ON THE PLACEMENT OF IMPERFECT SENSORS IN MUNICIPAL WATER NETWORKS

JONATHAN BERRY **ROBERT D. CARR** **WILLIAM E. HART**
VITUS J. LEUNG **CINDY A. PHILLIPS** **JEAN-PAUL WATSON**
Sandia National Laboratories
Albuquerque, New Mexico
{*jberry,rdcarr,wehart,vjleung,caphill,jwatson*}*@sandia.gov*

Abstract

We consider the problem of optimally placing water quality sensors in municipal water networks under the assumption that sensors may fail. We give a non-linear formulation of the problem, then a linearization of this formulation in the form of a mixed-integer program (MIP). We explore the scalability limits of this formulation, then use it as a bounding procedure for a local search heuristic that optimizes the same objective: minimizing the expected impact of a contamination event. This heuristic can find optimal or near-optimal solutions on networks with over ten thousand junctions.

Keywords

Sensor Placement, Contaminant Warning System Design, Imperfect Sensors.

1 Introduction

Previous work by this research team and others has generated sensor placement optimization models that optimally place *perfect* sensors in water distribution networks. This simplifying assumption has allowed us to make significant progress in our understanding of the fundamental optimization problem. However, it is clearly an unrealistic assumption. If sensors may fail, then a sensor layout determined by a perfect-sensor method will be flawed. Conceivably, for example, several injection scenarios of great expected impact could be mitigated by the strategic placement of a single perfect sensor. However, it would be unwise to assume that these potential injections are therefore mitigated. The sensor could fail, of course, and then the severe impact would occur in its entirety.

In other work, *Receiver Operator Curves* (ROC curves) are being generated to track the characteristics of various sensors at given locations throughout a water network (Klise and McKenna, 2005; McKenna et al., 2006). Assuming that the sensor returns continuous numeric values, say for the level of total organic carbon, the ROC curve plots *sensitivity* (the percentage of true positive events that are caught) versus 1-*specificity*, where specificity is the probability of a false positive reading. A point on the ROC curve is determined only after an event detection threshold has been set. For a pathological example, consider a threshold set at zero, where greater values are indicative of contamination. In this case, contaminant detection is signaled at every time step. In other words, all true events are caught (the sensitivity is 100%) and all non-events are also signaled as positives (the specificity is 0%). This holds whether the sensor is a random number generator or

an idealized perfect sensor. Dialing this threshold up to medium values, and then high values, we generate many points along this curve.

We assume that ROC curve information is available, and that experts have used human judgment to set a threshold that gives the highest possible sensitivity, given that the specificity must be very high. That is, we assume that false positive events are not tolerated more than once every few months. Higher false positive rates would probably cause a utility to start ignoring its sensor events. At threshold values with such high specificity, we may be forced to tolerate sensitivities in the 50% range or even lower. With sensors failing to detect true events (giving *false negatives*) so frequently, it becomes increasingly important to model their failures before committing to a placement.

2 Problem Input

As in previous work (Berry et al., 2005, 2006) , we take as input data a network model from which a program such as EPANET can calculate hydraulics and perform water quality simulations. In addition, we consider an ensemble of contamination events described by concentration, start time, duration, and location. New to the imperfect sensor models are two additional input parameters:

- We assign each node in the network to a *detection class*. For example, some areas of the network may experience large variations in water quality, while others may not. The first group of locations may be characterized by ROC curves that predict lower sensitivity than the second group.

- We assign *false negative probabilities* to each class of potential sensor locations by appealing to ROC curves.

With these data we can find an optimal or near-optimal solution to the sensor placement problem, assuming that a sensor placed at a location will fail to detect a true event with the false negative probability associated with that location.

In Section 3, we review the MIP formulations for our basic models – those that assume perfect sensors. Then in Section 4 we give our new model for handling imperfect sensors. The latter is an extension of the basic model.

