Simultaneous sensor and actuator placement for observation and 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
Indian Institute of Technology Madras

1 Scenario

1.1 Given

Specifications of water distribution network are provided as vulnerable nodes, demand nodes, and the sparse adjacency matrix. Time-delay in sensors of contaminant sensing, lengths of pipes, accuracy of sensing, etc. can be added onto this work easily, and are ignored.

1.2 Requirements to be satisfied

To find distribution of sensors on nodes and actuators on edges such that an attack can be identified in time and the contaminant can be prevented from reaching the demands.

2 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 to produce 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.

3 Hypotheses

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.

4 Method

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

5 Scenarios

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

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:

$$\begin{aligned} min \quad & \left(\sum x_i + \sum z_i\right) \\ sub \quad & \\ \mathbf{A}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{z}\end{bmatrix} \quad & \leq \quad \mathbf{b} \end{aligned}$$

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

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.

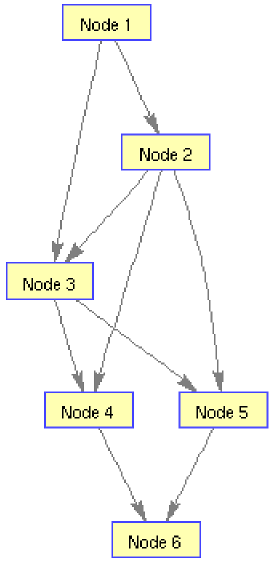

Figure 1: The example graph with 6 nodes and 9 edges

Formulating as binary integer optimization problem:

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

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.

$$A_2 = \left[\begin{array}{cc} J & I_{E\times E} \end{array}\right] = \left[\begin{array}{ccccccccccccccc} -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array}\right]$$

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.

$$A_{eq2} = \left[\begin{array}{ccccccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]; b_{eq2} = \left[\begin{array}{c} 0 \\ 0 \\ 1 \end{array}\right]$$

Results and verification

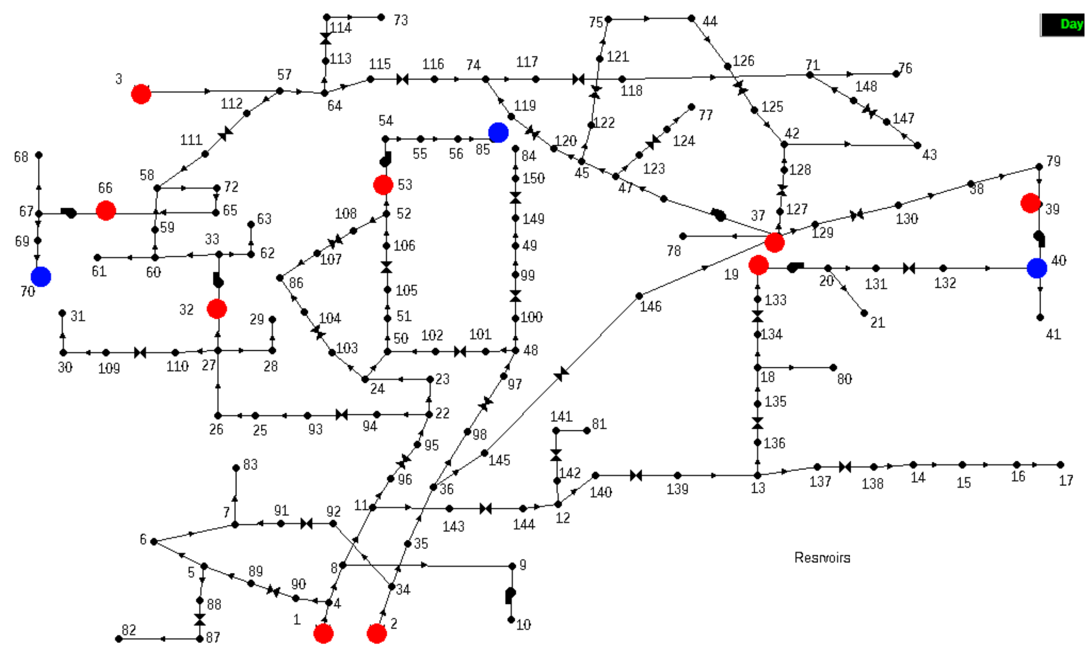

Figure 2: Graph abstraction with vulnerable nodes in red and sensors in blue.

