologies can be found in the related literature [8][9] to reduce the convergence time of consensus algorithms, only a few works [10][11][12] have addressed the minimization of its energy consumption. Existing approaches either confine themselves to undirected networks or do not consider the power consumption of each node and its energy capabilities at all. This motivates the design of new methods to improve the energy consumption of consensus algorithms in heterogeneous networks. In this work, we propose a new topology optimization methodology to maximize the lifetime of directed networks that are executing consensus-based tasks and are composed of nodes with different energy capabilities.

The remainder of this paper is structured as follows: some background on consensus algorithms is presented in Section 2. The formulation of the problem of network lifetime maximization is given in Section 3. In Section 4, we present an optimization procedure to find the topology with maximal network lifetime. Finally Section 5 presents some numerical results and Section 6 concludes the paper.

The work in this paper was supported by the PETROMAKS Smart-Rig grant 244205/E30 and the SFI Offshore Mechatronics grant 237896/O30.

2. CONSENSUS IN DIRECTED GRAPHS

The communication between nodes in an heterogeneous network can be modelled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set $\mathcal{V}$ of N nodes and a set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of directed links. A directed link e_{ij} denotes a data flow from node i to node j.

Definition 1. A directed graph is called *strongly connected* if and only if any two distinct nodes of the graph can be connected via a path that respects the orientation of the edges of the digraph.

The set of neighbors of a node i can be defined as $\mathcal{N}_i = \{j \in \mathcal{V} : e_{ij} \in \mathcal{E}\}$. Furthermore, let $\mathbf{A}$ be the $N \times N$ adjacency matrix of the graph $\mathcal{G}$, with entries $[\mathbf{A}]_{ij}$ equal to 1 if link e_{ij} is established and 0 otherwise. Then, the in-degree and out-degree of node i can be, respectively, defined as $d_i^{in} = \sum_{j=1}^{N} [\mathbf{A}]_{ji}$ and $d_i^{out} = \sum_{j=1}^{N} [\mathbf{A}]_{ij}$. The degree matrix $\mathbf{D}$ is the diagonal matrix, whose non-zero entries are given by the out-degree $[\mathbf{D}]_{ii} = d_i^{out}$ of each node i.

Definiton 2. A node i of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *balanced* if and only if its in-degree and out-degree are equal, i.e. $d_i^{in} = d_i^{out}$. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called *balanced* if and only if all of its nodes are *balanced*.

We denote by $\mathbf{L}$ the $N \times N$ Laplacian matrix of the graph, given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, whose entries are given by:

$$[\mathbf{L}]_{ij} = \begin{cases} [\mathbf{D}]_{ii} & \text{if } i = j \\ -[\mathbf{A}]_{ij} & \text{otherwise} \end{cases} \tag{1}$$

Fiedler defined in [13] the algebraic connectivity $a(\mathcal{G})$ of an undirected graph as the second smallest eigenvalue $\lambda_2(\mathbf{L})$ of its Laplacian matrix. In this work, we use its generalization to directed graphs proposed in [14] and [15], such that

$$a(\mathcal{G}) = \min_{\mathbf{x} \in \mathcal{P}} \mathbf{x}^T \mathbf{L} \mathbf{x} = \min_{\mathbf{x} \in \mathbb{R}^N, \mathbf{x} \neq 0, \mathbf{x} \perp \mathbf{e}} \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \tag{2}$$

where $\mathcal{P}$ is the set $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^N, \mathbf{x} \perp \mathbf{e}, ||\mathbf{x}|| = 1\}$, i.e. the set of real vectors of unit norm that are orthogonal to $\mathbf{e} = [1, \ldots, 1] \in \mathbb{R}^N$. For an undirected graph, the following well-known property holds [14]:

$$a(\mathcal{G}) = \min_{\mathbf{x} \in R^N, \mathbf{x} \neq \mathbf{0}, \mathbf{x} \perp \mathbf{e}} \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \lambda_2(\mathbf{L}).$$

An interesting operation that we will use later to formulate our problem is the mirror graph operation.