3 The Base Imperfect-Sensor MIP Model

In this section we review our base integer programming (IP) formulation of the sensor placement optimization problem. We assume a fixed budget of p sensors, each of which can be placed at any junction in a distribution network. We do not allow installation of sensors on pipes at this time because that would require water quality information along the pipes. We rely on water quality simulations from EPANET, which does not currently provide contaminant concentration information along pipes. We assume that sensors are capable of detecting contaminants at any concentration level, and we assume that a general alarm is raised when contaminant is first detected by a sensor, such that all further consumption is prevented.

We model a water distribution network as a graph $G = (V, E)$, where vertices in V represent junctions, tanks, or other sources, and edges in E represent pipes, pumps, and valves. In higher-granularity (i.e., skeletonized) network models, each vertex may represent an entire neighborhood or other geographic region. We assume that demands follow a small set of patterns, e.g., one pattern per hour throughout the day. Each pattern represents the demand during a particular time interval on a "typical" day. Because each pattern holds steady for one or more hours, we assume the gross flow characteristics induced by these demands holds steady during the time period associated with that pattern.

Let $\mathcal{A}$ denote the set of contamination scenarios against which a sensor configuration consisting of p sensors is intended to protect. A contamination scenario consists of individual contamination events, each of which can be characterized by quadruples of the form (v_x, t_s, t_f, X), where $v_x \in V$ is the origin of the contamination event, t_s and t_f are the contamination event start and stop times, and X is the contamination event profile, e.g., arsenic injected at a particular concentration at a given rate. The quadtuples can easily be extended to account for multiple coordinated contamination events. Let t_s^a and t_f^a respectively denote the start and stop times of the contamination event for scenario a. The impact of a given contamination scenario can be evaluated using water quality analysis software (e.g., EPANET (Rossman, 1999)) to compute the contaminant concentration at each junction in the network from time t_s^a to an arbitrary point $t_h \geq t_s^a$ in the future. The results of such an analysis are expressed in terms of concentration time-series τ_j for each $v_j \in V$, with samples at regular (arbitrarily small) intervals within $[t_s^a, t_h]$. Our discussion throughout the paper assumes that when contamination scenarios consist of multiple events, these events involve identical contaminant types. It should be clear from our definition of contamination events that this is not necessarily true, and thus the approach described here naturally generalizes.

It is usually straightforward to compute the total impact, $d_a(t)$, the total network-wide impact of a contamination scenario a at any given time $t \geq t_s^a$. A key characteristic of our base formulation is that it captures a wide range of possible definitions of impact including population exposed to contamination, volume of contaminant released from the network, total length of contaminated pipe, etc. In general, impact increases monotonically with time to detection. Let γ_{aj} denote the earliest time t at which a hypothetical sensor at junction v_j can detect contaminant due to a contamination scenario a. If no contaminant ever reaches v_j, then $\gamma_{aj} = t^*$, where t^* denotes the stop time imposed on the water quality simulations; otherwise, γ_{aj} can be easily computed from τ_j. We next define $d_{aj} = d_a(\gamma_{aj})$, i.e., the total impact of a contamination scenario a if the contaminant is first detected by a sensor at v_j. Finally, let q denote a "dummy" location that corresponds to failed detection of contamination scenario a. The impact d_{aq} is defined as the total impact of contamination scenario a if it is not detected before t^*.

Our formulation models the placement of p sensors on a set $L \subseteq V$ vertices, with the objective of minimizing the expected impact of a set $\mathcal{A}$ of contamination scenarios. A likelihood $\alpha_a \geq 0$ is assigned to each contamination scenario $a \in A$, such that $\sum_{a \in A} \alpha_a = 1$. Let $\mathcal{L}_a$ be the subset of vertices in $L \cup \{q\}$ that could possibly be contaminated by scenario a. The design objective is then expressed as:

$$\sum_{a \in \mathcal{A}} \alpha_a \sum_{i \in \mathcal{L}_a} d_{ai} x_{ai},$$

where x_{ai} is an indicator variable with value equal to 1 if location i raised the alarm (i.e., first detected contaminant) for contamination scenario a and 0 otherwise. If $x_{ai} = 1$, we say that location i *witnesses* (or is a witness for) contamination event a.

