A General Integer-Programming-Based Framework for Sensor Placement in Municipal Water Networks

Article · June 2004

DOI: 10.1061/40737(2004)455

J. Berry
SDR Venture Capital Partners
264 PUBLICATIONS **8,601** CITATIONS
SEE PROFILE

Cynthia A. Phillips
Sandia National Laboratories
150 PUBLICATIONS **6,273** CITATIONS
SEE PROFILE

James G Uber
Xylem Inc.
130 PUBLICATIONS **4,056** CITATIONS
SEE PROFILE

Sensor Placement in Municipal Water Networks

Jonathan Berry* Lisa Fleischer† William E. Hart*
Cynthia A. Phillips* Jean-Paul Watson*

Abstract

We present a model for optimizing the placement of sensors in municipal water networks to detect maliciously injected contaminants. An optimal sensor configuration minimizes the expected fraction of the population at risk. We formulate this problem as a mixed-integer program, which can be solved with generally available solvers. We find optimal sensor placements for three test networks with synthetic risk and population data. Our experiments illustrate that this formulation can be solved relatively quickly, and that the predicted sensor configuration is relatively insensitive to uncertainties in the data used for prediction.

1 Introduction

Public water distribution systems are inherently vulnerable to accidental or intentional water contamination because of their distributed geography. Major accidental contamination events, like the defining accident on the River Rhine in Germany, have highlighted these vulnerabilities (Brosnan, 1999). Although such accidents are of low probability, their immediate and long-term human health consequences are potentially severe. More recently, concerns over terrorist attacks have been heightened following the 9/11 attacks in the United States. These threats are potentially catastrophic, and the existence of such threats can impact public confidence in water supplies.

To address these concerns, the U.S. Environmental Protection Agency is working with community water systems to undertake a more comprehensive view of water safety and security. The development and implementation of on-line, real-time early warning systems (EWSs) is a key element of this effort. The general goal of an EWS is to identify a low probability/high impact contamination incident while allowing sufficient time for an appropriate response that mitigates any adverse impacts. An EWS complements utilities' conventional routine monitoring by quickly providing information on unusual threats to a water supply. Although several European countries have deployed EWSs to monitor riverine water supplies (Drage et al., 1998; Schmitz et al., 1994; Stoks, 1994), relatively few systems have been deployed for U.S. water supplies.

*Algorithms and Discrete Math Dept, Sandia National Laboratories, Mail Stop 1110, P.O. Box 5800, Albuquerque, NM, 87185-1110; PH (505)844-2217, (505) 845-7296; {jberry,wehart,caphill,jwatson}@sandia.gov. Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

†Dept. Operations Research and Math, 239 Posner Hall, Carnegie Mellon U, Pittsburgh, PA 15213; PH (412)268-5902; email: lkf@andrew.cmu.edu

The design of an effective EWS remains an active area of research. Appropriate sensor technologies need to be identified and developed. Further, the integration of information from on-line monitoring equipment remains a major challenge. For example, one possible approach for designing an EWS is to use conventional water monitors, using changes in water quality parameters (e.g. temperature, chlorine residual, color, conductivity, and pH) to infer the presence of a contaminant. However, the relationship between changes in water quality parameters and the presence of specific contaminants is not well understood, and it may be difficult to effectively characterize baseline statistics for common water quality parameters. Many questions also arise when considering the detection of intentional contamination events. Issues like the expected general effectiveness, the sensor density requirements and the expected false alarm rate may be qualitatively different when considering intentional contamination events.

We consider the deployment of sensors within a water utility's water distribution system. In this context, an effective deployment of an EWS requires a set of on-line monitors that ensures an adequate coverage of the network's flow for detection of contaminants. In particular, utilities wish to place on-line sensors such that the deployment cost is minimized and the level of protection afforded is maximized. However, such an EWS system will not necessarily provide warnings prior to the initial contamination of the water distribution system itself, as a traditional EWS would.

We present an approach for determining the placement of contaminant sensors within a municipal water network so as to minimize the expected fraction of the population exposed to the contaminant. The likelihood of a contamination is modeled as a fixed probability distribution across junctions in the network, which can be used to model the likelihood of either accidental or intentional attacks. We use integer programming techniques to find a globally optimal set of sensor placements. Our empirical results demonstrate that this approach is practical on two synthetic datasets and one real-world dataset. Further, our analysis of the optimal sensor placements for these datasets suggests that data uncertainties do not have a substantial impact on the optimal sensor placement.

