Lifetime Maximization of an Internet of Things (IoT) Network based on Graph Signal Processing

Holm, Josefine; Chiariotti, Federico; Nielsen, M; Popovski, Petar

Published in:
I E E E Communications Letters

DOI (link to publication from Publisher):
10.1109/LCOMM.2021.3085023

Publication date:
2021

Document Version
Accepted author manuscript, peer reviewed version

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2021.3085023, IEEE Communications Letters

Lifetime Maximization of an Internet of Things (IoT) Network based on Graph Signal Processing

Josefine Holm, Federico Chiariotti, *Member, IEEE*, Morten Nielsen, and Petar Popovski, *Fellow, IEEE*.

***Abstract*—The lifetime of an Internet of Things (IoT) system consisting of battery-powered devices can be increased by minimizing the number of transmissions per device while not excessively deteriorating the correctness of the overall IoT monitoring. We propose a graph signal processing based algorithm for partitioning the sensor nodes into disjoint sampling sets. The sets can be sampled on a round-robin basis and each one contains enough information to reconstruct the entire signal within an acceptable error bound. Simulations on different models of graphs, based on graph theory and on real-world applications, show that our proposal consistently outperforms state-of-the-art sampling schemes, with no additional computational burden.**

***Index Terms*—Graph signal processing, Internet of Things, sampling set selection.**

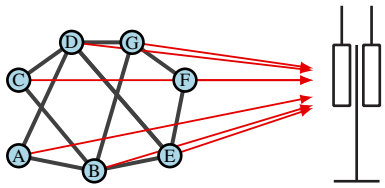


Fig. 1: Concept drawing of the system on a slice of the test network. Communication between base station and sensors (red arrows) and associations between sensors (black lines).

I. Introduction

The expansion of Internet of Things (IoT) systems leads to massive data produced from a vast variety of connected devices and sensors, providing unprecedented knowledge about the state and processes of the physical world [1] [2]. These IoT sensors are often cheap, wireless and battery-powered. They transmit data sporadically to a Base Station (BS), which forwards the data to a data analytics module. In order to increase the longevity of the massive IoT network, each sensor should try to minimize the number of transmissions, while not compromising the quality of the inference at the data analytics module. This letter addresses the optimization of the conflicting objectives of network lifetime and correctness of inference by casting the problem in the context of signal processing on graphs.

We consider a deployment of an IoT network in which the underlying graph structure is induced naturally by the physical deployment of the IoT devices. This concept is shown on Figure 1, where the IoT sensors are attached at the vertices of a water supply network. Other similar examples include IoT devices attached to other types of utility networks, sensors deployed along streets, etc. Each sensor transmits the data to a BS. Due to the physical setup, the readings of the sensors connected within the physical graph are correlated and sampling of one of them carries information about the potential reading of the other sensor. The objective is to partition the set of sensors into S subsets, such that the subsets are disjoint and all sensors are assigned to exactly one set. In each *sampling round* the data analytics module reconstructs the state of the graph based only on the transmission of a subset of sensors. This reduces the duty cycle requirements for each sensor, as it has to transmit S times less often. Partition sampling can then, in principle, increase the IoT network lifetime by S times, which can be equivalent to an order of magnitude increase. However, it comes at the cost of increased error in the reconstructed signal at the receiver, as the data are incomplete. Finding a partition that minimizes that error is critical to achieve the best possible balance between increased battery lifetime and reconstructed signal quality.

The reconstruction of the state can leverage the methods of the emerging field of signal processing on graphs [3], [4]: the graph structure, representing the correlation among the measurements from the different sensors, can be exploited to improve the reconstruction of the signal from each of these subsets. The optimal partition of the graph, i.e., the one that provides the lowest reconstruction Mean Square Error (MSE) for a given number of subsets S, is a combinatorial problem, as the value of adding a node to the sampling set depends on the other nodes already in the set. Most works in the literature have concentrated on finding a single set, and not a complete partition. To the best of our knowledge, the first work to do so was [5], which considered band-limited signals, and find a sampling set that allow for perfect reconstruction under certain conditions. A later work [6] aimed at finding the smallest possible set for a given MSE bound, while [7] concentrated on the opposite problem, i.e., finding the sampling set of fixed size with the minimum MSE. A modification of [6] that is resilient to packet losses was presented in [8], but the authors still do not take into account the fact that the data stemming from the graph nodes are a time series, while [9] considers the temporal aspect but neglects to consider the fact that the importance of a node depends on the other nodes in the subset.