Definition 3. The mirror graph operation $\mathcal{M}(\mathcal{G})$ transforms any directed graph $\mathcal{G}$ into an undirected graph $\hat{\mathcal{G}}$, such that

$$[\hat{\mathbf{A}}]_{ij} = [\hat{\mathbf{A}}]_{ji} = \frac{[\mathbf{A}]_{ij} + [\mathbf{A}]_{ji}}{2},$$

where $\hat{\mathbf{A}}$ is the adjacency matrix corresponding to the mirror graph $\hat{\mathcal{G}}$. The Laplacian matrix associated to the mirror graph is then given by $\hat{\mathbf{L}} = \hat{\mathbf{D}} - \hat{\mathbf{A}}$.

Let us define an additional operation $\mathcal{M}^{-1}(\mathbf{A}_1, \mathbf{A}_2)$ that takes a reference adjacency matrix $\mathbf{A}_1$ corresponding to a graph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ and a second adjacency matrix $\mathbf{A}_2$ corresponding to a subgraph $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2 \subseteq \mathcal{E}_1)$ and obtains the following matrix $(\mathbf{A}_2 \oslash \mathbf{A}_2) \odot \mathbf{A}_1$, where $\oslash$ and $\odot$ stands for the Hadamard division and product respectively. Clearly, $\mathcal{M}^{-1}(\mathbf{A}_1, \mathbf{A}_2)$ reverses the mirror operation by taking $\mathbf{A}_1 = \mathbf{A}$ and $\mathbf{A}_2 = \hat{\mathbf{A}}$.

Finally, let us assume that nodes have some initial data at time instant $t = 0$. We collect them in an initial vector $\mathbf{x}(0)$, whose average is $\mathbf{x}_{\text{avg}} = \frac{\mathbf{1}\mathbf{1}^T \mathbf{x}(0)}{N}$, where $\mathbf{1}$ denotes the all ones column vector. The linear update of the state of all sensors at time instant t, given by:

$$\dot{\mathbf{x}}(t) = -\mathbf{L}\mathbf{x}(t) \tag{3}$$

solves the average consensus problem if and only if the underlying digraph is strongly connected and balanced [4][15]. Moreover, the work in [15] shows that the dynamics in (3) globally asymptotically vanishes with a speed that is equal to $\lambda_2(\hat{\mathbf{L}})$. Accordingly, the convergence time can be defined as:

$$t(\mathbf{L}) = \frac{\mathcal{K}}{\lambda_2(\hat{\mathbf{L}})}, \tag{4}$$

where $\mathcal{K}$ groups all the irrelevant constants involved, such as the duration of a time slot and the error reduction factor.

3. ENERGY CONSUMPTION AND LIFETIME IN DIRECTED GRAPHS

When considering wireless networks, maintaining the underlying topology implies a cost on communications, which usually depends on the physical distance between the nodes. Thus, to maintain a particular topology, each node of the network needs to spend some power consumption for this. Such powers can be stored in a vector as follows:

$$\mathbf{p} = [\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_N], \tag{5}$$

where $[\mathbf{p}]_i = \mathrm{p}_i$ denotes the power consumption per communication step of node i. Let us assume a generic path loss model with path loss exponent γ, then the power that node i requires to successfully communicate with node j is given by $p_{ij} = p_{\text{min}}^{(j)} \cdot r_{ij}^{\gamma}$, where $p_{\text{min}}^{(j)}$ is the minimum power required at the receiver j to successfully decode the incoming information and r_{ij} is the distance between nodes i and j. Accordingly, the power consumption per communication step of node i is given by:

$$\mathrm{p}_i = \sum_{j \in \mathcal{N}_i} p_{ij} = \sum_{j \in \mathcal{V}} p_{ij} \cdot [\mathbf{A}]_{ij} = \sum_{j \in \mathcal{V}} p_{ij} \cdot [\hat{\mathbf{A}}]_{ij} \tag{6}$$

Notice that there exists a one-to-one mapping between p_i and the network topology.