2 Sensor Placement for an EWS

2.1 Technical Approach

A variety of technical approaches have been developed for sensor placement problems in water networks, including integer programming models (Lee and Deininger, 1992; Watson et al., 2004), combinatorial heuristics (Kessler et al., 1998; Kumar et al., 1999; Ostfeld and Salomons, 2004), and general-purpose metaheuristics (e.g. see Ostfeld and Salomons (2004)). Metaheuristics can be applied to complex simulations, including simulations that directly model sensor performance in detail (e.g. considering contaminant concentrations, flow turbulence, etc), as well as detailed health effects (e.g. the impact of accumulated exposure). Although these detailed models can capture many aspects of sensor performance, simulation-based optimization is extremely challenging because the total optimization time can be long in order to provide high confidence that near-optimal solutions are found.

In this paper, we consider sensor placement problems based on combinatorial optimization formulations. In particular, we consider models that can be solved as integer programs. Combinatorial models like integer programs can often be solved to optimality in practice, thereby providing the ability to ensure that the best solution is found. However, formulations like integer programs generally rely on simplifying assumptions to limit the number of design parameters and to enable the model to be solved with a particular method. We use the following assumptions to formulate a sensor placement problem as an integer program:

1. An attack occurs at a single point in the network
2. We consider the total population exposed, without reference to specific health impacts. Without sensors in the network, a population at a node is exposed if contaminant (with any concentration) can reach that point in a given flow period.
3. Sensors protect 'downstream' populations. A population is considered exposed if it could be reached by a flow path from the attack point without passing a sensor.
4. Transitions between time periods are ignored. Each time period is treated independently.

These simplifying assumptions allow us to (1) ignore temporal effects, (2) ignore concentration effects and (3) simplify health impacts. Thus, this model is well-suited for applications where contaminant flows rapidly through a network and there is a large volume of contaminant introduced. In this context, transitions between time periods are not so important, simple exposure is well-correlated with adverse health impacts, and sensors should reliably be able to detect the contaminant.

Although these assumptions clearly do not reflect all practical applications, our motivation here was to create combinatorial models that would scale well to large-scale real-world problems, while capturing many aspects of the real-world problem. Berry et al. (2004) describes our initial efforts to consider temporal effects directly, but our preliminary results suggest that temporal models are much larger and thus less amenable to exact solvers on large-scale problems. Concentration effects are ignored here because they seem much less relevant in a non-temporal model, and because we do not have precise information about how contaminant concentration relates to contaminant detection.

2.2 Detailed Modeling Issues

This section provides a more detailed discussion of the modelling assumptions made in our combinatorial model. Numerous measures can quantify the efficacy of sensor placements, reflecting various costs and risks of an attack on a network. For example, algorithms have been proposed for optimizing sensor placements with respect to detection coverage and probability (Lee and Deininger, 1992; Ostfeld and Salomons, 2004), volume of contaminated water consumed (Kessler et al., 1998), the extent of contamination (Watson et al., 2004) and time to detection (Kumar et al., 1999). However, public health protection is the primary goal of placing contaminant sensors in community drinking water systems.

The objective of our model is to minimize the expected fraction of the population that is at risk for some attack. We model an attack as the release of a harmful contaminant at a single point in the network with a single injection. For any particular attack, we assume that all points "downstream" of the release point (connected by a set of directed flows) can be contaminated. In general, we do not know a priori where this attack will occur, so our objective is to place sensors to provide a compromise solution across a set of weighted attack scenarios.

We assume that typical water demands throughout a day occur in one of a fixed set of patterns. The model makes no assumptions about how long each pattern holds, how often it appears, or the order in which the patterns appear. We use the water network simulator EPANET (Rossman, 1999) to determine a water flow given a set of available water sources, assuming each demand pattern holds steady for sufficiently long. Thus the set of sources and demands and the set of flow patterns are interchangeable concepts in this paper. We ignore the magnitude of water velocity, requiring only its direction and that it be sufficiently large.

For each flow, each junction is weighted by the number of people potentially consuming water at that point. We correlate flow patterns with approximate time of day to set these population numbers (e.g., to represent people at work during the day and at home in the evening). Note that these population numbers are not necessarily proportional to demand. For example, an industrial site could consume a lot of water although few people are on site and a public drinking fountain could consume a relatively small amount of water, but that water is directly ingested by many people.

Attack scenarios are defined by a probability distribution over all pairs of population-weighted flows and attack points (i.e., junctions in the network). This distribution might come from expert opinions, potentially taking into consideration knowledge of the network defenses (ease of access), location of assets within the network (e.g., location of a person or building that may be a likely target), degree of damage, and attacker psychology. For this paper, we generated these distributions synthetically.

