Sensor placement for fault location identification in water networks: A minimum test cover approach

Citation: Sela Perelman et al. "Sensor Placement for Fault Location Identification in Water Networks: A Minimum Test Cover Approach." Automatica 72 (October 2016): 166–176 © 2016 Elsevier Ltd

As Published: http://dx.doi.org/10.1016/j.automatica.2016.06.005

Publisher: Elsevier

Persistent URL: http://hdl.handle.net/1721.1/118402

Version: Original manuscript: author's manuscript prior to formal peer review

Sensor placement for fault location identification in water networks: a minimum test cover approach

Lina Sela Perelman [a], Waseem Abbas [b], Xenofon Koutsoukos [b], Saurabh Amin [a]

[a] *Massachusetts Institute of Technology*

[b] *Vanderbilt University*

Abstract

Resilient water networks aided by real-time sensing and analytics are crucial to ensure water security. Design of a network of sensors for pipe failure location identification is a salient feature of our work. We cast the fault location identification problem as the minimum test cover (MTC) problem. The MTC is a combinatorial optimization problem in which the objective is to select the minimum number of tests such that every event can be uniquely classified in one of the given categories based on selected tests' outcomes. In our setup, the set of outcomes of tests are sensors' states, events are pipe bursts, and classification categories are the location of the failed pipes. We consider two approaches to solve the NP-hard MTC problem. First, we transform the MTC problem to the counterpart minimum set cover (MSC) problem and using greedy heuristic compute our solution by exploiting submodularity of the set cover problem. Second, we present a *fast greedy* approach for solving the MTC that does not require a complete transformation of the MTC to the equivalent MSC and directly compute the value of the objective function in a greedy fashion. Finally, we suggest several metrics to evaluate the performance of the design, including detection, identification, and localization scores, and worst localization performance. We provide a detailed analysis of our approach along with the simulations for a benchmark and a real water distribution network.

Key words: Fault identification; Minimum test cover; Water infrastructure.

1 Introduction

Infrastructure deterioration, demand-supply uncertainty, and risk of disruptions pose new challenges in maintaining infrastructure resilience. Resilient urban infrastructures spanning water distribution systems, transportation networks, and electric grids are crucial for societal well-being. *Smart* infrastructure systems fortified with sensing technologies have been identified as one of the primary tools towards sustainable urban systems [21,13]. Through a network of sensors, a fault in a system component can be detected and localized, and rectification actions can be executed in response to the specific fault. Whereas network observability through sensor placement has been widely studied in the context of fault detectability, sensor placement for fault isolability, i.e. the ability to distinguish between faults, has not been commonly sought after, specially in the context of burst detection in water distribution networks. Design of a network of sensors for pipe failure location identification trough the minimum number of sensors is a salient feature of this work.

Several problem formulations and solution approaches have been developed and implemented to solve the sensor placement problem in water systems. A typical approach is to formulate the desired objective function through a set valued function $f(S)$ that depends on the choice of sensor set S. The maximization (or minimization) of f is desired through a sensor set for which heuristics or algorithms are developed. A key object in the efficient and practically feasible design of these algorithms is to carefully examine and exploit the properties of f. In [12], a mutual information criterion was proposed to select the most informative sensors to monitor a spatial phenomenon modeled by a Gaussian process. The submodular property of the criterion, as shown in [19], was then exploited to obtain a polynomial time algorithm guaranteeing a constant factor approximation of the optimal sensor set. In simple terms, the submodularity encapsulated a diminishing return behavior of the criterion. In a related work [10], to detect the presence of contaminants in large water distribution systems, the

Email addresses: linasela@mit.edu (Lina Sela Perelman), waseem.abbas@vanderbilt.edu (Waseem Abbas), Xenofon.Koutsoukos@vanderbilt.edu (Xenofon Koutsoukos), amins@mit.edu (Saurabh Amin).

notion of penalty reduction function was introduced to realize objective functions such as reduction of detection time, expected consumption of contaminated water, and the expected population affected. Submodularity of the penalty reduction function was then used to solve sensor placement problems efficiently and with provable guarantees. Moreover, solving sensor placement and sensor scheduling simultaneously to monitor spatial phenomena was shown to exhibit better performance, specially when sensing quality function could be measured and satisfied submodularity [11]. The facility location approach for sensor placement was suggested in [3], in which sensor placement problem for contaminant warning system was formulated as a well known discrete p-median problem from discrete location theory, and could be written as a mixed integer linear program. The problem of computing locations for sensors measuring contaminant concentrations was further addressed in [7]. A problem formulation was presented that included state-space representation of the propagation and reaction dynamics of the contaminant, coupled with the impact dynamics describing the damage caused by contamination. The problem was posed as single and multi-objective optimization problem suggesting genetic algorithms as a solution method.

Once the sensors are in place, data is collected and transmitted in real-time. Various data and model driven techniques are then applied for system state estimation and event detection and isolation [6,22,24,25]. The basic premise in these methods is to compute the difference between measurements, such as pressure [22] and flow [24], and their estimated values using the network hydraulic model. Model based leakage detection techniques are employed primarily on the operational side with the objective to efficiently utilize the available measurements along with the available system model to determine the system faults. In this work, we consider the optimal sensor placement from the design perspective for the detection as well as localization of pipe failures in water distribution networks.

In the urban water sector, majority of previous works focused on the sensor placement for detecting hypothetical contamination events assuming perfect sensors capable of detecting all types of contaminants. Only few works have considered the optimal sensor placement for systems failures, specially detection and localization of leakages and pipe failures. In this regard, our approach is somewhat related to [5,26] that consider pipe bursts as failure events. In [5], detection of events in networks is studied using distance decaying sensing function. The problem is formulated as a continuous p-median facility location problem and solved using a gradient descent algorithm. However, in contrast to [5], in which only the detection problem is considered, we consider detection as well as location identification of link failures. In [26] both the detection and location identification of failure events are considered in the problem formulation. Their framework is based on structural network analysis and the solution is attained through the depth-first branch and bound search. We cast the fault identification problem as a combinatorial optimization problem, employing the *minimum test cover* problem formulation, to solve the optimal sensor placement problem, which allows us to obtain performance guarantees on the obtained solution.

The minimum test cover (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 selected tests' outcomes [18]. In our sensor placement problem for water distribution networks, set of outcomes of tests are sensors states, the events are pipe bursts, and the classification categories are the location of the failed pipe, as described in Section 2. For the identification of a pipe failure, the objective is to determine a minimum number of sensors (tests) so that each pipe failure can be uniquely identified through selected sensors' outputs.

The MTC problem is NP-hard [18], and therefore no polynomial time algorithm can give an optimal solution. However, using the approach as in [8], we transform the MTC problem to the counterpart minimum set cover (MSC) problem first, which we solve using greedy approximation, as shown in Section 3. If n is the total number of possible failure events, the greedy algorithm gives an approximation of $(1 + 2\ln n)$ for the test cover problem, which is the best possible [18]. However, the event space of corresponding MSC instance consists of all pairs of events and therefore contains a total of $\binom{n}{2}$ events. This poses a major limitation in terms of the scalability, particularly for large n. Thus, we exploit submodular property of the set cover problem to obtain an efficient solution. In Section 4 we present a *fast greedy* approach for solving the MTC that does not require the complete transformation of the MTC to the equivalent MSC and directly computes the objective function in a greedy fashion. This algorithm is much faster than the straight-forward greedy approach, which is specially attractive for large-scale networks application.

Finally, in Section 5 we apply our approach on two water distribution networks. We suggest four metrics to evaluate the performance of the design including detection, identification, and localization scores, and worst localization performance. Section 6 summarizes main contributions of our work and future extensions.

2 Problem formulation

This work focuses on the design of network of sensors, continuously measuring hydraulic pressures, such that the fault location identification through event detection

is maximized. We consider n link failures, i.e. pipe bursts, as a set of failure events, $\mathcal{L} = \{\ell_1, \ldots, \ell_n\}$. For the ease of presentation without the loss of generality, let ℓ_j denote the location of the failure event. Additionally, we define 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 ith sensor. After occurrence of the failure event ℓ_j, transient system state is analyzed in terms of pressure, p, and flow, q, conditions. The output of sensors, denoted by $\mathbf{y}_{\mathcal{S}}$, are then evaluated at potential sensor locations. A combinatorial optimization problem for the sensor placement is then formulated to maximize failure events' location identification through the minimum number of sensors.

2.1 Optimal sensor placement

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, f, is maximized. The optimization problem is then formulated as:

$$\max_{S \subseteq \mathcal{S}} \{f(S; \mathcal{L}), |S| \leq M\} \tag{1}$$

where M is the number of available sensors and f is a set function $f : 2^{\mathcal{S}} \longrightarrow \mathbb{R}$ that assigns a real number to the selected set of sensors S.

