An Efficient Approach to Fault Identification in Urban Water Networks Using Multi-Level Sensing

Waseem Abbas
Vanderbilt University
waseem.abbas@vanderbilt.edu

Lina Sela Perelman
Massachusetts Institute of Technology
linasela@mit.edu

Saurabh Amin
Massachusetts Institute of Technology
amins@mit.edu

Xenofon Koutsoukos
Vanderbilt University
xenofon.koutsoukos@vanderbilt.edu

ABSTRACT

The objective of this work is to develop an efficient and practical sensor placement method for the failure detection and localization in water networks. We formulate the problem as the minimum test cover problem (MTC) with the objective of selecting the minimum number of sensors required to uniquely identify and localize pipe failure events. First, we summarize a single-level sensing model and discuss an efficient *fast greedy* approach for solving the MTC problem. Simulation results on benchmark test networks demonstrate the efficacy of the fast greedy algorithm. Second, we develop a multi-level sensing model that captures additional physical features of the disturbance event, such as the time lapsed between the occurrence of disturbance and its detection by the sensor. Our sensor placement approach using MTC extends to the multi-level sensing model and an improved identification performance is obtained via reduced number of sensors (in comparison to single-level sensing model). In particular, we investigate the bi-level sensing model to illustrate the efficacy of employing multi-level sensors for the identification of failure events. Finally, we suggest extensions of our approach for the deployment of heterogeneous sensors in water networks by exploring the trade-off between cost and performance (measured in terms of the identification score of pipe/link failures).

Keywords

fault localization; minimum test cover; water networks

1. INTRODUCTION

Sensor placement problems for fault detection, localization, and response arise in most engineering disciplines for secure and resilient system operations. For electric networks, placement of phasor measurement units monitoring voltage and current phasors of buses and incident branches has been suggested for the observability of electric networks under normal and critical contingency conditions such as line outages [10]. Intelligent transportation systems heavily rely on data collected by traffic sensors, where a limited number of sensors are typically placed to optimize travel times estimation for travel control [20]. Applications of sensors for water distribution has been extensively studied, primarily considering deliberate contamination of the water system maximizing detection likelihood and minimizing the volume of contaminated water [7, 8]. Several problem formulations and solution approaches have been developed and implemented to solve sensor placement problems in water, transportation, and electric systems. These approaches include integer and mixed integer programming based methods [2], submodular optimization techniques [12], greedy heuristic based solutions [11], and evolutionary algorithms [5], to name a few.

In our previous work [17], we formulated the sensors placement problem for fault location identification as a minimum test cover problem,which is known to be NP-hard [14]. MTC problem can be approximately solved by first transforming it to the well known minimum set cover (MSC) problem, and then solving MSC using greedy approach [6]. In [17] we proposed a *fast greedy* approach for solving the MTC problem that does not require a complete transformation of the MTC to the MSC, and directly computes the objective function in a greedy manner. The algorithm is faster than the straight-forward greedy approach, which makes it well suited for large-scale network applications.

In this work, we first discuss the sensor placement problem over a flow network for the purpose of identification and localization of link failures (Section 2.3). We then, evaluate our approach on benchmark test networks. Moreover, we compare our fast greedy algorithm and the straight-forward greedy algorithm in terms of the running times required to solve the MTC problem for various water networks (Section 2.4). Some of the other main features of this work include the following:

- *Single and multi-level sensing models (Section 3).* We develop a multi-level sensing model that extends our initial single-level model [17]. We generalize the *1-bit* sensors whose output consists of only a binary decision – *detected* or *not detected* failure event. The σ*-bit* sensor output is one of the multiple (more than two) possible outcomes. It means that such a sensor not only detects a disturbance once it occurs, but also encodes some of the physical features of the disturbance, such as the time lapsed between the occurrence of distur-

BuildSys'15, November 4–5, 2015, Seoul, South Korea.

ACM 978-1-4503-3981-0/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2821650.2821666.

bance and its detection by the sensor, or the intensity of the detected disturbance at the sensor. The advantage of this better sensing technology in the form of multi-level sensing is twofold: (1) The maximum number of link failures that can be detected by σ-bit sensors is greater than the 1-bit sensor. (2) For a given number of sensors, more link failures can be uniquely detected by σ-bit sensors as compared to 1-bit sensors. We illustrate these advantages through a specific case of bi-level sensing using the three water networks (Section 3.4).

- *Heterogeneous sensors (Section 4)*. We further explore inclusion of heterogeneous sensors, assuming a limited number of 1-bit and σ-bit sensors, and demonstrate that heterogeneous sensors achieve a better trade-off between the performance and the cost entailed (in terms of the numbers of 1-bit and σ-bit sensors used). Furthermore, we explore the applicability of network topology based criteria for the sensor placement for fault localization.

The outcome of our approach would be a decision support tool for the water utilities for sensor placement for failure detection and localization. Given a budget and the target performance score, the water utility can select the number, type, and location of sensors to be placed in the network. The main steps of the suggested approach involve: (1) defining the network, events, and sensing models, (2) formulating the failure identification problem as the minimum test cover problem, (3) solving the MTC problem, and (4) evaluating the performance based on multiple criteria.

2. PROBLEM FORMULATION

There are several technologies for active pipe burst detection, which can be categorized into surface, inline, and online technologies. Surface and inline technologies include acoustic, thermography, umbilical tools, and advanced autonomous robots, which are principally used to verify and pinpoint the location of the burst. Their operation is typically time consuming and costly, and they are also not suitable for continuous operation [19]. In this work we consider the use of online sensors continuously monitoring the system state, particularly hydraulic pressure sensors for analyzing the transient flow regime. Pipe burst causes rapid change in the flow, which moves through the system as a pressure wave with very high velocity known as water surge [18]. We consider the use of real-time event detection hardware-software system consisting of sensors continuously measuring water pressure and analyzing the transient flow regime for detecting the events.

In a previous work [17], we developed network dynamics in response to pipe failure events, formulated a sensor placement problem for detection and localization as minimum set and minimum test cover problems respectively, solved the minimum test cover problem using transformed greedy approach, and suggested a fast greedy solution approach. To provide background for our models in Sections 3 and 4, we summarize the sensor placement problem and discuss the fast greedy solution approach.

2.1 System model

Given a set of sensors $\mathcal{S}$ and a set of events $\mathcal{L}$, the sensor placement problem is to find a subset of sensor locations $S \subseteq \mathcal{S}$, such that the sensor network performance function, which is the location identification of pipe failure events, is maximized while using the minimum number of sensors. Consider a set of n pipe failure events, $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$, where ℓ_j denotes the location of the event and a set of m sensors that can be placed at the nodes of the network, $\mathcal{S} = \{S_1, \ldots, S_m\}$, where S_i denotes the location of the i^{th} sensor. Let $p(t)$ be a vector of the sensed pressures in the system, and $y_{S_i}(t, \ell)$ be the state (output) of the S_i sensor at time t. Assume $\xi : p(\cdot) \longrightarrow \mathbb{R}$ be some function characterizing the distance between the expected and the measured data, then we define $y_S(t, \ell) \in \{0, 1\}$ to be a discrete sensor state vector – 0 representing a normal operating state and 1 representing otherwise. The sensor outcome can then be formulated as:

$$y_{S_i}(t, \ell) = \{1 \mid \xi\left(p_{S_i}(\tau_\ell)\right) \geq \varepsilon\} \tag{1}$$

where ε is a threshold value and τ_ℓ is the time taken by the sensor to detect the event ℓ. An example of this simple sensing model would be case in which sensor S_i indicates occurrence of the event whenever change in pressure is above some threshold value ε.

