Simultaneous sensor and actuator placement for containment of contaminants in a water distribution network

Abstract

A multi objective integer linear optimization formulation for finding sensor and actuator placement strategies assuming perfect sensors and actuators is developed and implemented. The goal is to ensure observability of contamination while preventing contaminant from reaching demands.

Suhas Gundimeda, Shankar Narasimhan
Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai

1 Introduction

Water Distribution Networks (WDNs) are networks of pipes, tanks, pumps, valves, and other components that are used to distribute water from sources like reservoirs, to consumers. These networks are identified to be vulnerable to contamination[7, 8]. Suggested detection and response methods are varied, and one such is the placing of online sensors and actuators (valves) for detection and containment of contaminants. The sensor detects the contaminant, and the actuator shuts down the network, preventing further spread. The task then becomes the computing of optimal sensor and actuator placement strategies.

1.1 Previous work

Sensor placement using the principle that there must exist a unique non-zero set of sensors for each set of vulnerable nodes that can be affected was done by Palleti et all[1]. The work also augmented the graph abstraction with vulnerable nodes representing multiple real nodes and used sets of affected nodes to implement identifiable sensor placement.

Given sensor network, minimal actuator placement on edges to achieve shut down of network to contain contaminant was performed in [2]. The current work combines these methods in a multi-objective optimization formulation.

The approach in [4] attempts to optimize different objectives: maximizing detection likelihood and minimizing expected time to detection and solves it using genetic algorithms. The current work assumes both of these objectives are binary, i.e. no time delay in detection and accurate sensing.

We hypothesize that simultaneous sensor and actuator planning can be achieved and is more efficient than iterative non-linear optimization sequentially solving the two placement problems, i.e. these are not independent problems, and these can be formulated as an integer linear program and solved.

We first develop a formulation and algorithm for each case, implement a simulation in MATLAB, and compare with results from previous works to check veracity and improvement.

2 Problem formulation

Specifications of the water distribution network are provided as vulnerable nodes, demand nodes, and the sparse adjacency matrix. The demand nodes are the nodes we want to prevent the contamination of. Time-delay in sensors of contaminant sensing, lengths of pipes, accuracy of sensing, etc. can be added onto this work easily, and are not considered here.

2.1 Requirements to be satisfied

To find distribution of sensors on nodes and actuators on edges such that an attack can be identified and the network shutdown, preventing the contaminant from reaching the demand nodes.

3 Scenarios

Case 1: Shutting down the network effectively stops the contaminant beyond the actuator as well.

In this simpler case, there are no additional constraints on the actuator placement problem beyond the being a min-cut of the entire graph. The sensor and actuator placement problems are independent. As long as the sensor network can detect the contaminant before it reaches the demands and the actuation can happen immediately, the requirements are satisfied.

The network is assumed to be static, the hydraulic results from 00:00 hours are taken.

Formulation

A binary integer optimization problem is presented:

$$\begin{array}{rcc} & min & (\sum x_i + \sum z_i) \\ & sub & \\ \mathbf{A}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} & \leq & \mathbf{b} \\ \mathbf{A_{eq}}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} & = & \mathbf{b_{eq}} \end{array}$$

Where $\mathbf{x} \equiv [x_i]$ is 1 if there exists a sensor at i^{th}node, 0 otherwise.
$\mathbf{y} \equiv [y_i]$ is 1 if i^{th}node is in the demands side of the actuators, 0 otherwise.
$\mathbf{z} \equiv [z_i]$ is 1 if there exists an actuator at i^{th}edge, 0 otherwise.
Hence the vector $\begin{bmatrix} \mathbf{x} & y & z \end{bmatrix}$ is of length $2 * N + E$.
$A = \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix}$, $A = \begin{bmatrix} A_{eq1} & \mathbf{0} \\ \mathbf{0} & A_{eq2} \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, $b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \end{bmatrix}$.

