Resource Optimization in a Wireless Sensor Network with Guaranteed Estimator Performance

Ling Shi*, Agostino Capponi**, Karl H. Johansson† and Richard M. Murray‡

Abstract

New control paradigms are needed for large networks of wireless sensors and actuators in order to efficiently utilize system resources. In this paper we consider the problem of discrete-time state estimation over a wireless sensor network. Given a tree that represents the sensor communications with the fusion center, we derive the optimal estimation algorithm at the fusion center, and provide a closed-form expression for the steady-state error covariance matrix. We then present a tree reconfiguration algorithm that produces a sensor tree that has low overall energy consumption and guarantees a desired level of estimation quality at the fusion center. We further propose a sensor tree construction and scheduling algorithm that leads to a longer network lifetime than the tree reconfiguration algorithm. Examples are provided throughout the paper to demonstrate the algorithms and theory developed.

$*$: Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Email: eesling@ust.hk.

$**$: Computer Science, California Institute of Technology, Pasadena, CA 91125. Email: acapponi@cs.caltech.edu.

†: School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. Email: kallej@ee.kth.se.

‡: Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91106. Email: murray@cds.caltech.edu.

The work by L. Shi is supported by grant DAG08/09.EG06. The work by A. Capponi is supported by a fellowship granted by the Social and Information Sciences Laboratory at Caltech. The work by K. H. Johansson is supported by the Swedish Research Council and the Swedish Foundation for Strategic Research. The work by R. Murray is supported in part by AFOSR grant FA9550-04-1-0169.

Some preliminary results [1] of this paper were presented by the same authors (excluding the 2nd author) at the 46th IEEE Conference on Decision and Control at New Orleans, 2007.

I. Introduction

Wireless sensor networks (WSN) have attracted much attention in the past decade which can be used for environment and habitat monitoring, health care, home and office automation, traffic control, etc. [2]. In WSN, there is an economic incentive towards using off-the-shelf sensors and standardized communication solutions. A consequence of this is that the individual hardware components might be of relatively low quality and that communication resources are quite limited. Thus a single sensor may not be enough to provide a desired level of estimation quality, and data fusion from multiple sensors is often required.

Estimation and control over such resource-constrained networks thus require new design paradigms beyond traditional sampled-data control. For example, consider the problem of state estimation over such a network using a Kalman filter. The Kalman filter [3] is a well-established methodology for model-based fusion of sensor data [4]. Kalman filtering under certain information constraints, such as decentralized implementation, has been extensively studied [5]. The interaction between Kalman filtering and how data is routed on a network seems to be less studied.

Another issue inherent with WSN is the limited energy resource available at each sensor node which is typically battery-powered. Periodically changing the battery is often difficult and expensive, and sometimes even impossible. Thus any good design must fully consider the energy resource constraint and minimize the sensor energy consumption as much as possible.

Sensor network energy minimization is typically done via efficient MAC protocol design [6], or via efficient scheduling of the sensor states [7]. A sensor transmitting scheduling was suggested by Chen et al. [8]. Lai et al. [9] proposed a scheme to divide the deployed sensors into disjoint subsets of sensors such that each subset can complete the mission, and then maximized the number of such disjoint subsets.

In this paper, we consider the problem of centralized state estimation while considering sensor energy constraint. The main contributions are summarized as follows.

1) Given a tree that represents the sensors' communications with the fusion center, we derive the optimal estimation algorithm at the fusion center, and we provide a closed-form expression on the steady-state error covariance matrix.
2) We present a tree reconfiguration algorithm that produces a sensor tree having low overall energy consumption and providing a desired level of estimation quality at the fusion center.
3) We propose a sensor tree construction and scheduling algorithm that leads to a longer network lifetime than the tree reconfiguration algorithm.

Routing protocols have been widely investigated in the literature. The main efforts have concentrated toward defining protocols that discover routes on demand using local and scalable technique, while avoiding the overhead of storing routing tables or other expensive information such as link costs or topology changes. The main challenge in those works is in discovering paths that are both time and energy efficient, meaning that the information is delivered across the network in a reasonable amount of time and at the minimum cost. Some examples include energy aware routing, attributed based routing, rumor routing and directed diffusion. We refer the reader to [10] for a more detailed treatment. The focus of our paper is different since we want to simultaneously solve an estimation and energy minimization problem. More specifically, we want to find the most efficient network topology given constraints on the estimation performance measured by the estimation error covariance matrix. Differently from the works mentioned above, our network topology is static and recovered once for all as the solution of an optimization problem.

All algorithms have low complexity which leads to efficient design and implementation in practice. Furthermore, the low complexity brings the plug-and-play feature to the network, i.e., a new tree can be calculated and dynamically formed when new sensors join the network and existing sensors quit from the network, or when the performance requirement is time-varying (e.g., see the example in Section IV-C).

The rest of the paper is organized as follows. In Section II, we give the mathematical models of the considered problems, and provide some preliminary results on Kalman filtering to facilitate the analysis in remaining sections. In Section III, we derive the optimal estimation algorithm at the fusion center for a given sensor tree. In Section IV, we present the a sensor tree reconfiguration algorithm. Then in Section V, we propose a sensor tree construction and scheduling algorithm that leads to a longer lifetime than the sensor tree reconfiguration algorithm. Concluding remarks as well as future work are given in the end.

II. Problem Setup and Preliminaries

A. Problem Setup

Consider the problem of state estimation over a wireless sensor network (Figure 1). The process dynamics is described by

$$x_{k+1} = Ax_k + w_k. \tag{1}$$

A wireless sensor network consisting of N sensors $\{S_1, \ldots, S_N\}$ is used to measure the state. When S_i takes a measurement of the state in Eqn (1), it returns

$$y_k^i = H_i x_k + v_k^i. \ (i = 1, \ldots, N.) \tag{2}$$

In Eqn (1) and (2), $x_k \in \mathbb{R}^n$ is the state vector in the real n-dimensional vector space, $y_k^i \in \mathbb{R}^{m_i}$ is the observation vector at S_i, $w_k \in \mathbb{R}^n$ and $v_k^i \in \mathbb{R}^{m_i}$ are zero-mean Gaussian random vectors with $\mathbb{E}[w_k w_j{}'] = \delta_{kj}Q,\ Q \geq 0$, $\mathbb{E}[v_k^i v_t^{i}{}'] = \delta_{kt}\Pi_i,\ \Pi_i > 0$, $\mathbb{E}[v_k^i v_t^{j}{}'] = 0 \ \forall\, t, k$ and $i \neq j$, $\mathbb{E}[w_k v_t^{i}{}'] = 0 \ \forall\, i, t, k$, where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ otherwise. We assume that $(A, \sqrt{Q})$ is controllable, and (A, C_{all}) is observable, where $C_{\text{all}} = [H_1; \cdots ; H_N]$, i.e., the joint measurement matrix of all sensors.

Each sensor can potentially communicate via a single-hop connection with a subset of all the sensors by adjusting its transmission power. We assume the communication links are perfect in this paper in the sense that data packets traveling on the links will not be dropped. Thus we will not consider the effect of interference or fading, etc. Let us introduce a fusion center which we denote as S_0, and consider a tree T with root S_0 (see Figure 2). We suppose that there is a non-zero single-hop communication delay, which is smaller than the sampling time of the process. All sensors are synchronized in time, so the data packet transmitted from S_i to S_0 is delayed one sample when compared with the parent node of S_i. We also assume that S_i aggregates the previous time data packets from all its child nodes with its current time measurement into a single data packet. Therefore only one data packet is sent from S_i to its parent node at each time k.