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 contained in the vulnerable side of actuator network

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.

This formulation assumes in addition vulnerable nodes are always attacked simultaneously.

Formulation

Formulating as binary integer optimization problem:

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

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.

Γ 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 Γ denotes the sensor partition and 1 implies the demand partition. Then we construct constraints for actuator placement for the i^{th}node as follows: $\begin{bmatrix} ... & 0 & -(\Gamma + \mathbf{S}) & 0 & ... \end{bmatrix} \mathbf{w}_i + \left[a + \frac{a}{\Gamma}\right] \leq \begin{bmatrix} -\Gamma \end{bmatrix}$. 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

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

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, 9
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.

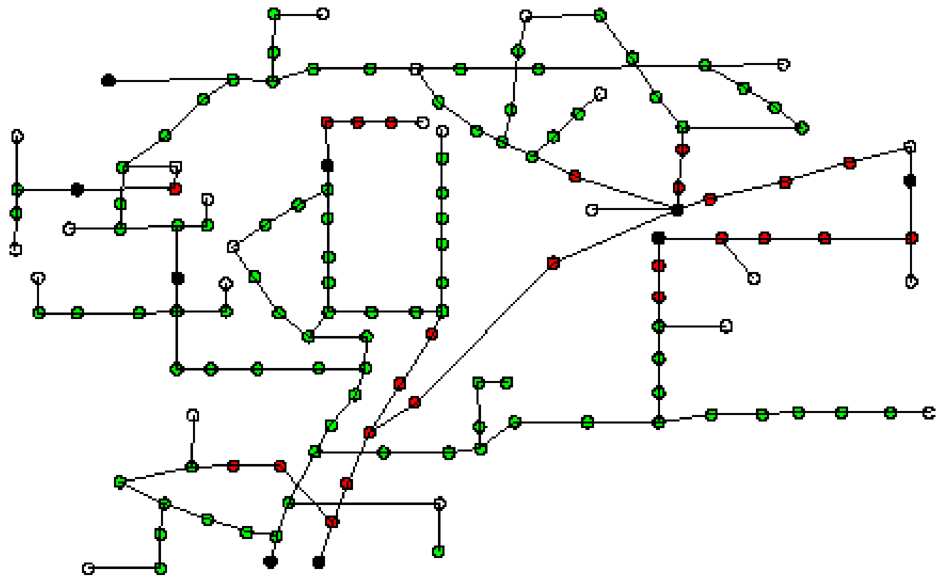

Figure 3: Network figure.
Green – Demand partition
Red – Vulnerable partition
Black – Vulnerable nodes
White – Demand nodes

Case 3: Vulnerable nodes can be attacked individually

Formulation

Formulating as binary integer optimization problem:

$$\begin{array}{ll} min & (\sum x_i + \sum z_i) \\ sub & \end{array}$$

$$\mathbf{A}\begin{bmatrix}\mathbf{x}\\\mathbf{y}\\\mathbf{z}\\\mathbf{d}\\\mathbf{w}\end{bmatrix} \leq \mathbf{b}$$

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

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, 0 otherwise.

$\mathbf{z} \equiv [z_i]$ 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 $\Gamma-$the distance to detection for a particular vulnerable node.

$\mathbf{w} \equiv [w_i]$ is 1 if i^{th}node is on the demands side of the actuators, a large number (Γ) otherwise.

Γ is a large number greater in order than the distances int he network.

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

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

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.