The possibility of exploiting graph structures and correlations in the data from different IoT sensors has already been considered for water distribution networks [10], which are uniquely suited for this thanks to the strong correlation between nodes and lack of high-frequency dynamics in the graph. In general, reducing the activity of each sensor can give huge benefits in IoT networks, as sensors are generally battery-powered and expected to last for several years.

This work extends the heuristic from [6] towards finding the largest complete partition, all subsets of which respect the MSE constraint. Our main contribution is a new algorithm that allows

J. Holm, F. Chiariotti, and P. Popovski are with the Department of Electronic Systems, Aalborg University, Denmark (email: jho@es.aau.dk, fchi@es.aau.dk, petarp@es.aau.dk). M. Nielsen is with the Department of Mathematical Sciences, Aalborg University, Denmark (email: mnielsen@math.aau.dk).

us to find the largest partition of the set of nodes in the graph that respects an MSE constraint. While our algorithm is a heuristic, it can be very close to the optimal by considering the effect of the existing nodes in each subset when adding nodes. This work shares some similarities with [9], but we take the correct importance of each node, considering the elements already in the various subsets, instead of simplifying the problem by assuming its independence from other elements. Our algorithm can run with a similar computational cost to [6] even though we find a complete partitioning, and reduces the average MSE by approximately 5%.

II. System Model

Consider a network of IoT devices wirelessly transmitting to a Base Station. Such a network can be described as a graph where the sensors are represented with nodes and the edges represent similarity between the sensors in terms of what values they observe. The edges can be organized into an adjacency matrix $A \in \mathbb{R}^{N\times N}$, where N is the number of nodes and $A_{i,j} \neq 0$ if and only if there is an edge between node i and j. The observations of the sensors are considered a signal, this signal exist in both time and space, such that for each sensor there is a time series and for each timestamp there is a graph signal. We will denote the graph signals by $x \in \mathbb{R}^N$. In this letter we propose a method of prolonging the lifespan of the sensors. We split the sensors in disjoint sets and, at a given timestamp, the Base Station (BS) samples only one set. If the sets are sampled on a round-robin basis, then the channel usage is reduced and thus the network lifetime is prolonged. The problem is how to do this and still be able to reconstruct the full graph signal at each timestamp within a reasonable margin of error.

In order to solve this problem we utilize concepts from graph signal processing [3]. The graph Fourier transform is defined in terms of the of the eigenvectors of the graph Laplacian defined as $L = D - A$, where D is the degree matrix, i.e. a diagonal matrix whose i-th element is the sum of the weights going into vertex i. We write the eigen decomposition as $L = VEV^H$, where $V = [v_1 \cdots v_N]^H$ is a matrix with the eigenvectors of L and E is a diagonal matrix with the corresponding eigenvalues. We use the eigen decomposition to define the Fourier transform of x on the graph as $\bar{x} = V^H x$.

If we assume that $x \in \mathbb{R}^N$ is a spectrally sparse signal on the graph, i.e., $x = V_{\mathcal{K}} \bar{x}_{\mathcal{K}}$, where $|\mathcal{K}| \ll N$ and $V_{\mathcal{K}} \in \mathbb{R}^{N\times|\mathcal{K}|}$ are the columns of V with index in the set $\mathcal{K}$, then compression is obtained in the graph Fourier domain. Each node i can measure one component of the signal, with an additional noise component: the fully sampled signal is then $y = x + w$, where the noise, w, follows a zero mean circular distribution with $\Lambda_w = \text{diag}(\lambda_{w,i})$. We remark that this noise is inherent in the measurement operation, and is not due to the communication link with the BS. It is further assumed that the noise has full rank. Even if the sampling set contains every node in the graph, the reconstruction of the signal at the BS will never be perfect due to this measurement noise. If we consider sampling sets that are strict subsets of the graph, the reconstruction quality will degrade accordingly.

