

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

February 12, 2016

## Abstract

A multi objective integer linear optimization strategy assuming perfect sensors and actuators is developed and implemented.

Suhas Gundimeda, Indian Institute of technology MadrasCH12B026

## 1 Scenario

### 1.1 Given

Specification of water distribution network – vulnerable nodes, demand nodes, the 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 the attack can be identified 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. The work augmented the graph abstraction with vulnerable nodes representing multiple real nodes and used sets of affected nodes to force sensor placement. Actuator placement on edges to achieve a balanced min-cut, between the sensor nodes and demands, to contain contaminant[1]. It is also this work's primary reference.

The approach in [3] attempts to optimize different objectives: maximizing detection likelihood and minimizing expected time to detection and solves it

using genetic algorithms. This work assumes both of these objectives are binary – no time delay in detection and accurate sensing.

### 3 Hypotheses

Simultaneous sensor and actuator distribution can be achieved and is more efficient i.e. these are not independent problems, and these can be formulated as a ILP and solved.

### 4 Method

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

### 5 Implementation

#### 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 simultaneously, 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:

$$\begin{array}{ll}
 \min & (\sum x_i + \sum z_i) \\
 \text{sub} & \\
 \mathbf{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} & \leq \quad \mathbf{b} \\
 \mathbf{A}_{\text{eq}} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} & = \quad \mathbf{b}_{\text{eq}}
 \end{array}$$

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  $\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix}$  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.

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

$$A_{eq1} = b_{eq1} = \mathbf{0}$$

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

$b_2 = [\mathbf{0}]$ . 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.

## Results and verification

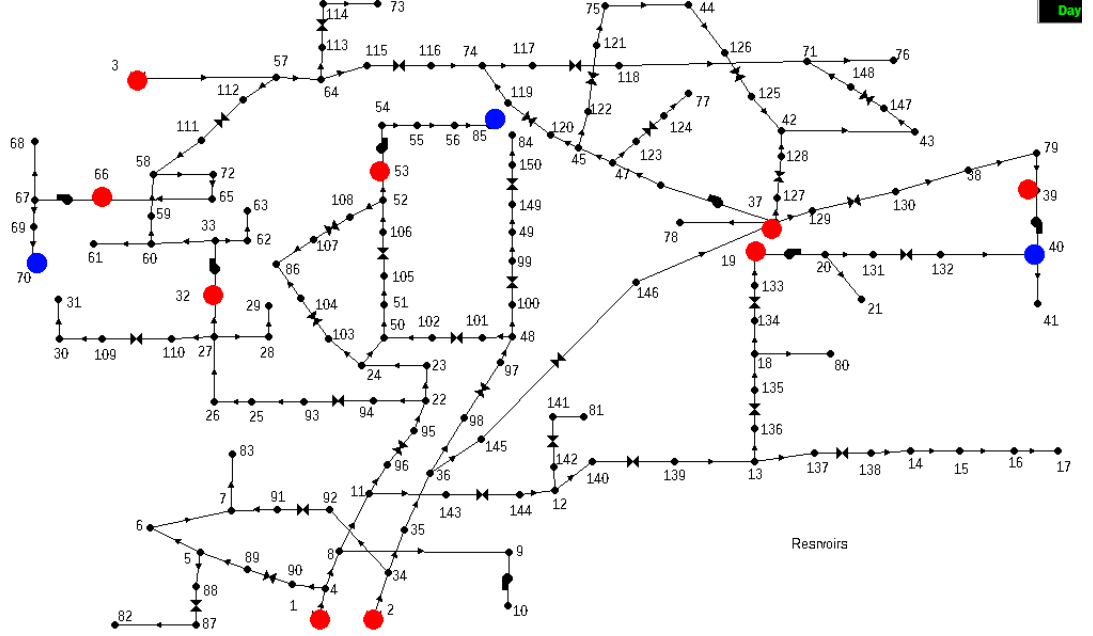


Figure 1: Graph abstraction with vulnerable nodes in red and blue are sensors

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 12 actuators: on edges 2, 12, 16, 30, 38, 40, 48, 51, 52, 67, 71, 76, 79, 87, 103, 108, 110.