A_1 is a matrix of vulnerable nodes vs. affected nodes, with $A_{1ij} = -1$ if i^{th} vulnerable node affects the j^{th} node in graph.

$$b_1 = \begin{bmatrix} . \\ . \\ . \\ -1 \\ . \\ . \\ . \end{bmatrix}$$

$A_{eq1} = b_{eq1} = \mathbf{O}$

$A_2 = \begin{bmatrix} J & I_{E \times E} \end{bmatrix}$ where J is the incidence matrix.

$b_2 = [\mathbf{O}]$. This links the partitioning variables $\mathbf{y}$ to the variables in the cost function, $\mathbf{z}$.

A_{eq2}, b_{eq2} provide bounds to force the decision variables to reality, i.e. the demand nodes being in the second partition and the vulnerable nodes being in the first.

Small network example

Consider the graph represented by Figure 1, with demand node 6 and vulnerable nodes 1 and 2.

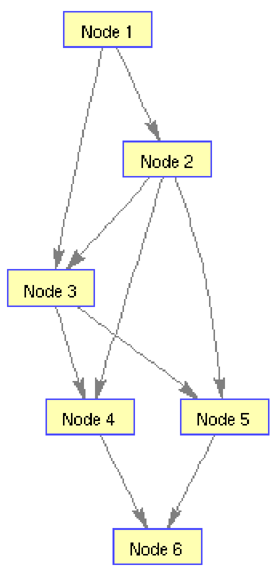

Figure 1: The example graph with 6 nodes and 9 edges

Formulating as binary integer optimization problem:

$$\begin{array}{ccc} & min & (\sum x_i + \sum z_i) \\ & sub & \\ \mathbf{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} & \leq & \mathbf{b} \end{array}$$

$$\mathbf{A_{eq}} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} \qquad = \qquad \mathbf{b_{eq}}$$

Hence the vector $\begin{bmatrix} \mathbf{x} & y & z \end{bmatrix}$ is of length $2 * N + E = 21$.

$A = \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix}$, $A = \begin{bmatrix} A_{eq1} & \mathbf{0} \\ \mathbf{0} & A_{eq2} \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, $b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \end{bmatrix}$.

A_1 is a matrix of vulnerable nodes vs. affected nodes, with $A_{1ij} = -1$ if i^{th} vulnerable node affects the j^{th} node in graph.

$$A_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$A_{eq1} = b_{eq1} = \mathbf{O}$

The following maps the actuator placement problem to one of classification into source or demand partition.

$$A_2 = \begin{bmatrix} J & I_{E \times E} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

where J is the incidence matrix.

$b_2 = [\mathbf{O}]$. This links the partitioning variables $\mathbf{y}$ to the variables in the cost function, $\mathbf{z}$. An edge between two different partitions causes the cost function to assume the presence of an edge.

A_{eq2}, b_{eq2} provide bounds to force the decision variables to reality, i.e. the demand nodes being in the second partition and the vulnerable nodes being in the first.

$$A_{eq2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \; b_{eq2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Results and verification