Besides having to ensure a minimum power $p_{\text{min}}^{(j)}$ at the receiver j, there exists another inherent limitation on radio communications, consisting of a maximum power transmission $p_{\text{max}}^{(i)}$ at each transmitter i. Both constraints determine the maximum distance $r_{\text{max}}^{(i,j)}$ for which a transmitter i can correctly transmit a message in absence of interference to a receiver j, which is given by:

$$r_{\text{max}}^{(i,j)} = \left(\frac{p_{\text{max}}^{(i)}}{p_{\text{min}}^{(j)}} \right)^{1/\gamma} \tag{7}$$

The total energy consumption of a node i, after a single consensus process, is the product between the consumed power p_i at each time step and the convergence time $t(\mathbf{L})$. Formally, we have the following:

$$\mathcal{E}_i(\hat{\mathbf{A}}) = \mathrm{p}_i \cdot t(\hat{\mathbf{L}}) = \mathcal{K} \frac{\sum\limits_{j \in \mathcal{V}} p_{ij} \cdot [\hat{\mathbf{A}}]_{ij}}{\lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}}))}$$

In this work, we focus on the optimization of the network topology, given by the matrix $\mathbf{A}$, which is our main optimization variable. As a cost function, we consider the lifetime of the network, defined in this paper as:

$$\mathcal{L}(\hat{\mathbf{A}}) = \max_i \left\{ \frac{\mathcal{C}_i}{\mathcal{E}_i(\hat{\mathbf{A}})} \right\} \tag{8}$$

where $\mathcal{C}_i$ denotes the energy budget of node i. Then, expression in (8) gives the total number of consensus processes that can be executed before the first node runs out of battery.

Let us denote $\mathbf{A}_{\text{max}}$ the adjacency matrix associated to the maximally connected topology, where all nodes are using the maximum transmission range given by (7). Accordingly, our problem (**P0**) can be cast as follows:

$$\begin{array}{ll}
\max._{\{\hat{\mathbf{A}}\}} & \mathcal{L}(\hat{\mathbf{A}}) \\
\text{s. t.} & \xi \leq \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) \\
& [\hat{\mathbf{A}}]_{ij} = [\hat{\mathbf{A}}]_{ji} \;\; \forall i, j \in \mathcal{V} \\
& (\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}}))\mathbf{1} = (\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}})^T)\mathbf{1} \\
& [\hat{\mathbf{A}}]_{ij} \in \{0, \frac{1}{2}\} \;\text{ if }\; [\mathbf{A}_{\text{max}}]_{ij} \neq [\mathbf{A}_{\text{max}}]_{ji} \\
& [\hat{\mathbf{A}}]_{ij} \in \{0, 1\} \;\text{ if }\; [\mathbf{A}_{\text{max}}]_{ij} = [\mathbf{A}_{\text{max}}]_{ji} = 1 \\
& [\hat{\mathbf{A}}]_{ij} = 0 \;\text{ if }\; [\mathbf{A}_{\text{max}}]_{ij} = [\mathbf{A}_{\text{max}}]_{ji} = 0
\end{array}$$

where ξ is an arbitrary small positive constant to ensure that the resulting value of $\lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}}))$ is greater than zero, i.e. the resulting graph is connected. The second and the third constraints ensure that the mirror solution graph and its corresponding reconstruction are symmetric and balanced respectively. Finally, the number of entries of $\hat{\mathbf{A}}$ that can be different from zero is limited by (7), reducing considerably the number of variables of the problem.

Unfortunately, **P0** is clearly a combinatorial problem because of the binary variables constraint. In the next section, we will show how to alleviate this difficulty.

4. OPTIMIZATION PROCEDURE

In order to obtain a tractable problem, we introduce in **P0** a convex relaxation to allow the entries of the matrix $\hat{\mathbf{A}}$ to be real values, leading to a relaxed problem that we call **P1**. Due to this relaxation procedure, the matrix coefficients do not determine directly the presence or absence of a specific link. A procedure to project the relaxed solution to the original feasible set is proposed later.

In order to solve the resulting relaxed problem **P1**, we introduce the following function [16]:

$$h(\mu, \hat{\mathbf{A}}) = \max_{\hat{\mathbf{A}}} \left\{ \max_i \left\{ \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) - \mu \frac{\mathrm{p}_i(\hat{\mathbf{A}})}{\mathcal{C}_i} \right\} \right\},$$

where μ is a real positive parameter that controls the trade-off between the power consumption p_i of node i and the convergence time $t(\hat{\mathbf{L}})$. The values of $\hat{\mathbf{A}}$ that maximize $h(\mu, \hat{\mathbf{A}})$ are the same ones that maximize our objective function [16]. Furthermore, the value of μ that makes $h(\mu, \hat{\mathbf{A}}) = 0$ is also the maximum of our cost function. Then, by applying standard optimization tools [17] and introducing the function $h(\mu, \hat{\mathbf{A}})$ to **P1**, we obtain (**P2**):