The *detection* problem is to find the minimum number of sensors and their locations such that every link failure can be detected by at least one sensor. The objective function, f_D, assigns a positive integer value quantifying the number of failure events that are detected by a given set of sensors. The location *identification* problem is to find the minimum number of sensors and their locations, such that every link failure can be *uniquely* identified, i.e., distinguished from any other link failure. The objective function, f_I, quantifies the number of events that are uniquely distinguishable by a given subset of sensors.

To emphasize the difference between detection and identification objectives, consider a *super*-sensor installed at one of the network nodes and is capable of detecting all failure events. From the detection point of view, this is the best possible solution, since a single sensor is sufficient to detect all failure events. However, from the identification point of view, this is the worst possible solution, since we cannot distinguish, localize, and respond to any of the events. We focus on the sensor placement for the detection as well as localization of pipe bursts in water distribution networks.

2.2 Network dynamics

A water distribution system can be represented by a network graph composed of nodes (supply and demand) connected by links (pipes, valves, and pumps). The network flow model is characterized by flows over network links, q, and hydraulic heads over network nodes, h. The hydraulic head is equal to the pressure, p, and elevation of the node, z, i.e. $h = p + z$. The system's steady state conditions can be described using mass conservation at network nodes and energy conservation along network links [27]. Physical failures of the infrastructure, such as pipe bursts, cause rapid change in the flow, which moves through the system as a pressure wave known as *water hammer* or surge with very high velocity varying typically in the range of $600 - 1500[\frac{m}{s}]$. This implies that the steady state analysis is inadequate and requires analyzing the transient system state between the initial and the final steady state conditions.

Consider a pipe burst event $\ell \subseteq \mathcal{L}$, initiated at time t_0 inducing a transient system response, denoted by $h_\ell(t_0)$. Let ℓ be the location of the event, $h(t)$ be the vector of hydraulic heads at time t, and $h_\ell(t)$ be the hydraulic head at location ℓ and time t. The transient system state can be described by mass and momentum partial differential equations that are commonly solved using the method of characteristics. A full derivation of equations describing changes in pressures can be found in [28] and is demonstrated in the Appendix A. Moreover, changes in flow conditions can be described by equations (2):

$$h_\ell(t_0 + 1) = P\left(C_d A_\ell(t_0)\right) \tag{2a}$$

$$h(t+1) = E_h h(t) + Bq(t) + RH\left(q(t)\right) \tag{2b}$$

$$q(t+1) = B_h h(t+1) - E_q h(t) + B_q q(t) - R_q H\left(q(t)\right) \tag{2c}$$

where (2a) describes the burst boundary conditions induced by event in location ℓ; (2b) and (2c) describe the change in heads and flows during the transient state; $C_d A_\ell$ is burst coefficient; $E_h, B, R, B_h, E_q, B_q, R_q$ are sparse matrices representing the grid of characteristics and the corresponding impedance and resistance coefficients, and P, H are nonlinear functions describing the burst boundary condition and the hydraulic head loss.

We consider the use of online hydraulic pressure sensors (see Appendix B) continuously monitoring the system state for detecting pipe bursts by analyzing the transient flow regime.

Remark 1 *In a transient flow regime, both pressure and flow measurements can be used. However, most flow meters do not react instantaneously to changes in flow, hence are not suitable for transient analysis [23].*

Let $p(t) := h(t) - z$ 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. The capability of the sensor to detect a pressure anomaly depends on the impact of the fault ℓ at the location of the sensor S_i. Let $y_S(t, \ell) \in \{0, 1\}$ be a discrete sensor state vector – 0 representing a normal operating state and 1 representing otherwise. Moreover, assume $\xi : p(\cdot) \longrightarrow \mathbb{R}$ be some function characterizing the distance between the expected and the measured

data, e.g. the error between model based and measured pressures at a given time and location. The sensor outcome can then be formulated as:

$$y_{S_i}(t,\ell) = \{1 \mid \xi\left(p_{S_i}(t_0+\tau_\ell)\right) \geq \varepsilon\} \tag{3}$$

where ε is a threshold value and τ_ℓ is the time passed since t_0, i.e., the time it takes the sensor to detect the event ℓ. An example for simple sensing model would be where the sensor S_i indicates an event if the change in the pressure is above some threshold value ε. Our main objective in obtaining the sensing model is to acquire the sensor's output as a result of some event ℓ. By allowing the sensor S_i to detect event ℓ in a certain finite time T, we can neglect the time dependency of the sensor to detect the event, and hence can restate the output of the sensor as:

$$\mathbf{y}_{S_i}(\ell) = \begin{cases} 1 & \text{if } \; y_{S_i}(t,\ell) = 1, \quad \text{for some } t \in [t_0 \quad T] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Let, $\mathbf{y}_{\mathcal{S}}(\ell) = [\mathbf{y}_{S_1}(\ell), \cdots, \mathbf{y}_{S_m}(\ell)]$ be a boolean vector of the outputs of sensors in the set $\mathcal{S}$ for a failure event ℓ.

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, $\mathbf{y}_{\mathcal{S}}(\ell), \ell \in \mathcal{L}$, and $\mathcal{M}_{ij} = 1$ indicates that a sensor S_i detected a failure in link ℓ_j, and $\mathcal{M}_{ij} = 0$ otherwise, as:

$$\mathcal{M}\left(\mathcal{L},\mathcal{S}\right) = \begin{bmatrix} \mathbf{y}_{\mathcal{S}}(\ell_1) \\ \mathbf{y}_{\mathcal{S}}(\ell_2) \\ \vdots \\ \mathbf{y}_{\mathcal{S}}(\ell_n) \end{bmatrix} \tag{5}$$

For the set of link failures, $\mathcal{L}$, and the set of all available sensors, $\mathcal{S}$, we define $C_i = \{\ell_j \in \mathcal{L} \mid \mathbf{y}_{S_i}(\ell_j) = 1\}$, for each sensor S_i, and $\mathcal{C} = \{C_i : \forall i\}$. 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.

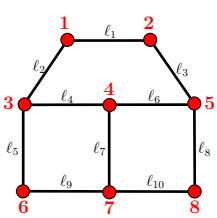


Fig. 1. Illustrative example layout

Example 2 *To illustrate network dynamics, consider a small network having* 8 *nodes connected by* 10 *links as shown the Fig. 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].*

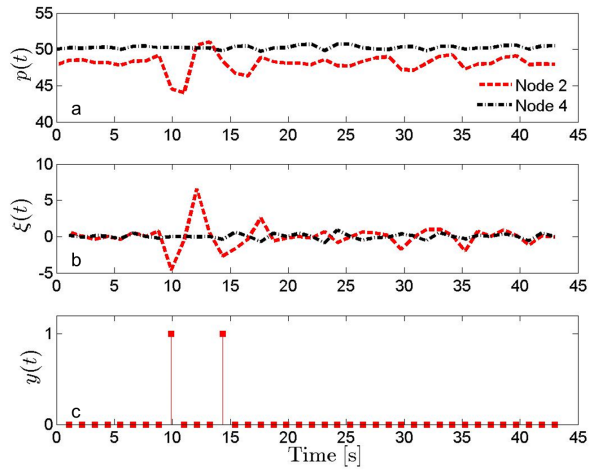


Fig. 2. Network dynamics for failure event generated in pipe ℓ_1 in the small example – (a) $p(t)$ pressure head $[m]$, (b) $\xi(t)$ change in the pressure $[m]$, and (c) $y_{S_2}(t,\ell_1)$, i.e., sensor S_2 output in response to event ℓ_1.

Fig. 2(a) shows the pressure head, $p(t)$*, generated at nodes 2 and 4 in response to the pipe failure* ℓ_1*. As explained previously, a burst event causes rapid change in the pressure head, hence a straightforward estimation of a potential event is through observing the difference in pressure during subsequent time periods. Fig. 2(b) shows the difference in the pressure where* $\xi\left(p_{S_i}(t+1)\right) = p_{S_i}(t+1) - p_{S_i}(t)$*. It is important to note that this simplified difference function* ξ *is for demonstration purposes only, and more sophisticated methods are typically applied for outlier detection in online data streams. Fig. 2(c) shows the sensor state indicating an event if the pressure drop is larger than a threshold of* $2.5[m]$*. We can observe that the sensor installed at node 2 is able to detect the event originated in pipe 1, whereas the sensor at node 4 is not detecting the event. Thus for* $S_A = \{S_2, S_4\}$ *the sensor state is* $\mathbf{y}_{S_A}(\ell_1) = [1, 0]$*. If we consider placing sensors at all nodes of the system, failure is detected by sensors located at nodes* $\{1, 2, 3, 5\}$*, then the sensor state is* $\mathbf{y}_{\mathcal{S}}(\ell_1) = [1, 1, 1, 0, 1, 0, 0, 0]$*.*

Next, if we consider a single failure event in each pipe of the network with $\mathcal{L} = \{\ell_1, \cdots, \ell_{10}\}$*, then, for* S_2 *for example,* $C_2 = \{\ell_1, \ell_2, \ell_3, \ell_6, \ell_8\}$*. Moreover, the corresponding influence matrix is:*