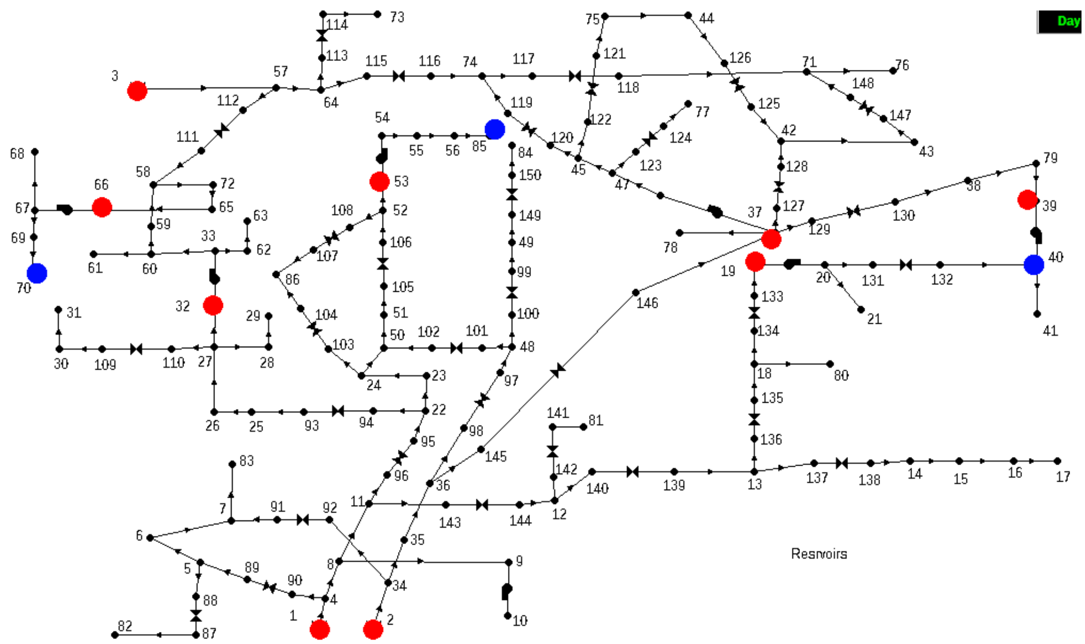

Figure 2: Graph abstraction with vulnerable nodes in red and sensors in blue.

Vulnerable Nodes	Sensor Nodes
1	41, 70, 85
2	41, 85
3	70
19	41
32	70
37	41
39	41
53	85
66	70

Table 1: Vulnerable nodes and their corresponding sensor nodes

It takes three sensors to detect contaminant with no other requirements.

This is the same as [1].

The closing of network requires 7 actuators: on edges 1, 12, 49, 50, 71, 80, 106.

Case 2: The contaminant must be contained in the vulnerable side of actuator network; attacks occur on all vulnernerable nodes at once

This case is not trivial as the positions of the sensors must be used as input, i.e. they must be on the vulnerable side of actuators.

Formulation

Formulating as binary integer optimization problem:

$$\begin{array}{ccc} & min \quad (\sum x_i + \sum z_i) & \\ & sub & \\ \mathbf{A}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ a \\ \mathbf{w} \end{bmatrix} & \leq & \mathbf{b} \\ \mathbf{A_{eq}}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ a \\ \mathbf{w} \end{bmatrix} & = & \mathbf{b_{eq}} \end{array}$$

Where $\mathbf{x} \equiv [x_i]$ is 1 if there exists a sensor at i^{th}node, 0 otherwise.
$\mathbf{y} \equiv [y_i]$ is 1 if i^{th}node is in the demands side of the actuators, 0 otherwise.
$\mathbf{z} \equiv [z_i]$ is 1 if there exists an actuator at i^{th}edge, 0 otherwise.
a is a decision variable which stores the maximum distance the set of sensors are to the set of vulnerable nodes.
$\mathbf{w} \equiv [w_i]$ is 1 if i^{th}node is in the demands side of the actuators, a large number(N) otherwise.
M is a large number.
Hence the vector $\begin{bmatrix} \mathbf{x} & y & z & a & w \end{bmatrix}$ is of length $3 * N + E + 1$.
$A = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_2 & \mathbf{0} \\ & A_3 & \end{bmatrix}$, $A = \begin{bmatrix} A_{eq1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{eq2} & \mathbf{0} \\ \mathbf{0} & A_{eq3} & \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$, $b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \\ b_{eq3} \end{bmatrix}$.
A_1 is a matrix of vulnerable nodes vs. affected nodes, with $A_{1ij} = -1$ if i^{th} vulnerable node affects the j^{th} node in graph.

$$b_1 = \begin{bmatrix} . \\ . \\ . \\ -1 \\ . \\ . \\ . \end{bmatrix}$$

$A_{eq1} = b_{eq1} = [\mathbf{O}]$
$A_2 = \begin{bmatrix} J & I_{E\times E} \end{bmatrix}$ where J is the incidence matrix.
$b_2 = [\mathbf{O}]$. This links the partitioning variables $\mathbf{y}$ to the variables in the cost function, $\mathbf{z}$.