Our complete base model formulation – which we denote by BSP – is easily expressed as the following MIP:

$$\text{(BSP)} \quad \text{minimize} \quad \sum_{a \in \mathcal{A}} \alpha_a \sum_{i \in \mathcal{L}_a} d_{ai} x_{ai}$$

$$\text{where} \quad \begin{cases} \sum_{i \in \mathcal{L}_a} x_{ai} = 1 & \forall a \in \mathcal{A} \\ x_{ai} \leq s_i & \forall a \in \mathcal{A}, i \in \mathcal{L}_a - \{q\} \\ \sum_{i \in L} s_i \leq p & \\ s_i \in \{0, 1\} & \forall i \in L \\ 0 \leq x_{ai} \leq 1 & \forall a \in \mathcal{A}, i \in \mathcal{L}_a \end{cases}$$

The binary decision variable s_i for each potential sensor location $i \in L$ equals 1 if a sensor is placed at location i and 0 otherwise. The first set of constraints assures that exactly one sensor raises the alarm for (witnesses) each contamination scenario. The second set requires that a location cannot raise any alarm

unless there is an installed sensor. The last constraint enforces the limit on the total number of sensors. The objective function chooses the best (lowest impact, eligible given the constraints) sensor as a witness for each contamination scenario a.

BSP was first described by Berry et al. (2004). Remarkably, the BSP is identical to the well-known p-median facility location problem (Mirchandani and Francis, 1990). In the p-median problem, p facilities (e.g., central warehouses) are to be located on m potential sites such that the sum of distances d_{aj} between each of n customers (e.g., retail outlets) a and the nearest facility j is minimized. In contrasting the BSP and p-median problems, we observe equivalence between (1) sensors and facilities, (2) contamination scenarios and customers, and (3) contamination impacts and distances. While the BSP allows placement of *at most* p sensors, p-median formulations generally enforce placement of all p facilities; in practice, the distinction is irrelevant unless p approaches the number of possible locations.

Finally, we use a slightly revised formulation of BSP in teva-sp and our computational experiments. We have observed that for any given contamination scenario a, there are often many total impacts d_{aj} that have the same value. If the contaminant reaches two junctions at approximately the same time, then these two junctions could witness the contamination event with the same impact values. For example, this occurs frequently when we use a coarse reporting time-step with the water quality simulation. This observation has led us to consider the following generalization of BSP:

$$\text{(cBSP)} \quad \text{minimize} \quad \sum_{a \in \mathcal{A}} \alpha_a \sum_{i \in \hat{\mathcal{L}}_a} d_{ai} x_{ai}$$

$$\text{where} \quad \begin{cases} \sum_{i \in \hat{\mathcal{L}}_a} x_{ai} = 1 & \forall a \in \mathcal{A} \\ x_{ai} \leq s_i + \sum_{j \in \mathcal{L}_a \setminus \hat{\mathcal{L}}_a : d_{aj} = d_{ai}} s_j & \forall a \in \mathcal{A}, i \in \hat{\mathcal{L}}_a \\ \sum_{i \in L} s_i \leq p & \\ s_i \in \{0,1\} & \forall i \in L \\ 0 \leq x_{ai} \leq 1 & \forall a \in \mathcal{A}, i \in \hat{\mathcal{L}}_a \end{cases}$$

where $\hat{\mathcal{L}}_a \subseteq \mathcal{L}_a$ such that $d_{ai} \neq d_{aj}$ for all $i, j \in \hat{\mathcal{L}}_a$. This revised formulation treats sensor placement locations as equivalent if their corresponding contamination impacts are the same for a given contamination event. In doing so, the fundamental structure of this formulation changes only slightly, but this IP may require significantly less memory (by eliminating duplicate d_{ai} values). However, it is important to note that $cBSP$ and BSP have the same set of feasible solutions, so they can be used to find the same optimal sensor placements. In preliminary experiments, $cBSP$ was often ten times smaller than BSP, and corresponding reductions in optimization runtime have been observed.