Single-level sensing: First we introduce the single-level sensing model, in which during a given time horizon (much longer than the characteristic time of water surge) the sensor either detects the event or not. We achieve this by allowing sensor S_i to detect event ℓ in a certain finite time T, thus, neglecting time dependency of the sensor to detect the event. Hence, we can restate the output of sensor as:

$$\mathbf{y}_{S_i}(\ell) = \begin{cases} 1 & \text{if } y_{S_i}(t, \ell) = 1, \text{ for some } t \in [0 \quad T] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Consequently, for a sensor set $\mathcal{S}$ and the set of events $\mathcal{L}$, we can instantiate a boolean matrix of dimensions $|\mathcal{L}| \times |\mathcal{S}|$ called as *influence matrix* and denoted by $\mathcal{M}$. The rows of the matrix are sensor outputs, $\mathbf{y}_{\mathcal{S}}(\ell), \ell \in \mathcal{L}$. Moreover, $\mathcal{M}_{ij} = 1$ indicates that sensor S_i detects a failure in the link ℓ_j, and $\mathcal{M}_{ij} = 0$ implies otherwise. Thus,

$$\mathcal{M}\left(\mathcal{L}, \mathcal{S}\right) = \begin{bmatrix} \mathbf{y}_{\mathcal{S}}(\ell_1) \\ \mathbf{y}_{\mathcal{S}}(\ell_2) \\ \vdots \\ \mathbf{y}_{\mathcal{S}}(\ell_n) \end{bmatrix}. \tag{3}$$

For the set of link failures $\mathcal{L}$, and the set of all available sensors $\mathcal{S}$, we define a collection of subsets of links, denoted by $\mathcal{C} = \{C_i : \forall i\}$, where $C_i = \{\ell_j \in \mathcal{L} | \; \mathbf{y}_{S_i}(\ell_j) = 1\}$, for each sensor S_i. In other words, $C_i \subseteq \mathcal{L}$ is the set of link failure events detected by the sensor S_i and $\mathcal{C}_S$ is a collection of C_i's corresponding to the set of sensors in $S \subseteq \mathcal{S}$.

Example 1a: Consider a small network having 8 nodes connected by 10 links as shown the Figure 1. A pipe burst event is simulated in the middle of pipe ℓ_1 and system response at network nodes is recorded. For the ease of notations, we designate the failure events as pipes' ids, ℓ_j. The transient simulations were computed using the HAMMER software [1].

The first plot in Figure 2 shows the pressure head, $p(t)$, generated at nodes 2 and 5 in response to the pipe failure ℓ_1. A sharp pressure drop is observed at both nodes. Since the disturbance arrives first at node 2, the pressure drop at node 2 is larger and occurs before the pressure drop at node 5. The second plot in Figure 2 shows the sensors' states

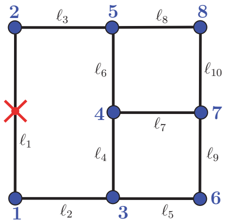


Figure 1: Illustrative example layout.

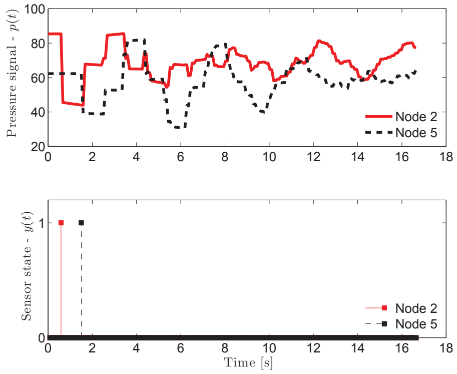


Figure 2: Network dynamics for failure event generated in pipe ℓ_1 in the small example – $p(t)$ pressure head $[m]$ and $y_{S_2,S_5}(t,\ell_1)$ – sensors S_2 and S_5 output in response to event ℓ_1.

indicating detection of the event at node 2 and node 5 approximately $0.5[sec]$ and $1.5[sec]$ respectively after the pipe burst. Based on the single-level model, both sensors located at nodes 2 and 5, can detect the failure in pipe ℓ_1. Thus, for $S_A = \{S_2, S_5\}$, the sensor state is $\mathbf{y}_{S_A}(\ell_1) = [1,1]$. If sensor are placed at all the nodes of the system, then the failure would be detected by sensors located at nodes $\{1,2,3,5\}$, and the sensor state would be $\mathbf{y}_{\mathcal{S}}(\ell_1) = [1,1,1,0,1,0,0,0]$. Next, if we consider a single failure event in each pipe of the network with $\mathcal{L} = \ell_1, \cdots, \ell_{10}\}$, then the corresponding influence matrix is:

$$\mathcal{M}(\mathcal{L},\mathcal{S}) = \begin{array}{c} \\ \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \\ \ell_5 \\ \ell_6 \\ \ell_7 \\ \ell_8 \\ \ell_9 \\ \ell_{10} \end{array} \begin{array}{c} \begin{array}{cccccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \qquad (4)$$

2.2 Identification as the minimum test cover

For a set of events $\mathcal{L}$ and a set of sensors $\mathcal{S}$, the objective of the identification problem is to maximize the number of uniquely identified link failures through the minimum number of sensors. In other words, based on unique outputs of sensors, we improve the localization of the event. One way to achieve that is by selecting a set of sensors S, such that for every pair of events ℓ_i and ℓ_j, sensors' states $\mathbf{y}_S(\ell_i)$ and $\mathbf{y}_S(\ell_j)$ differ by at least one element, i.e. sensors' state vector $\mathbf{y}_S(\ell_i)$ is unique for every event $\ell_i \in \mathcal{L}$. In terms of the influence matrix of the network, this is possible whenever for every row pair ℓ_i, ℓ_j, $i \neq j$, there exists a column C_u with different i and j row entries (i.e., $\mathcal{M}_{iu} \neq \mathcal{M}_{ju}$). The objective of the identification problem, denoted by $f_I(S;\mathcal{L})$, is then to find the minimum number of sensors that can distinguish between the maximum number of *pair-wise* events.

This problem can be formulated as the *minimum test cover* (MTC) problem. The MTC problem is a combinatorial optimization problem in which the objective is to select the minimum number of tests from a collection of tests such that every event involved can be uniquely classified in one of the given categories based on the outcomes of selected tests [14]. In our sensor placement problem for water distribution networks, set of outcomes of tests are sensors' states, events are pipe bursts, and classification categories are locations of the failed pipes.

Definition 1. (Minimum test cover (MTC)) Given a finite set $\mathcal{L}$ and a collection of subsets $\mathcal{C} = \{C_i : C_i \subseteq \mathcal{L}\}$. The minimum test cover problem is to find $\mathcal{C}_t \subseteq \mathcal{C}$ with the minimum cardinality such that if for a pair of elements $\{\ell_u, \ell_v\} \in \mathcal{L}$ there exists $C_i \in \mathcal{C}$ that contains exactly one of ℓ_u and ℓ_v, then there exists some $C_j \in \mathcal{C}_t$ that also contains exactly one of ℓ_u and ℓ_v.