A_{eq2}, b_{eq2} provide bounds to force the decision variables to reality, viz. the demand nodes being in the second partition and the vulnerable nodes being in the first.

We use shortest path lengths from all the vulnerable nodes (simulating an attack on all of them) to all the nodes in the graph to model "farther away in time". Let this vector be $\mathbf{S}$. The first set of constraints of A_3 is this vector of shortest paths multiplied with each column of $\mathbf{I} \times \mathbf{x}$, storing the maximum distance of that particular sensor configuration in a. The actuators need to be at least that much distance away from the set of vulnerable nodes.

This can be implemented in two ways: constraining the actuators to be placed after a particular distance, or constraining the demand partition to start after a particular distance. Since generally speaking $N < E$ in a lot of real world networks, we show the node implementation.

We transform the $\mathbf{y}$ vector to $\mathbf{w}$, where M denotes the sensor partition and 1 implies the demand partition. Then we construct constraints for actuator placement for the i^{th}node as follows: $\begin{bmatrix} \ldots & 0 & -(M+\mathbf{S}) & 0 & \ldots \end{bmatrix} \mathbf{w}_i + \begin{bmatrix} a + \frac{a}{M} \end{bmatrix} \leq \begin{bmatrix} -M \end{bmatrix}$. This completes the formulation.

Small network example

Consider the same network in Figure 1, with the two vulnerbale nodes (1, 2) and one demand node (6).

The additional constraints introduced in this case are $\mathbf{S}_i \mathbf{x}_i - a \leq 0$ for each node i. This becomes $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \mathbf{x} -$ $\begin{bmatrix} a \\ a \\ a \\ a \\ a \\ a \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ for this network. The constraints for ensuring desired partitioning are

$$\begin{bmatrix} -(M) & 0 & 0 & 0 & 0 & 0 \\ 0 & -(M) & 0 & 0 & 0 & 0 \\ 0 & 0 & -(M+1) & 0 & 0 & 0 \\ 0 & 0 & 0 & -(M+1) & 0 & 0 \\ 0 & 0 & 0 & 0 & -(M+1) & 0 \\ 0 & 0 & 0 & 0 & 0 & -(M+2) \end{bmatrix} \mathbf{w} + \begin{bmatrix} a + \frac{a}{M} \\ a + \frac{a}{M} \\ a + \frac{a}{M} \\ a + \frac{a}{M} \\ a + \frac{a}{M} \\ a + \frac{a}{M} \end{bmatrix} \leq \begin{bmatrix} -M \\ -M \\ -M \\ -M \\ -M \\ -M \end{bmatrix}$$

This ensures that $\mathbf{S}_i$ is greater than a for all nodes i.

Results and verification

The network is verified to have no paths from vulnerable to demand after removing edges.

The distance to detection in this design is 0 units.

Vulnerable Nodes	Time to sense	Sensor nodes	Actuator edges
1,2,3,19,32,37,39,53,66	0	19,39,53,66	1, 13, 27, 30, 38, 39, 40, 44, 51, 52, 68, 72, 80, 87, 90, 103, 150, 149
1,2,3	0	1,2,3	1,4,68
131,20	0	131	27, 30, 37

Table 2: The distance to detection for different sets of vulnerable nodes

In the previous work [2], the number of actuators suggested was 19 viz. 2, 12, 16, 24, 30, 38, 40, 48, 51, 52, 67, 71, 76, 86, 79, 89, 108, 110, 127 with 3 sensors (at nodes 40, 54, 67) and the same 9 vulnerable nodes.