We can achieve further reduction in problem size by stronger *witness aggregation*: grouping witnesses whose impacts are close to each other, but not equal. The impact of the group is the impact of its worst member (conservatively). The optimal solution to this problem is only approximately optimal for the original problem. We can trade off size of formulation (and speed of solution) for approximation quality.

For simplicity of presentation, our subsequent discussion will refer to BSP when describing these two formulations. However, the $cBSP$ is the actual MIP model used in teva-sp.

4 ROC-Based Imperfect Sensor Model

As suggested above, the new model presented in this section assumes that each sensor location has a false negative probability and a false positive rate. We assume that sensor failures are independent. We expect that the performance of a sensor will depend on its environment. It can also depend on the components within the sensor (if the sensor is really a package of multiple sub-sensors). However, for this first discussion, we will assume that there is only one type of sensor and only a small fixed number of environments (say, low, medium, and high water quality variation). Though a real network may have many microenvironments

depending upon the sensitivity of the sensor, we expect that initially at least, those who generate input data will use this kind of course granularity. We assume we can derive probabilities for false negatives and false positives (false alarm rate) from sources of sensor uncertainty data, such as ROC curves or vendor-specific time-between-failure information. We treat false negatives and false positives separately in Sections 4.1 and 4.2, respectively.

4.1 False Negatives

In the base sensor placement formulation (BSP), we place sensors to minimize the expected impact taken over the given contamination scenarios. Each scenario has a single (best) witness if sensors are perfect. Given a probability of detection (or sensitivity) at junction j, p_j, we can change the precise impact for a scenario to an expected impact. For scenario a, sort the potential witnesses by increasing impact (better witnesses have lower impact on the network and population). Let $\mathcal{L}_{aj} \subset \mathcal{L}_a$ be the set of junctions preceding and including junction j in the ordered witness list for the contamination scenario a (witnesses that would be at least as good as location j). Let x_{aj} be the probability that a sensor placed at junction j is the first to detect the contamination scenario. Thus the x_{aj} are no longer binary, but are still bound between 0 and 1. The objective function doesn't change.

Computing the witness probabilities directly involves products of the sensor-placement variables and is therefore nonlinear:

$$x_{aj} = p_j s_j \prod_{i \in \mathcal{L}_{aj} - \{j\}} (1 - p_i s_i),$$

Ignoring the sensor-placement variables s_i for the moment, this expression is the probability that all the witnesses that are better than j fail times the probability that the sensor at location j succeeds in detecting the event. We multiply by the sensor-placement variables to account for actual placement (a previous witness doesn't have to fail if it has no sensor; location j cannot succeed if it has no sensor).

We now describe one way to linearize these constraints for the x_{aj}. If the number of sensors and environments (number of probabilities/behaviors) is constant, then this linearization does not explode combinatorially. We demonstrate this for three classes of environments, which we equivalently call sensor types. This shows reasonable generality without being too cumbersome. Let the three false negative probabilities for types 1, 2, and 3 be p_α, p_β, and p_γ respectively.

Consider injection scenario a. This injection will have only one witness, but what the witness is now depends not only on the placement of the sensors but also on the (probabilistic) instantiations of the false negatives. Let's assume the set of potential witnesses for injection a, $\mathcal{L}_a$ are sorted by increasing impact, so the best witness is first in the list. If there is a sensor at that first location and the sensor detects injection a, then it is the witness. No other sensors have an opportunity to be a witness. If the first sensor fails, then the second sensor has an opportunity to be the witness. If it succeeds, then it is the witness with no other locations having a chance, and so on. Thus the possible outcomes of the random sensing events are disjoint.