From the above definition and discussion, we can state the following [17]:

PROPOSITION 1. *Maximizing the identification of link failures in networks is equivalent to solving the minimum test cover problem.*

Example 1b Following example 1a, consider two sensors placed at nodes 2 and 4, $S_A = \{S_2, S_4\}$. We observe that although sensors S_2 and S_4 *cover* all link failures with $C_2 \cup C_4 = \mathcal{L}$, they are not sufficient for identification. Sensors S_2 and S_4 generate only three unique states for the 10 events, which makes it impossible to distinguish between all link failures. For instance, given sensors' outputs, i.e., $\mathbf{y}_{S_A}(\ell_1) = [1,0]$, we can uniquely identify or localize the failure on link ℓ_1 (see (4)). Given the output $[1,1]$, we can not uniquely localize the failure, but identify a set of links ℓ_2, ℓ_3, ℓ_6, or ℓ_8, that are associated with this output of sensors. We later define this set of links as the *localization set.*

However, for the set of sensors $S_B = \{S_1, S_2, S_3, S_5\}$, which solve the MTC problem in example 1a, the output is unique for each link failure and 10 distinct indicator vectors corresponding to each failure event are obtained.

2.3 Fast greedy MTC solution

The minimum test cover problem is NP-hard [6]. One approach to solve the MTC problem is by first transforming it to the minimum set cover problem (MSC) [3]. Given an instance of MTC, i.e., $\mathcal{L}$ and $\mathcal{C}$, we obtain a new matrix $\mathcal{M}^t(\mathcal{L}^t, \mathcal{S})$ of dimensions $\binom{n}{2} \times m$ in which each row corresponds to a *pairwise link failure* and each column represents sensor's output. The row element is equal to 1 if the corresponding column sensor distinguishes ℓ_i from ℓ_j by producing different outputs for ℓ_i and ℓ_j respectively, i.e., if the sensor output is 1 in case of ℓ_i and 0 for another event ℓ_j, and vice versa. Next, the MSC can be efficiently solved and a solution of the original problem can be directly obtained from the solution of the MSC problem.

The MSC has been extensively studied in the context of sensor placement for detectability. The detection problem can be posed as the MSC problem where the objective is to maximize the detection of failure events with the minimum number of sensors. A straight-forward way to solve the MSC is by the *greedy approach*. In each iteration, a sensor that detects the maximum number of uncovered link failures is selected. This continues until all link failures are covered, or no further link failure can be detected by any sensor. The MSC is submodular, i.e. the objective function depicts the diminishing return behavior. The submodularity property can be exploited to further improve the efficiency of the greedy solution approach by getting rid of the redundant computations in each iteration [13]. The greedy approach to solve MTC yields the best possible $(1 + 2 \ln n)$ approximation ratio, where n is the total number of link failures. Details on the transformation of the MTC problem to the MSC problem can be found in [3] and accelerated greedy MSC solution in [13, 12].

However, if n failure events need to be localized, the corresponding set cover instance contains $\binom{n}{2}$ events, and the time complexity of the greedy approach is $\mathcal{O}\left(m\binom{n}{2}\right)$, where m is the total number of sensors. Even for small networks with a small number of link failures, this approach becomes extremely inefficient owing to a large number of computations required. Moreover, employing accelerated greedy also does not solve the problem due to a large number of events in the event space. Next, we propose a fast greedy approach to improve the time complexity of the identification solution.

2.3.1 Main algorithm

The main idea of the *fast greedy* approach is to avoid the complete transformation of the MTC to the MSC and reducing the number of comparisons made in each step. Instead of $\mathcal{O}\left(m\binom{n}{2}\right)$ comparisons, in our algorithm, the number of comparisons made for each sensor in a single iteration is always bounded by $\mathcal{O}\left(K\binom{k}{2}\right)$, where k is the maximum number of link failures detected by any sensor, and K is the number of sensors that are included in the test cover until that iteration. Since k is typically much smaller than n, a large number of computations are thus avoided in each iteration.

Before explaining our algorithm, we note that:

- A sensor i that detects k link failures, detects $k(n-k)$ pairwise link failures, since a sensor can distinguish between k detected events and $(n-k)$ undetected events. Unlike the detection problem in which a sensor with a large k is desirable for the detection purposes, a sensor that detects a large number of failures is not always useful for the identification.

- A distinction between the occurrence of events ℓ_u and ℓ_v is not possible through the sensor i whenever both $\ell_u, \ell_v \in C_i$, or $\ell_u, \ell_v \notin C_i$. Thus, if a sensor i is selected in the test cover, then sensor(s) that can distinguish between events $\ell_u, \ell_v \in C_i$ also need to be included in the test cover.

Based on these observations, two components contribute to the overall utility of adding a sensor S_i to the test cover: (i) the number of pairwise events corresponding to the links that are not already covered by existing sensors (i.e. sensors selected in previous iterations) that can be detected by the sensor S_i and (ii) the number of undetected pairwise events by existing sensors that can be detected by sensor S_i. We denote the overall utility by w_i and the individual contributions, (i) and (ii), as x_i and y_i.

Consider $\mathcal{L}$ the set of link failures, $\mathcal{S}$ the set of available sensors, $C_u \subseteq \mathcal{L}$ the set of link failures detected by the sensor S_u, and $\mathcal{C}$ the collection of all such C_u's. We define $C^* \subseteq \mathcal{C}$ to be the test cover under construction, and $\mathcal{C}_{cov}$ be the set of link failures detected by the sensors that are included in the test cover under the construction, i.e., $\mathcal{C}_{cov} = \bigcup_{C_u \in C^*} C_u$. Then the utility of adding sensor S_i to the test cover, i.e. adding C_i to C^* is computed as the sum of x_i and y_i. In each iteration j, we compute x_i and y_i as:

- x_i – if n_j is the number of link failures that are not yet included in $\mathcal{C}_{cov}$, (i.e., $n_j = n - |\mathcal{C}_{cov}|$), and C_i contains k_i of such link failures, then $x_i = k_i(n_j - k_i)$.

- y_i – if a sensor S_u is already included in the test cover, then the pairwise link failures corresponding to the links in C_u remain undetected. Thus, y_i computes how many of such pair-wise link failures can be detected by the inclusion of sensor S_i in the test cover.

A sensor S_{i^*} that maximizes this overall utility w_{i^*}, will then be included in the test cover, and $\mathcal{C}_{cov}$ will be updated to $\mathcal{C}_{cov} \leftarrow \mathcal{C}_{cov} \cup C_{i^*}$.

The steps of the fast greedy algorithm for the solution of the MTC are given in Algorithm 1. Additionally, to support Algorithm 1, we define:

$$\beta(X) = \text{set of all 2-element subsets of } X,$$

and

$$\alpha(Y, \beta(X)) = \{a \in \beta(X) : \ |Y \cap a| = 1\},$$

where X and Y are two sets,$\alpha(Y, \beta(X))$ is a set consisting of such 2-element subsets of X that have exactly one common element with Y. For instance, if $X = \{1, 2, 3\}$ and $Y = \{1, 3\}$, then $\beta(X) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, and $\alpha(Y, \beta(X)) = \{\{1, 2\}, \{2, 3\}\}$.