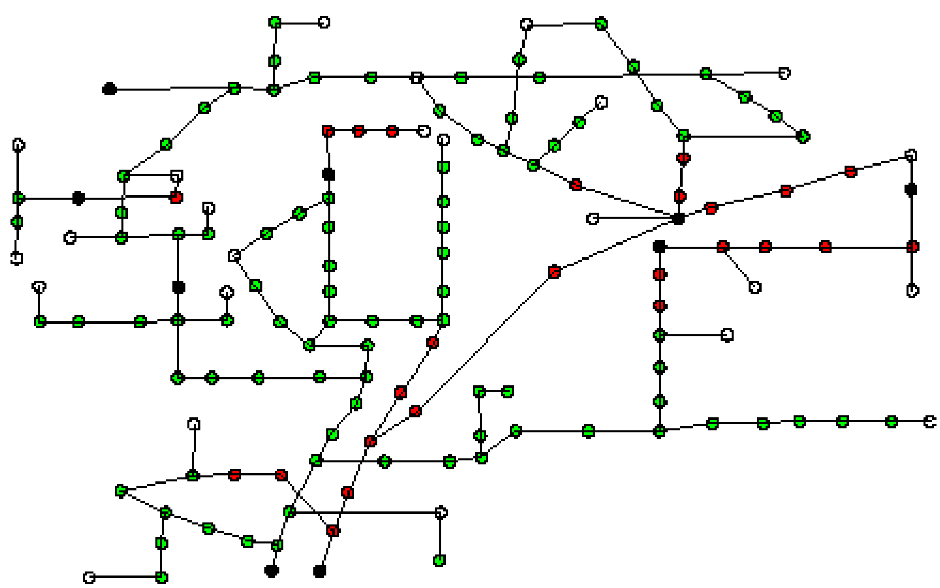

Figure 3: Network figure.
Green – Demand partition
Red – Vulnerable partition
Black – Vulnerable nodes
White – Demand nodes

Case 3: Vulnerable nodes might be attacked individually

Solution formulation

Formulating as mixed integer linear optimization problem:

$$\min \quad (\sum x_i + \sum z_i)$$
$$sub$$
$$\mathbf{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{z}\\ \mathbf{d}\\ \mathbf{w}\\ \mathbf{v}\end{bmatrix} \leq \mathbf{b}$$
$$\mathbf{A_{eq}}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{z}\\ \mathbf{d}\\ \mathbf{w}\\ \mathbf{v}\end{bmatrix} = \mathbf{b_{eq}}$$

Where $\mathbf{x} \equiv [x_i]$ is a decision variable vector whose components are 1 if there exists a sensor at i^{th}node, 0 otherwise.

$\mathbf{y} \equiv [y_i]$ is a decision variable vector whose components are 1 if i^{th}node is in the demands side of the actuators(demand partition), 0 otherwise (source partition).

$\mathbf{z} \equiv [z_i]$ is a decision variable vector whose components are 1 if there exists an actuator at i^{th}edge, 0 otherwise.

$\mathbf{d} \equiv [d_i]$ is a decision variable vector whose components store $M-$the "distance to detection" for a particular vulnerable node.

$\mathbf{w} \equiv [w_i]$ is 1 if i^{th}node is in the demand partition, a large number (M) otherwise.

$\mathbf{v} \equiv [v_{ji}]$ is 1 if the j^{th} vulnerable node's distance to detection is inequal to the distance to the i^{th} (sensor3) node.

M is a large number greater in order than the distances in the network.

Hence the decision vector $\begin{bmatrix} \mathbf{x} & y & z & a & w & \mathbf{v} \end{bmatrix}$ is of length $3*N+E+V+N*V$, where V is the number of vulnerable nodes, N is the number of nodes, E is the number of edges.

$$A = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_2 & \mathbf{0} \\ & A_3 & \end{bmatrix}, A_{eq} = \begin{bmatrix} A_{eq1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{eq2} & \mathbf{0} \\ & A_{eq3} & \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, b_{eq} = \begin{bmatrix} b_{eq1} \\ b_{eq2} \\ b_{eq3} \end{bmatrix}.$$

A_1 is a matrix of vulnerable nodes vs. affected nodes, with $A_{1ij} = -1$ if i^{th} vulnerable node affects the j^{th} node in graph.

$$b_1 = \begin{bmatrix} . \\ . \\ . \\ -1 \\ . \\ . \\ . \end{bmatrix}$$

$A_{eq1} = b_{eq1} = [\mathbf{O}]$

To link the partitioning variables $\mathbf{y}$ to the variables in the cost function, $\mathbf{z}$:

$A_2 = \begin{bmatrix} J & I_{E\times E} \end{bmatrix}$ where J is the incidence matrix.