2.3 Integer Programming Model

In this section we give a more detailed description of the input data and formulate the problem as a *mixed-integer program* (MIP). A MIP is the minimization (or maximization) of a linear objective function subject to a set of linear and integrality constraints on the variables. In this case, the integrality constraints represent decisions of where to place a limited number of sensors.

We model a water network as a graph $G = (V, E)$. E is a set of edges representing pipes. V is a set of vertices, or nodes, where pipes meet. Vertices can represent sources, such as reservoirs or tanks, where water is introduced, and sinks (demand points) where water is consumed. In general, the network is represented at some scale or granularity, where nodes represent neighborhoods or regions of a city. Each pipe connects two vertices v_i and v_j and is usually denoted (v_i, v_j).

We consider risk under a fixed number of flow patterns, where we require only the direction of the flow on each edge. Thus a flow specifies for each edge (v_i, v_j) whether the flow is i-to-j, j-to-i or essentially zero (based on a minimal threshold for the flow). We require the following input data:

- $G = (V, E)$, the network. $V = v_1, \dots v_n$ and E $= e_1, \dots e_m$.
- α_{ip}, the probability of an attack at node v_i during flow pattern p conditional on exactly one attack on a node during some flow pattern. We have $\sum_{v_i \in V, p \in 1 \dots P} \alpha_{ip} = 1$, where

P is the number of flow patterns.

- δ_{ip}, the density (number of people) at node v_i while flow p is active.
  One can replicate a flow and associate multiple population densities with it, for example, when flows are associated with time periods and appear multiple times in a day. $\delta_{ip} = 0$ if node v_i is not a demand node during flow p.

- $f_{ijp} \equiv f_{ep} \in \{0,1\}$. These parameters describe flow pattern p. $f_{ijp} = 1$ if there is positive flow along (directed) edge $e = (v_i, v_j)$ during flow pattern p and are 0 otherwise. Water cannot flow in both directions of a pipe, so we have $f_{ijp} f_{jip} = 0$.

- S_{max}, the maximum number of sensors we can place.

Given a single attack on node v_i during flow pattern p, a node $v_j \neq v_i$ is contaminated if there is a path from v_i to v_j without a sensor for which all edges have "positive" flow during flow p. More specifically, v_j is contaminated if there is a path $v_i \equiv v_1, v_2, \ldots, v_j \equiv v_l$ such that $(v_k, v_{k+1}) \in E$ and $f_{k(k+1)p} = 1$ for all $k = 1 \ldots l-1$ and we place no sensors on any edge in the path. If a demand node v_j is contaminated during flow p, then all the people at node v_j during time p are exposed. We wish to minimize the expected number of exposed people.

We introduce the following variables for the MIP formulation of our sensor-placement problem:

- Decision variable $s_{ij} = 1$ if we place a sensor on (undirected) edge (i,j) and 0 otherwise. A sensor on edge (i,j) detects contaminants moving in either direction. For ease of exposition, we will use both variables s_{ij} and s_{ji}, but they will be equal and, as a pair, represent the placement of only one sensor.

- Derived variables $c_{ipj} = 1$ if node v_j is contaminated by an attack at node v_i during flow pattern p, and 0 otherwise.

The mathematical formulation of the MIP is:

$$\text{(SP1)} \quad \text{minimize} \quad \sum_{i=1}^{n} \sum_{p=1}^{P} \sum_{j=1}^{n} \alpha_{ip} c_{ipj} \delta_{jp}$$

$$\text{where} \quad \begin{cases} c_{ipi} = 1 & \forall i = 1 \ldots n, p = 1 \ldots P \\ s_{ij} = s_{ji} & \forall i = 1 \ldots n-1, i < j \\ c_{ipj} \geq c_{ipk} - s_{kj} & \forall (k,j) \in E \text{ s.t. } f_{kjp} = 1 \\ \sum_{(i,j) \in E, i<j} s_{ij} \leq S_{max} & \\ s_{ij} \in \{0,1\} & \forall (i,j) \in E \end{cases}$$

The first set of constraints ensures that when a node is directly attacked, it is contaminated. The second set indicates that a single sensor covers a pipe for flow in both directions. The third set propagates contamination from a node v_k to a node v_j if node v_k is contaminated, there is positive flow along a directed edge from v_k to v_j and there is no sensor on that edge. The next constraint enforces the limit on total number of sensors. The final set forces integrality of the sensor-placement decisions. If these variables are set integrally, then the contamination indicator variables c_{ipj} are also integral, even though they are not explicitly forced to binary values in the MIP. The objective function exerts pressure to minimize these variables. The first and third set of constraints propagate values of 1 whenever there are no sensor to prevent the propagation (otherwise the contamination variables can stay at 0).