This imperfect reconstruction from smaller sampling sets is the one we consider: as in [9], the goal is to divide the sensors into disjoint sampling sets, where each set produce good enough representation of the entire graph signal on its own. The approach in [9] is to calculate the importance of of each node individually and then use that to group the nodes. However, that approach does not take into account that the importance of the nodes depends on which nodes they are grouped with. This is the main difference from our approach, as we recalculate the importance at each step to take maximum advantage of the graph structure. To this end, we use the reconstruction MSE for a sampling set as in [6]. The authors in [6] define the optimal linear interpolation operator to obtain the reconstructed signal $\hat{x} = By$, accounting for the fact that the importance of the nodes depends on the other nodes in the set; however, the objective is to construct only one sampling set. Specifically, the goal in [6] is to make the smallest possible sampling set with reconstruction error below a given threshold. An optimal linear interpolation operator B^* can be found for each sampling set by minimizing the interpolation error covariance matrix

$$B^* = \underset{B}{\text{argmin}} \ \mathbb{E}[(x - By)(x - By)^H | x, w]. \tag{1}$$

In the following, we use the simplified notation $B^* y = \hat{x}$ to indicate the reconstructed vector after interpolation. Using the approach from [6] as a starting point and putting it in the context of network longevity, we create several disjoint sampling sets that are all sufficiently good to reconstruct the original graph signal below the error tolerance. We will denote the partitioning G, G_p is set number p and $G_{p,j}$ is iteration j of set p. It is defined as $\text{MSE}(\hat{x}) = E\|x - \hat{x}\|_2^2$. As we follow the same model as defined in [6], we can exploit their derivation of the maximum MSE for the sampling set:

$$\text{MSE}(G_p) = Tr[K(G_p)], \tag{2}$$

where

$$K(G_p) = V_{\mathcal{K}} \left(\Lambda^{-1} + \sum_{i \in G_p} \lambda_{w,i}^{-1} v_i v_i^H \right)^{-1} V_{\mathcal{K}}^H. \tag{3}$$

In [6, Prop 1] it is shown that (3) is the error covariance matrix of $\hat{x}$ if we use the optimal interpolation $\hat{x} = B^* y$ from (1) given a sampling set. This can be rewritten to calculate the MSE iteratively as follows:

$$Tr[K(G_p)] = V_{\mathcal{K}}^H V_{\mathcal{K}} \left(\Lambda^{-1} + \sum_{i \in G_p} \lambda_{w,i}^{-1} v_i v_i^H \right)^{-1}, \tag{4}$$

because trace is rotational, i.e., $Tr[ABCD] = Tr[CDAB]$. We then define:

$$A = \Lambda^{-1} + \sum_{i \in G_p} \lambda_{w,i}^{-1} v_i v_i^H. \tag{5}$$

Let K_j be a shorthand for $K(G_p)$ where G_p has j nodes. We will use this for the iterative calculation of $K(G_p)$.

$$Tr[K_{j-1}] = Tr[V_{\mathcal{K}}^H V_{\mathcal{K}} A^{-1}] \tag{6}$$

$$Tr[K_j] = Tr[V_{\mathcal{K}}^H V_{\mathcal{K}} (A + \lambda_{w,s}^{-1} v_s v_s^H)^{-1}]. \tag{7}$$

We then use the matrix inversion lemma

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U \left(C^{-1} + V A^{-1} U \right)^{-1} V A^{-1}, \tag{8}$$

using $A = A$, $U = v_s$, $V = v_s^H$ and $C = \lambda_{w,s}^{-1}$ we get

$$Tr[K_j] = Tr[K_{j-1} - V_{\mathcal{K}}^H V_{\mathcal{K}} \frac{A^{-1} v_s v_s^H A^{-1}}{(\lambda_{w,s}^{-1})^{-1} + v_s^H A^{-1} v_s}]. \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2021.3085023, IEEE Communications Letters

Therefore, the iterative calculation of of the MSE becomes:

$$\text{MSE}(G_{p,j}\cup v_s)=Tr[K_j]-\frac{v_s^H K_j V_{\mathcal{K}}^H V_{\mathcal{K}} K_j v_s}{\lambda_{w,s}+v_s^H K_j v_s}, \quad (10)$$

where

$$K_j=K_{j-1}-V_{\mathcal{K}}^H V_{\mathcal{K}}\frac{K_{j-1}v_u v_u^H K_{j-1}}{\lambda_{w,u}+v_u^H K_{j-1} v_u}, \quad (11)$$