For ease of exposition, we define a set of auxiliary variables z_{ai} to be the probability that all sensors in $\mathcal{L}_{ai}$ fail to detect injection a. This probability considers both sensor placement decisions and detection failure. In the actual IP model, we express the z_{ai} in terms of other variables. To compute the z_{ai}, with a polynomial number of variables, we take advantage of the small number of sensor types. We do not care which individual locations in $\mathcal{L}_{ai}$ have sensors, we only care about the number of each type that have sensors. In particular, if there are j type-1 sensors (with sensitivity p_α), k type-2 sensors (with sensitivity p_β), and l type-3 sensors (with sensitivity p_γ), then we have

$$z_{ai} = (1 - p_\alpha)^j (1 - p_\beta)^k (1 - p_\gamma)^l.$$

We now give constraints that allow the IP to determine these counts. This will require a number of related classes of new variables:

- Let c_{aijkl} be a binary variable that is 1 if and only if the IP places sensors so that locations $\mathcal{L}_{ai}$ have exactly j type-1 sensors, k type-2 sensors, and l type-3 sensors.
- Let $\mathcal{L}_{ai}^{\alpha} \subseteq \mathcal{L}_{ai}$ be the type-1 locations in $\mathcal{L}_{ai}$.
- Let $\mathcal{L}_{ai}^{\beta} \subseteq \mathcal{L}_{ai}$ be the type-2 locations in $\mathcal{L}_{ai}$.
- Let $\mathcal{L}_{ai}^{\gamma} \subseteq \mathcal{L}_{ai}$ be the type-3 locations in $\mathcal{L}_{ai}$.
- Let c_{aij}^{α} equal 1 if there are j type-1 sensors in $\mathcal{L}_{ai}$ and 0 otherwise.
- Let c_{aik}^{β} equal 1 if there are k type-2 sensors in $\mathcal{L}_{ai}$ and 0 otherwise.
- Let c_{ail}^{γ} equal 1 if there are l type-3 sensors in $\mathcal{L}_{ai}$ and 0 otherwise.

Any given $\mathcal{L}_{ai}$ and sensor placement s_i, has a specific count of sensor types. So exactly one of the c_{aijkl} is equal to 1. We do not need to enforce this explicitly since it will be implied by the other constraints.

We do, however, explicitly enforce this constraint on the indicator counts for the separate sensor types:

$$\sum_{j=0}^{|\mathcal{L}_{ai}^{\alpha}|} c_{aij}^{\alpha} = 1 \tag{1}$$

$$\sum_{k=0}^{|\mathcal{L}_{ai}^{\beta}|} c_{aij}^{\beta} = 1 \tag{2}$$

$$\sum_{l=0}^{|\mathcal{L}_{ai}^{\gamma}|} c_{aij}^{\gamma} = 1 \tag{3}$$

Equations 1-3 specify (decide) the number of sensors of each type in $\mathcal{L}_{ai}$. We make sure the individual sensor placement decisions give the correct counts of each type:

$$\sum_{j=0}^{|\mathcal{L}_{ai}^{\alpha}|} j c_{aij}^{\alpha} = \sum_{m \in \mathcal{L}_{ai}^{\alpha}} s_m \tag{4}$$

$$\sum_{k=0}^{|\mathcal{L}_{ai}^{\beta}|} j c_{aij}^{\beta} = \sum_{m \in \mathcal{L}_{ai}^{\beta}} s_m \tag{5}$$

$$\sum_{l=0}^{|\mathcal{L}_{ai}^{\gamma}|} j c_{aij}^{\gamma} = \sum_{m \in \mathcal{L}_{ai}^{\gamma}} s_m \tag{6}$$

For example, in equation 4, the left side is equal to the number of type-1 sensors selected by the c_{aij}^{α} variables and the right side is the number of type-1 locations that receive a sensor.