$$\begin{array}{ll}
\max._{\{s, \hat{\mathbf{A}}\}} & s \\
\text{s. t.} & \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) - \mu \frac{\mathrm{p}_1(\hat{\mathbf{A}})}{\mathcal{C}_1} \geq s \\
& \vdots \\
& \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) - \mu \frac{\mathrm{p}_N(\hat{\mathbf{A}})}{\mathcal{C}_N} \geq s \\
& \xi \leq \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) \\
& [\hat{\mathbf{A}}]_{ij} = [\hat{\mathbf{A}}]_{ji} \;\; \forall i, j \in \mathcal{V} \\
& (\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}}))\mathbf{1} = (\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}})^T)\mathbf{1} \\
& 0 \leq [\hat{\mathbf{A}}]_{ij} \leq \frac{1}{2} \;\text{ if }\; [\mathbf{A}_{\text{max}}]_{ij} \neq [\mathbf{A}_{\text{max}}]_{ji} \\
& 0 \leq [\hat{\mathbf{A}}]_{ij} \leq 1 \;\; [\mathbf{A}_{\text{max}}]_{ij} = [\mathbf{A}_{\text{max}}]_{ji} = 1 \\
& [\hat{\mathbf{A}}]_{ij} = 0 \;\text{ if }\; [\mathbf{A}_{\text{max}}]_{ij} = [\mathbf{A}_{\text{max}}]_{ji} = 0
\end{array}$$

which is a convex parametric problem, whose solution is a function of the parameter μ.

Algorithm 1 solves the problem **P1** by applying **P2** iteratively, where ϵ controls the accuracy of the solution (a relaxed topology matrix $\hat{\mathbf{A}}_{\text{relax}}$ corresponding to some mirror graph). Then, before projecting $\hat{\mathbf{A}}_{\text{relax}}$ to the original feasible set, we need to undo the mirror graph operation by applying $\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}}_{\text{relax}})$. We denote by $\mathbf{A}_{\text{relax}}$ the result of this operation, which is finally projected to the original feasible set, leading to the projected solution $\mathbf{A}_{\text{int}}$, see Fig. 1.

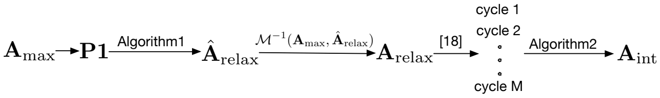
$\mathbf{A}_{\text{max}}$
P1
Algorithm1
$\hat{\mathbf{A}}_{\text{relax}}$
$\mathcal{M}^{-1}(\mathbf{A}_{\text{max}}, \hat{\mathbf{A}}_{\text{relax}})$
$\mathbf{A}_{\text{relax}}$
[18]
cycle 1
cycle 2
cycle M
Algorithm2
$\mathbf{A}_{\text{int}}$

Fig. 1. Steps followed by our optimization procedure to obtain $\mathbf{A}_{\text{int}}$ from $\mathbf{A}_{\text{max}}$.

Algorithm 1 Solves **P1**, obtaining $\hat{\mathbf{A}}_{\text{relax}}$

```
Require: $\epsilon$
Ensure: $\max_i \left\{ \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) - \mu \frac{\mathrm{p}_i(\hat{\mathbf{A}})}{\mathcal{C}_i} \right\} \le \epsilon$
  Set matrix $\hat{\mathbf{A}}$ as a feasible solution
  while $\max_i \left\{ \lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}})) - \mu \frac{\mathrm{p}_i(\hat{\mathbf{A}})}{\mathcal{C}_i} \right\} > \epsilon$ do
    Set $\mu$ as $\max_i \left\{ \frac{\lambda_2(\hat{\mathbf{L}}(\hat{\mathbf{A}}))\mathcal{C}_i}{\mathrm{p}_i(\hat{\mathbf{A}})} \right\}$
    Solve P2 with the current $\mu$, obtaining $\hat{\mathbf{A}}^*$
    Set $\hat{\mathbf{A}} = \hat{\mathbf{A}}^*$
  end while
```