B. Problems of Interest

We are interested in the following problems. The first one is how should the fusion center process the measurement data from the sensors which arrive at different times (due to the multi-hop communications) such that the estimation error is minimized?

The second problem is related to the energy constraint on the sensor nodes. Apparently, to minimize the energy consumption, sensors should only use minimum transmission energy and communicate with their nearby neighbors; on the other hand, the many short-hop communications introduce many delays when delivering the data to the fusion center. As delays deteriorate the estimation quality, there is a clear tradeoff between how much energy the sensors should spend and how good the estimation quality is at the fusion center. We are thus interested in seeking a low-energy sensor tree which still guarantees a desired level of estimation quality at the fusion center.

When all sensors need to participate in the estimation, minimizing the total energy consumption might not lead to a longer lifetime of the network as demonstrated by the example in Section V-C. Therefore we are also interested in schemes that can maximize the network lifetime.

In Section III, IV, and V we provide answers to the above three problems respectively. Before we present the main result of the paper, we briefly introduce the standard Kalman filtering upon which our optimal estimation algorithm for solving the first problem relies.

C. Kalman Filtering Preliminaries

Consider the process in Eqn (1) with the following single sensor measurement equation:

$$y_k \quad = \quad C_k x_k + v_k, \tag{3}$$

where v_k is zero-mean Gaussian random vectors with $\mathbb{E}[v_k {v_j}'] = \delta_{kj} R_k,\ R_k > 0$, and $\mathbb{E}[w_k {v_j}'] = 0 \ \forall\ j, k$. Notice that we consider time-varying C_k and R_k here. The Kalman filter in its most general form can assume time-varying A and Q. The special form we look at here suffices for deriving the optimal estimation algorithms in later sections.

Assume a linear estimator receives y_k and computes the optimal state estimate at each time k. Let $\mathbf{Y}_k$ denote all measurements available at the estimator at time k. Further define:

$$\hat{x}_k \quad \triangleq \quad \mathbb{E}[x_k | \mathbf{Y}_k], \tag{4}$$

$$P_k \quad \triangleq \quad \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)' | \mathbf{Y}_k], \tag{5}$$

$$\overline{P} \quad \triangleq \quad \lim_{k \to \infty} P_k, \text{if the limit exists.} \tag{6}$$

It is well known that $\hat{x}_k$ and P_k can be computed as

$$(\hat{x}_k, P_k) = \mathbf{KF}(\hat{x}_{k-1}, P_{k-1}, y_k, C_k, R_k),$$

where **KF** denotes the Kalman filter which consists of the following update equations at time k:

$$\hat{x}_k^- \quad = \quad A \hat{x}_{k-1}, \tag{7}$$

$$P_k^- \quad = \quad A P_{k-1} A' + Q, \tag{8}$$

$$K_k \quad = \quad P_k^- C_k' [C_k P_k^- C_k' + R_k]^{-1}, \tag{9}$$

$$\hat{x}_k \quad = \quad \hat{x}_k^- + K_k (y_k - C_k \hat{x}_k^-), \tag{10}$$

$$P_k \quad = \quad (I - K_k C_k) P_k^-. \tag{11}$$

Let $\mathbb{S}_+^n$ be the set of n by n positive semi-definite matrices. For functions $f_1, f_2 : \mathbb{S}_+^n \to \mathbb{S}_+^n$, define $f_1 \circ f_2$ as

$$f_1 \circ f_2(X) \triangleq f_1\big(f_2(X)\big). \tag{12}$$

Further define the functions $h, \tilde{g}_{[C,R]}, g_{[C,R]} : \mathbb{S}^n_+ \to \mathbb{S}^n_+$ as

$$\begin{aligned} h(X) &\triangleq AXA' + Q, && (13)\\ \tilde{g}_{[C,R]}(X) &\triangleq X - XC'[CXC' + R]^{-1}CX, && (14)\\ g_{[C,R]}(X) &\triangleq h \circ \tilde{g}_{[C,R]}(X). && (15) \end{aligned}$$

We write $g_{[C,R]}(X)$ and $\tilde{g}_{[C,R]}$ as g_C and $\tilde{g}_C$ if there is no confusion on the underlying parameters R.

III. State Estimation over a Sensor Tree

Let us define the following state estimate and other quantities at the fusion center S_0. For a given tree T rooted at S_0, define $\hat{x}_k(T), P_k(T), \overline{P}(T)$ at S_0 similar as that in Eqn (4)–(6). We write $\hat{x}_k(T)$ as $\hat{x}_k$, etc., if there is no confusion on the underlying tree T. In this section, we shall compute $\hat{x}_k$ and P_k for a given T.

Assume the tree T has depth D. Define $\mathcal{Y}_k^{k-i}$ as the set of all measurements available at the fusion center for time $k-i$ at time $k, i = 0, \ldots, D-1$. Let S_{i_j} be the sensor node that is j hops away from the fusion center. Define

$$\begin{aligned} \Gamma_j &\triangleq [H_{1_j}; H_{2_j}; \cdots], \quad j = 1, \ldots, D\\ C_i &\triangleq [\Gamma_1; \cdots; \Gamma_i], \quad i = 1, \ldots, D\\ \Upsilon_j &\triangleq \text{diag}\{\Pi_{1_j}, \Pi_{2_j}, \cdots\}, \quad j = 1, \ldots, D\\ R_i &\triangleq \text{diag}\{\Upsilon_1, \ldots, \Upsilon_i\}, \quad i = 1, \ldots, D. \end{aligned}$$

Then the following theorem presents the optimal estimation algorithm over a sensor tree and characterizes the steady-state error covariance matrices in closed-form expression.

Theorem 3.1 ([1]): Consider a sensor tree T with depth D that is rooted at the fusion center. Then

1) $\hat{x}_k$ and P_k can be computed from D Kalman filters in sequence as

$$\begin{aligned} (\hat{x}_{k-D+1}, P_{k-D+1}) &= \textbf{KF}(\hat{x}_{k-D}, P_{k-D}, \mathcal{Y}_k^{k-D+1}, C_D, R_D),\\ &\vdots\\ (\hat{x}_{k-1}, P_{k-1}) &= \textbf{KF}(\hat{x}_{k-2}, P_{k-2}, \mathcal{Y}_k^{k-1}, C_2, R_2),\\ (\hat{x}_k, P_k) &= \textbf{KF}(\hat{x}_{k-1}, P_{k-1}, \mathcal{Y}_k^{k}, C_1, R_1). \end{aligned}$$

2) Furthermore, the steady-state error covariance matrix $\overline{P}$ satisfies

$$\overline{P} = \tilde{g}_{C_1} \circ g_{C_2} \circ \cdots \circ g_{C_{D-1}}(P_\infty), \tag{16}$$

where P_∞ is the unique solution to $g_{C_D}(P_\infty) = P_\infty$.

IV. Minimum-energy Sensor Tree

In this section, we seek a low-energy sensor tree that guarantees a desired level of estimation quality at the fusion center. The following definition are used in the remaining of the paper. Define $\text{Node}(T)$ as all the nodes of T, $\text{Fam}_T(S_i)$ as the subtree of T that is rooted at S_i, $\text{Par}_T(S_i)$ as the parent node of S_i in T, and $\text{Edge}(T)$ as the edges of T, i.e.,

$$\text{Edge}(T) \triangleq \big\{(S_i, S_j) : S_i \in \text{Node}(T), S_j = \text{Par}_T(S_i)\big\}.$$

We sometimes omit the subscript T if there is no confusion on the underlying tree T, e.g., we write $\text{Fam}_T(S_i)$ simply as $\text{Fam}(S_i)$.