$K_0=\Lambda$ and u is the index for the most recently added node. Note that $V_{\mathcal{K}}^H V_{\mathcal{K}}=I$ for $|\mathcal{K}|=N$, as well as:

$$\text{MSE}(v_s)=\text{MSE}(\emptyset\cup v_s). \quad (12)$$

The main problem we will propose a solution to is formulated as:

$$\begin{aligned}&\text{maximize} \quad k\\ &\text{subject to} \quad \text{MSE}(G_p)\leq\epsilon \quad \forall\, p=0,...,k.\end{aligned} \quad (13)$$

III. Algorithm

The problem in (13) is combinatorial, as the MSE of each subset depends on all the elements of the set, and cannot be solved efficiently. For this reason, we propose a heuristic Joint Iterative Partitioning (JIP) algorithm which builds on the basic idea of the Individual Iterative Partitioning (IIP) algorithm from [9, Algorithm 1]. However, the importance of each node is not considered to be fixed, but we compute the possible gain from adding it to a sampling set considering the nodes already in the set: in this way, we can improve the balance between the sets by putting each node in the subset where it can improve the MSE the most.

The idea behind JIP is to start by calculating the MSE for all possible sampling sets with one node and sorting the nodes based on this. We take the largest node and add it to the first set. If MSE $<\epsilon$ the process is done, otherwise we search for the best node to add. If more than one node makes MSE $<\epsilon$ we choose the one with the smallest MSE alone and the set is finished. If no such node exists, we choose the one that minimizes the MSE for the set. Then we find the next node to add to the set. An example of how this partition is made can be seen in Fig. 2. The graph is divided into three sets by JIP, which iteratively adds the nodes to the sets. The numbers on the nodes in the figure indicate the order in which they were added, while their color indicates the set in which they were placed. The first node to be assigned is the one with the most neighbors, which makes sense, as its low distance from several others means that the reconstruction drawn only from that node has the lowest MSE. The second node added covers a part of the graph that is far from the first node, but it is still connected to several other nodes. After the third node, the green set is over the threshold, so the algorithm moves to a new set. As the best nodes are selected first, the number of nodes in the sets increases, and the red set has 7 nodes. The last node is allocated to a random set, as it cannot form a set on its own and all existing sets are below the threshold MSE. The allocation of these "orphan" nodes, which cannot form another set with the required MSE, can be performed in various ways, with the effect of further reducing the MSE, but it cannot increase the number of sets in the partition.

The algorithm requires $O(n^2|\mathcal{K}|^2)$ operations, the same as [6] in the worst case scenario. The pseudo-code for our approach is given in Algorithm 1.

Ones the sampling sets are selected we create the sampling matrix S by setting $S_{i,j}$ as 1 if $i\in G_j$ and 0 otherwise, and then get the samples $y_j=S_j x$.

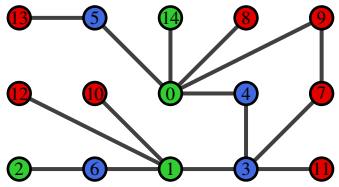


Fig. 2: Graph partitioned into 3 sets.

Algorithm 1 Joint Iterative Partitioning (JIP)