The practical motivation for formulating our sensor placement problem as a MIP is that in practice, many MIPs are solved exactly using intelligent variations of branch and bound. There are a number of commercial and free software packages for solving MIPs such as CPLEX (an ILOG product) and PICO (Eckstein et al., 2001). Generic branch and bound for MIPs uses linear programming (LP) as a lower bound, obtained from the original MIP by relaxing the integrality constraints. LP is efficiently solvable theoretically, and LPs can be effectively solved in practice with commonly available codes.

3 Methods

We have evaluated our sensor-placement strategy experimentally using two networks from the EPANET test set and one real network. Because information on population density and risk was unavailable for the EPANET networks and only partially available for the other, we used plausible synthetic data. For each of these datasets, we used EPANET to determine flow patterns during four six-hour time periods within a twenty-four hour time period. We will sometimes use “time period” interchangeably with “flow pattern” in the following discussion.

Dataset 1: We adapted the first dataset from “Example Network 2” provided with EPANET 2.0. This network has 36 nodes and 40 pipes, with one pump station. We divided the nodes in this dataset into four fictitious categories: pump station, residential neighborhood, business district, and industrial district. We considered twelve sensor placement problems for this dataset. In each problem, we considered one of the four groups of nodes at risk for attack (four attack scenarios), and we set the sensor limit to either 3, 5, or 7. To create an attack distribution, we selected four flow patterns: one from each six-hour time periods in a 24-hour EPANET simulation. We assumed a constant population of 500 across all of the nodes in each time period, with shifts between nodes that reflect the likely behaviors of a typical town. The *relative* probability of attack across nodes is constant within each attack scenario: the node group at risk has uniformly high probability, and all other nodes have low probabilities. The relative (total) probability of each flow pattern is weighted by the perceived likelihood that a attack would be successful within that time period; for example, if a residential neighborhood is at risk, an attack is more likely during the day than in the evening, when more people are in the neighborhood.

Dataset 2: We adapted the second dataset from “Example Network 3” provided with EPANET 2.0. This network has 97 nodes and 117 pipes, with 2 reservoirs and 3 tanks. We partitioned the nodes in this dataset into five fictitious categories: residential neighborhood, the mall, downtown neighborhood, industrial district, and other. The experiments were identical to those with dataset 1 except that the aggregate population, still constant through time, was 200000.

Dataset 3: We adapted the third dataset from a real-world network. This network has 470 nodes and 621 pipes, with 3 pumps and 4 tanks. We divided the nodes in this dataset into three fictitious categories: residential neighborhood, business district, and industrial district. We considered six sensor-placement problems where the number of sensors was either 10, 50, 100, 150, 250 or 300. We assumed a maximum total population of 7600, though we did not keep this number constant within each time period. In each time period, we considered all three groups of nodes at risk, though with different relative probabilities. Thus the relative probability of attack at each node depended on the time of day as well as the node category. This represents a blending of the individual (category-based) attack scenarios.

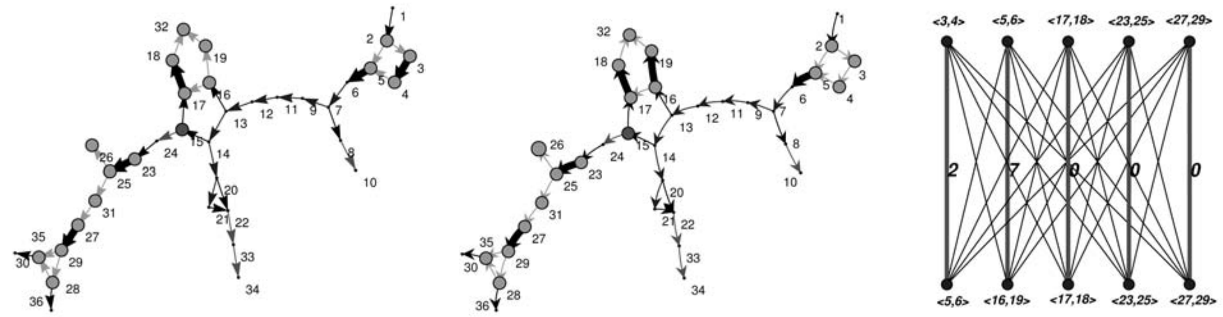
For each of the problems defined by datasets 1, 2 and 3, we used EPANET 2.0 to calculate the flow directions for the attack scenarios. Dataset 1 has relatively few changes in its flows from one time period to the next. In this dataset, the network is almost linear (chainlike), and since it has only one water source, water simply flows from one end to the other. For dataset 2, water enters the network from two distinct sources. Changes in demand thus lead to some gross shifts in the water flow. Dataset 3 exhibits flow direction changes in roughly 90% of the pipes. In this network, the neighborhood, business and industrial nodes are somewhat segregated, so flow changes tend to be localized.