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

2.3 Detection as minimum set cover

For a set of events $\mathcal{L}$ and a set of sensors $\mathcal{S}$, the detection problem is to place the minimum number of sensors within the network such that whenever an event $\ell \in \mathcal{L}$ occurs, at least one sensor detects the event. Following the above formulation, row indices corresponding to 1 in each column of $\mathcal{M}$ represent link failure events that can be detected by the sensor placed in the corresponding location. To optimally solve the detection problem, it is then required to select the minimum number of columns of $\mathcal{M}$ such that all the rows are covered, i.e., for every row, there is at least one column that contains 1 at the corresponding row index. Equivalently, objective function, f_D, of the detection problem can be formulated as:

$$f_D(S;\mathcal{L}) = \left| \bigcup_{C_i \in \mathcal{C}_S} C_i \right| \tag{6}$$

In fact, maximizing the detection score (6) by selecting the minimum number of sensors is related to well known *minimum set cover problem* which is defined as [4]:

Definition 2.1 *(Minimum set cover (MSC)) Let $\mathcal{L}$ be a finite set of elements, and $\mathcal{C} = \{C_i : C_i \subseteq \mathcal{L}\}$ be the collection of subsets of $\mathcal{L}$. The minimum set cover is to find $\mathcal{C}_s \subseteq \mathcal{C}$ with the minimum cardinality such that $\bigcup_{C_i \in \mathcal{C}} C_i = \bigcup_{C_j \in \mathcal{C}_s} C_j$.*

If $\mathcal{L}$ is the set of link failures and $\mathcal{C}$ is the collection of C_i's corresponding to all the available sensors, then the minimum set cover, denoted by $\mathcal{C}_s$, gives the minimum number and locations of sensors that solve the detection problem. Thus, we get the following:

Proposition 2.1 *Maximizing the detection of link failures in a network is equivalent to solving the minimum set cover problem.*

2.4 Identification as minimum test cover

In the identification problem, the objective is to maximize the number of uniquely identified link failures through the minimum number of sensors. One way to achieve that is by selecting a set of sensors S, such that the sensors' state vector, $\mathbf{y}_S(\ell_i)$, is unique for every $\ell_i \in \mathcal{L}$. In other words, for every pair of events ℓ_i and ℓ_j, sensors' states $\mathbf{y}_S(\ell_i)$ and $\mathbf{y}_S(\ell_j)$ should differ by at least one element. In terms of the influence matrix of the network, this is possible whenever for every row pair $\ell_i, \ell_j,\ i \neq j$, 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* problem. Given a set of events $\mathcal{L}$ and a collection of tests $\mathcal{C}$, a test C_i *covers an event pair* $\{\ell_i, \ell_j\}$ (or differentiates between the pair of events ℓ_i and ℓ_j) if either $\ell_u \in C_i$ or $\ell_v \in C_i$, i.e. $|C_i \cap \{\ell_u, \ell_v\}| = 1$ [4].

Definition 2.2 *(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 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_u, then there exists some $C_j \in \mathcal{C}_t$ that also contains exactly one of ℓ_u and ℓ_u.*

The identification problem is to find a subset $\mathcal{C}_t \subseteq \mathcal{C}$ of minimum cardinality, or equivalently the corresponding subset of sensors $S \subseteq \mathcal{S}$, such that if $\mathbf{y}_{\mathcal{S}}(\ell_j)$ is unique with respect to set of all sensors $\mathcal{S}$, then $\mathbf{y}_S(\ell_j)$ is also unique with respect to a subset of sensors S, which is precisely the MTC problem defined above. Thus, we can state:

Proposition 2.2 *Maximizing the identification of link failures in networks is equivalent to solving the minimum test cover problem.*

Example 3 *Following example 2, consider two sensors placed at nodes 2 and 4, $S_A = \{S_2, S_4\}$. For the detection purpose, we note that $C_2 \cup C_4 = \mathcal{L}$. In other words, at least one of the sensors S_2 and S_4 has a state 1 whenever a link fails. Thus, sensors S_2 and S_4 cover all link failures and solve the detection problem. For identification, sensors S_2 and S_4 are not sufficient as the sensors generate only three unique states associated with the 10 events, which makes it impossible to distinguish between all link failures. For example, the state $\mathbf{y}_{S_A}(\ell_1) = \{1, 0\}$ is uniquely associated with a failure in link ℓ_1, whereas, the state $\{1, 1\}$ can be associated with a failure in any of the links ℓ_2, ℓ_3, ℓ_6, or ℓ_8. However, for the set of sensors $S^* = \{S_1, S_2, S_3, S_5\}$ that solves the MTC of problem of example 2, the output is unique for each link failure, i.e. 10 distinct indicator vectors corresponding to each failure event.*

3 Main results

In this section, we present solutions of sensor placement problems for the detection and identification of link failures by solving the minimum set cover and the minimum test cover problems, respectively.

3.1 Detection solution

Both problems, minimum set cover and minimum test cover, are NP-hard [15]. MSC has been studied extensively owing to its wide variety of applications in theoretical as well as practical domains. A straight-forward way to solve the MSC is by the *greedy approach*. In our detection problem, the greedy approach is to select, in

each iteration, a sensor that detects the maximum number of uncovered link failures. This continues until all link failures are covered, or no further link failure can be detected by any sensor. If n is the total number of link failures, m is the total number of sensors, then *greedy* algorithm gives the best approximation ratio of $\mathcal{O}(\ln n)$ for the set cover problem [14,15]. In fact, if k is the maximum number of link failures that can be detected by any sensor, then the greedy algorithm has an approximation ratio of $\mathcal{O}(\ln k)$, which is the best possible (unless P=NP).

The greedy approach, although gives the best approximation ratio, requires a large number of function evaluations. The running time of greedy approach is a function of the number of sensors and events, $\mathcal{O}(mn)$. For large scale systems and data intensive applications, in which n and m are very large numbers, this simple greedy approach becomes infeasible owing to a large number of function evaluations, even if evaluating a function is not expensive. However, greedy algorithm can be made extremely fast by reducing the number of function evaluations if a certain property, known as *submodularity*, is satisfied by the function involved [16]. Submodularity depicts the diminishing return behavior of the function.

Definition 3.1 *(Submodularity) Let $\mathcal{C}$ be a finite set and f be a set function, $f : 2^{\mathcal{C}} \longrightarrow \mathbb{R}$. Moreover, $\mathcal{C}_s \subseteq \mathcal{C}_r \subseteq \mathcal{C}$, and $C_i \in \mathcal{C} \setminus \mathcal{C}_r$, then f is submodular whenever*

$$f(\mathcal{C}_s \cup \{C_i\}) - f(\mathcal{C}_s) \geq f(\mathcal{C}_r \cup \{C_i\}) - f(\mathcal{C}_r) \quad (7)$$

In the context of detection problem, this means that as the number of link failures detected by the sensors selected increase, the utility of adding a sensor to the cover would decrease. If $\mathcal{L} = \{\ell_1, \cdots, \ell_n\}$ is the set of link failures, $\mathcal{C} = \{C_1, \cdots, C_m\}$ is the collection of C_i's where C_i is the set of link failures detected by the sensor S_i, and $\mathcal{C}_s \subseteq \mathcal{C}$ is a given subset of $\mathcal{C}$, then in the greedy algorithm for the detection problem, the objective function f is computed as:

$$f(\mathcal{C}_s) = \left| \bigcup_{C_i \in \mathcal{C}_s} C_i \right| \quad (8)$$

which is exactly the detection problem objective as in (6). Note that $\mathcal{C}$ is a finite set in which each element C_i is a subset of $\mathcal{L}$.

Lemma 3.1 *The set function f as described in (8) is submodular.*

Proof – Let $\mathcal{C}_s \subseteq \mathcal{C}_r \subseteq \mathcal{C}$, and $C_i \in \mathcal{C} \setminus \mathcal{C}_r$, then we need to show

$$f(\mathcal{C}_s \cup \{C_i\}) - f(\mathcal{C}_s) \geq f(\mathcal{C}_r \cup \{C_i\}) - f(\mathcal{C}_r)$$

Assume that $C_i' = C_i \setminus \bigcup_{C_j \in \mathcal{C}_s} C_j$, then

$$f(\mathcal{C}_s \cup \{C_i\}) = f(\mathcal{C}_s \cup \{C_i'\}) = f(\mathcal{C}_s) + f(\{C_i'\}) \quad (9)$$

Moreover, let $\lambda = \left(\bigcup_{C_k \in \mathcal{C}_r} C_k \right) \setminus \left(\bigcup_{C_j \in \mathcal{C}_s} C_j \cup C_i' \right)$, and $\mu = \bigcup_{C_k \in \mathcal{C}_r} C_k \cap C_i'$, then

$$f(\mathcal{C}_r \cup \{C_i\}) = f(\mathcal{C}_r \cup \{C_i'\}) = f(\mathcal{C}_s \cup \{C_i'\}) + f(\{\lambda\}), \quad (10)$$

and

$$f(\mathcal{C}_r) = f(\mathcal{C}_s) + f(\{\lambda\}) + f(\{\mu\}). \quad (11)$$