Algorithm 2 Projects $\mathbf{A}_{\text{relax}}$, obtaining $\mathbf{A}_{\text{int}}$

```
Require: $LCycles(1, \dots, N)$
Ensure: $a(\mathcal{G}(\mathbf{A}_{\text{int}})) > 0$ and $\mathbf{A}_{\text{int}}\mathbf{1} = \mathbf{A}_{\text{int}}^T\mathbf{1}$
  Every node $v \in \mathcal{V}$ is tagged with its id$= 1, \dots, N$
  Every cycle $C \in LCycles$ is marked as unclassified
  while $\exists C \in LCycles(1, \dots, N)$ unclassified do
    Choose $v \in \mathcal{V} : tag(v) = \max_i\{tag(i)\}$
    if $\exists C \in LCycles(v)$ marked as unclassified then
      Choose $C \in LCycles(v)$ with best coefficient
      Mark $C$ as active and any other cycle $C' : v \in C', C'$ unclassified as inhibited
    else
      Choose randomly $C \in LCycles(v), C$ inhibited
      Mark $C$ as active and $C' : v \in C'$ with $C'$ being unclassified or active as inhibited
    end if
    Update all $tags$ and $\mathbf{A}_{\text{int}}$ accordingly
  end while
```

A common projection procedure in this context is to compare the entries of $\mathbf{A}_{\text{relax}}$ with a given threshold so that the entries that are larger than it are set to one and the rest are set to zero. Although this simple procedure has given good results in the past [11] [12], it can be shown to produce unbalanced graphs in heterogeneous networks. A suitable matrix $\mathbf{A}_{\text{int}}$ should satisfy the following three conditions: i) $a(\mathcal{G}(\mathbf{A}_{\text{int}})) > 0$ ii) $\mathbf{A}_{\text{int}}\mathbf{1} = \mathbf{A}_{\text{int}}^T\mathbf{1}$ and iii) $\mathbf{A}_{\text{int}} = \max_{\mathbf{A}}\{\mathcal{L}(\mathbf{A})\}$. In other words, it should correspond to a connected and balanced digraph that maximizes the network lifetime.

Our projection procedure, summarized in Algorithm 2, accepts N lists of simple (binary) cycles $LCycles(1, \dots, N)$ and builds $\mathbf{A}_{\text{int}}$ from them. A particular list $LCycles(i)$ includes all cycles containing node i. In order to obtain such lists, we use the algorithm proposed in [18] with two important variations: 1) its input is $\mathbf{A}_{\text{relax}}$, so that the simple cycles with higher coefficients are found and returned first[1] and 2) cycles containing links with a small coefficient in $\mathbf{A}_{\text{relax}}$ are discarded[2]. A list of N integers is used to control the number of components in the graph, so we can ensure that the resulting graph is strongly connected. Moreover, note that as $\mathbf{A}_{\text{int}}$ is built from the union of cycles. the graph is ensured to be balanced. Thus, both i) and ii) are satisfied. In the following we present some numerical results to illustrate how close our solution falls from the true optimal.

5. NUMERICAL RESULTS

Our setting includes $N = 100$ nodes randomly deployed in a square area of side $L = 200$m. A path loss exponent $\gamma = 3$ is used and a minimum power $p_{\text{min}} = 10^{-8}$ mW. is required at the nodes to decode the signal, such that $p_{ij} = p_{\text{min}} r_{ij}^{\gamma}$, expressed in mW. Finally, $\mathcal{C}_i$ follows a normal distribution of mean 1000 and variance 250, expressed in Joules. Lets denote by $\mathbf{A}_{\text{opt}}$ the combinatorial optimal topology matrix that gives the maximum network lifetime $\mathcal{L}(\mathbf{A}_{\text{opt}})$, then the convex relaxation provides a matrix $\mathbf{A}_{\text{relax}}$ with an associated network lifetime $\mathcal{L}(\mathbf{A}_{\text{relax}})$ at least as large as $\mathcal{L}(\mathbf{A}_{\text{opt}})$ (since the integer solution is also a feasible solution to the relaxed problem). The projection procedure then uses the solution of the convex relaxation to generate an integer solution $\mathbf{A}_{\text{int}}$ with (possibly suboptimal) value $\mathcal{L}(\mathbf{A}_{\text{int}})$. The analysis of the algorithm leads to a comparison of these three quantities which satisfy $\mathcal{L}(\mathbf{A}_{\text{relax}}) \ge \mathcal{L}(\mathbf{A}_{\text{opt}}) \ge \mathcal{L}_{\text{int}}(\mathbf{A}_{\text{int}})$, see Fig. 2.