We used the AMPL modeling language (Fourer et al., 2002) to formulate the MIP (SP1). In all cases, we solved this MIP on Solaris and Linux workstations using AMPL 9.0, which applied the CPLEX 9.0 MIP solver. We measure the efficacy of our solution by considering the run time as well as the expected percentage of the maximum population that is at risk for the optimal solution.

We expect that it will be difficult to assess the population densities and attack risks in real-world applications accurately. Consequently, we also studied the sensitivity of our model to uncertainties in this data. More specifically, we consider how the value of the optimal solution and optimal sensor configuration change given changes in the population density and attack risks. We consider three noise levels: 5%, 10% and 25%. For each problem, we altered each element of the population densities and risk probabilities by multiplying by a uniformly distributed value in $[1.0 - \epsilon, 1.0 + \epsilon]$, where ϵ is the noise level. We then renormalized the population densities and risk probabilities to ensure that the total population size was not changed in any attack scenario and that the total risk probabilities sum to one. Let us define an *experiment* to be a set of trials of our MIP model for a fixed dataset, attack scenario, noise level, and number of sensors. The experiments for datasets 1 and 2 each consisted of thirty trials, while those for the larger dataset 3 each comprised five trials.

To measure the sensitivity of the optimal sensor configuration, we define a distance measure between two sensor placements and compare each trial (optimization under noise) to the baseline without noise. We defined this distance by matching sensors in one configuration to the sensors in another. For each pair of matched sensors, the distance between those sensors is the number of nodes that are traversed in the shortest path between their two pipes within the water network. The distance of the entire matching is the sum of the pairwise distances taken all pairs of sensors in the matching. We define the distance between two sensor placements to be the value of the minimum matching between the two sensor placements, divided by the number of sensors in the original configuration. This corresponds to the minimum average movement for each sensor. We can efficiently compute this distance using a minimum-weight bipartite matching algorithm (Cook and Rohe, 1999), with nodes corresponding to the two sets of sensor placements and edge weights corresponding to the distance between each sensor pair.

For example, consider Figure 1. Figure 1a shows an optimal placement of 5 sensors for dataset 1 where there is an attack on residential districts; edges with thick lines (such



(a) (b) (c)

Figure 1: (a) A noiseless optimal sensor placement for dataset 1, (b) an optimal placement in the presence of noise, and (c) the bipartite graph used to compute the distance between the placements.

as (3,4)) have sensors. Figure 1b shows the optimal placement in the presence of slightly perturbed data, while Figure 1c depicts the weighted bipartite graph constructed to compute the distance, which in this case is 9/5. Thickened edges show the minimum-weight matching, and this figure shows distances only on the matching edges. The diameter of a node is proportional to its current population. In particular, node 1 has no population. We used the *LINK* system (Berry et al., 2000) to compute the minimum-weight matching and create custom graphics.

4 Numerical Results

Tables 1a, 1b and 1c summarize the values of the optimal solutions found for datasets 1, 2 and 3. The values in each cell of these tables show the expected percentage of population at risk, and the number of linear program solves required to compute the optimal sensor configuration. Further, the scenarios in Tables 1a and 1b indicate the set of nodes that are primarily at risk in these scenarios. These results confirm that the expected percentage of the population that is at risk goes down as the number of sensors is increased. For these datasets, a large fraction of the population can be protected with a limited number of sensors.

Tables 1a, 1b and 1c also summarize the number of subproblems CPLEX 9.0 required to solve the corresponding MIP. In many cases these problems were solved at the root of the branch-and-bound tree. Thus simply solving a linear program may be sufficient to solve the MIP in some of these cases.

The column with zero sensors provides a baseline for assessing the impact of using sensors to protect the population. Consider the example of "pump station" scenario in Table 1a. There was roughly a 90% probability that the attack would occur at the pump station (Node 1) sometime during the 24 hour simulation. Without any sensors, all of the population would be exposed during an attack at that node. However, the expected percent exposed is less than 100% since there is a nonzero probability of attack at the other nodes. On the other hand, the extremely low objective value for Table 1c is an artifact of topological characteristics of the dataset 3. The flow in this network looks like a main trunkline with bush subnetworks that are tree-like. Consequently, most nodes are leaves or nearly leaves, and thus most attacks will impact a small number of "downstream" nodes.