$b_2 = [\mathbf{O}]$. The pesence of an edge from source to demand forces the corresponding $\mathbf{z}_i$ to switch to one, indicating the presence of an actuator on that edge.

A_{eq2}, b_{eq2} provide bounds to force the decision variables to reality, viz. the demand nodes necessarily being in the demand partition and the vulnerable nodes being in the source. We might also include constraints to consider a few steps after vulnerable node to be necessarily in the source partition, as done in [2].

We use the shortest distances from each of the vulnerable nodes to all other nodes in the graph to model time to contamination. We're assuming the actual times are linear functions of distance. Let this matrix be $\mathbf{D}$. The first set of constraints of A_3 is one for each entry in $\begin{matrix} M-\mathbf{D} \end{matrix}$ matrix, each of which is multiplied with $\mathbf{x}$. Each constraint is balanced by $-d_j$, where j is the vulnerable node under consideration in that constraint. $b_3 = [\mathbf{O}]$.

Now, d_j is greater than or equal to $M - D_{ij}$. We add constraints to force equality to the distance to the closest sensor: $D_{ij} + d_j - Mv_{ij} \le M$ for all sensors. More constraints for making sure non-sensor nodes don't use positive v_{ij}: $-x_i + v_i \le 0$. For ensuring at least one equality of D_{ij} and d_j, per j: $A_{1j}\mathbf{x} - A_j\mathbf{v}_j \le -1$.

The actuators need to be at least $M - d_j$ distance away from the each of the vulnerable nodes. This can be implemented in two ways: constraining the actuators to be placed after a particular distance, or constraining the demand partition to start after a particular distance. Since generally speaking $N < E$ in a lot of networks, we show the node implementation.

Transform the $\mathbf{y}$ vector to $\mathbf{w}$, where $w_i = M$ denotes the sensor partition and $w_i = 1$ implies the node is in demand partition. Construct constraints for actuator placement for the i^{th}node as follows: $\begin{bmatrix} ... & 0 & -(M + D_{ij} - 1) & 0 & ... \end{bmatrix} [\mathbf{w}] + \begin{bmatrix} -1 \end{bmatrix} d_j \le \begin{bmatrix} \\ -2*M \\ \\ \end{bmatrix}$. This ensures that when $\mathbf{w}_i$ is 1, the constraint is activated and only satisfied if the corresponding $\mathbf{D_{ij}} > \mathbf{d_j}$. When $\mathbf{w}_i$ is M, the constraint is always satisfied as $M^2$is several orders of magnitude greater than $2*M$.

Small network example

Consider the same network in Figure 1, with the two vulnerable nodes (1, 2) and one demand node (6).

The additional constraints over case 2 introduced in this case are:

$$\begin{bmatrix} M-0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M-1 & 0 & 0 & 0 & 0 \\ 0 & 0 & M-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & M-2 & 0 & 0 \\ 0 & 0 & 0 & 0 & M-2 & 0 \\ 0 & 0 & 0 & 0 & 0 & M-3 \end{bmatrix} \mathbf{x} - \begin{bmatrix} d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} M-M & 0 & 0 & 0 & 0 & 0 \\ 0 & M-0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & M-1 & 0 & 0 \\ 0 & 0 & 0 & 0 & M-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & M-2 \end{bmatrix} \mathbf{x} - \begin{bmatrix} d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This gets us variables storing $M - d_i$, the value after subtracting the minimum distance to sensor from the large number for each vulnerable node.