To compute y_i, first we compute the set of common link failures in C_i and $\mathcal{C}_{cov}$ and call it as $Y_i = C_i \cap \mathcal{C}_{cov}$. Now, if S_u is already in the test cover, and $G_u \subseteq \beta(X_u)$ is the set of undetected pair-wise link failures corresponding to the links in $X_u \subseteq C_u$, then

$$y_i = \sum_{C_u \in C^*} |\alpha(Y_i, G_u)|$$

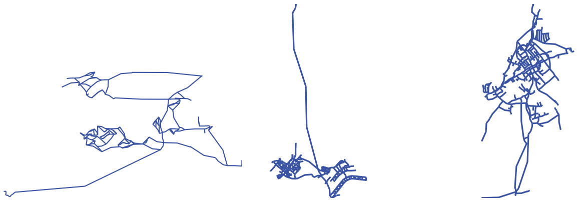
Algorithm 1 is basically an efficient implementation of the greedy approach for the MTC problem, and has the same approximation ratio as the standard greedy algorithm.

2.4 Applications

In all our simulations we: (1) consider a single failure event occurring at the center of each pipe; (2) enumerate all possible failure events in each network; (3) consider the shortest distance threshold model, $\mathbf{y}_{S_i}(\ell) = \{1 \mid d(S_i, \ell) \leq \varepsilon\}$ [4] with $\varepsilon = 1[km]$. We assume that the disturbance in pressure can be sensed while within a specified distance from the location of the burst, where d is the length of the shortest path between two locations S_i and ℓ and ε is the detectable travel distance; and (4) solve the identification problem for the single-level sensing model using the transformed greedy (TG) and the fast greedy (FG) approaches.

Algorithm 1 Minimum Test Cover - Fast Greedy

```
Input: $\mathcal{C} = \{C_1, \cdots, C_m\},\ C_i \subseteq \mathcal{L}$
Output: MTC: $\mathcal{C}^* \subseteq \mathcal{C}$
Initialization: $\mathcal{C}_{cov} = \emptyset;\ \mathcal{C}^* = \emptyset;\ G_0 = \emptyset;\ j = 1;\ w_{i^*} = 1;$
while $w_{i^*} > 0$ do
    $n_j \leftarrow n - |\mathcal{C}_{cov}|$
    for all $i$ do
        $X_i \leftarrow (C_i \setminus \mathcal{C}_{cov})\, ; k_i \leftarrow |X_i|$
        $x_i \leftarrow k_i(n_j - k_i)$
        $Y_i \leftarrow C_i \cap \mathcal{C}_{cov}$
        $y_i \leftarrow \sum_{t=0}^{j-1} |\alpha(Y_i, G_t)|$
        $w_i = x_i + y_i$
    end for
    $w_{i^*} \leftarrow \max w_i$
    if $w_{i^*} > 0$ then
        $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{C_{i^*}\}$
        $\mathcal{C}_{cov} \leftarrow \mathcal{C}_{cov} \cup C_{i^*}$
        $G_j \leftarrow \beta(X_{i^*})$
        for $t = 0$ to $j - 1$ do
            $G_t \leftarrow G_t \setminus \alpha(Y_{i^*}, G_t)$
        end for
        $j \leftarrow j + 1$
    end if
end while
```



(a) Net 1 (b) Net 2 (c) Net 3

Figure 3: Layouts of networks

Table 1: Networks data and simulation results

Network	No. of pipes	No. of nodes	No. of sensors	TG [sec]	FG [sec]
Net1	168	126	48	14.03	4.94
Net2	366	269	98	143.69	34.77
Net3	496	420	134	415.83	98.76

TG - transformed greedy; FG - fast greedy;

We evaluate the performance of the sensor placement based on four scores:

1. *Identification* – is the number of uniquely identified pairs of events by a given set of sensors S. The identification score is the objective function of our optimization problem. The normalized identification score, $I_I(S; \mathcal{L})$, is the ratio of the detected pair-wise events to the total number of pair-wise events.

2. *Detection* – is the number of events that are detected by at least one sensor in the given set of sensors S. The normalized detection score, denoted by $I_D(S; \mathcal{L})$, is then the ratio of the total number of detected events given a sensor set S to the total number of event failures $|\mathcal{L}|$.

3. *Localization* – is defined as the number of localization sets, where each localization set is associated with a unique sensors' state. Let $L \subseteq \mathcal{L}$ be a localization set under the sensor configuration S if for all $\ell_i \in L$, the outputs of sensors in S remain the same. In other words, it is not possible to distinguish between failure events in a localization set based on sensors' outputs. The normalized localization score, denoted by $I_L(S; \mathcal{L})$, is the ratio of the number of unique localization sets formed under the sensor configuration S to the total number of event failures.

Ideally, all three normalized performance scores should be equal to 1, indicating that each fault can be detected and uniquely identified. We apply our approach on three real water systems [9, 15]. Net1 is a benchmark system that has been previously extensively studied in the context of sensor placement for water quality [15]. The system consists of 168 pipes, 126 nodes, one reservoir, one pump, and two storage tanks and its layout is shown in Figure 3a. The system supplies a daily demand of $5.15 \times 10^3 [m^3/day]$ and has a total pipe length of $37.5[km]$. Net2 and Net3 were originally collected by the Kentucky Infrastructure Authority. The systems supply 7.6 and $8.58 \times 10^3 [m^3/day]$ daily demand of and have a total pipe length of 91.3 and $96.6[km]$, respectively. Network layout are shown in Figures 3b -3c and main features are summarized in Table 1. Full data and hydraulic model can be obtained from [9].

We first formulate the identification problem as the MTC (Section 2.2) based on the single-level sensing model. We then solve the optimization problem using the transformed greedy (TG) and the fast greedy (FG) approaches (Section 2.3), and compare solutions and running times of the two approaches. The solutions based on TG and FG approaches were identical, as suggested in Section 2.3. Figure 4 shows the normalized identification score $I_I(S; \mathcal{L})$ as a function of the number of sensors for the three networks. The identification score function exhibits the submodular property, i.e. diminishing return behavior. The number of sensors required for the maximum identification, and running times $[sec]$ required for each network are listed in Table 1. The results indicate that, as expected for networks with larger pipe length and failure events, the required number of sensors is larger as well. Additionally, the FG approach was found to be 2.8 to 4.2 times faster than the TG approach for the given networks and require significantly less memory.

Next, for each network we evaluate the detection and localization scores as a function of the number of sensors for each solution of the identification problem. We observe that the detection score (Figure 5a) behaves similarly to the identification score (Figure 4), although not truly submodular.