Scenario	Num Sensors			
	0	3	5	7
Pump Station	93.56 (1)	1.16 (1)	0.92 (1)	0.76 (1)
Residential	36.12 (1)	11.35 (1)	9.87 (3)	8.57 (1)
Business	55.16 (1)	6.03 (1)	4.23 (1)	3.38 (1)
Industry	10.49 (1)	2.75 (1)	2.39 (1)	2.11 (1)

(a)

Scenario	Num Sensors			
	0	3	5	7
Industry	14.77 (1)	3.44 (1)	2.66 (1)	2.16 (1)
Downtown	3.33 (1)	2.15 (1)	2.10 (20)	2.06 (1)
Residential	19.34 (1)	8.54 (1)	6.81 (3)	5.54 (1)
Mall	45.54 (1)	12.87 (9)	5.22 (1)	1.48 (1)

(b)

	Num Sensors						
	0	10	50	100	150	250	300
Values	3.77 (1)	1.35 (4)	0.52 (1)	0.32 (1)	0.24 (112)	0.17 (32)	0.15 (161)

(c)

Table 1: Summary of the values of optimal sensor configurations for (a) dataset 1, (b) dataset 2 and (c) dataset 3.

The run time to solve the MIP in these experiments varied significantly with the problem size. For datasets 1, 2 and 3, these problems were solved on average within a second, a minute and a half hour respectively. In preliminary experimentation, we had difficulty solving problems substantially larger than dataset 3. Further investigation indicated the limitation was due to the size of the LPs; intelligent preprocessing of the MIP, e.g., by eliminating constraints between pairs of vertices that are not reachable, largely avoids this problem.

The optimal solution value (percentage of the population that is at risk) showed very little sensitivity to noise for all three datasets. In virtually all cases, the mean value of the modified problem was within 0.1% of the baseline value, and usually within 0.05%. The variance of the modified problem values was also quite low. Even with noise levels of 25%, variance was almost always at or below 0.02%.

However, the configuration of the optimal solution did exhibit sensitivity to noise. The data in Table 2 demonstrates that with increasing uncertainty in the data, the uncertainty in the optimal sensor configuration increases. In these tables, the first two values in each cell show the mean and average variance of the distance between the optimal sensor configuration for the baseline data and the optimal sensor configuration for perturbed data. The third entry for each cell is the average consensus, the percentage of sensors which do not move at all.

As the noise level increases, the average distance from the baseline solution increases as well as the variance of this distance. However, even for noise levels of 25%, the average distance is less than 2 in all cases. Thus we should expect to move each sensor at most two edges from our baseline solution configuration even when the input data for the model changes by as much as 25%.

As a function of number of sensors, for datasets 1 and 3, sensor placement sensitivity to noise at all tested levels and all scenarios consistently climbs to a peak at some middle value of number of sensors, and then drops as the number of sensors climbs. Dataset 2 shows this trend weakly, except for scenario 4. We'll discuss a plausible explanation in Section 5.

5 Discussion

In this section we present possible explanations for the data from our pilot study and discuss some implications and directions for further study.

One plausible explanation for the rise, then decline of sensor placement sensitivity is that the number of sensors falls into one of three *regimes* with respect to the network and data sets. With very few sensors, the best strategy is to protect the most valuable asset(s). Eventually, with more sensors, there are more choices for secondary assets to protect, and these choices may be quite sensitive to variations in attack probabilities and population densities. Finally, when there are enough sensors to essentially protect everything, sensors are always placed in core locations.

For example, in Table 2 (a), all four scenarios exhibit this regime change, with 3 sensors representing the first regime, 5 sensors representing the second, and 7 sensors representing the last. The pattern is exhibited again in the scenarios 1 and 2 of Table 2 (b), and strongly so in Table 2 (c), with 250 sensors representing the turning point. The only scenarios that do not appear to exhibit a regime change are scenarios 3 and 4 in Table 2 (b). We conjecture that in the former case, the regime change point is at a level greater than 7 sensors, and in the latter case, 3 sensors are sufficient to assume the highest regime.

Verifying regime changes and predicting their location (number of sensors) as a function of network and data set is a topic for further study. Some networks may show multiple regime changes depending upon the number of major, localized assets.