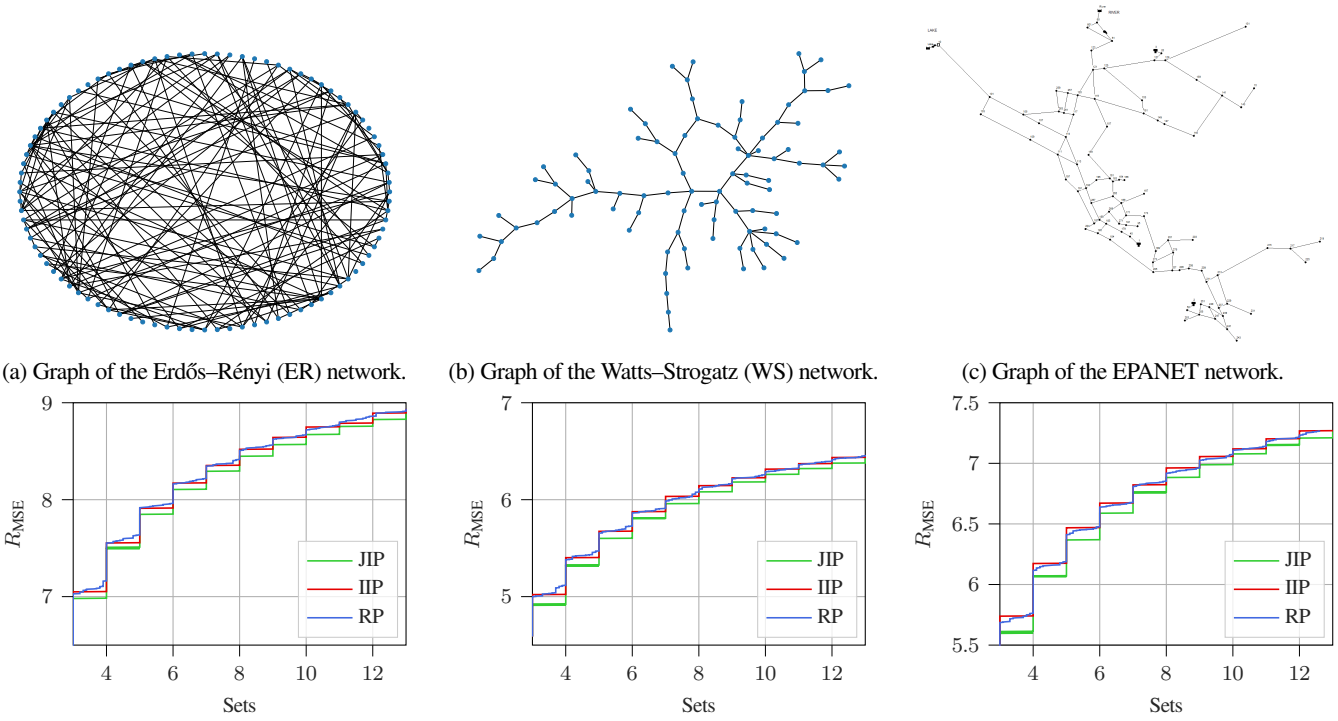
1: Input: $\Lambda,\Lambda_w,V,\epsilon$
2: $K_0=\Lambda, p=0, j=1$
3: $L_s=\text{MSE}(v_s)$
4: $L\_index=argsort(L)$ (largest first)
5: **while** $L\_index\neq\emptyset$ **do**
6: $u=pop(L\_index_{-1})$
7: $G_{p,j}=u$
8: **while** $\text{MSE}(G_{p,j})>\epsilon$ and $L\_index\neq\emptyset$ **do**
9: Calculate K_j according to (11)
10: **for** i in L_index **do**
11: **if** $\text{MSE}(G_{p,j}\cup L_i)<\epsilon$ **then**
12: $G_{p,j+1}=\{G_{p,j}\cup L_i\}$
13: $j=0, p=p+1, delete(L\_index=i)$
14: goto 5
15: $G_{p,j+1}=\{G_{p,j}\cup\text{argmin}(\text{MSE}(L))\}$
16: remove chosen node from L_index, $j=j+1$
17: **if** $\text{MSE}(G_{-1,-1})>\epsilon$ **then**
18: Split the nodes in G_{-1} among the other sets
19: Return G

Adaptation of IIP for comparison: In this section we adapt the IIP algorithm from [9] to this setting. The setting is similar in that we also want to minimize the energy consumption of the network by splitting the nodes into disjoint sampling sets. The Authors in [9] assumes that data from the sensors are available and use this to infer a graph structure. In this paper we assume that the graph structure is given, but we have no assumption about the availability of data from the nodes. Therefore, we will skip step 2 and 3 in IIP and instead add the known graph structure to the initialization step. Once a graph structure has been established, the algorithm orders the nodes in descending order of sampling importance in steps 4 and 5. We make a slight modification to this step, as the original algorithm used a Root MSE (RMSE) measure to sort the nodes, while we take the expected MSE which can be computed by creating the sampling matrix for the known graph structure. Once the nodes are sorted, they are iteratively assigned to sampling sets following a greedy approach: the first nodes, i.e., the ones with the lowest RMSE, are put in the first set, until the RMSE of the reconstruction is below the threshold, in steps 10 and 11. As we use the expected value of the RMSE instead of the statistical average, we cannot preform a reconstruction to perform this check, so we just check if the expected RMSE is below the threshold. With this adaptation, IIP from [9] can be compared fairly to the proposed algorithm.