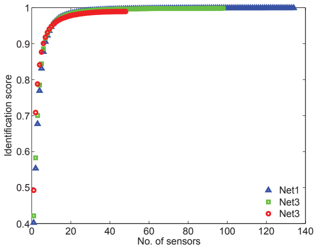
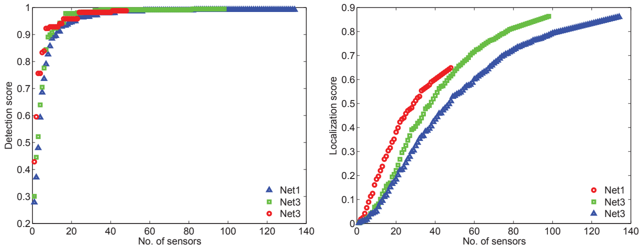


Figure 4: Identification score as a function of the number of sensors for – Net1, Net2, and Net3



(a) Detection

(b) Localization

Figure 5: Detection and localization evaluation as a function of the number of sensors for – Net1, Net2, and Net3

The identification and detection scores are improved to a greater extent with the first few sensors $|S| < 20$, and only minor improvements are seen when more sensors are added. The localization score (Figure 5b) exhibits a more graduate improvement with each added sensor and does not reach the theoretical maximum of 1, even when more sensors are added. This can be primarily attributed to the single-level sensing model that captures only a single-level of the event dynamics. Additional factors that affect this behavior are network structure and the dynamics of the failure event.

3. HIGH RESOLUTION SENSORS

Previously, we have considered single-level sensing model, i.e. *1-bit* sensors whose output consisted of only one binary decision – failure event is *detected* and failure event is *not detected*. However, a more general scenario would be to have the sensor output consisting of more than two possible outcomes, in which case multiple output bits would be needed. For instance, the time stamp and intensity of the detected event might also be encoded in the sensor output. We define this as *multi-level* or σ*-bit* sensing model.

3.1 Multi-level sensing model

In a simplistic scenario, we assume that a sensor, if able to detect, can only detect a failure event within the T time duration of the occurrence of the event. This is a valid assumption as the signal (such as pressure wave) generated as a result of the failure event typically decays with time as it travels more distance within the network. Thus, the output of the sensor can be made of *higher resolution* by dividing the interval $[0 \;\; T]$ into further σ sub-intervals such as $[0 \;\; t_1), [t_1 \;\; t_2), \cdots, [t_{\sigma-1} \;\; T]$, where $0 < t_1 < t_2 < \cdots, t_{\sigma-1} < T$. The time taken by the sensor to detect an event would then lie in one of these intervals. Thus, there are $(\sigma+1)$ possible outcomes of a sensor including the outcome corresponding to the no detection of the event. Note that at least $\lceil \log_2(\sigma+1) \rceil$ bits would be needed to encode the sensor output in this case. For instance, if the interval $[0 \;\; T]$ is divided into two sub-intervals $[0 \;\; t_1)$ and $[t_1 \;\; T]$, where $t_1 < T$, then we get the following possible sensor outputs:

1. Event is not detected at all, corresponding to the two bit output of $(0 \;\; 0)$,

2. Event is detected *early*, i.e., within a certain time threshold $[0 \;\; t_1)$ after its occurrence, represented by the two-bit output of $(1 \;\; 0)$,

3. Event is detected *later*, i.e., within the interval $[t_1 \;\; T]$, represented by the output $(0 \;\; 1)$.

Now, exploiting a direct relationship between the time taken by the signal (such as pressure wave) to reach the sensor and the distance traveled by the signal from the point of origin to the sensor, we can develop a *high resolution* version of the *single-level* sensing model discussed previously. In the single-level model, a 1-bit sensor S_i could *detect* a failure on a link ℓ whenever $\xi(p_{S_i}(t)) \leq \varepsilon$, i.e.,

$$y_{S_i}(t,\ell) = \{1 \mid \; \xi(p_{S_i}(t)) \leq \varepsilon \text{ for some } t \in [0 \;\; T]\}\,.$$

Now, let τ_ℓ be the time taken by the sensor to detect the event ℓ after its occurrence, then τ_ℓ lies in one of the sub-intervals of $[0 \;\; T]$. Note that τ_ℓ depends on the event ℓ. Based on the sub-interval in which τ_ℓ lies in, a $2-bit$ output of the sensor can be obtained as follows:

$$\mathbf{y}_{S_i}(\ell) = \left\{ \begin{array}{ll} (1\,0) & \text{if } y_{S_i}(\tau_\ell,\ell) = 1, \text{ for some } \tau_\ell \in [0 \;\; t_1) \\ (0\,1) & \text{if } y_{S_i}(\tau_\ell,\ell) = 1, \text{ for some } \tau_\ell \in [t_1 \;\; T] \\ (0\,0) & \text{otherwise.} \end{array} \right. \tag{5}$$

The model can be extended to σ-bit output in a similar way by dividing the interval $[0 \;\; T]$ into σ further sub-intervals. As previously, if $\mathbf{y}_{\mathcal{S}}(\ell) = [\mathbf{y}_{S_1}(\ell), \cdots, \mathbf{y}_{S_m}(\ell)]$ is a boolean vector with $2m$ entries corresponding to the outputs of m sensors in response to the failure event ℓ, then a *2-bit influence matrix*, denoted by $\breve{\mathcal{M}}(\mathcal{L},\mathcal{S})$, could be defined as,

$$\breve{\mathcal{M}}(\mathcal{L},\mathcal{S}) = \begin{bmatrix} \mathbf{y}_{\mathcal{S}}(\ell_1) \\ \mathbf{y}_{\mathcal{S}}(\ell_2) \\ \vdots \\ \mathbf{y}_{\mathcal{S}}(\ell_n) \end{bmatrix} \tag{6}$$

Example 1c: Continuing with the example network in Section 2.2, we illustrate the 2-bit model. A sensor can either detect a failure event within $[0 \; 1.5][sec]$, or not at all. This interval is further divided into two sub-intervals, $[0 \;\; 0.75)$ and $[0.75 \;\; 1.5][sec]$. The detection of failure event within $t \in [0 \;\; 0.75)[sec]$ of its occurrence results in an output of $(1 \quad 0)$, whereas, detection within $t \in [0.75 \;\; 1.5][sec]$ renders the output $(0 \quad 1)$. For instance, the failure in ℓ_1 is detected by sensors S_1, S_2 within $0.5[sec]$, and by sensors S_3, S_5 within $1.5[sec]$. Thus, for $S_A = \{1,2,3,5\}$, we get $\mathbf{y}_{S_A} = [1,0,1,0,0,1,0,1]$. The 2-bit influence matrix $\breve{\mathcal{M}}(\mathcal{L},\mathcal{S})$ for the single failure events in the network is,

$$\begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0
\end{pmatrix}$$

where each 2-bit sensor output is represented by a pair of columns.

3.2 Identification through multi-level sensing

The objective of identification of link failures through σ-bit sensors remains the same as with the 1-bit sensors, i.e. *maximizing the identification of link failures using the minimum number of sensors.* The basic idea is same – select the minimum number of sensors to maximize the *detection of pair-wise* link failures. However, as a result of multiple output bits, more pairwise link failures can now be detected by the multiple output bits sensor as compared to the 1-bit sensor as explained below. This leads to the lesser number of sensors required to achieve the same level of identification in the network. For instance, in the above example, in comparison to the four sensors required in the case of 1-bit sensors for complete identification, three sensors $\{S_2, S_3, S_4\}$ are sufficient when 2-bit sensors are used.