The low sensitivity of the objective function implies that one may be able to predict a reasonable sensor budget as a function of desired protection level. However, it may be difficult to determine a single set of sensor locations that will work for a variety of related datasets. In a setting where sensor placement is sensitive, network planners may prefer a formulation that explicitly addresses data uncertainties. One possible method is to incorporate the noise into additional attack scenarios. However, one must then tolerate sparse sampling or solve huge problem instances. Another possibility is to find a solution that uses a limited number of additional sensors to guarantee robustness. For example, for a goal number of sensors g_s, place $g_s + k$ sensors in such a way that this static sensor placement performs as well as a *mobile* set of g_s, always (near) optimally placed sensors would perform.

Finding an optimal sensor placement for full-sized networks containing 100000 or more nodes is likely to require parallelization using, for example, the PICO massively-parallel MIP code (Eckstein et al., 2001). Parallel LP solvers are being incorporated into PICO, which will enable the solution of MIPs whose LP solves are too large for workstations. PICO also has facilities for exploiting problem structure in order to speed search and reduce the number of subproblems. An analyst can stop the computation early once a solution is provably within

Scenario	Num Sensors	Noise Level								
		0.05			0.10			0.25		
1	3	0.00	(0.00)	100%	0.00	(0.00)	100%	0.01	(0.01)	99%
	5	0.01	(0.01)	99%	0.01	(0.01)	99%	0.36	(1.65)	83%
	7	0.00	(0.00)	100%	0.00	(0.00)	100%	0.06	(0.13)	96%
2	3	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%
	5	0.81	(4.50)	83%	1.01	(4.49)	78%	1.39	(3.15)	67%
	7	0.13	(0.09)	87%	0.18	(0.25)	85%	0.32	(0.79)	82%
3	3	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%
	5	0.01	(0.01)	99%	0.05	(0.04)	95%	0.10	(0.05)	90%
	7	0.00	(0.00)	100%	0.00	(0.00)	100%	0.05	(0.24)	98%
4	3	0.01	(0.01)	99%	0.04	(0.04)	96%	0.12	(0.08)	88%
	5	0.07	(0.67)	99%	0.22	(1.84)	96%	0.67	(3.98)	87%
	7	0.04	(0.03)	96%	0.05	(0.03)	95%	0.10	(0.06)	90%

(a)

Scenario	Num Sensors	Noise Level								
		0.05			0.10			0.25		
1	3	0.00	(0.00)	100%	0.41	(7.35)	97%	1.66	(23.41)	86%
	5	0.11	(0.05)	89%	0.10	(0.05)	90%	0.37	(4.92)	83%
	7	0.06	(0.04)	94%	0.07	(0.07)	93%	0.18	(0.25)	82%
2	3	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%
	5	0.00	(0.00)	100%	0.00	(0.00)	100%	0.39	(3.54)	92%
	7	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%
3	3	0.00	(0.00)	100%	0.01	(0.01)	99%	0.10	(0.07)	90%
	5	0.24	(0.45)	92%	0.28	(0.46)	91%	0.38	(0.52)	87%
	7	0.62	(11.12)	89%	0.90	(13.59)	85%	1.95	(15.64)	67%
4	3	0.00	(0.00)	100%	0.13	(0.22)	93%	0.09	(0.16)	96%
	5	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%
	7	0.00	(0.00)	100%	0.00	(0.00)	100%	0.00	(0.00)	100%

(b)

Num Sensors	Noise Level								
	0.05			0.10			0.25		
10	0.00	(0.00)	100%	0.00	(0.00)	100%	0.26	(3.58)	93%
50	0.03	(0.19)	98%	0.12	(0.45)	93%	0.54	(1.77)	86%
100	0.07	(0.57)	96%	0.20	(1.62)	94%	0.50	(4.44)	85%
150	0.16	(0.87)	92%	0.30	(0.69)	85%	0.43	(2.20)	80%
250	0.37	(2.77)	89%	0.41	(1.72)	87%	0.61	(5.07)	79%
300	0.14	(0.83)	93%	0.20	(1.24)	92%	0.34	(1.54)	82%

(c)

Table 2: Summary of the effect of noise on the sensor configuration for (a) datasets 1, (b) dataset 2 and (c) dataset 3.

a desired tolerance of the global optimum.

In Section 4, we normalized the optimal number of people at risk by the maximum population. A alternative normalization factor is the value of the MIP with zero sensors, the average number of people exposed within a particular attack scenario. The normalized objective would then be the expected number of people saved by the sensors compared with this baseline.