We assume to have an energy sensor model regulating the amount of energy expenditure for transmission and reception. Further assume that the total energy used by two sensors (one sending and the other receiving) increases as the distance between the two sensors increases [11]. Since at each time, each sensor sends and/or receives fixed number of data packets, without loss of generality, let $e^i_{tx}(T)$ be the energy cost for S_i sending a measurement packet to $\text{Par}_T(S_i)$ and $e^i_{rx}(T)$ as the energy cost for S_i receiving measurement packets from its children. The total energy cost of T per time is then given by

$$e(T) = \sum_{S_i \in T} e^i_{tx}(T) + e^i_{rx}(T). \tag{17}$$

Denote $\mathcal{T}_{\text{all}}$ as the set of all sensor trees, and let $P_{\text{desired}} \in \mathbb{S}^n_+$ be given. Since the sensors operate on batteries, it is natural to let the network operate at an energy level that is as low as possible. Thus we are interested in the following problem:

Problem 4.1: How can we choose the sensor tree that has the least overall energy consumption yet still provides certain desired level of estimation quality? i.e.,

$$\min_{T \in \mathcal{T}_{\text{all}}} e(T)$$

$$\text{subject to}$$

$$\overline{P}(T) \leq P_{\text{desired}}$$

where the inequality is in the matrix sense, i.e., $P_{\text{desired}} - \overline{P}(T)$ is positive semi-definite. Cayley [12] showed that the number of all possible trees is N^{N-2}, thus solving Problem 4.1 via exhaustive search is intractable when N is large. It is also non-convex, thus finding the global optimal solution is in general difficult. To approximate the global optimal solution, we present the following tree reconfiguration algorithm.

A. Tree Reconfiguration Algorithm

The proposed *Tree Reconfiguration Algorithm* (Figure 3) consists of three subroutines. The first one is the *Tree Initialization Algorithm* which forms an initial tree T_0 (the top rectangular block). Depending on whether T_0 provides the required estimation quality, the *Switching Tree Topology Algorithm* (the middle-right rectangular block) and the *Minimum Energy Subtree Algorithm* (the bottom rectangular block) are executed respectively. These algorithms are presented in detail next.

Tree Initialization Algorithm: The idea contained in the *Tree Initialization Algorithm* is that the fusion center S_0 first establishes direct connections with its neighbor sensors using minimum transmission power level Δe. After that, its neighbor sensors establish further connections with their own neighbor sensors also using minimum transmission power level Δe. This process continues until a tree of depth D is formed. As a result, the complexity of the algorithm is $O(D)$. The algorithm is presented graphically in Figure 4.

Switching Tree Topology Algorithm: For a given tree T_t, if $\overline{P}(T_t) \nleq P_{\text{desired}}$, the tree needs to be adjusted in a way that the estimation quality is improved. The *Switching Tree Topology Algorithm* provides such a way (Figure 5). The idea is that a sensor node in T_t that is two-hops away from the fusion center is reconfigured to directly connect with it, hence becomes only one-hop away from the fusion center. As we prove shortly, this reconfiguration always improves the estimation quality at the fusion center.

We define $\pi(T_t, S_i)$ as the new tree obtained by removing the edge $\big(S_i, \text{Par}_T(S_i)\big)$ and inserting (S_i, S_0). Further define

$$\mathcal{S}_{j-hop} \triangleq \{S_i : S_i \text{ is } j\text{−hop away from } S_0\}. \tag{18}$$

The algorithm is given as follows, where $\text{Tr}(X)$ denotes the trace of the matrix X.

Algorithm 1 SWITCHING TREE TOPOLOGY

```
Init: $T_t$.
Compute
    $S_p = \arg\min_{S_i \in \mathcal{S}_{2-hop}} \text{Tr}\big(\overline{P}(\pi(T_t, S_i))\big).$
Return $T_{t+1} := \pi(T_t, S_p)$.
```

Minimum Energy Subtree Algorithm: For a given tree T with $\overline{P}(T) \leq P_{\text{desired}}$, the *Minimum Energy Subtree Algorithm* finds the subtree T' rooted at S_0 with the property that $\overline{P}(T') \leq P_{\text{desired}}$, and $e(T') \leq e(\tilde{T})$ for any subtree $\tilde{T}$ of T rooted at S_0. The idea is that all possible subtrees $\tilde{T}$ rooted at S_0 and satisfying

$$\overline{P}(\tilde{T}) \leq P_{\text{desired}}$$

are found in an efficient way utilizing the structure of T. Then the subtree T' which has the least overall energy cost is returned. The details are as follows.

To make the presentation clear and easy to follow, we divide the algorithm into several key steps and provide an example to illustrate each step. Before introducing the algorithm, let us define $S(i_1 i_2 \cdots i_p)$ as the subtree that consists of the sensor nodes $\{S_{i_1}, S_{i_2}, \cdots S_{i_p}\}$. We further define $\Omega(i_1 i_2 \cdots i_p)$ as the complementary tree of $S(i_1 i_2 \cdots i_p)$ in T, i.e.,

$$\Omega(i_1 i_2 \cdots i_p) = T \setminus S(i_1 i_2 \cdots i_p).$$

We assume $i_1 \leq i_2 \leq \cdots \leq i_p$. The following example is provided to illustrate the algorithm.

Example 4.2: Consider the tree T with four sensor nodes in Figure 6. Assume the following:

1) $\overline{P}(T) \leq P_{\text{desired}}$, i.e., T provides the desired estimation quality.

2) $\overline{P}(S(i)) \nleq P_{\text{desired}}, i = 1, 2, 3, 4$, i.e., no single sensor provides the desired estimation quality.

3) $\overline{P}(S(ij)) \leq P_{\text{desired}}$ iff $\{i, j\} = \{1, 4\}$, i.e., among the two sensor pairs, only $\{S_1, S_4\}$ can provide the desired estimation quality.

4) $\overline{P}(\Omega(i)) \leq P_{\text{desired}}, i = 2, 3, 4$, i.e., any three sensors except $\{S_2, S_3, S_4\}$ can provide the desired estimation quality.

5) The energy cost of a single-hop communication in T is Δe.

By the above assumptions, it is easy to see that the minimum energy subtree T' is given by $\tilde{T}_4$ with $e(\tilde{T}_4) = 2\Delta e$.

Let us examine the case when we take T as an input to the *Minimum Energy Subtree Algorithm* which consists of the following key steps.

Step 1

- Init: T
- $l := 0, \mathcal{D}_l := \{S_{i_p} \in T : \overline{P}(\Omega(i_p)) \leq P_{\text{desired}}\}$.

In this step, $\mathcal{D}_0$ holds all individual sensors without which the remaining sensors still satisfy the estimation quality constraint. Therefore in Example 4.2, $\mathcal{D}_0 = \{S_2, S_3, S_4\}$.

```
Step 2
• $l := l + 1, \mathcal{D}_l := \mathcal{D}_{l-1}$
• $\forall\ S_{i_p} \in \mathcal{D}_{l-1}$ with $\overline{P}(\Omega(i_p)) \leq P_{\text{desired}}$
    - $\forall\ q > p$ and $S_{i_q} \notin \text{Fam}(S_{i_p})$,
        if $\overline{P}(\Omega(i_p i_q)) \leq P_{\text{desired}}$, $\mathcal{D}_l := \mathcal{D}_l \bigcup S(i_p i_q)$.
```