Substituting (11) into (10) gives,

$$f(\mathcal{C}_s \cup \{C_i\}) - f(\mathcal{C}_s) - f(\{\mu\}) = f(\mathcal{C}_r \cup \{C_i\}) - f(\mathcal{C}_r)$$

The required result follows directly. □

Next, this submodularity of f in (8) can be exploited to obtain the *lazy greedy* algorithm as in [16]. The basic idea behind accelerated greedy is to get rid of the redundant computations in each iteration, and can be explained as follows: let $F_\kappa(C_i)$ be the utility of adding a sensor i to the cover in the κth iteration, then by the submodularity of f, we know that $F_{\kappa+1}(C_i) \leq F_\kappa(C_i)$. Moreover, without loss of generality, we assume that $F_\kappa(C_1) \geq F_\kappa(C_2) \geq \ldots$, then C_1 is the greedy choice in the κth iteration. However, in the next iteration, if $F_{\kappa+1}(C_2) \geq F_\kappa(C_3)$, then $F_{\kappa+1}(C_2) \geq F_{\kappa+1}(C_j)$, $\forall j \geq 3$, which means that there is no need to compute $F_{\kappa+1}(C_j)$, $\forall j \geq 3$. This saves a large number of computations, thus accelerating greedy algorithm, which is crucial for large scale systems.

3.2 Identification solution

The objective of the identification problem formulated as MTC is to find the minimum number of sensors and their locations, that detect the most pairwise link failures. One of the approaches to solve the minimum test collection problem is by first transforming it to the minimum set cover problem [4], and then solving the MSC problem using efficient heuristics, as explained before. In fact, a greedy approach to solve MTC yields a $(1 + 2\ln n)$ approximation ratio algorithm, which is the best possible [18], where n is the total number of link failures. The solution of the counterpart MSC is the solution to the original MTC. Thus, a straight-forward way to solve the identification problem for link failures is to first obtain an equivalent detection problem, and then utilize the greedy approach to solve the corresponding detection problem. Next, we illustrate the transformation of the MTC problem to the MSC problem as outlined in [4].

3.2.1 Transformation of MTC to MSC

Given an instance of MTC, i.e., $\mathcal{L}$ and $\mathcal{C}$, where $C_i \subseteq \mathcal{L}$, transform the MTC to the MSC by taking the following two steps:

- *Create new set of events* – $\mathcal{L}^t = \{\ell^t_{12}, \cdots, \ell^t_{(n-1)n}\}$. For each unordered pair $\{\ell_i, \ell_j\}$, define an element ℓ^t_{ij}, then $\mathcal{L}^t$ consists of all such ℓ^t_{ij}'s.
- *Create new sets of sensors' outputs* – $\mathcal{C}^t = \{C^t_1, \cdots, C^t_m\}$, where $C^t_\kappa = \{\ell^t_{ij} : \quad |\{\ell_i, \ell_j\} \cap C_\kappa| = 1\}$, $\forall k \in \{1, \cdots, m\}$. In other words, $\ell^t_{ij} \in C^t_\kappa$ whenever exactly one of ℓ_i or ℓ_j is in C_κ.

Hence, we obtain a new matrix $\mathcal{M}^t(\mathcal{L}^t, \mathcal{S})$ of dimension $\binom{n}{2} \times m$ in which each row corresponds to a *pairwise link failure* and each column represents sensor's output. If the uth row in $\mathcal{M}^t$ represents the pair ℓ_i, ℓ_j, then the vth column entry of the corresponding row in $\mathcal{M}^t$ is simply the *exclusive OR* of the $(i, v)^{th}$ and $(j, v)^{th}$ entries of the influence matrix $\mathcal{M}$. The above point illustrates the fact that to localize an event ℓ_i, there is always a sensor that distinguishes ℓ_i from ℓ_j by producing different outputs for ℓ_i and ℓ_j respectively, i.e., if sensor output is 1 (or 0) in case of ℓ_i, then to differentiate ℓ_i from another event ℓ_j, the sensor output is 0 (or 1) in case of ℓ_j. This should be true for all $j \neq i$.

Then, in the greedy algorithm of the identification problem, $f_I(S; \mathcal{L})$ is the number of uniquely identified *pairs* of failure events by the given sensor configuration S and is essentially computed as the detection score of the corresponding MSC instance:

$$f_I(\mathcal{S}; \mathcal{L}) = f_D(S; \mathcal{L}^t) \tag{12}$$

Where f_D is computed as in (6), but with the transformed $\mathcal{L}^t$ and $\mathcal{C}^t$. The normalized identification score is computed by dividing f_I by the total number of pairwise events, $|\mathcal{L}^t|$.

3.2.2 Greedy approach based solution

Once the test cover problem has been transformed to the set cover, a straightforward way to obtain a solution is to employ the greedy heuristics as outlined in Algorithm 1.

Algorithm 1 Minimum Test Cover - Simple Greedy

1: **Input:** $\mathcal{C} = \{C_1, \cdots, C_m\}$, $C_i \subseteq \mathcal{L}$
2: **Output:** MTC: $\mathcal{C}^* \subseteq \mathcal{C}$
3: **Initialize:** $\mathcal{C}^* \leftarrow \emptyset$
4: **Transform:** the test cover instance to the set cover instance, i.e., from a given $\mathcal{L}$ and $\mathcal{C}$, obtain a corresponding $\mathcal{L}^t$ and $\mathcal{C}^t$ (Section 3.2.1).
5: **Solve:** using greedy algorithm
   (a) Select $C^t_{i^*} \in \mathcal{C}^t$ (i.e., the sensor i^*) covering the most uncovered elements in $\mathcal{L}^t$.
   (b) $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{C_{i^*}\}$
   (c) Repeat until all elements in $\mathcal{L}^t$ are covered or no new element in $\mathcal{L}^t$ can be covered by any $C^t_i \in \mathcal{C}^t$.

To accelerate the greedy process, lazy greedy approach, which exploits the submodularity property of the set cover problem, can be utilized. However, if there are n link failures (events) that need to be localized, then the corresponding set cover instance contains $\binom{n}{2}$ events, and the time complexity of the greedy approach in Algorithm I 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 lazy greedy also does not solve the problem due to a large number of events in the event space. Thus, the objective is to improve upon the time complexity of the identification solution to make it more scalable.

4 Fast greedy MTC solution

The main idea of the fast greedy approach is achieve a computationally efficient approximation algorithm. We do so by avoiding the complete transformation of the MTC to the MSC and directly evaluating the objective function, thus eliminating the need to *pre*-compute the identification matrix $\mathcal{M}^t(\mathcal{L}^t, \mathcal{S})$.

In the standard greedy heuristic for the test cover problem, in each iteration, a sensor that covers the most pairwise link failures from a total of $\binom{n}{2}$ pairwise failures, is selected. Thus $\mathcal{O}\left(\binom{n}{2}\right)$ comparisons are made by each sensor in a single iteration. We aim to avoid this by significantly reducing the number of comparisons made in each step. In fact, in our algorithm, the number of comparisons made for each sensor in a single iteration are 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.

In this context, it is observed that a sensor i that detects k link failures (i.e., $|C_i| = k$), detects $k(n-k)$ pairwise link failures (i.e., $|C^t_i| = k(n-k)$), 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. Fig. 3 shows the pairwise detection as a function of sensor detection capability for a single sensor. We can observe that the maximum number of pairwise link failure detections are realized when $k = n/2$.

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

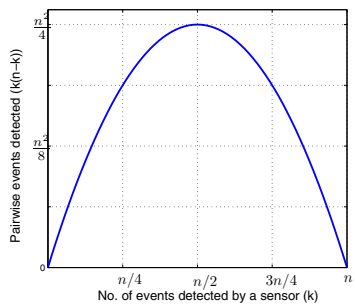


Fig. 3. Utility of a single sensor for the identification problem as a function of its detection capability.

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 this observation, we suggest an efficient greedy approach to compute the test cover without computing the $\binom{n}{2}$ event pairs apriori.

Restating our notations, $\mathcal{L}$ is the set of link failures, $\mathcal{S}$ is the set of available sensors, $C_u \subseteq \mathcal{L}$ is the set of link failures detected by the sensor S_u, and $\mathcal{C}$ is the collection of all such C_u's. Moreover, let $C^* \subseteq \mathcal{C}$ 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., $\mathcal{C}_{cov} = \bigcup_{C_u \in C^*} C_u$. Thus, in the light of previous discussion, the utility of adding C_i to C^* (i.e., adding sensor S_i to the test cover) is based on the two factors:

- How many pairwise link failures corresponding to the links not in $\mathcal{C}_{cov}$ can be detected by C_i? We call this value as x_i.
- How many undetected pairwise link failures corresponding to the links in $\mathcal{C}_{cov}$ can be detected by C_i? We call this value as y_i.