6. CONCLUSIONS

This paper presents a topology optimization methodology to maximize the network lifetime. The approach relies on the notion of mirror graph and solves iteratively convex programs to obtain an optimal relaxed solution. A novel projection procedure is proposed to recover integer solutions. Numerical results showcase the benefits of the proposed scheme.

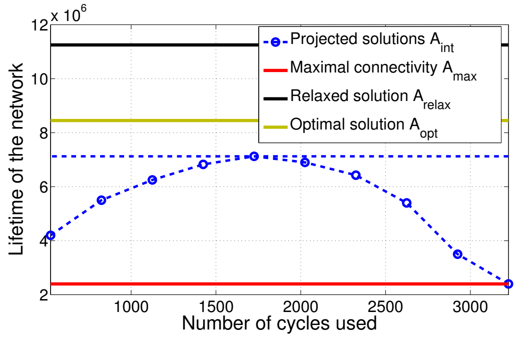

Fig. 2. Network lifetime for $\mathbf{A}_{\text{int}}$, $\mathbf{A}_{\text{max}}$, $\mathbf{A}_{\text{relax}}$ and $\mathbf{A}_{\text{opt}}$.

[1] This can be implemented by choosing the next node in the path according to the adjacency coefficient instead of using any other random criteria.

[2] This stops the recursion before exploring all the branches, which reduces considerably the computational cost of the algorithm.

7. REFERENCES

[1] L. Mainetti, L. Patrono and A. Vilei, “Evolution of wireless sensor networks towards the internet of things: A survey,” in *Proc. International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, pp., 1–6.

[2] D. Alonso-Román and B. Beferull-Lozano, “Adaptive consensus-based distributed detection in wsn with unreliable links,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp., 4438–4442.

[3] D. Alonso-Román and B. Beferull-Lozano, “Adaptive consensus-based distributed kalman filter for wsns with random link failures,” in *Proc. International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, pp., 187–192.

[4] R. Olfati-Saber, J.A. Fax and R.M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[5] S. Xiong, J. Li, and L. Yu, “Maximize the lifetime of a data-gathering wireless sensor network,” in *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON’09)*. IEEE, pp., 1–9.

[6] O. Cayirpunar, B. Tavli, E. Kadioglu-Urtis and S. Uludag, “Optimal mobility patterns of multiple base stations for wireless sensor network lifetime maximization,” *IEEE Sensors Journal*, vol. 17, no. 21, pp. 7177–7188, 2017.

[7] N. H. Mak and W. KG Seah, “How long is the lifetime of a wireless sensor network?,” in *Proc. International Conference on Advanced Information Networking and Applications (AINA’09)*. IEEE, pp., 763–770.

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

[9] C. Asensio-Marco and B. Beferull-Lozano, “Accelerating consensus gossip algorithms: Sparsifying networks can be good for you,” in *Proc. IEEE International Conference on Communications (ICC)*. IEEE, pp., 1–5.

[10] B. Bejar, and M. Vetterli, “Topology optimization for energy-efficient communications in consensus wireless networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp., 5085–5089.

[11] 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, 2012.

[12] 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, 2015.

[13] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[14] Ch. Wah Wu, “Algebraic connectivity of directed graphs,” *Linear and multilinear algebra*, vol. 53, no. 3, pp. 203–223, 2005.

[15] R. Olfati Saber, and R. M Murray, “Agreement problems in networks with directed graphs and switching topology,” in *Proc. IEEE Conference on Decision and Control*. IEEE, vol., pp., 4126–4132.

[16] W. Dinkelbach, “On nonlinear fractional programming,” *Management science*, vol. 13, no. 7, pp. 492–498, 1967.

[17] S. Boyd,and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[18] D. B Johnson, “Finding all the elementary circuits of a directed graph,” *SIAM Journal on Computing*, vol. 4, no. 1, pp. 77–84, 1975.