6 Conclusions

We have demonstrated that the MIP model described in this paper can be used to effectively solve large-scale sensor placement problems. Although we have limited our discussion to a particular MIP formulation for sensor placement, there clearly are a variety of related models that address different performance objectives, consider alternative placement locations, address temporal modeling issues and consider data uncertainties. Although a detailed discussion of these issues is beyond the scope of this paper, we summarize here some ideas of how our current model can be generalized:

- **Temporal Effects:** As we noted earlier, the current model is well suited for applications where water flow is quick, or where water flow does not change direction. We have developed another MIP formulation for sensor placement that more directly models temporal effects, using a qualitatively different modeling strategy (Berry et al., 2004). Our preliminary results indicate that this MIP requires considerable data about flow patterns and contaminant properties, and that it can be much more difficult to solve for a given network size.

- **Placement Locations:** Although our current model places sensors on edges, it is easy to adapt this model to place sensors on nodes of the network, or a mixture of both (e.g. see the models discussed in Watson et al. (2004)).

- **Sensor Costs:** Our current model treats cost issues by simply limiting the total number of sensors. It is straightforward to generalize this approach to consider installation costs, as well as maintenance costs (which may differ on the location and type of sensor).

- **Performance Objective:** Although protecting public health is the primary goal of community drinking water systems, MIPs can be used to formulate sensor placement problems for a variety of related objectives. Early work by Lee and Deininger (1992) a considered detection coverage metric, and more recently Watson et al. (2004) have formulated MIPs for metrics like the extent of contamination and time to detection. Although we have proved that sensor placement is NP-hard for the population exposed and time-to-detection metrics (e.g. see Berger-Wolf et al. (2003)), our computational experience suggests that such MIP formulations can be effective in practice.

References

T. Y. Berger-Wolf, W. E. Hart, and J. Saia. Discrete sensor placement problems in distributed networks. Technical report, Sandia National Laboratories, Albuquerque, NM, 2003.

J. Berry, N. Dean, M. Goldberg, G. Shannon, and S. Skiena. Graph computation with LINK. *Software Practice and Experience*, 30:1285–1302, 2000.

J. Berry, W. E. Hart, C. A. Phillips, and J. Uber. A general integer-programming-based framework for sensor placement in municipal water networks. In *Proceedings of the World Water and Environment Resources Conference*, 2004.

T. B. Brosnan, editor. *Early Warning Monitoring to Detect Hazardous Events in Water Supplies.* International Life Sciences Institute Risk Science Institute. ILSI PRESS, Washington, DC, 1999.

W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal on Computing*, 11:138–148, 1999. available at `citeseer.nj.nec.com/cook98computing.html`.

B. E. Drage, J. E. Upton, and M. Purvis. On-line monitoring of micropollutants in the river trent (UK) with respect to drinking water abstraction. *Water Science and Technology*, 38 (11):123–130, 1998.

J. Eckstein, W. E. Hart, and C. A. Phillips. PICO: An object-oriented framework for parallel branch and bound. In D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 219–265. Elsevier Science Publishers, Amsterdam, The Netherlands, 2001.

R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Brooks/Cole, Pacific Grove, CA, second edition, 2002.

A. Kessler, A. Ostfeld, and G. Sinai. Detecting accidental contaminations in municipal water networks. *Journal of Water Resources Planning and Management*, 124(4):192–198, 1998.

A. Kumar, M. L. Kansal, and G. Arora. Discussion of 'detecting accidental contaminations in municipal water networks'. *Journal of Water Resources Planning and Management*, 125(4):308–310, 1999.

B. H. Lee and R. A. Deininger. Optimal locations of monitoring stations in water distribution system. *Journal of Environmental Engineering*, 118(1):4–16, 1992.

A. Ostfeld and E. Salomons. Optimal layout of early warning detection stations for water distribution systems security. *Journal of Water Resources Planning and Management*, 130 (5):377–385, 2004.

L. A. Rossman. The EPANET programmer's toolkit for analysis of water distribution systems. In *Proceedings of the Annual Water Resources Planning and Management Conference*, 1999. Available at `http://www.epanet.gov/ORD/NRMRL/wswrd/epanet.html`.

P. Schmitz, F. Krebs, and U. Irmer. Development, testing and implementation of automated biotests for the monitoring of the river rhine, demonstrated by bacteria and algae tests. *Water Science and Technology*, 29:215–221, 1994.

P. G. Stoks. Water quality control in the production of drinking water from river water. In M. Adriaanse, J. van der Kraats, P.G. Stoks, and R.C. Ward, editors, *Proceedings: Monitoring Tailor-made*, RIZA, Lelystad, The Netherlands, 1994. (ISBN 9036945429).

J.-P. Watson, H. J. Greenberg, and W. E. Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of the World Water and Environment Resources Conference*, 2004.