IV. Numerical Results

Figure 3d shows the results of three partitioning methods used on the ER graph structure. It shows a small gain from using JIP rather

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2021.3085023, IEEE Communications Letters



(a) Graph of the Erdős–Rényi (ER) network. (b) Graph of the Watts–Strogatz (WS) network. (c) Graph of the EPANET network.

(d) Relative error for different number of sets created for the ER graph structure. (e) Relative error for different number of sets created for the WS graph structure. (f) Relative error for different number of sets created for the EPANET graph structure.

Fig. 3: The graphs and the results.

Algorithm 2 Random Partitioning (RP)

Input: Λ,Λ_w,V,ϵ
$K_0=\Lambda$,$p=0$,$j=1$
$L\_index=[0,1,...,N]$
while $L\_index\neq\emptyset$ **do**
  $r=random(0,|L\_index|)$
  $u=pop(L\_index_r)$
  $G_{p,j}=u$
  while MSE$(G_p)>\epsilon$ and $L\_index\neq\emptyset$ **do**
    Calculate K_j according to (11)
    $r=random(0,|L\_index|)$
    $u=pop(L\_index_r)$
    $G_{p,j+1}=\{G_{p,j}\cup u\}$
    if MSE$(G_{p,j+1})<\epsilon$ **then**
      $j=0$,$p=p+1$
      goto 4
    $j=j+1$
if MSE$(G_{-1,-1})>\epsilon$ **then**
  Split the nodes in G_{-1} among the other sets
Return G

than IIP or Random Partitioning (RP). However, for the method to work well we need to have some structure to the graph. The advantage of the JIP method over simple IIP is that it considers the relations between the nodes already in a sampling set and the new node to be added, so the difference between the methods is expected to be starker for graphs with highly local structures, i.e., more clustered ones. Figure 3e shows the results for a WS graph structure with $\beta=0.5$. As expected the gain from using JIP is mush bigger and the relative error is lower for all partitioning methods.

In this section we compare the proposed JIP algorithm to the IIP algorithm and a RP algorithm. The RP algorithm creates one set at a time by adding randomly chosen nodes to the set until it is below the given threshold, the pseudo-code for the RP algorithm is given in Algorithm 2. We will compare the three methods on three different graph structures, namely, an ER graph, a WS graph and a graph simulated with EPANET, [11], a tool made for simulation of water supply networks, as shown in Figure 3c. The EPANET graph can represent a real-world application of the use of our graph sampling technique in IoT networks, although it is still simulated. All graphs are undirected, unweighted and has a similar edge density.

The sampling noise λ_w is included in the calculation of the MSE, therefore we want to present the results of the different algorithms and graphs relative to the MSE sampling set that is only affected by noise i.e. the set with all nodes, F: $R_{\text{MSE}}(G)=\frac{\text{MSE}(G)}{\text{MSE}(F)}$.

Lastly, Figure 3f shows the results for the graph simulated with the EPANET tool. This graph has some clustering, but not as mush as the WS graph, and the results show that both the relative error and the gain form using JIP is somewhere in between the results for ER and WS, which is what we can expect from most real world graphs.

In order to show the optimality gap of our heuristic, we have performed an Exhaustive Search (ES) for all possible partitions on two graphs with 15 nodes, a WS graph and an ER graph. Performing ES on larger graphs is infeasible, because the computation time grows exponentially. Figure 4 shows the MSE of the best possible partitioning with two, three and four sets, measured by the maximum MSE for any set in the partitioning. For the three greedy sampling algorithms it shows the lowest possible threshold that gives the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2021.3085023, IEEE Communications Letters