We now link the indicator variables for the individual sensor types to the overall count variables:

$$c^{\alpha}_{aij} = \sum_{k=0}^{|\mathcal{L}^{\beta}_{ai}|} \sum_{l=0}^{|\mathcal{L}^{\gamma}_{ai}|} c_{aijkl}, 0 \leq j \leq |\mathcal{L}^{\alpha}_{ai}| \quad (7)$$

$$c^{\beta}_{aik} = \sum_{j=0}^{|\mathcal{L}^{\alpha}_{ai}|} \sum_{l=0}^{|\mathcal{L}^{\gamma}_{ai}|} c_{aijkl}, 0 \leq k \leq |\mathcal{L}^{\beta}_{ai}| \quad (8)$$

$$c^{\gamma}_{ail} = \sum_{j=0}^{|\mathcal{L}^{\alpha}_{ai}|} \sum_{k=0}^{|\mathcal{L}^{\beta}_{ai}|} c_{aijkl}, 0 \leq l \leq |\mathcal{L}^{\gamma}_{ai}| \quad (9)$$

For example, Equation 7 enforces that if c^{α}_{aij} is 1 for some j (the IP has decided there will be j type-1 sensors in $\mathcal{L}_{ai}$), then the indicator variable specifying all counts $c_{aij'k'l'} = 1$ must have the appropriate type-1 count ($j' = j$).

Finally, we finish by linking across pairs of types:

$$c^{\beta}_{aik} + c^{\gamma}_{ail} - 1 \leq \sum_{j=0}^{|\mathcal{L}^{\alpha}_{ai}|} c_{aijkl}, 0 \leq k \leq |\mathcal{L}^{\beta}_{ai}|, 0 \leq l \leq |\mathcal{L}^{\gamma}_{ai} \quad (10)$$

$$c^{\alpha}_{aij} + c^{\gamma}_{ail} - 1 \leq \sum_{k=0}^{|\mathcal{L}^{\beta}_{ai}|} c_{aijkl}, 0 \leq j \leq |\mathcal{L}^{\alpha}_{ai}|, 0 \leq l \leq |\mathcal{L}^{\gamma}_{ai} \quad (11)$$

$$c^{\alpha}_{aij} + c^{\beta}_{aik} - 1 \leq \sum_{l=0}^{|\mathcal{L}^{\gamma}_{ai}|} c_{aijkl}, 0 \leq j \leq |\mathcal{L}^{\alpha}_{ai}|, 0 \leq k \leq |\mathcal{L}^{\beta}_{ai} \quad (12)$$

For example, Inequality 10 only has an effect if the IP has decided there will be k type-2 sensors and l type-3 sensors in $\mathcal{L}_{ai}$). In this case, then the indicator variable specifying all counts $c_{aij'k'l'} = 1$ must have the appropriate type-2 and type-3 counts ($k' = k$ and $l' = l$).

Once the (complete) count variables are correctly set, we can compute the z_{ai}:

$$z_{ai} = \sum_{j=0}^{|\mathcal{L}^{\alpha}_{ai}|} \sum_{k=0}^{|\mathcal{L}^{\beta}_{ai}|} \sum_{l=0}^{|\mathcal{L}^{\gamma}_{ai}|} c_{aijkl}(1-p_{\alpha})^{j}(1-p_{\beta})^{k}(1-p_{\gamma})^{l} \quad (13)$$

We are now prepared to compute (via constraints) the witness probabilities x_{ai}. We can compute the first witness probability directly, since the first sensor (as long as it exists), has an opportunity to sense the event. It succeeds with probability p_1 (where p_1 is one of the three sensing probabilities $p_{\alpha}, p_{\beta}, p_{\gamma}$.