In this step, $\mathcal{D}_1$ holds all single-sensor or two-sensor pairs without which the remaining sensors still satisfy the estimation quality constraint. The third line of step 2 eliminates the redundancy in listing the subtrees as $S(i_p i_q) = S(i_q i_p)$, and if S_{i_p} is removed from a tree, so is $\text{Fam}(S_{i_p})$. Therefore in Example 4.2, $\mathcal{D}_1 = \{S_2, S_3, S_4, S(23)\}$.

```
Step 3
• $l := l + 1, \mathcal{D}_l := \mathcal{D}_{l-1}$
• $\forall\ S(i_p i_q) \in \mathcal{D}_{l-1}$ with $\overline{P}(\Omega(i_p i_q)) \leq P_{\text{desired}}$
    - $\forall\ o > q$ and $S_{i_o} \notin (\text{Fam}(S_{i_p}) \bigcup \text{Fam}(S_{i_q}))$,
        if $\overline{P}(\Omega(i_p i_q i_o)) \leq P_{\text{desired}}$,
            $\mathcal{D}_l := \mathcal{D}_l \bigcup S(i_p i_q i_o)$.
```

Similar to step 3, $\mathcal{D}_2$ holds all single-sensor, two-sensor pairs or three-sensors without which the remaining sensors still satisfy the estimation quality constraint. The algorithm continues in this way until $\mathcal{D}_r = \mathcal{D}_{r-1}$ at some step $r \leq D$.

```
Step $\mathbf{r+1}$
• Return $T' = \arg\min_{\Omega(\cdot) \in \mathcal{D}} e(\Omega(\cdot))$
```

In Example 4.2, $\mathcal{D}_2 = \{S_2, S_3, S_4, S(23)\} = \mathcal{D}_1$. Hence the algorithm stops and returns $T' = \Omega(23) = S(14) = \tilde{T}_4$ with $\overline{P}(T') \leq P_{desired}$ and $e(T') = 2\Delta e$.

Remark 4.3: In general, the global minimum energy tree depends on the initial tree that we start with. The particular initial tree that we choose is certainly arbitrary but has a low energy consumption. Star tree (e.g., all sensor nodes connect to the fusion center directly) could be another choice, which provides the least estimation error. However it is unlikely to be the minimum energy tree. A better approach may be that start from a few random initial trees and run the algorithms simultaneously. In the end choose the minimum energy tree from all outcomes of the algorithms. This will be the essential idea in the next section when we consider maximizing network lifetime.

B. Performance Analysis of the Algorithms

The performance of the previous algorithms are summarized in the following algorithm.

Theorem 4.4 ([1]): (1) Given a tree T_t, the *Switching Tree Topology Algorithm* returns $T_{t+1} \in \mathcal{T}_{\text{all}}$ such that

$$\overline{P}(T_{t+1}) \leq \overline{P}(T_t) \ .$$

(2) Given a tree T with $\overline{P}(T) \leq P_{\text{desired}}$, the *Minimum Energy Subtree Algorithm* returns $T' \subset T$ rooted at S_0 such that

$$\overline{P}(T') \leq P_{\text{desired}} \text{ and } e(T') \leq e(\tilde{T})$$

for any other $\tilde{T} \subset T$ that is rooted at S_0.

(3) If $\exists\, T \in \mathcal{T}_{\text{all}}$ such that $\overline{P}(T) \leq P_{\text{desired}}$, then the output T' from the *Tree Reconfiguration Algorithm* satisfies $\overline{P}(T') \leq P_{\text{desired}}$.

C. Example

In this section, we provide an example to demonstrate the use of the tree reconfiguration algorithm. Consider the following process with three sensors. The dynamics of the process and sensor measurement equations are as follows:

$$\begin{aligned} x_k &= 0.9x_{k-1} + w_{k-1}, \\ y_k^1 &= x_k + v_k^1, \\ y_k^2 &= x_k + v_k^2, \\ y_k^3 &= x_k + v_k^3, \end{aligned}$$

with $Q = 1, \Pi_1 = 1.5, \Pi_2 = 1$, and $\Pi_3 = 0.5$.

The sensors positions are illustrated in Figure 7. Assume that if S_i is connected to $S_{i-1}, i = 1, 2, 3$, the energy of communication is Δe; if S_i is connected to $S_{i-2}, i = 2, 3$, the energy is $4\Delta e$ and if S_3 is connected to S_0, the energy is $8\Delta e$. Without loss of generality, for the remaining examples, we only calculate the total transmission energy.

Suppose the following performance specification is received by the fusion center:

$$\begin{aligned}\overline{P} &\leq 0.75, 1 \leq k \leq 100,\\ \overline{P} &\leq 0.25, 101 \leq k \leq 200,\\ \overline{P} &\leq 1.0, 201 \leq k \leq 300,\\ \overline{P} &\leq 0.75, 301 \leq k \leq 500.\end{aligned}$$

Then the fusion center can find the corresponding minimum energy tree that fulfills the performance requirement. Figure 8 shows the simulation result when the fusion center uses the same tree ($T_0 \setminus S_3$) all the time, and Figure 9 shows when it reconfigures the trees according to the performance specification. It is easy to see that when $101 \leq k \leq 200$, the total energy usage increases from $2\Delta e$ to $13\Delta e$. However, the error becomes much smaller; when $201 \leq k \leq 300$, the total energy usage reduces to just Δe. Although in this case the error becomes much larger, the performance specification is still satisfied.

V. Towards Maximizing Sensor Network Lifetime

We say the sensor network is functioning if there are sufficient number of sensors that can provide the estimation equality, i.e., $\overline{P} \leq P_{\text{desired}}$. We define the network lifetime as the first time that the sensor network stops functioning, i.e., after some sensors die due to running out of battery, the remaining sensors cannot provide the estimation equality.

In some applications, all sensors might be needed (or some high quality sensors are always needed) for guaranteeing the estimation quality at the fusion center. In those scenarios, although the tree configuration algorithm in the previous section minimizes the total energy consumption of the sensor nodes, it may not maximize the lifetime of the network, which is given by in this case the first time that a sensor dies due to running out of battery.

For example, consider a network that consists of two sensors (Figure 10). Assume both T_1 and T_2 in Figure 10 satisfy

$$\overline{P}(T_i) \leq P_{\text{desired}}, i = 1, 2.$$

Further assume that

$$\overline{P}(S_i) \nleq P_{\text{desired}}, i = 1, 2.$$

Let e_{ij} be the total energy cost for S_i in $T_j, i, j = 1, 2$, and let $\mathcal{E}_i$ be the initial energy for S_i. Consider the following parameters.

$$E = [e_{ij}] = \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix}, \mathcal{E}_1 = \mathcal{E}_2 = 1000.$$

Denote the lifetime of the network as L. It is easy to verify that $L = 100$ when the *Tree Reconfiguration Algorithm* is executed, as T_1 is the only tree used.

It turns out that we can increase L by mixing the use of T_1 and T_2. Let $0 \leq \alpha \leq 1$ denote the portion of times that T_1 is used, we can show that if $0 < \alpha < 1$, then $L > 100$. It is also easy to verify that L attains its maximum value at 181 when $\alpha = 0.5$.

From this example, we see that simply minimizing the total energy consumption of the sensors may not maximize the network lifetime, which is the focus of this section.

We point out in Section IV that the set of all possible trees has cardinality N^{N-2}. Thus optimal scheduling on the N^{N-2} trees is intractable when N is large. We therefore restrict our attention to a set of $M << N^{N-2}$ trees, and optimally schedule those M trees instead. It turns out that choosing a set of M trees that maximizes the lifetime is NP-complete. The complete proof is provided in Section A in the appendix. We therefore propose a tree construction algorithm that generates a set of M trees followed by a scheduling algorithm on the M trees. We show that these algorithms lead to a longer lifetime than the previous tree reconfiguration algorithm.

A. Tree Construction Algorithm

The proposed tree construction algorithm consists of three main subroutines which are the *Random Initialization Algorithm*, the *Topology Improvement Algorithm*, and the *Tree Reconfiguration Algorithm* from Section IV. The overall algorithm is presented in Figure 11.