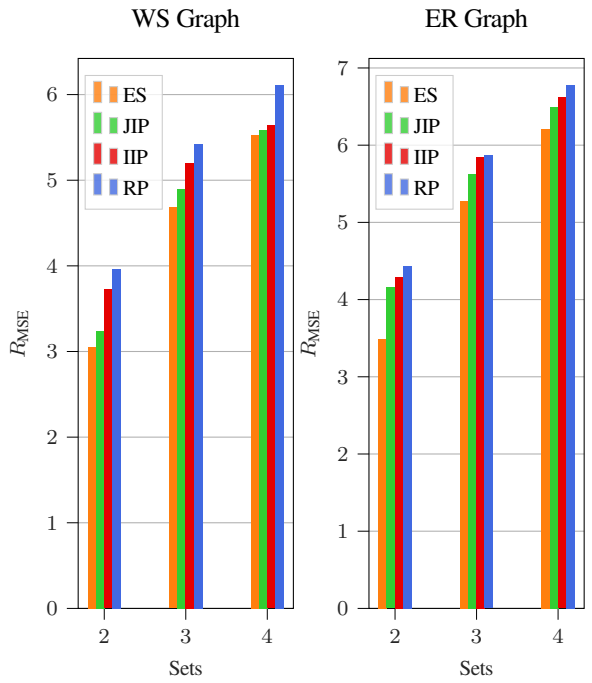


Fig. 4: Exhaustive search for best partition on two graphs with 15 nodes.

same amount of sets. As the figure shows, JIP is significantly closer to the optimum than IIP or RP in most cases, particularly on the WS graph structure. For the ER graph all sampling algorithms struggled with the partition in two sets, but for three and four sets we lower the MSE threshold by 30-40% compared to IIP and 40-50% compared to RP. For a theoretical analysis of the optimality of the proposed algorithm we refer to [6]. They perform an in-depth analysis of the approximate supermodularity of using (2) for greedy optimization.

The main advantage of our proposed scheme can be summarized as follows. Given that the underlying graph structure is known *a priori*, the algorithm can be run before the sensors start to collect any data and they can thus be preconfigured. Although very infrequently, the graph structure may change over time; it is then important to update the sampling scheme accordingly. However, how to do that efficiently is a subject of future work.

The method works well for cases where the graph structure can be induced from some physical aspects of the setup. An example of this is the case of a network of water pipes, here the connections of the pipes gives a natural way of connecting the nodes. Because the sampling scheme is derived before deployment and all sensors sample periodically, as seen from the sensor perspective, the strain on the sensors from computations is practically nonexistent. To reconstruct the entire dataset we must assume that the server has a reasonable amount of computation power and memory available and that we allow for some error ϵ.

V. Conclusion

We have considered the problem of energy-efficient sampling of IoT sensors that are deployed in a system with an underlying graph structure. We leverage this structure through a graph signal processing framework to obtain the maximal number of disjoint sets to be sampled, while the the expected mean square error for each set is below a given error threshold. We have evaluated the algorithm in a scenario where the underlying graph is build according to a water network. The results show that JIP gives a lower error than IIP and RP. Although the gains are not particularly high, the methodologically novel approach shows promises through its consistent improvement over the other approaches and incurs no additional computational cost. Furthermore, our results show that the gain from using our scheme increases for more clustered graphs like the WS small-world model, making JIP particularly suited for those scenarios. As a future work, we will generalize the approach to allow for efficient update of the partitions upon changes in the underlying physical graph.

Acknowledgment

This work has, in part, been supported by the Danish Council for Independent Research (Grant No. 8022-00284B SEMIOTIC).

References

[1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5g: Physical and mac-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.

[2] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.

[3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging filed of signal processing on graphs," *IEEE Signal Processing Magazine*, 2013.

[4] G. B. Ribeiro and J. B. Lima, "Graph signal processing in a nutshell," *Journal of Communication and Information Systems*, vol. 33, no. 1, 2018.

[5] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3864–3868.

[6] L. F. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2017.

[7] F. Wang, Y. Wang, and G. Cheung, "A-optimal sampling and robust reconstruction for graph signals via truncated neumann series," *IEEE Signal Processing Letters*, vol. 25, no. 5, pp. 680–684, 2018.

[8] B. Güler, A. Jayawant, A. S. Avestimehr, and A. Ortega, "Robust graph signal sampling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7520–7524.

[9] A. Chiumento, N. Marchetti, and I. Macaluso, "Energy efficient wsn: a cross-layer graph signal processing solution to information redundancy," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 645–650.

[10] Z. Wei, A. Pagani, G. Fu, I. Guymer, W. Chen, J. McCann, and W. Guo, "Optimal sampling of water distribution network dynamics using graph Fourier transform," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1570–1582, 2019.

[11] L. A. Rossman *et al.*, "EPANET 2: users manual," 2000.