Constraints for ensuring the conditions around d_j's:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \end{bmatrix} - M \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \\ v_{16} \end{bmatrix} \leq M \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} M & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \end{bmatrix} - M \begin{bmatrix} v_{21} \\ v_{22} \\ v_{23} \\ v_{24} \\ v_{25} \\ v_{26} \end{bmatrix} \leq M \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

For v_i's are 1 only when i is a sensor: $-x_i + v_i \leq 0$. At least one equality constraint being satisfied: $\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{bmatrix} \leq \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

The constraints for ensuring desired partitioning are

$$\begin{bmatrix} -(M+0-1) & 0 & 0 & 0 & 0 & 0 \\ 0 & -(M+1-1) & 0 & 0 & 0 & 0 \\ 0 & 0 & -(M+1-1) & 0 & 0 & 0 \\ 0 & 0 & 0 & -(M+2-1) & 0 & 0 \\ 0 & 0 & 0 & 0 & -(M+2-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & -(M+3-1) \end{bmatrix} \mathbf{w} - \begin{bmatrix} d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \\ d_1 \end{bmatrix} \leq \begin{bmatrix} -2M \\ -2M \\ -2M \\ -2M \\ -2M \\ -2M \end{bmatrix}$$

$$-1 \times \begin{bmatrix} M+M-1 & 0 & 0 & 0 & 0 & 0 \\ 0 & M+0-1 & 0 & 0 & 0 & 0 \\ 0 & 0 & M+1-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & M+1-1 & 0 & 0 \\ 0 & 0 & 0 & 0 & M+1-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & M+2-1 \end{bmatrix} \mathbf{w} - \begin{bmatrix} d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \end{bmatrix} \leq \begin{bmatrix} -2M \\ -2M \\ -2M \\ -2M \\ -2M \\ -2M \end{bmatrix}$$

Results and verification

The network is verified to have no paths from vulnerable to demand after removing edges. The distance to detection is always lesser than the distance to the first actuators.

Vulnerable Nodes	Time to sense	Sensor nodes	Actuator edges
1 2 3 19 32 37 39 53 66	0 0 0	1 2 3 19 32 37 39 53 66	1 4 40 48 51 52 68 150 151 152 153 149
1 2 3	0 0 0	1,2,3	1,4,68
1 2 3 19 32 37 39 53 66[1]	0 4 1 0 1 0 0 1 0	1 7 19 33 37 39 54 57 66	2 8 12 30 38 40 48 51 52 67 69 77 78 89 103 108 110

Table 3: The distance to detection for different sets of vulnerable nodes

In the previous work [2], the similar task was partial shutdown of WDN. The number of actuators suggested was 20 viz. 2, 12, 16, 24, 30, 38, 40, 48, 51, 52, 67, 71, 76, 86, 79, 89, 108, 110, 127, 112 with 3 sensors (at nodes 40, 54, 67) and the same 9 vulnerable nodes. The nodes directly downstream of vulnerable are taken as source nodes. Here for task 4, we get a solution with 26 cardinality; 17 actuators, 9 sensors. The previous work does not require that no demand node be compromised though.