## Case 2: The contaminant contained only 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 ensuring they are on the vulnerable side of actuators.

### Formulation

Formulating as binary integer optimization problem:

$$\begin{aligned}
 & \min \quad (\sum x_i + \sum z_i + \frac{1}{\Gamma}a) \\
 & \text{sub} \\
 & \mathbf{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ a \\ \mathbf{w} \end{bmatrix} \leq \mathbf{b} \\
 & \mathbf{A}_{eq} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ a \\ \mathbf{w} \end{bmatrix} = \mathbf{b}_{eq}
 \end{aligned}$$

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.

$\Gamma$  is a large number.

Hence the vector  $[\mathbf{x} \ y \ z \ a \ w]$  is of length  $3 * N + E + 1$ .

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

$A_1$  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.

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

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

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

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

So 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”. Let the vector be  $\mathbf{S}$ . The first constraint of  $A_{eq3}$  will be this vector of shortest paths multiplied with each row of  $\mathbf{I} \times \mathbf{x}$  to get the maximum distance of that particular sensor configuration, stored 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 the  $N < E$  in a lot of real world networks, we show the node implementation.

After transforming the  $\mathbf{y}$  vector to  $\mathbf{w}$ , we construct constraints for the  $i^{th}$  node as follows:  $\begin{bmatrix} \dots & 0 & -(\Gamma + \mathbf{S}) & 0 & \dots \end{bmatrix} [\mathbf{x}] \leq \begin{bmatrix} -\Gamma \end{bmatrix}$ . This completes the formulation.

## 6 Example

For example, take the graph generated by  $adjGraph = sparse([1 \ 1 \ 2 \ 2 \ 2 \ 3 \ 3 \ 4 \ 5], [2 \ 3 \ 4 \ 5 \ 3 \ 4 \ 5 \ 6 \ 6], [2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2], 6, 6);$

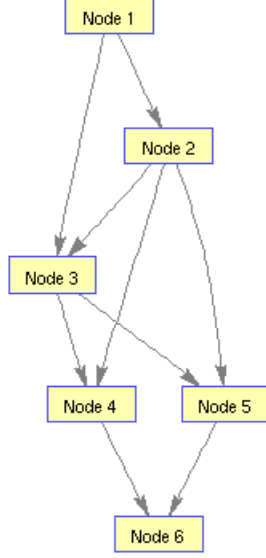


Figure 2: The test graph

Node 1,2 are vulnerable and node 6 is demand node.  
Formulating as binary integer optimization problem:

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

Where  $\mathbf{x} \equiv [x_i]$  is 1 if there exists a sensor at  $i^{\text{th}}$  node, 0 otherwise.

$\mathbf{y} \equiv [y_i]$  is 1 if  $i^{\text{th}}$  node is in the demands side of the actuators, 0 otherwise.

$\mathbf{z} \equiv [z_i]$  is 1 if there exists an actuator at  $i^{\text{th}}$  edge, 0 otherwise.

Hence the vector  $\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix}$  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  $i^{\text{th}}$  vulnerable node affects the  $j^{\text{th}}$  node in graph.

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

$$\begin{aligned}
b_1 &= \begin{bmatrix} -1 \\ -1 \end{bmatrix} \\
A_{eq1} &= b_{eq1} = \mathbf{O} \\
A_2 &= \begin{bmatrix} J & I_{E \times E} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}
\end{aligned}$$

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.

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

## References

- [1] V. Reddy, 2015 - Sensor network design for contaminant detection and identification in water distribution networks.
- [2] On the structure of all min cuts in a network
- [3] Optimization of Contaminant Sensor Placement in Water Distribution Networks: Multi-Objective Approach
- [4] Review of Sensor Placement Strategies for Contamination Warning Systems in Drinking Water Distribution Systems 10.1061/ ASCE WR.1943-5452.0000081
- [5] Sensor Placement Methods for Contamination Detection in Water Distribution Networks: A Review