3.2.1 1-bit vs. σ-bit sensors

If the number of events detected by a 1-bit sensor is k, then the number of pair-wise link failures detected by the sensor, denoted by $\mathcal{P}_1$, is

$$\mathcal{P}_1 = k(n-k) \tag{7}$$

where n is the total number of events. 1-bit sensor can be replaced by a σ-bit sensor, in which the threshold of detection is divided into σ further levels, i.e., $[0\ \ T]$ interval is further divided into σ sub-intervals. Each of the σ output bits corresponds to one of the sub-intervals and is 1 whenever the event is detected within the sub-interval, and 0 otherwise. Let k_i be the number of events detected by the i^{th} output bit in the σ-bit sensor, i.e., the number of events detected by the sensor in the i^{th} sub-interval of $[0\ \ T]$. Note that the total number of link failures detected by the 1-bit sensor and the corresponding σ-bit sensor described here are same, i.e, $\sum_{i=1}^{\sigma} k_i = k$. However, the number of pair-wise link failures detected by the σ-bit sensor, denoted by $\mathcal{P}_\sigma$, are

$$\mathcal{P}_\sigma = \sum_{i=1}^{\sigma} \left(k_i(n-k_i)\right) - \sum_{x=1}^{\sigma-1}\sum_{y>x} (k_x k_y) \tag{8}$$

which is the number of pair-wise link failures detected by each i^{th}bit minus the number of pair-wise link failures mutually detected by the pairs of output bits.

To compare pair-wise link failures detected by the 1-bit and the corresponding σ-bit sensors, we use (7) and (8).

$$\begin{aligned}\mathcal{P}_1 = k(n-k) &= \left(\sum_{i=1}^{\sigma} k_i\right)\left(n - \sum_{i=1}^{\sigma} k_i\right) \\ &= \left(\sum_{i=1}^{\sigma} k_i(n-k_i)\right) - 2\left(\sum_{x=1}^{\sigma-1}\sum_{y>x}^{\sigma} k_x k_y\right) \qquad (9) \\ &= \mathcal{P}_\sigma - \left(\sum_{x=1}^{\sigma-1}\sum_{y>x}^{\sigma} k_x k_y\right)\end{aligned}$$

Thus, the σ-bit sensor detects $\sum_{x=1}^{\sigma-1}\sum_{y>x}^{\sigma} k_x k_y$ more pair-wise link failures as compared to the corresponding 1-bit sensor. Moreover, depending on the values of k_i, $\forall i \in \{1, \cdots, \sigma\}$, such that $\sum_{i=1}^{\sigma} k_i = k$, we get the following,

$$\mathcal{P}_1 + (\sigma - 1)\left(k - \frac{\sigma}{2}\right) \ \le\ \mathcal{P}_\sigma \ \le\ \mathcal{P}_1 + \left(\frac{k^2(\sigma-1)}{2\sigma}\right) \tag{10}$$

Note that the lower bound is achieved when $k_i = 1$ for all i but one, for which the value is $k-\sigma+1$. Similarly, the upper bound is achieved when $k_i = \frac{k}{\sigma}$ for all $i \in \{1, \cdots, \sigma\}$. For a 2-bit sensor, in which k_1 and k_2 are the number of detected events by the first and second bits respectively, the number of pair-wise link failures detected is

$$\mathcal{P}_2 = k_1(n-k_1) + k_2(n-k_2) - k_1 k_2 \tag{11}$$

Comparing it to the pair-wise link failures detected by the corresponding 1-bit sensor in which $k = k_1 + k_2$ is the number of detected link failures, we get

$$\mathcal{P}_1 + (k-1) \ \le\ \mathcal{P}_2 \ \le\ \left(\mathcal{P}_1 + \frac{k^2}{4}\right) \tag{12}$$

3.3 Solving the multi-level MTC problem

To solve identification problem through σ-bit sensors, we can modify the *transformed greedy* algorithm for the test cover problem. The algorithm consists of two main steps – transforming the test cover to the equivalent set cover problem and solving the set cover problem using greedy approach.

Given a σ-bit influence matrix $\breve{\mathcal{M}}(\mathcal{L}, \mathcal{S})$ of dimension $n \times \sigma m$, in which rows correspond to the n events in $\mathcal{L}$ and columns correspond to the m number of σ-bit sensors in $\mathcal{S}$. Note that the consecutive columns indexed by $\{(\sigma u + 1), (\sigma u + 2), \cdots, (\sigma u + \sigma)\}$ represent the output of sensor S_{u+1}, where $u \in \{0, 1, \cdots, m-1\}$.

1. *Transformation of the Test Cover to Set Cover:* From $\breve{\mathcal{M}}(\mathcal{L}, \mathcal{S})$, we obtain a new matrix $\breve{\mathcal{M}}_T(\mathcal{L}, \mathcal{S})$ of dimensions $\binom{n}{2} \times m\sigma$, in which rows correspond to the *pair-wise* link failures, i.e.

   $$\{\ell_1\ell_2,\ \ell_1\ell_3, \cdots, \ell_1\ell_n,\ \ell_2\ell_3,\ \ell_2\ell_4, \cdots, \ell_{n-1}\ell_n\}$$

   In $\breve{\mathcal{M}}_T$, the row entry corresponding to $\ell_i\ell_j$ pair-wise event of the $(\sigma u + p)^{th}$ column represents the outcome of p^{th} output bit of sensor S_{u+1} for $u = \{0, 1, \cdots, m-1\}$ and $p = \{1, \cdots, \sigma\}$. This entry is 1 whenever p^{th} output of sensor S_{u+1} detects either ℓ_i or ℓ_j, but not both, i.e., either $\breve{\mathcal{M}}_{i,(\sigma u+p)} = 1$, or $\breve{\mathcal{M}}_{j,(\sigma u+p)} = 1$, where $\breve{\mathcal{M}}$ is the σ-bit influence matrix. This implies that S_{u+1} can distinguish between events ℓ_i and ℓ_j through its p^{th} output bit.

2. *Solving Set Cover Using Greedy:* Once $\breve{\mathcal{M}}_T$ is constructed from $\breve{\mathcal{M}}$, the objective is to select the minimum number of sensors so that all pairwise events are detected, which is basically a set cover problem. We select iteratively, a sensor S_{u+1} that detects the maximum number of uncovered pair-wise events, i.e., the set of columns $\{(\sigma u + 1), (\sigma u + 2), \cdots, (\sigma u + \sigma)\}$ in $\breve{\mathcal{M}}_T$ for $u \in \{0, 1, \cdots, m-1\}$ that collectively covers the maximum number of uncovered pair-wise events.

3.4 Simulation results

In this section, we illustrate the significance of using the multi-level sensing model. In particular, we formulate and solve the identification problem considering 2-bit sensors for the three networks. The results demonstrate the advantage of using 2-bit sensors in terms of improvement in localization of the failure events and the required number of sensors compared to the 1-bit sensors.

The maximum number of link failures that can be uniquely identified by sensors primarily depends on the network topology and the sensing model, i.e. either 1-bit or σ-bit. It is possible that complete identification might not be achieved even if a sensor is placed at every node within the network.