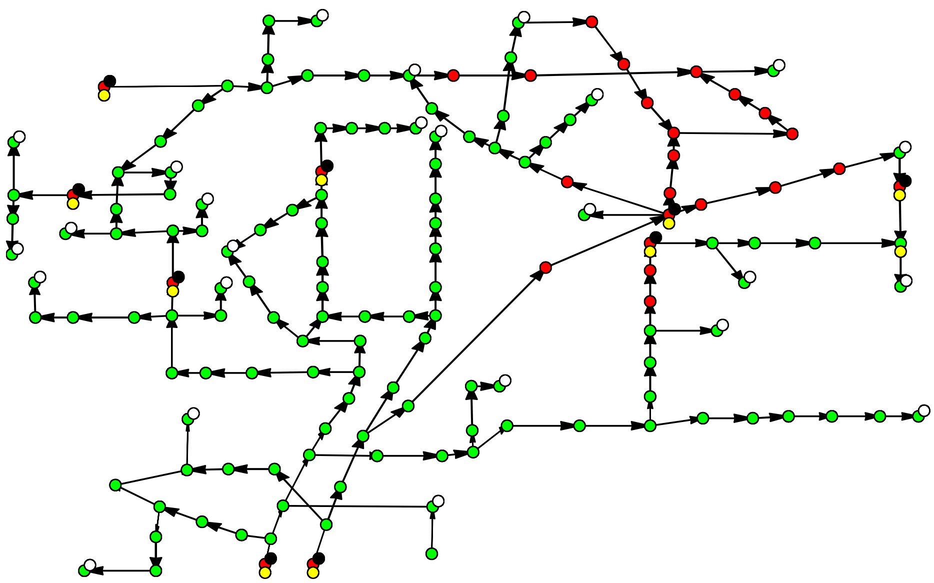

Figure 4: Sensor and actuator placement strategy for 9 vulnerable nodes
Green – Demand partition
Red – Vulnerable partition
Black – Vulnerable nodes
White – Demand nodes
Yellow – sensor nodes

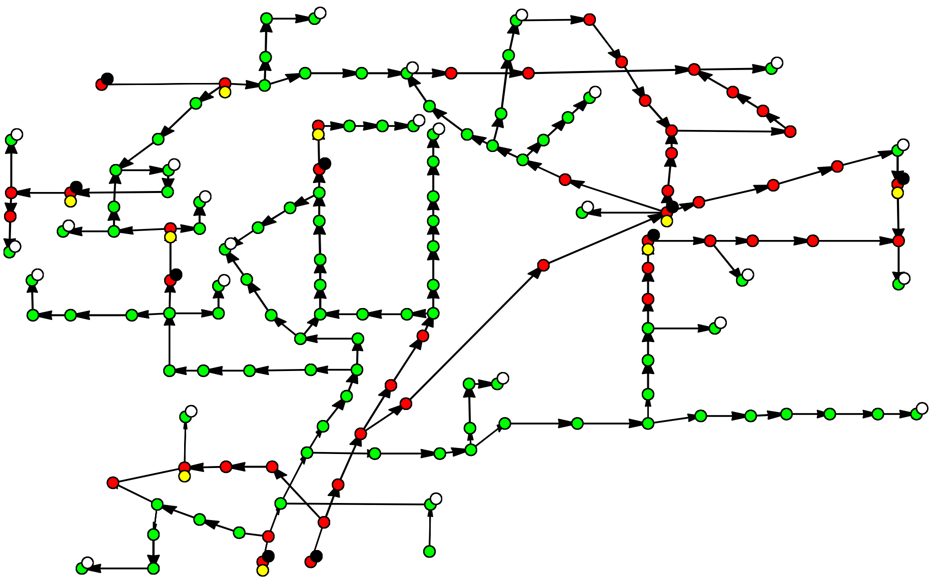

Figure 5: Sensor and actuator placement strategy for 9 vulnerable nodes and instant downstream contamination
Green – Demand partition
Red – Vulnerable partition
Black – Vulnerable nodes
White – Demand nodes
Yellow – sensor nodes

Conclusion and future work

References

[1] Palleti, V. R.; Narasimhan, S.; Rengaswamy, R.; Teja, R.; Bhallamudi, S. M. Sensor network design for contaminant detection and identification in water distribution networks. Computers and Chemical Engineering 2016, 87, 246 – 256.

[2] Venkata Reddy Palleti, Varghese, Shankar Narasimhan and Raghunathan Rengasamy: Actuator network design to mitigate contamination effects in water distribution networks

[3] On the structure of all min cuts in a network

[4] Optimization of Contaminant Sensor Placement in Water Distribution Networks: Multi-Objective Approach

[5] Review of Sensor Placement Strategies for Contamination Warning Systems in Drinking Water Distribution Systems 10.1061/ ASCE WR.1943-5452.0000081

[6] Sensor Placement Methods for Contamination Detection in Water Distribution Networks: A Review

[7] Kessler, A.; Ostfeld, A.; Sinai, G. Detecting Accidental Contaminations in Municipal Water Networks. Journal of Water Resources Planning and Management 1998, 124, 192–198.

[8] Ostfeld, A. et al. The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms. Journal of Water Resources Planning and Management 2008, 134, 20 556–568.