Random Initialization Algorithm: For a given T that is rooted at S_0, define $\mathcal{S}^c(T)$ as

$$\mathcal{S}^c(T) \triangleq \{S_i : S_i \text{ is not in } T\}.$$

The intuitive idea of the *Random Initialization Algorithm* is that $\mathcal{S}_{j-hop}, j = 1, \ldots, D$, defined in Eqn (18), are randomly determined in sequence until all S_i's are included in the tree.

After the execution of the *Random Initialization Algorithm*, an initial tree of depth D is constructed with $|\mathcal{S}_{j-hop}| = n_j, j = 1, \ldots, D$, and $\sum_{j=1}^{D} n_j = N$.

Remark 5.1: If $n_1 = N$, then the algorithm returns $T^{\star}$, i.e., all sensor nodes connect to S_0 directly.

Topology Improvement Algorithm: Since the previous algorithm randomly constructs the initial tree, some sensor communication paths may be established inefficiently, i.e., some sensors use more energy yet need more hops to communicate with S_0. The *Topology Improvement Algorithm* aims to remove this inefficiency.

Algorithm 2 RANDOM INITIALIZATION

```
$D := 0$
$T := \{S_0, \emptyset\}$
$\forall j \ \mathcal{S}_{j-hop} := \emptyset$
$\mathcal{S}^c = \{S_1, \dots, S_N\}$
while $(\mathcal{S}^c \neq \emptyset)$ do
    $D := D + 1$
    Pick $n_D$ from $(1, |\mathcal{S}^c|)$ uniformly randomly.
    $l := 1$
    while $(l \leq n_D)$ do
        Pick any $S_p \in \mathcal{S}^c$ and any $S_q \in \mathcal{S}_{(D-1)-hop}$ uniformly randomly.
        Connect $S_p$ to $S_q$.
        $\mathcal{S}^c := \mathcal{S}^c \setminus \{S_p\}$
        $T := T \cup \{S_p, (S_p, S_q)\}$
        $\mathcal{S}_{D-hop} := \mathcal{S}_{D-hop} \cup \{S_p\}$
        $l := l + 1$
    end while
end while
```

When S_i is connected to S_p, we define $\tau_{i,p}$ as the number of hops between S_i and the fusion center S_0, and $e_{i,p}$ as the transmission energy cost of S_i. We further define τ_0 and e_0 for S_i in the initial tree constructed by the *Random Initialization Algorithm*.

We consider modifying the path of S_i in the initial tree, where $S_i \in \mathcal{S}_{j-hop}, j \geq 2$, only if there exists S_p in the same tree and $S_p \in \mathcal{S}_{j-hop}, j \leq \tau_0 - 1$ such that either $e_{i,p} < e_0$ or $e_{i,p} = e_0$ and $\tau_{i,p} < \tau_0$. In these cases, S_i is connected to S_p. The first condition corresponds to reducing the energy cost of S_i yet not making the hops between S_i and S_0 larger; the second condition corresponds to making the hops between S_i and S_0 smaller yet not increasing its energy cost. Define F_i as the indicator function for S_i, and $F_i = 1$ means that S_i has already been examined for possible improvement and $F_i = 0$ otherwise. The full algorithm is presented below.

Notice that F_i is set to be 1 for all $S_i \in \mathcal{S}_{j-hop}, j \leq 1$, as for those sensor nodes that are one hop away from S_0, no improvement can be made that further reduces the energy cost (and maintains the same hop

Algorithm 3 Topology Improvement

```
$\forall i \; F_i := 0$
$\forall S_i \in \mathcal{S}_{j-hop}, j \leq 1, F_i := 1$
**while** $\exists F_i = 0$ **do**
    $F_i := 1$
    $\Sigma := \{S_p : S_p \in \mathcal{S}_{j-hop}, j \leq \tau_0 - 1, e_{i,p} \leq e_0\}$
    **if** $\Sigma \neq \emptyset$ **then**
        $\tau_{i,q} := \min\{\tau_{i,p} : S_p \in \Sigma\}$
        **if** $e_{i,q} < e_0$ or ($e_{i,q} = e_0$ and $\tau_{i,q} < \tau_0$) **then**
            reconnect $S_i$ to $S_q$
            update $\mathcal{S}_{j-hop}, j \leq \tau_{i0}$
        **end if**
    **end if**
**end while**
```

numbers) or reduces the hop numbers.

At this step, we have constructed a set of M randomized initial trees. We then use them as input to the *Tree Reconfiguration Algorithm* from Section IV-A (ignoring its tree initialization algorithm subroutine) to make sure that each tree provides the desired estimation quality.

Remark 5.2: The randomized algorithm here to a certain extent guarantees that the constructed M trees will have different energy cost of the individual sensor nodes, hence through the scheduling algorithm presented in the next section, the overall lifetime of the network is maximized.

B. Tree Scheduling Algorithm

Up to now, we have constructed a set of trees $\mathcal{T}$ and for each $T_j \in \mathcal{T}$,

$$\overline{P}(T_j) \leq P_{\text{desired}}.$$

Let T_0 be the low-energy tree from the tree reconfiguration algorithm in Section IV. Denote θ as a scheduling policy on $\mathcal{T} \cup \{T_0\}$, and $t_j(\theta)$ as the time that T_j is used for the policy θ. Then the network lifetime $L(\theta)$ can be computed as

$$L(\theta) = \sum_{j=0}^{M} t_j(\theta).$$

Consider the following optimization problem:

Problem 5.3:

$$\max_{t_0,\ldots,t_M} \sum_{j=0}^{M} t_j$$

subject to

$$\sum_{j=0}^{M} t_j e_{ij} \leq \mathcal{E}_i, i = 1, \ldots, N,$$

where e_{ij} is the energy consumption for sensor S_i in tree T_j, and $\mathcal{E}_i$ is the initial energy available at S_i.

Problem 5.3 can be solved efficiently via linear programming, as both the objective function and constraints are linear functions of the variables. We also have the following result which shows that the tree construction and scheduling algorithm leads to a longer lifetime than using the tree reconfiguration algorithm.

Lemma 5.4: Let the lifetime of the network via solving Problem 5.3 be L^*, and via T_0 alone be $L(T_0)$. Then

$$L(T_0) \leq L^*,$$

with $L(T_0) = L^*$ iff $t_j^* = 0$ for all $j = 1, \ldots, M$.

Proof: Assume $L(T_0) > L^*$. Then setting $t_j^* = 0$ for all $j = 1, \ldots, M$ leads to a better solution than L^*, i.e., $L(T_0)$. This violates the optimality assumption of L^*. ■

C. Example

In this section, we provide an example to demonstrate the theory and algorithms developed so far. We start by describing the process and sensor models.

Process and Sensor Models: We consider the process in Eqn (1) with

$$A = \begin{bmatrix} 1 & 0.1 & 0.05 & 0.0002 \\ 0 & 1 & 0.1 & 0.05 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $Q = 0.1I$. There are three sensors available. The measurement equations are given by

$$\begin{aligned} y_k^1 &= [\ 1 \quad 0 \quad 0 \quad 0\ ]x_k + v_k^1, \\ y_k^2 &= [\ 0 \quad 1 \quad 0 \quad 0\ ]x_k + v_k^2, \\ y_k^3 &= [\ 1 \quad 0 \quad 1 \quad 0\ ]x_k + v_k^3, \end{aligned}$$

with $\Pi_1 = 0.5, \Pi_2 = 0.25$, and $\Pi_3 = 0.1$. Assume the sensors are placed in a line (Figure 12) with relative distance

$$d_{1,0} = 2, d_{2,1} = 1, d_{3,2} = 1,$$

where $d_{p,q}$ is the distance between S_p and S_q.