The overall utility of adding sensor S_i to the test cover, denoted by w_i, is the sum of x_i and y_i. A sensor i^* that maximizes this overall utility, say w_{i^*}, will then be included in the test cover, and $\mathcal{C}_{cov}$ will be updated to $\mathcal{C}_{cov} \leftarrow \mathcal{C}_{cov} \cup C_{i^*}$. Now, we will see how to compute x_i and y_i in a single iteration j.

Computing 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 $x_i = k_i(n_j - k_i)$. Note that computing x_i is very straight forward and does not require computing pair-wise link failures from a given set of link failures.

Computing 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. To make it precise, we proceed as follows:

If X and Y are two sets, then 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 $\alpha(Y, \beta(X))$ 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

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

The complete algorithm with all the steps is stated below.

Algorithm 2 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

Example 4 *Consider the network shown in Fig. 1. In the first iteration, the event space is $n = 10$ and the detection capability of the sensors is $k = \{5, 5, 7, 9, 7, 6, 7, 6\}$, where $k_1 = 5$, i.e. sensor S_1 detects 5 failure events obtained from $C_1 \subseteq \mathcal{C}$. Then the new number of pairwise link detection for each sensor is $x_i = k_i(n - k_i)$. Since*

there are no sensors in the test cover set yet, $y_i = 0$ for all the sensors. The maximum w_{i^*} is attained for the sensors S_1, S_2 with $k_1 = k_2 = n/2$ and $x_1 = x_2 = 5(10-5) = 25$. We select S_1 to the test cover. We update the set of all undetected pairwise events, G_1, for sensor S_1. Finally, we update the number of covered events $\mathcal{C}_{cov} = \{1, 2, 3, 4, 5\}$ and the event space $n_2 = 5$.

A complete account of the states of variables of the algorithm for the example 2 is given in Table C.1 in the Appendix. The algorithm returns the test cover consisting of sensors $\{1, 2, 3, 5\}$ (in that order) that uniquely identifies all link failures.

Note that the Algorithm 2 is basically an efficient implementation of the greedy approach for the test cover problem. In fact, it produces the same solution as the straightforward greedy approach, which implies that if the greedy solution is unique, then the solution obtained by the Algorithm 2 and standard greedy are identical. Thus, Algorithm 2 has the same approximation ratio as the standard greedy algorithm, which has been proven to be the best possible.

As a result of avoiding a large number of computations, Algorithm 2 is more efficient than the simple greedy. In contrast to the $\mathcal{O}\left(\binom{n}{2}\right)$ comparisons performed for a sensor in each iteration in a simple greedy, $\mathcal{O}\left(\sum_i^{m_j}\binom{k_i}{2}\right)$ comparisons are done by a sensor in Algorithm 2 in each iteration. Here, n is the total number of link failures, k_i is the number of link failures detected by the sensor i, and m_j is the number of sensors included in the test cover until that iteration. Thus, if $k = \max(k_i)$, then Algorithm 2 is at least n/k times faster than the simple greedy approach as shown below. Moreover, typically $k << n$ in the case of link failure detection in water distribution networks, thus n/k factor turns out to be a significant improvement.

Proposition 4.1 *Let $\sum_i k_i = n$, and $k = \max(k_i)$, then*

$$\sum_i \binom{k_i}{2} \leq \frac{k}{n}\binom{n}{2} \tag{13}$$

Proof –

$$\sum_i \binom{k_i}{2} = \frac{1}{2}\left(\sum_i k_i^2 - \sum_i k_i\right) \leq \frac{1}{2}\left(k\sum_i k_i - n\right)$$
$$= \frac{1}{2}(kn - n) \leq \frac{1}{2}(kn - k) = \frac{k}{n}\binom{n}{2}.$$

□

We mention here that Algorithm 2 is somewhat similar to the two-step greedy algorithm presented in [4]. However, in our approach, both x_i and y_i are computed in the same iteration resulting in a more efficient implementation.

5 Applications

We apply our approach on two real water systems [20,9]. For 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, as in [5], for the influence model assuming that the disturbance in pressure can be sensed while within a specified distance from the location of the burst, $\mathbf{y}_{S_i}(\ell) = \{1 \mid d(S_i, \ell) \leq \varepsilon\}$, where d is the length of the shortest path between two locations S_i and ℓ and ε is some threshold. The distance based approach emulates the intensity of the burst, i.e., larger bursts create transients with higher amplitude that travel to larger distances in the network before dissipating. Simulation and practical applications have shown that transients can travel long distances $(2-3[km])$ in transmission mains and shorter $(< 2[km])$ in the distribution network. (4) Solve the MTC problem by transforming the MTC to the counterpart MSC and solving using the greedy approach, as described in section 3.2.

5.1 Network 1

We first test our approach on a medium-size infrastructure network. Network 1 is a benchmark system that has been previously extensively studied in the context of sensor placement for water quality [20]. The system consists of 126 nodes, 168 pipes, one reservoir, one pump, and two storage tanks and its layout is shown in Fig. 4. The system supplies a daily demand of $5.15 \times 10^3[m^3/day]$ and has a total pipe length of $37.5 \times 10^3[m]$ as listed in Table 1.

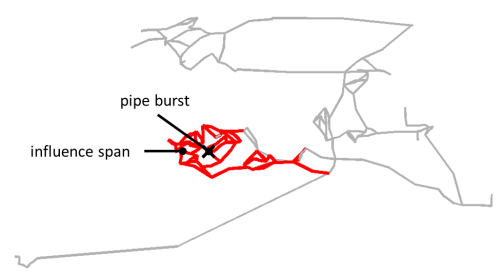


Fig. 4. Layout of Network 1 and influence span of failure in LINK-126

Assuming that a sensor can be placed at any of the 126 network nodes and any of the 168 network pipes can fail, we formulate and solve the MTC problem, as described previously in sections 2.4, 3.2, given a threshold distance of $\varepsilon = 1000[m]$. Fig. 4 shows an example of the influence range (in red) of a burst in LINK-126 of the network for a threshold distance of $\varepsilon = 1000[m]$, i.e., a sensor located in the red region can detect the pipe failure.

Table 1
Data for water systems

System	Pipe length $[km]$	Demand $10^3[m^3/day]$	No. of events	No. of sensors
Network 1	37.5	5.15	168	126
Network 2	260.3	5.68	1156	959

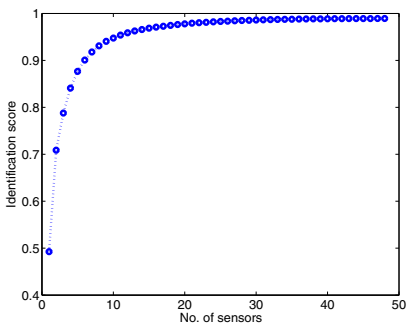


Fig. 5. Identification score for Network 1

Figure 5 shows the normalized identification score as a function of the number of sensors using the greedy approach. The normalized identification score, $I_I(S;\mathcal{L})$, is computed by dividing $f_I(S;\mathcal{L})$ by the total number of pairwise events, $|\mathcal{L}^t| = \binom{168}{2} = 14028$. We can observe that, as shown in section 3, the identification score function exhibits a diminishing return property, which results in a concave performance function. The maximum identification score of 0.99 is attained with 48 sensors.

Observing the identification score of the network is not sufficient to evaluate the quality of the design, hence, we suggest two complementary metrics for evaluating the performance of the sensor network design – localization score and localization-set-size.

(1) *Localization score* – is defined as the number of localization sets, where each localization set is associated with a unique sensors states. Let $L \subseteq \mathcal{L}$ be a localization set under the sensor configuration S if $\forall \ell_i \in L$ the outputs of sensors in S remain the same. It simply means that it is not possible to distinguish between the failure events in a localization set by merely observing the outputs of sensors. The normalized localization score, denoted by $I_L(S;\mathcal{L})$, is then the ratio of the total number of localization sets formed under the sensor configuration S to the total number of event failures. This number indicates how well the location of the fault is classified into one of the network pipes and can be viewed as the sensitivity of the design. Ideally, the normalized localization score should be equal to 1, indicating that each fault can be uniquely identified.

To demonstrate, in example 3, given a two-sensor design $S_A = \{S_2, S_4\}$, three localization sets are formed, i.e. $L_1 = \{\ell_1\}, L_2 = \{\ell_4, \ell_5, \ell_7, \ell_9, \ell_{10}\}, L_3 = \{\ell_2, \ell_3, \ell_6, \ell_8\}$. The normalized localization score is thus $I_L(S;\mathcal{L}) = 3/10$.

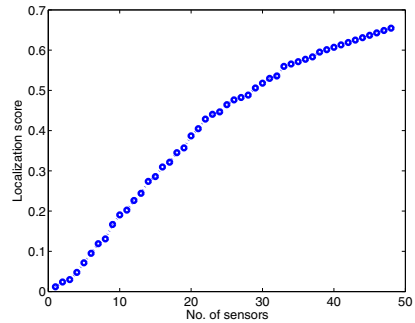


Fig. 6. Localization score for Network 1

Ideally the localization score would be 1, meaning that each pipe is uniquely identified, which is achieved with $S_B = \{S_1, S_2, S_3, S_5\}$, where we observe ten unique sensor states associated with each failure event.