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

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

To links 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.

$b_2 = [\mathbf{O}]$.

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 the shortest distances from vulnerable nodes to all other nodes in the graph to model "distance in time". Let this matrix be $\mathbf{D}$. The first set of constraints of A_3 is a padded $\Gamma - \mathbf{D}$ matrix, which is multiplied with $\mathbf{x}_i$. This is balanced by a column of $-I_{V\times 1}$, which correspond to coefficients of $\mathbf{d}$. $b_3 = [\mathbf{O}]$. Thus, the $\mathbf{d}$ vector now stores $\Gamma - t$ where $t \in [0, u]$ and u is the minimum distance to sensor for each vulnerable node. The actuators need to be at least that much 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 real world networks, we show the node implementation.

Transform the $\mathbf{y}$ vector to $\mathbf{w}$, where $w_i = \Gamma$ denotes the sensor partition and $w_i = 1$ implies the demand partition.

Construct constraints for actuator placement for the i^{th}node as follows: $\begin{bmatrix} \ldots & 0 & -(\Gamma+\mathbf{D}) & 0 & \ldots \end{bmatrix} \times [\mathbf{w}] + \begin{bmatrix} -1-\frac{1}{\Gamma} \end{bmatrix} \times \mathbf{d} \leq \begin{bmatrix} \ldots \\ -2*\Gamma-1 \\ \ldots \end{bmatrix}$. 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 Γ, the constraint is always satisfied as $\Gamma^2$is several orders of magnitude greater than $-2*\Gamma-1$.

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

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

$$\begin{bmatrix} \Gamma-\Gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma-0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Gamma-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma-1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Gamma-1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma-2 \end{bmatrix} \mathbf{x} - \begin{bmatrix} d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \\ d_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This gets us variables storing the $\Gamma - d_i$, the value after subtracting the minimum distance to sensor from the large number for each vulnerable node. The constraints for ensuring desired partitioning are

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

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

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	41,70,85	1, 13, 27, 30, 38, 39, 40, 44, 51, 52, 68, 72, 80, 87, 9
1,2,3	0	72,150	1,4,68
131,20	0	132	27, 30, 37

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

In the previous work [2], the equivalent task was partial shutdown of WDN. 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. Here we get a solution with 18 actuators, 3 sensors.

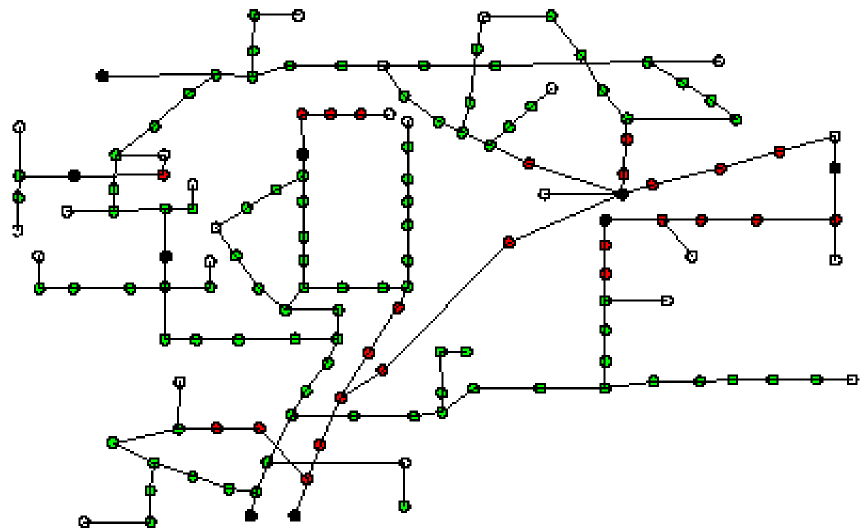

Figure 4: Network figure.
Green – Demand partition
Red – Vulnerable partition
Black – Vulnerable nodes
White – Demand nodes

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