Let $e_{tx}(S_p, S_q)$ be the energy cost for S_p transmitting a packet to S_q and $e_{rx}(S_p, S_q)$ be the energy cost for S_q receiving such a packet from S_p. We use the following simplified energy model

$$e_{rx}(S_p, S_q) = 1, e_{tx}(S_p, S_q) = d_{p,q}^2, \forall 1 \le p, q \le 3, p \neq q.$$

Assume the initial energy $\mathcal{E}_i$ available at S_i is known and given by

$$\mathcal{E}_1 = \mathcal{E}_2 = \mathcal{E}_3 = 2000.$$

Let the performance specification at the fusion center be

$$\mathrm{Tr}\big(\overline{P}(T_k)\big) \le 1.75 \; \forall \; k.$$

It is easy to verify that

$$\mathrm{Tr}\big(\overline{P}(T \setminus S_2)\big) = 2.7062, \;\; \mathrm{Tr}\big(\overline{P}(T \setminus S_3)\big) = 3.1110$$

and $(A, [H_2; H_3])$ is not observable. Therefore all three sensors are needed in order to satisfy the estimation quality constraint.

Tree Construction Algorithm: Initially, we run the tree reconfiguration algorithm which returns the initial tree T_0 as seen from Figure 13. It is easy to verify that $\mathrm{Tr}\big(\overline{P}(T_0)\big) = 1.5752$ which satisfies the estimation quality constraint.

We further construct three trees, i.e., $M = 3$ here. Figure (14)–(16) demonstrate the use of the tree construction algorithm. As a result,

$$\mathcal{T} = \{T_1, T_2, T_3\}$$

is returned with $\mathrm{Tr}\big(\overline{P}(T_1)\big) = 1.6773, \mathrm{Tr}\big(\overline{P}(T_2)\big) = 1.3777, \mathrm{Tr}\big(\overline{P}(T_3)\big) = 1.5023$, and energy cost

$$E = [e_{ij}] = \begin{bmatrix} 4 & 1 & 0 & 5 \\ 10 & 11 & 9 & 1 \\ 1 & 1 & 16 & 16 \end{bmatrix},$$

where $i = 1, 2, 3$ and $j = 0, 1, 2, 3$. Notice that during the construction of T_1 to T_3, only the topology improvement algorithm modifies the input tree.

Tree Scheduling Algorithm: Let t_j be the time that T_j will be used. In order to maximize the lifetime of the network, we solve the following scheduling problem:

$$\max_{t_0,\ldots,t_3} \sum_{j=0}^{3} t_j$$

$$\text{subject to}$$

$$\sum_{j=0}^{3} t_j e_{ij} \leq 2000, i = 1, 2, 3,$$

$$t_j \geq 0, j = 0, 1, 2, 3.$$

Solving the above problem via standard LP toolbox gives the following optimal value:

$$t^* = [186 \;\; 0 \;\; 0 \;\; 131].$$

Therefore only T_0 and T_3 will be used, and the maximum network lifetime L^* is given by

$$L^* = \sum_{j=0}^{3} t_j = t_0 + t_3 = 301.$$

It is also to compute that

$$L(T_0) = 200, L(T_1) = 181, L(T_2) = 222, L(T_3) = 125.$$

Hence the network lifetime is indeed increased.

VI. Conclusions

In this paper, we consider the problem of discrete-time state estimation over a wireless sensor network. We first study the problem of optimal estimation over a sensor tree, and showed that the optimal estimator is a chain of Kalman filters and the length of the chain corresponds to the depth of the tree. Closed-form expression on the steady-state error covariance is obtained, which suggests how much each sensor contributes to the overall estimation quality. Then we present a tree reconfiguration algorithm to establish a sensor tree that has low overall sensor energy consumption and also guarantees a desired level of estimation quality. After that, we propose a tree construction and scheduling algorithm which has a longer lifetime compared with the tree reconfiguration algorithm. The idea is that a set of low energy trees with different energy cost of individual sensors are constructed, and those trees are then scheduled in a way that the network lifetime is maximized.

There are many interesting directions along the line of the current work that will be pursued in the future.

We have assumed the communication links are perfect in the current paper in the sense that data packets traveling on the links will not be dropped. However, in many cases, especially in wireless communications, packet drops are often seen, e.g., due to interference, fading, etc. We have studied the tradeoffs between measurement communication and estimate communication for a fixed sensor tree subject to random packet drops on the communication links in [13]. We will further take a look at the tradeoff between the estimation quality, the underlying graph that represents the sensor communication, the quality of the communication link, and the energy cost of the sensors. We assumed synchronization of all sensor nodes in the current work and we plan to relax this assumption in the future work. For the algorithms presented in the paper, we will give bounds on how far the solution obtained is from the global optimal solution, and also look for better algorithms. Closing the loop using the estimation algorithms developed in paper is also interesting.

Appendix

A. The Optimal Scheduling is NP-Complete

In this section we prove the following.

Problem 1: Show that finding the family which maximizes the network lifetime, among all families consisting of M trees, is NP-complete.

Before formalizing the problem of interest, we introduce some notation. Given a set $S = \{s_1, \ldots, s_N\}$ of vertices, let us denote by $\mathcal{T}$ the family of all trees having S as vertex set. For any given integer M, we denote by $\mathcal{F}_M$ the family of all subfamilies consisting of M trees, with each tree belonging to $\mathcal{T}$. Formally speaking

$$\mathcal{F}_M = \{G_i : G_i \subset \mathcal{T}, |G_i| = M\} \tag{19}$$

Let $f : 2^{\mathcal{T}} \to \mathbb{R}^+$, where $2^{\mathcal{T}}$ denotes the power set of $\mathcal{T}$. Moreover, let us denote by $(\mathcal{F}_M, f_M)$ the family $\mathcal{F}_M$ endowed with the function f_M which is obtained projecting f on $\mathcal{F}_M$, meaning restricting the domain of f to $\mathcal{F}_M$.

We now have all ingredients needed to formalize our optimization problem of interest:

Problem 2: Given $(\mathcal{F}_M, f_M)$, where $f_M(G_i)$ is computable in polynomial time for any $G_i \in \mathcal{F}_M$, find

$$\max\{f_M(G_i) : G_i \in \mathcal{F}_M\} \tag{20}$$

Before proceeding with the proof of the NP-completeness, we want to relate the formal problem (2) to our original problem (1) of interest. The correspondence is as follows.

$$\begin{aligned} S &\rightarrow \text{sensors of the network} \\ \mathcal{T} &\rightarrow \text{set of of possible trees of sensors} \\ f_M(G_i) &\rightarrow \text{LP}(G_i) \end{aligned} \tag{21}$$

where LP(G_i) is the solution of the linear programming problem applied to the family G_i of sensor trees, which is computable in polynomial time using, for example, the ellipsoid method. Using the correspondence given in Eqn (21), it is straightforward to check that problem (2) is the formalization of problem (1). We next proceed with the proof of the NP-completeness. Since NP-completeness deals with decision problems, we reformulate problem (2) as the following decision problem

$\mathbf{\Pi_{scheduling}}$ Given $(\mathcal{F}_M, f_M)$ where $f_M(G_i)$ is computable in polynomial time for any $G_i \in \mathcal{F}_M$, and a real number k, where $k \geq 0$, is

$$\{f_M(G_i) : G_i \in \mathcal{F}_M\} \geq k? \tag{22}$$

If $G_i \in \mathcal{F}_M$ is such that $f_M(G_i) \geq k$, then we say that G_i *satisfies* the decision problem $\Pi_{scheduling}$.