Figure 6 shows the normalized localization score as a function of the number of sensors. The highest localization score of 0.65 is achieved when 48 sensors are installed. This results indicates that 110 unique states are associated with the 168 failure events. The localization score does not exhibit submodular property as the identification score in Figure 5.

(2) *Localization set size* is the number of faults associated with each unique sensor state. We characterize the localization performance of the sensor design by quantifying the minimum, median, and maximum of the localization sets for a given set of sensors, S. The worst localization set size, $I_W(S;\mathcal{L})$, i.e. the maximum, implies about the specificity of the design and quantifies the uncertainty in the location of the fault. The higher the size of the localization set the harder it will be to identify the location of the fault and might require additional local inspection methods. Ideally, the worst localization set size would be 1, indicating, again, that all faults are uniquely localized.

Figure 7 demonstrates the worst, median, and minimum localization set sizes as a function of the number of sensors for Network 1. We can observe that initially localization-sets contain large number of pipes that rapidly decrease with the number of sensors, until the worst localization-set-size reaches a plateau at 20 sensors and does not improve further. This implies that adding more sensors, might improve local performance, but will not improve the overall network localization performance, making additional sensors not cost effective for the water utility.

5.2 Network 2

We test our approach on a larger system – *Network 2*, that was originally collected by the Kentucky Infrastructure Authority. The network consists of 1,156 pipes, 959 nodes, two pumps, four tanks and one reservoir. Network layout and main features are shown in Fig. 8 and Ta-

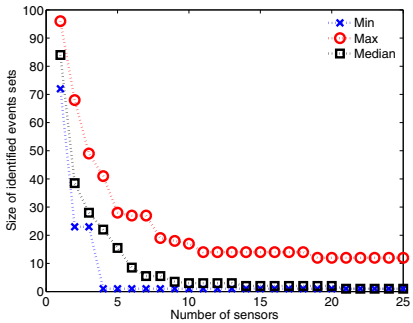


Fig. 7. Localization-set-size for Network 1

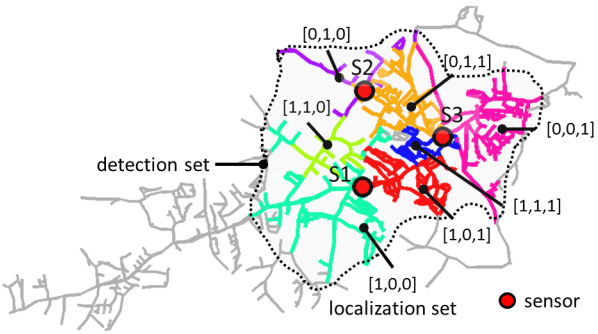


Fig. 8. Layout of Network 2 and example of the detection and localization sets of three sensors placed in the network

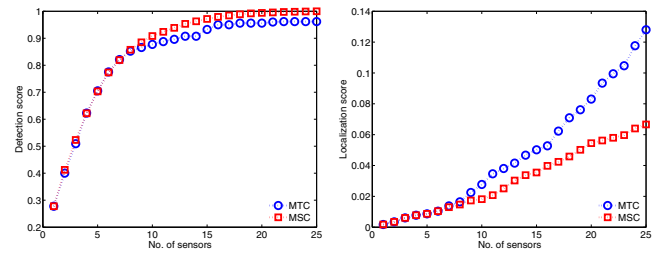
ble 1 and full data and hydraulic model can be obtained from [9]. Although the system supplies similar daily demand as Network 1, it is spatially more distributed with approximately 260 $[km]$ of pipes.

We first formulate and solve the MTC problem using the greedy algorithm assuming that faults can occur at any network pipe and sensors can be placed in any node of the network, and setting the distance threshold to $\varepsilon = 2000[m]$. The optimal identification score of 0.99 is attained with 261 sensors installed in the network. The localization score is then 0.91.

5.2.1 *MSC vs. MTC*

Next, we compare optimal solutions computed for the identification problem versus the optimal solutions computed for detection problem. As described previously in sections 2.3 and 3.1, the fault detection problem can be stated as a minimum *set* cover problem. We next compare the design attained from solving the MTC and MSC problems.

Fig. 8 schematically illustrates a difference between the MTC and MSC problem formulations in the context of Network 2. Consider three sensors installed in the network, Fig. 8 demonstrates the seven localization sets corresponding to seven unique sensor states, $[0,0,1], \cdots, [1,1,1]$ and the detection set, being the union of the localization sets. Whereas, the detection problem tries to maximize the detection set, the identification problem aims to identify unique subsets.



(a) Detection (b) Localization

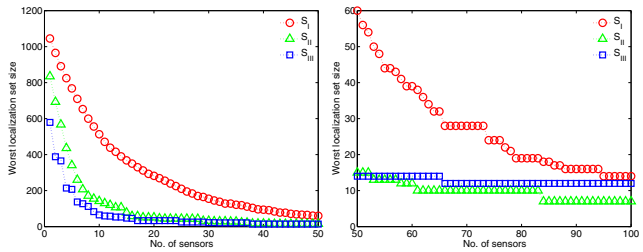
Fig. 9. MTC versus MSC performance for Network 2

Fig. 9 demonstrates the comparison between the detection and localization scores for the MTC (blue) and MSC (red) designs. For the detection problem, 25 sensors are sufficient to cover the entire system, hence, we select the first 25 sensors for the identification problem and compare their performance. From Fig. 9a it can be seen that the two designs overlap for the first 7 sensors and the MSC design only slightly outperforms the MTC design for higher number of sensors. Whereas the MTC design significantly outperforms the MSC design when comparing localization scores, as shown in Fig. 9b.

5.2.2 *Variable distance thresholds*

Next, we vary the sensing thresholds in the influence model and investigate the resulting designs, with S_I, S_{II}, S_{III} corresponding to $\varepsilon = 1000, 2000$, and $3000[m]$. The theoretical number of sensors that need to be installed for maximum identification performance is 359, 261, and 237, respectively, with corresponding localization score of $I_L(S_I;\varepsilon_1) = 0.87, I_L(S_{II};\varepsilon_2) = 0.91, I_L(S_{III};\varepsilon_3) = 0.91$. The theoretical number of sensors for maximum localization score is extremely high, requiring to install sensors on 24-37% of network nodes, which is not practical for the water utilities in terms of capital and maintenance costs. Since, installing such large number of senors is impractical, we look at the number of sensors required for relative performance based on the detection, localization score and set size measures.

First, we show the worst set size performance, I_W, of the three designs. Fig. 10 demonstrates the worst localization set size as a function of the number of sensors (a) 1-50 and (b) 50-100 for the three thresholds. The results demonstrate that the reduction in the size of the worst set is generally greater for events with larger influence distances, although at the final solution, the large distance design, S_{III}, is slightly less sensitive than the medium distance design, S_{II}. For example, given 20 sensors, the largest number of pipes in all localization sets will be 25 for S_{III}, 30 for S_{II} and 350 for S_I. This implies that in order to detect weaker failure events, e.g.,



(a) No. of sensors: 1 - 50 (b) No. of sensors: 50 - 100

Fig. 10. Worst localization-set-size for Network 2 for: 1000 (red), 2000(green), and 3000(blue) influence distance $[m]$.

smaller pipe burst, more sensors need to be installed, compared to detecting larger pipe bursts. We can observe that around 100 sensors, all three models converge to a similar localization-set-size.

The results for all performance metrics for all simulations are summarized in Table 2. For each of the designs, S_I, S_{II}, S_{III}, Table 2 lists the number of sensors required to (1) to detect 90 and 95 % of all faults, i.e. $I_D = \{0.9, 0.95\}$, (2) to identify 50 and 75% unique sensor states, i.e., localization score $I_L = \{0.5, 0.75\}$, and (3) to achieve a worst localization set size of 30 and 20 pipes. The results demonstrate that:

(1) The number of sensors required solely for detection purpose is significantly lower than the number of sensors required for localization, for both performance metrics. For example, consider the design S_I (first row in Table 2), then to detect 90% of the events, 37 sensors is sufficient, whereas to achieve $I_L = 0.5$ we require 137 sensors, and 66 to achieve $I_W = 66$.

(2) Between the two localization measures, I_L and I_W, the localization score is more conservative than the worst set size, resulting in much higher number of sensor. As above, for the design S_I, installing 137 sensors achieves $I_L = 0.5$, i.e. 50% of unique sensor states. Whereas with 66 sensors, the worst size of the corresponding localization sets would consist of 30 pipes. This is observed for all tested designs. Combining the two observations, for this application we have $|S(I_L, \varepsilon)| \geq |S(I_W, \varepsilon)| \geq |S(I_D, \varepsilon)|$ for any given ε.