Thus, maximum number of link failures that can be uniquely detected or identified is of interest in the context of sensor placement. Another metric to quantify the performance of sensor placement is the number of sensors required to achieve the maximum identification. Thus, σ-bit sensors for $\sigma > 1$ would be better than 1-bit sensors if

1. The maximum number of link failures that can be uniquely detected by σ-bit sensors is greater than in the case of 1-bit sensors.
2. For a given number of sensors, more link failures can be uniquely detected by σ-bit sensors as compared to 1-bit case.

As illustrated in the following simulations for the 2-bit sensing model, both (1) and (2) are achieved in the three water networks. We consider the sensor placement for localization in three real water networks as defined in Section 2.4 using 2-bit sensors. For a single failure event ℓ occurring at the center of each pipe, the output of a 2-bit sensor S_i, denoted by $\mathbf{y}_{S_i}(\ell)$, is defined as follows:

$$\mathbf{y}_{S_i}(\ell) = \begin{cases} (1\ 0) & \text{if } d(S_i, \ell) < d_1 \\ (0\ 1) & \text{if } d_1 \leq d(S_i, \ell) \leq d_2 \\ (0\ 0) & \text{otherwise} \end{cases} \tag{13}$$

where $d(S_i, \ell)$ is the length of the shortest path between S_i and ℓ, $d_1 = 0.5[km]$, and $d_2 = 1[km]$. The transformation from time interval to length interval can be easily deduced through the propagation velocity of the pressure wave. Note that the previous threshold of $1[km]$, which corresponds to the maximum time within which an event ℓ can be detected, is now divided into two further thresholds, which correspond to the two sub-intervals of time within which the event can be detected by the sensor.

Figures 6a-6c compare the localization performance achieved through 1-bit and 2-bit sensing models for the three networks. Table 2 summarizes the maximum localization score and the number of sensors required based on the two sensing models. Note that using 2-bit sensors, the localization score increases significantly compared to 1-bit sensors. Moreover, for Net2 and Net3, the number of sensors required to achieve maximum localization is smaller compared to the corresponding cases of 1-bit sensors. This two-fold advantage entailed by 2-bit sensors, which are the simplest instance of multi-bit sensors, make them particularly attractive for the fault localization in urban water networks.

Table 2: Optimal solutions for single and bi-level sensing

	1-bit		**2-bit**	
Network	**No. of sensors**	**Max LS**	**No. of sensors**	**Max LS**
Net1	48	110	48	150
Net2	98	317	80	351
Net3	134	427	106	461

LS - localization score;

The results demonstrate that the minimum test cover formulation provides a promising decision support tool for the identification problem. The decision on the number of sensors is affected, in addition to budget constraint, by the detection and localization performance. The number of sensors required for the localization of failure events is significantly large as compared to detection of the events. In the next section, we extend the single-level sensing model to a multi-level sensing model that considers higher resolution sensors and event detection schemes.

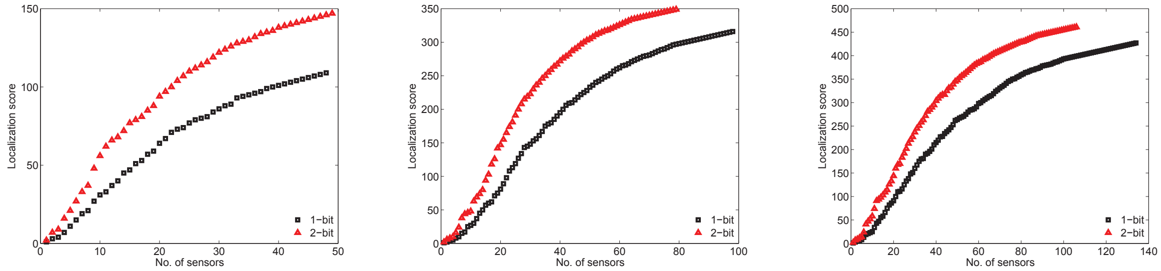
4. HETEROGENEOUS SENSORS

Multi-level sensing can improve the localization performance in terms of increasing the maximum number of link failures that can be uniquely identified while minimizing the required number of sensors. However, multi-level sensors may incur additional cost by requiring higher sampling and data transmission rates. To have a trade-off between cost and localization performance, another approach could be to utilize *heterogeneous* sensors, rather than using same sensors throughout the network. In this regard, some of the issues of great significance are as follows:

- How can we define and characterize heterogeneous sensors, specially in the context of urban water networks?
- Does deploying heterogeneous sensors allow a trade-off between performance and cost, which might be depicted by the number of sensors required?
- Where should these heterogeneous sensors be deployed within the network? In particular, how can we use the underlying network structure to determine potential locations for heterogeneous sensors?

Heterogeneity can be viewed along a number of dimensions, including the classification of sensors based on the physical aspects of the network being measured by the sensors, such as pressure, flow, or water quality [16]. Another distinction, even among the sensors which measure the same physical quantity, can be made based on their detection ranges and the detection algorithms used. For instance, sensors may be different in terms of their ability to detect the pressure signal generated at various distances.Yet another aspect of heterogeneity, which we use in the remainder of this section, can be construed in terms of the sensing levels, such that the value of σ is different for various sensors, as we explained in the setup of multi-level sensing. Note that in this case, sensors might have same detection ranges, but they are different in terms of the number of levels into which they classify their outputs such as 1-bit, 2-bits, etc.

In regards to the second question above, we illustrate using networks 1 and 2, and 1-bit and 2-bit sensors, that heterogeneous sensors leverage a trade-off between localization performance and the associated cost. The cost can be viewed in terms of the number of deployed sensors of various levels. Low level sensing is typically assumed to incur a smaller cost as compared to the high level sensing. In the case of all 1-bit sensors, the maximum localization scores (using greedy approach) achieved in networks 1 and 2 are 110 and 317 respectively (Table 2). This required 48 sensors for the network 1 and 98 for the network 2. How much could these scores be improved if few of these 1-bit sensors are replaced by the 2-bit ones? As shown in Table 2, the maximum localization scores with all 2-bit sensors are 150 and 351 for the networks 1 and 2, respectively, we expect the maximum scores to be in the ranges 110–150 for the network 1 and 317–351 for the network 2 when a mix of 1-bit and 2-bit sensors are used. We fix the total number of sensors, 50 and 100 for the networks 1 and 2, respectively, and compute the



(a) Net 1

(b) Net 2

(c) Net 3

Figure 6: Localization performance for single-level (black squares) and bi-level (red triangles) sensing models

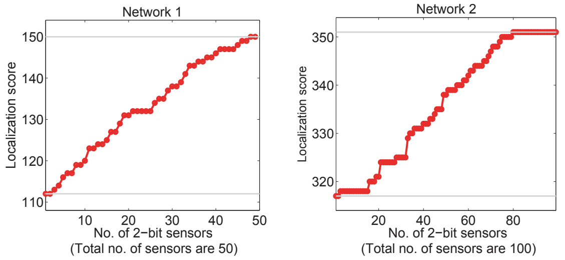


Figure 7: Localization scores obtained using various numbers of 1-bit and 2-bit sensors. The upper and lower horizontal lines indicate respectively the maximum localization scores that can be achieved when all 1-bit and all 2-bit sensors are used.

maximum scores for various combinations of heterogeneous sensors. Figure 7 shows the localization score as a function of the number of 2-bit sensors based on the results. These plots demonstrate the trade-off between localization performance and the cost in terms of the number of heterogeneous sensors employed.