We briefly recall the definition of NP-completness and refer the reader to [14] for more details. We start with the following definitions

Definition 1: Let Π be a decision problem. Then Π is said the belong to the class *NP* if, given a candidate solution s for the problem Π, it is possible to verify in polynomial time that s satisfies the decision problem Π.

Definition 2: Let Π_1 and Π_2 be two decision problems. We say that Π_1 is polynomially reducable to Π_2 (notation: $\Pi_1 \leq_p \Pi_2$), whenever any instance I_1 of Π_1 can be transformed in polynomial time to an instance I_2 of Π_2 such that I_1 satisfies Π_1 if and only if I_2 satisfies Π_2.

Roughly speaking, Definition 2 says that Π_1 is a special case of Π_2. Thus, if $\Pi_1 \leq_p \Pi_2$, then there exists a polynomial time algorithm that transforms an instance for Π_1 into an instance for Π_2, that does not change the outcome.

A decision problem Π is said to be *NP-complete* if the following holds:

(a) Π is in *NP*

(b) $\Pi_1 \leq_p \Pi$ for any decision problem Π_1 in *NP*.

We first establish (a), i.e. that $\Pi_{scheduling}$ is in *NP*. Suppose that we are given a candidate solution, let us call it $G_{sol} \in \mathcal{F}_M$, for our problem. Since we can evaluate f_M on G_{sol} in polynomial time, then we can verify in polynomial time whether $f(G_{sol}) \geq K$. Thus we can verify in polynomial time whether G_{sol} satisfies $\Pi_{scheduling}$.

We next prove (b). We will show that the satisfiability problem can be reduced to $\Pi_{scheduling}$ in polynomial time. This will directly imply (b) since the satisfiability problem is well known to be NP-complete, therefore for any decision problem Π_1 in *NP*, we would have:

$$\Pi_1 \leq_p SAT \leq_p \Pi_{scheduling}, \quad \forall \Pi_1 \in NP \tag{23}$$

which clearly implies

$$\Pi_1 \leq_p \Pi_{scheduling}, \quad \forall \Pi_1 \in NP \tag{24}$$

Before proceeding further, we give the formulation of the satisfiability decision problem.

SAT: Given $(\psi, \{0,1\}^N)$, where ψ is a boolean formula consisting of n literals $x_1, x_2, \ldots, x_N$, find an assigment $\boldsymbol{y} \in \{0,1\}^N$ such that $\psi(\boldsymbol{y}) = 1$.

We next show that we can map an instance of $\mathbf{SAT}$ to an instance of $\mathbf{\Pi_{scheduling}}$ as follows.

Cayley [12] proved that the number of spanning trees of a complete simple graph with n vertices is n^{n-2}. We use the result by Prufer [15] who noticed the fact that n^{n-2} is the number of ways to write down a string of length $n-2$ from a set S of n numbers and constructed a code (called Prufer's code) that maps polynomially such strings to labeled trees in a one-to-one correspondence.

Let $s \in \{0,1\}^{(n-2)M}$ be a string, with $s = s_1 s_2 \ldots s_M$, i.e. s is obtained concatenating M strings, each having length $(n-2)$. We can associate to any string s_i its corresponding tree $T_i := \phi(s_i)$ given by the Prufer code. This gives us a family of trees of size M defined as

$$\mathcal{G}_s = \{\phi(s_1), \phi(s_2), \ldots, \phi(s_M)\} \tag{25}$$

Since the time required to construct the Prufer's code for each substring s_i, $i = 1 \ldots M$, is polynomial in the length n of the substring, it follows that the above construction is polynomial in n. The function f_M associated to the constructed family G_s would be

$$f_M(G_s) = \psi(s) \tag{26}$$

where $\psi(s)$ indicates the output of the evaluation of the boolean formula ψ on the string s. Since evaluating a boolean formula of n literals can be done polynomially, any instance s of **SAT** can be polynomially reduced to an instance $(G_s, f_M(G_s))$ of $\mathbf{\Pi_{scheduling}}$. We set the decision boundary k in $\mathbf{\Pi_{scheduling}}$ to 1.

In order to complete the proof, we need to show that a string instance s satisfies **SAT** if and only if the corresponding instance G_s satisfies $\mathbf{\Pi_{scheduling}}$. Assume first that a string instance s satisfies **SAT**. Then $\psi(s) = 1$. Since $f_M(G_s) = \psi(s)$ by construction and since the decision boundary $k = 1$, we would have that G_s satisfies $\mathbf{\Pi_{scheduling}}$. Assume now that G_s satisfies $\mathbf{\Pi_{scheduling}}$. This means that $f_M(G_s) = 1$. Since $f_M(G_s) = \psi(s)$ by construction, we would have that the boolean formula ψ in **SAT** evaluates to one on the string instance s, thus it is satisfiable.

Having proven both (a) and (b), we can conclude that $\mathbf{\Pi_{scheduling}}$ is NP-complete.

References

[1] L. Shi, K. H. Johansson, and R. M. Murray, "Change sensor topology when needed: How to efficiently use system resources in control and estimation over wireless networks." Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA, Dec 2007.

[2] N. P. M. (Ed.), *Sensor Networks and Configuration*. Springer, 2007.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME Journal of Basic Engineering*, pp. 35–45, March 1960.

[4] B. Anderson and J. Moore, *Optimal Filtering*. Prentice Hall, 1990.

[5] D. Siljak, *Large-Scale Dynamic Systems: Stability and Structure*. North-Holland, New York, 1978.

[6] M. J. Miller and N. H. Vaidya, "A mac protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228 – 242, May-June 2005.

[7] N. A. Vasanthi and S. Annadurai, "Sleep schedule for fast and efficient control of parameters in wireless sensor-actor networks." First International Conference on Communication System Software and Middleware, Jan 2006.

[8] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin, "Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2294–2309, May 2007.

[9] C.-C. Lai, C.-K. Ting, and R.-S. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications." IEEE Congress on Evolutionary Computation, Sept 2007, pp. 3531 – 3538.

[10] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004.

[11] N. P. M. (Ed.), *Sensor Networks and Configuration*. Springer, 2007.

[12] A. Cayley, "A theorem on trees," *Quart. J. Math*, vol. 23, pp. 376–378, 1889.

[13] L. Shi, K. H. Johansson, and R. M. Murray, "Estimation over wireless sensor networks: Tradeoff between communication, computation and estimation quality." The 17th International Federation of Automatic Control, Seoul, Korea, July 2008.

[14] C. Papadimitriou, *Computational Complexity*. Addison Wesley, 1994.

[15] H. Prufer, "Neuer Beweis eines Satzes uber Permutationen," *Arch. Math. Phys.*, vol. 27, pp. 742–744, 1918.