(3) More conservative designs, assuming lower distance threshold, requires larger number of sensors to achieve the same performance for all criteria, i.e. for $\varepsilon_1 < \varepsilon_2 < \varepsilon_3 \rightarrow |S_I(I, \varepsilon_1)| \geq |S_{II}(I, \varepsilon_2)| \geq |S_{III}(I, \varepsilon_3)|$ for any given I.

6 Conclusions and future work

In this work we focus on the sensor placement for fault location identification in water networks. Our main contributions are three folded, we: (1) consider pipe bursts as failure events, (2) cast the identification problem as the test cover problem, and (3) propose four metrics to evaluate the performance of the sensor network design. Additionally, we suggest a fast greedy approach for solving the minimum test cover problem and analyze the performance of the common detection versus identification designs. The outcome of our approach provides a better diagnosis of failure events in terms of improved localization and response to failure events in operational mode. Future work will consider more sophisticated event and sensing models and robustness to sensor failures.

Table 2
Performance scores for Network 2

Design	No. of sensors					
	I_D		I_L		I_W	
	0.9	0.95	0.5	0.75	30	20
S_I	37	51	137	241	66	79
S_{II}	13	18	79	147	31	38
S_{III}	8	11	62	120	25	38

Acknowledgements

This work was supported in part by FORCES (Foundations Of Resilient CybEr-Physical Systems), which receives support from the National Science Foundation (NSF award numbers CNS-1238959, CNS-1238962, CNS-1239054, CNS-1239166) and the AFRL LABLET - Science of Secure and Resilient Cyber-Physical Systems (Contract ID: FA8750-14-2-0180, SUB 2784-018400).

References

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

[2] Visenti. `http://www.visenti.com/`. Accessed: 2015-04-14.

[3] 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.

[4] Koen MJ De Bontridder, Bjarni V Halldórsson, Magnús M Halldórsson, Cor AJ Hurkens, Jan Karel Lenstra, R Ravi, and Leen Stougie. Approximation algorithms for the test cover problem. *Mathematical Programming*, 98(1-3):477–491, 2003.

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

[6] DG Eliades, TP Lambrou, CG Panayiotou, and MM Polycarpou. Contamination event detection in water distribution systems using a model-based approach. *Procedia Engineering*, 89:1089–1096, 2014.

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

[8] Bjarni V. Halldórsson, Magnús 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.

[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] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne Vanbriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

[11] Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous optimization of sensor placements and balanced schedules. *IEEE Transactions on Automatic Control*, 56(10):2390–2405, 2011.

[12] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008.

[13] Zheng Liu and Y. Kleiner. Computational intelligence for urban infrastructure condition assessment: Water transmission and distribution systems. *IEEE Sensors Journal*, 14(12):4122–4133, Dec 2014.

[14] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[15] R Garey Michael and S Johnson David. Computers and intractability: a guide to the theory of np-completeness. *WH Freeman & Co., San Francisco*, 1979.

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

[17] Dalius Misiunas. *Failure monitoring and asset condition assessment in water supply systems.* PhD thesis, LUND University, Sweden, 2005.

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

[19] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical Programming*, 14(1):265–294, 1978.

[20] Avi Ostfeld, James G. Uber, Elad Salomons, Jonathan W. Berry, William E. Hart, Cindy A. Phillips, Jean-Paul Watson, Gianluca Dorini, Philip Jonkergouw, Zoran Kapelan, Francesco di Pierro, Soon-Thiam Khu, Dragan Savic, Demetrios Eliades, Marios Polycarpou, Santosh R. Ghimire, Brian D. Barkdoll, Roberto Gueli, Jinhui J. Huang, Edward A. McBean, William James, Andreas Krause, Jure Leskovec, Shannon Isovitsch, Jianhua Xu, Carlos Guestrin, Jeanne VanBriesen, Mitchell Small, Paul Fischbeck, Ami Preis, Marco Propato, Olivier Piller, Gary B. Trachtman, Zheng Yi Wu, and Tom Walski. The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms. *Journal of Water Resources Planning and Management-asce*, 134, 2008.

[21] A. Pandharipande, F. Calabrese, H.B. Lim, and R. Rajagopal. Guest editorial special issue on sensing technologies for intelligent urban infrastructures. *IEEE Sensors Journal*, 14(12):4121–4121, Dec 2014.

[22] Ramon Perez, Gerard Sanz, Vicenc Puig, Joseba Quevedo, M Cuguero Escofet, F Nejjari, Jordi Meseguer, G Cembrano, J Mirats Tur, and R Sarrate. Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in barcelona [applications of control]. *IEEE Control Systems*, 34(4):24–36, 2014.

[23] 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.

[24] José Ragot and Didier Maquin. Fault measurement detection in an urban water supply network. *Journal of Process Control*, 16(9):887–902, 2006.

[25] Albert Rosich, Erik Frisk, Jan Aslund, Ramon Sarrate, and Fatiha Nejjari. Fault diagnosis based on causal computations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 42(2):371–381, 2012.

[26] R. Sarrate, F. Nejjari, and A. Rosich. Sensor placement for fault diagnosis performance maximization in distribution networks. In *Control Automation (MED), 2012 20th Mediterranean Conference on*, pages 110–115, July 2012.

[27] E. Todini and L. Rossman. Unified framework for deriving simultaneous equation algorithms for water distribution networks. *Journal of Hydraulic Engineering*, 139(5):511–526, 2013.

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

A Transient modeling

Unsteady state flow in a closed conduit can be described by mass and momentum equations formulated as [28]:

$$\frac{\partial h}{\partial t} + \frac{a^2}{gA}\frac{\partial q}{\partial x} = 0 \tag{A.1}$$

$$\frac{1}{gA}\frac{\partial q}{\partial t} + \frac{\partial h}{\partial x} + \frac{cq|q|}{2gDA^2} = 0 \tag{A.2}$$

where h is the hydraulic head $[m]$, q is the volumetric flow rate $[\frac{m^3}{sec}]$, g is the gravitational acceleration $[\frac{m}{sec^2}]$, x distance along the pipe $[m]$, t is the time $[sec]$, a is the wave speed in the conduit $[\frac{m}{sec}]$, c is a friction factor, D is the pipe diameter $[m]$, and A is the pipe cross sectional area $[m^2]$.

The method of characteristics (MOC) is one of the most common numerical techniques used to approximate the solution of the hydraulic transient. Additional techniques used are finite differences and node characteristic method. A detailed derivation of the governing equations and the solution scheme can be found in [28,17]. The MOC transforms partial differential equations into ordinary differential equations that apply along specific lines (characteristics), C^+ and C^-, in the *space-time*, x-t, plane. Two characteristic equations are solved explicitly to compute the head and flow, h_*, q_*, at new point in time and space, $(\cdot)_*$, given that the conditions at a previous time step along the characteristic grid are known, i.e., h_+, q_+ and h_-, q_-. For a given pipe, the

two comparability equations are formulated as:

$$C^+ : \frac{a}{gA}\left(q_* - q_+\right) + \left(h_* - h_+\right) + \frac{c\Delta x}{2gDA^2} q_+|q_+| = 0 \tag{A.3a}$$

$$C^- : \frac{a}{gA}\left(q_* - q_-\right) - \left(h_* - h_-\right) + \frac{c\Delta x}{2gDA^2} q_-|q_-| = 0 \tag{A.3b}$$

Rearranging equations (A.3a) and (A.3b) we get:

$$C^+ : h_* = C_P - bq_* \tag{A.4a}$$

$$C^- : h_* = C_M + bq_* \tag{A.4b}$$

where

$$C^+ : \; C_P = h_+ + q_+\left(b - r|q_+|\right) \tag{A.5a}$$

$$C^- : \; C_M = h_- - q_-\left(b - r|q_-|\right) \tag{A.5b}$$

and

$$b = \frac{a}{gA} \tag{A.6}$$

$$r = \frac{c\Delta x}{2gDA^2} \tag{A.7}$$

b is a function of the physical characteristics of the pipe and the wave speed of the fluid in the conduit. The parameter b that can be viewed as the characteristic impedance, which is associated with the transient state. r is a function of the physical characteristics of the pipe, that can be viewed as pipe's resistance coefficient, and is associated with the steady state. If $b = 0$ the set of equations (A.4) is reduced to the steady state equations, where the head losses along the pipe occur only due to friction.