To address the last question, we examine sensor placement purely on network topology based metrics, in which the preference to deploy resources is given to the nodes that are *important* purely from the underlying graph perspectives, such as node degree[1] and betweenness centrality[2]. To illustrate this, for both networks 1 and 2, we first rank their nodes based on the network topology based measures of degree and betweenness centrality separately. We then fix the total number of sensors and assign the available number of 2-bit and 1-bit sensors to nodes in the decreasing order of their ranks, while giving a preference to the 2-bit sensors. The localization scores are then computed for various number of available 2-bit and 1-bit sensors. For comparison purposes, in a separate experiment, we assign the given number of 2-bit and 1-bit sensors randomly (as per uniform distribution), and compute the localization scores of the resulting sensor configurations. Finally, we compare the results with the greedy selection of the nodes for the 2-bit and 1-bit sensors (as computed in Figure 7). In Figure 8, a comparison of approaches (network topology based, random, greedy selection) for the placement of 2-bit and 1-bit sensors is shown in terms of the computed localization scores.

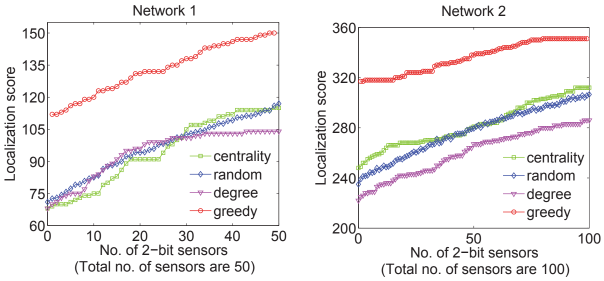


Figure 8: Comparisons of localization scores for 1-bit and 2-bit sensors placed using different topology metrics

In fact, our simulations illustrate that purely network based metrics serve as a bad choice for the sensor placement, in which higher level sensors are placed on the central locations in the underlying network graph. In physical networks, such as water distribution systems, the dynamics of physical process are crucial along with the network structure information for the most effective distribution of resources, including high and low level sensors for the failure detection. Thus, more carefully designed network based measures that also incorporate the physical aspects of the system are needed for the placement of heterogeneous sensors for the fault localization.

5. CONCLUSIONS

In this work, we proposed an efficient and practical method for sensor placement for failure detection and localization in water networks. The main contribution of this work are: (1) formulating the identification problem as minimum test cover problem, (2) suggesting an efficient fast greedy algorithm to solve the optimization problem, (3) introducing a multi-level sensing model to capture different types of available sensors and event detection schemes, and (4) controlling network performance by considering heterogeneous sensors. Multi-level sensing improves fault localization both in terms of improving the localization scores and reducing the required number of sensors, though it may incur additional cost. Heterogeneous sensing, on the other hand, leverages a trade-off between cost and performance. So, which of the following options serve as the best choice for sensor placement for fault localization in urban water networks: (1) all low-level sensors, (2) all high-level sensors, or (3) heterogeneous

[1] The degree of a node in a graph is the number of nodes (or edges) that are adjacent the node.

[2] The betweenness centrality of a node is the number of shortest paths from all nodes to all others that pass through that node.

sensors. There might not be a straight-forward answer here as a number of factors play a significant role in the selection. Among others, some of these factors include the cost of network reinforcement, the quality of online detection and corresponding cost of delay in response, robustness to random failures in the sensor network, and resilience to strategic failures in the sensor network. Thus, sensors' selection and placement criteria that can incorporate simultaneously all major aspects of the fault localization performance are crucial for a design of a comprehensive decision support tool for the fault localization in water networks, and is also our future work.

Acknowledgements

This work was supported in part by the National Science Foundation (award numbers: CNS-1453126, CNS-1238959, CNS-1238962, CNS-1239054, CNS-1239166) and Air Force Research Laboratory (contract ID: FA8750-14-2-0180, SUB 2784-018400).

6. REFERENCES

[1] Bentley, Water Hammer and Transient Analysis Software. `http://www.bentley.com/en-US/Products/HAMMER/`. Accessed: 2015-04-14.

[2] J. Berry, W. Hart, C. Phillips, J. Uber, and J. Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management*, 132(4):218–224, 2006.

[3] K. M. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. Hurkens, J. K. Lenstra, R. Ravi, and L. Stougie. Approximation algorithms for the test cover problem. *Mathematical Programming*, 98(1-3):477–491, 2003.

[4] A. Deshpande, S. E. Sarma, K. Youcef-Toumi, and S. Mekid. Optimal coverage of an infrastructure network using sensors with distance-decaying sensing quality. *Automatica*, 49(11):3351–3358, 2013.

[5] D. Eliades and M. Polycarpou. A fault diagnosis and security framework for water systems. *IEEE Transactions on Control Systems Technology*, 18(6):1254–1265, Nov 2010.

[6] B. V. Halldórsson, M. M. Halldórsson, and R. Ravi. On the approximability of the minimum test collection problem. In *Proceedings of the 9th Annual European Symposium on Algorithms*, ESA '01, pages 158–169, London, UK, UK, 2001. Springer-Verlag.

[7] W. Hart, J. Berry, E. Boman, R. Murray, C. Phillips, L. Riesen, and J. Watson. *The TEVA-SPOT Toolkit for Drinking Water Contaminant Warning System Design*, chapter 511, pages 1–12.

[8] W. Hart and R. Murray. Review of sensor placement strategies for contamination warning systems in drinking water distribution systems. *Journal of Water Resources Planning and Management*, 136(6), 2010.

[9] M. D. Jolly, A. D. Lothes, S. Bryson, and L. Ormsbee. Research database of water distribution system models. *Journal of Water Resources Planning and Management*, 140(4):410–416, 2014.

[10] T. Kim and S. J. Wright. PMU placement for line outage identification via multiclass logistic regression. arXiv:1409.3832v1 [math.OC].

[11] A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

[12] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[13] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.

[14] B. M. E. Moret and H. D. Shapiro. On minimizing a set of tests. *SIAM Journal on Scientific and Statistical Computing*, 6(4):983–1003, 1985.

[15] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, Phillips, et al. The battle of the water sensor networks: A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management*, 134(6):556–568, 2008.

[16] R. Puust, Z. Kapelan, D. A. Savic, and T. Koppel. A review of methods for leakage management in pipe networks. *Urban Water Journal*, 7(1):25–45, 2010.

[17] L. Sela Perelman, W. Abbas, X. Koutsoukos, and S. Amin. Sensor placement for fault location identification in water networks: a minimum test cover approach. 2015. http://arxiv.org/abs/1507.07134 [cs.SY].

[18] E. Wylie, V. Streeter, and L. Suo. *Fluid transients in systems*. Prentice Hall, 1993.

[19] T. T. Zan, H. B. Lim, K.-J. Wong, A. J. Whittle, and B.-S. Lee. Event detection and localization in urban water distribution network. *IEEE Sensors Journal*, 14(12):4134–4142, 2014.

[20] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, Dec 2011.