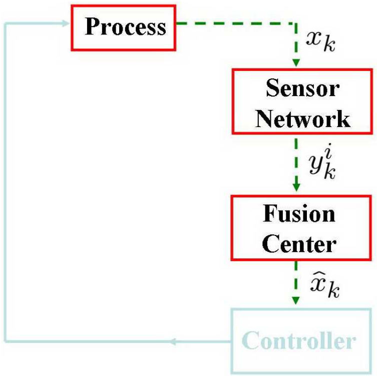

Figure 1. State Estimation Using a Wireless Sensor Network

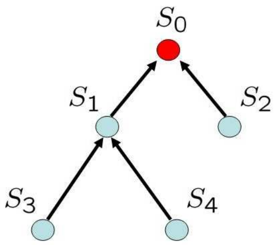

Figure 2. An Example of a Sensor Tree

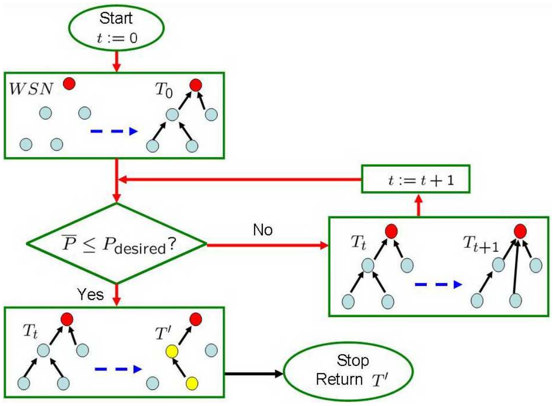

Figure 3. Tree Reconfiguration Algorithm

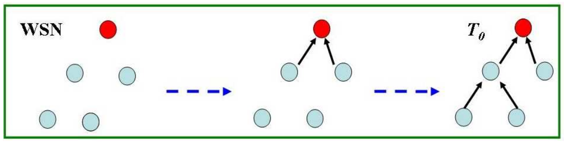

Figure 4. Tree Initialization Algorithm: Intuitive Idea

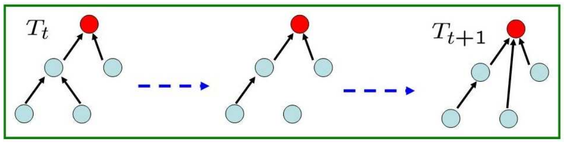

Figure 5. Switching Tree Topology Algorithm: Intuitive Idea

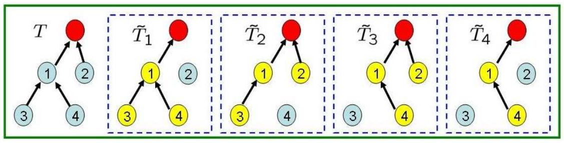

Figure 6. Tree T and Some Subtree $\tilde{T}$s

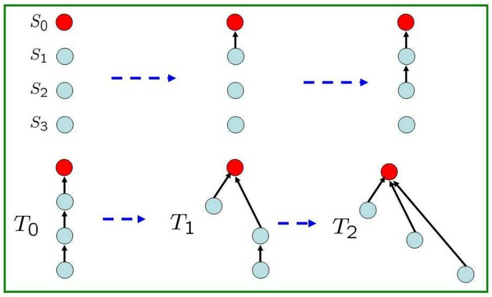

Figure 7. Different Trees Formed by the Tree Reconfiguration Algorithm

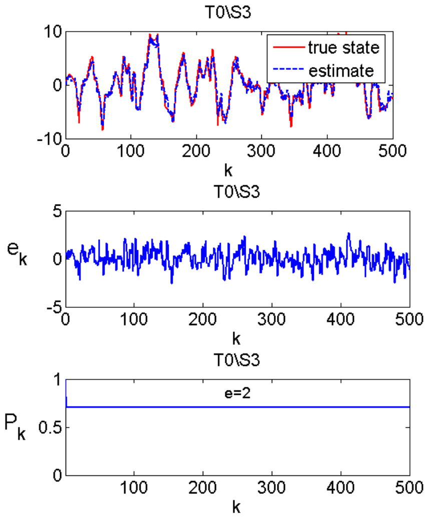

Figure 8. State and Error Evolution without Tree Reconfiguration

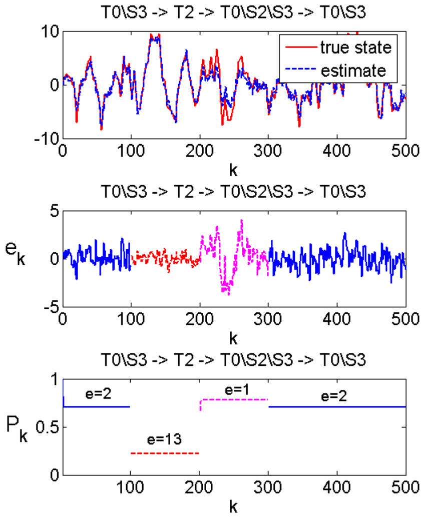

Figure 9. State and Error Evolution with Tree Reconfiguration

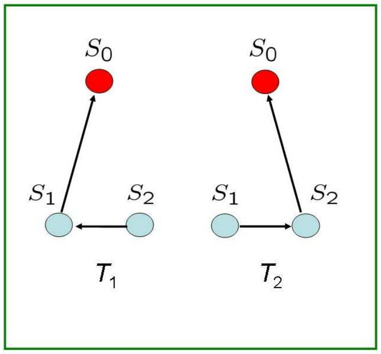

Figure 10. Network with Two Sensors

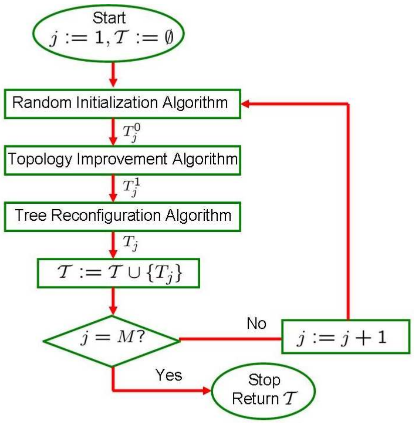
Start
$j := 1, \mathcal{T} := \emptyset$
Random Initialization Algorithm
T_j^0
Topology Improvement Algorithm
T_j^1
Tree Reconfiguration Algorithm
T_j
$\mathcal{T} := \mathcal{T} \cup \{T_j\}$
$j = M?$
No
$j := j + 1$
Yes
Stop
Return $\mathcal{T}$

Figure 11. Tree Construction Algorithm

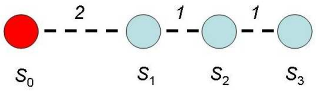

Figure 12. Initial Sensor Topology

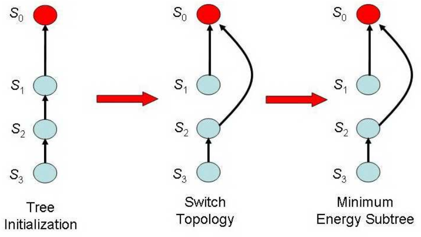

Figure 13. T_0

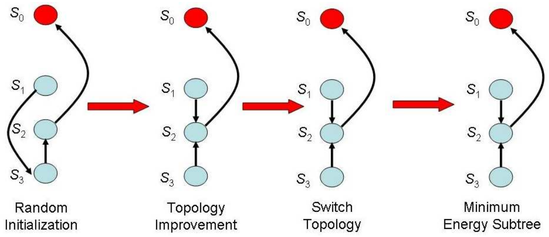

Figure 14. T_1

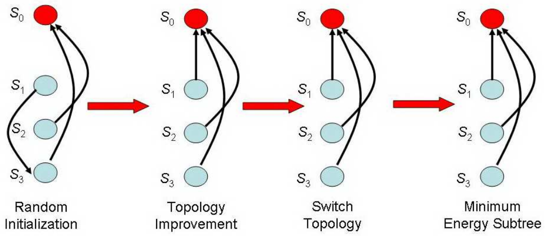

Figure 15. T_2

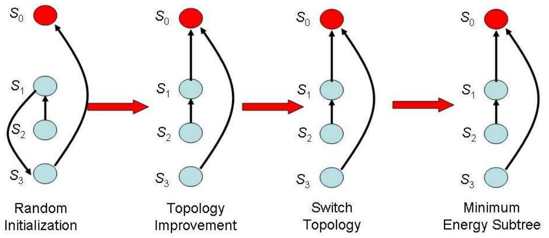

Figure 16. T_3