We designate the points $(\cdot)_+, (\cdot)_-, (\cdot)_*$ over a *space-time* grid of characteristics. If i and t are indices for space and time, respectively, then: $(\cdot)_* \rightarrow (h_{i,t+1}, q_{i,t+1}), (\cdot)_+ \rightarrow (h_{i-1,t}, q_{i-1,t}), (\cdot)_- \rightarrow (h_{i+1,t}, q_{i+1,t})$. Then solving first for $h_{i,t+1}$, by eliminating q_* in (A.4), for a single node in the numerical grid, we get:

$$h_{i,t+1} = \frac{1}{2}\big[h_{i-1,t} + h_{i+1,t} + b\left(q_{i-1,t} - q_{i+1,t}\right) + r\left(q_{i+1,t}|q_{i+1,t}| - q_{i-1,t}|q_{i-1,t}|\right)\big] \tag{A.8}$$

And in compact form:

$$h(t+1) = E_h h(t) + Bq(t) + RH\left(q(t)\right) \tag{A.9}$$

where h is a vector of hydraulic heads at the points of the grid of characteristics, E_h is a sparse matrix representing the connectivity of the grid, B is a sparse matrix whose elements represent the characteristic impedance b, R is a sparse matrix representing the resistance r, and H is the head loss function.

The flows $q(t+1)$ are then calculated directly from (A.4), as:

$$q(t+1) = B_h h(t+1) - E_q h(t) + B_q q(t) - R_q H\left(q(t)\right) \tag{A.10}$$

where the B_h, E_q, B_q, R_q are sparse matrices representing the grid of characteristics and the corresponding impedance and resistance coefficients.

At the boundaries specific conditions need to be defined describing the head-flow relation. Common boundary conditions, such as cross-connections and control valves, can be found in [28]. We give example for boundary condition introducing pipe burst at location ℓ using the orifice head-flow equation:

$$q_\ell = C_d A_\ell \sqrt{2gh_\ell} \tag{A.11}$$

where q_ℓ is the flow through the orifice, C_d is orifice discharge coefficient, and A_ℓ is the cross-section area of the orifice, and h_ℓ is the hydraulic head at the burst.

The burst event is simulated using a variable lumped burst coefficient $C_d A_\ell$. Before the burst occurs, the coefficient $C_d A_\ell$ is equal to zero, and it increases as a function of time to its finite value. Substituting the orifice, mass continuity, and compatibility equations, the hydraulic head at the burst can be formulated as:

$$h_\ell(t+1) + \frac{b_{\ell+1}b_{\ell-1}}{b_{\ell+1} + b_{\ell-1}} C_d A_\ell(t)\sqrt{2gh_\ell(t+1)} - \frac{C_M b_{\ell+1} + C_P b_{\ell-1}}{b_{\ell+1} + b_{\ell-1}} = 0 \tag{A.12}$$

where $b_{\ell+1}, b_{\ell-1}$ are the impedance coefficients up and down stream of the location of the burst ℓ. In a compact form, equation (A.12) can be rewritten as:

$$h_\ell(t+1) = P\left(C_d A_\ell(t)\right) \tag{A.13}$$

where P is a nonlinear function describing the burst boundary condition at location ℓ in terms of hydraulic head.

B Sensing pressure

Fig. B.1 shows a raw pressure signal recorded by Visenti [2] online sensor during a pipe burst event with $250[Hz]$ sampling frequency. Fig. B.1 shows the dynamic nature of the pressure, the sharp drop in the pressure during a pipe burst event, and the rapid return to normal operating range. The duration of the drop in the pressures occurs just under a few seconds, hence cannot be detected using a more traditional methods such as supervisory control and data acquisition (SCADA) systems, which typically operate in minutes scales.

Table C.1
Illustrative example demonstrating the steps in the fast greedy solution of the MTC problem

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$\mathcal{C}_{cov}$	$\emptyset$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5,6,8\}$	$\{1,2,\cdots,9\}$	$\{1,2,\cdots,10\}$
n_j	10	5	3	1	0
X_1, Y_1	$\{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5}\}$, $\emptyset$	–	–	–	–
X_2, Y_2	$\{1,2,3,6,8\}$, $\emptyset$	$\{\mathbf{6},\mathbf{8}\}, \{\mathbf{1},\mathbf{2},\mathbf{3}\}$	–	–	–
X_3, Y_3	$\{1,2,4,5,6,7,9\}$, $\emptyset$	$\{6,7,9\}, \{1,2,4,5\}$	$\{\mathbf{7},\mathbf{9}\}, \{\mathbf{1},\mathbf{2},\mathbf{4},\mathbf{5},\mathbf{6}\}$	–	–
$X_4, Y4$	$\{2,3,\cdots,10\}$, $\emptyset$	$\{6,\cdots,10\}, \{2,3,4,5\}$	$\{7,9,10\}, \{2,\cdots,6,8\}$	$\{10\}, \{2,3,\cdots,9\}$	$\emptyset$, $\{2,3,\cdots,10\}$
X_5, Y_5	$\{1,3,4,6,7,8,10\}$, $\emptyset$	$\{6,7,8,10\}, \{1,3,4\}$	$\{7,10\}, \{1,3,4,6,8\}$	$\{\mathbf{10}\}, \{\mathbf{1},\mathbf{3},\mathbf{4},\mathbf{6},\mathbf{7},\mathbf{8}\}$	–
X_6, Y_6	$\{2,4,5,7,9,10\}$, $\emptyset$	$\{7,9,10\}, \{2,4,5\}$	$\{7,9,10\}, \{2,4,5\}$	$\{10\}, \{2,4,5,7,9\}$	$\emptyset$, $\{2,4,5,7,9,10\}$
X_7, Y_7	$\{4,5,\cdots,10\}$, $\emptyset$	$\{6,\cdots,10\}, \{4,5\}$	$\{7,9,10\}, \{4,5,6,8\}$	$\{10\}, \{4,5,\cdots,9\}$	$\emptyset$, $\{4,5,\cdots,10\}$
X_8, Y_8	$\{3,6,\cdots,10\}$, $\emptyset$	$\{6,\cdots,10\}, \{3\}$	$\{7,9,10\}, \{3,6,8\}$	$\{10\}, \{3,6,7,8,9\}$	$\emptyset$, $\{3,6,\cdots,10\}$
$x_1,\ y_1$	$\mathbf{25},\mathbf{0}^*$	–	–	–	–
$x_2,\ y_2$	$25, 0$	$\mathbf{6},\mathbf{6}^*$	–	–	–
$x_3,\ y_3$	$21, 0$	$6, 4$	$\mathbf{2},\mathbf{3}^*$	–	–
$x_4,\ y_4$	$9, 0$	$4, 4$	$0, 2$	$0, 1$	$0, 0$
$x_5,\ y_5$	$21, 0$	$6, 6$	$2, 3$	$\mathbf{0},\mathbf{3}^*$	–
$x_6,\ y_6$	$24, 0$	$6, 6$	$0, 2$	$0, 1$	$0, 0$
$x_7,\ y_7$	$21, 0$	$6, 6$	$0, 0$	$0, 0$	$0, 0$
$x_8,\ y_8$	$24, 0$	$4, 4$	$0, 2$	$0, 0$	$0, 0$
G_0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
G_1	$\left\{\begin{array}{l}\{1,2\},\{1,3\},\{1,4\},\\ \{1,5\},\{2,3\},\{2,4\},\\ \{2,5\},\{3,4\},\{3,5\},\\ \{4,5\}\end{array}\right\}$	$\left\{\begin{array}{l}\{1,2\},\{1,3\},\{2,3\},\\ \{4,5\}\end{array}\right\}$	$\{\{1,2\},\{4,5\}\}$	$\emptyset$	$\emptyset$
G_2	–	$\{\{6,8\}\}$	$\emptyset$	$\emptyset$	$\emptyset$
G_3	–	–	$\{\{7,9\}\}$	$\emptyset$	$\emptyset$
G_4	–	–	–	$\{10\} \rightarrow \emptyset$	$\emptyset$

* is the selected sensor with the maximum utility, i.e. $w_{i^*} \leftarrow \max w_i$.

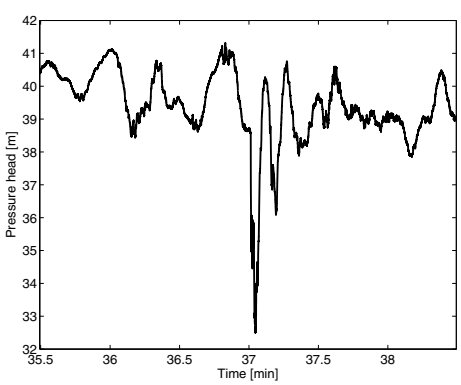


Fig. B.1. Pressure signal during a burst event recorded from online sensor installed in a water system

C Fast greedy MTC solution - illustrative example (cont.)

In each iteration, for every sensor S_i not in the test cover, C_i is decomposed into two sets namely, $X_i = C_i \setminus \mathcal{C}_{cov}$ and $Y_i = C_i \cap \mathcal{C}_{cov}$. The utility of including a sensor in the test cover is calculated in terms of x_i and y_i. x_i computes the number of pair-wise link failures detected by C_i corresponding to the links not in $\mathcal{C}_{cov}$, whereas y_i computes the undetected pair-wise link failures corresponding to the links in $\mathcal{C}_{cov}$ that can be detected by C_i. A complete account of variables is shown in Table B.1. At the end of the algorithm, sensors in $\{S_1, S_2, S_3, S_5\}$ are included in the test cover.