$$x_{a1} = p_1 s_1 \quad (14)$$

Because there is only one witness for each injection (outcomes are disjoint), we have the following set of constraints:

$$z_{ai} = 1 - \sum_{r=1}^{i} w_{ar}, \forall i = 1 \ldots |\mathcal{L}_a| \quad (15)$$

This says that the probability we have not detected an event by the time the first i potential witnesses have had an opportunity to sense is equal to one minus the probability that one of these potential witnesses succeeded. Because the IP has "computed" each of the z_{ai} directly from the count variables and the value

of w_{a1} directly, these constraints completely determine the other witness probability variables in order from 2 on up.

Thus the formulation for incorporating only false negatives adds constraints 1-15 to the BSP formulation. We can remove the constraints $x_{ai} \leq s_i$, since those are implied by other constraints.

Unfortunately, this formulation has a large number of binary variables, so we must consider modifications to make it more practical. It's possible this version will be tractable for two sensor types.

4.2 False Positives

We can consider false positives under both normal operating conditions and during (or right before) an event.

4.2.1 Limiting False Positives under Normal Operation

We can enforce a global limit on the false positive rate. We assume we have (independent) false positive rates f_i for each possible sensor location i. These rates will be predicted by the ROC curves associated with each possible sensor location. Thus we simply enforce the false positive tolerance directly:

$$\sum_{i \in L} f_i s_i \leq \rho. \tag{16}$$

4.2.2 Penalizing False Positive Interference with Event Detection

We now consider the extremely unlikely case in which a false positive occurs during an injection event. Recall that we have assumed that the event detection thresholds are set to limit the false positive rates to very low numbers, such as one per three months. Still, if a false positive actually occurs during a real injection event, it would be a potentially serious problem. If we assume that the water utility is willing and able to send out only one truck to verify the event with a manual sample, it would be a disaster if, during a real injection, the truck responded to a false positive reading and went to an uncontaminated section of the network. The utility would have apparent confirmation of a false alarm, when a real event was in progress. Thus, we penalize such a false positive during an injection by forcing a worst case impact, as if the injection had never been detected. This holds even if other sensors later detect the injection.

Consider the faulty-sensor IP that handles false negatives and enforces a tolerance on false positives during normal operation. We assume all optimal solutions to that IP will have the false-positive constraint (16) satisfied with equality. That is, in order to minimize impact, the IP will almost certainly be pushing the boundary on false positives. Therefore, we assume that the false positive rate in the sensor system is simply the constant ρ. We can think of this as an arrival rate and compute (offline) the probability of a false positive during any particular time interval (this is independent of the actual time; as with all Poisson processes, it will depend only on the interval length). We denote this probability $p(t)$, where t is the length of the time interval in minutes.

We can model the effect of false positives that occur during the window of the injection. However, we currently only discuss the highest-order effect. If we assume a delay of a full day between responses, then any false positive occurring within 24 hours of the injection start time will remove any possibility of detecting that injection. The probability of this is $p(1440)$, given that there are 1440 minutes in a day. Therefore, we need only modify Equation 15, subtracting $p(1440)$ from the right side. This reflects the initially reduced probability of witnessing injection a.

5 A Local Search Heuristic for the Imperfect Sensor Model

In Watson et al. (2005), variations of a GRASP local search heuristic are applied to the sensor placement problem under perfect-sensor assumptions (Resende and Werneck, 2004). These heuristics operate by iteratively improving an initial (possibly random) solution, in this case a sensor placement, and then evaluating every "neighboring" solution. The neighborhood operation might consist, for example, of trying to replace each sensor in the current solution with a sensor at some currently uncovered location. The best neighboring solution then becomes the current solution, and the neighborhood search process is repeated until a locally optimal solution is identified.

Fortunately, the structure of the data associated with water sensor placement problems is remarkably amenable to solution via the GRASP heuristic. In fact, during the processing of many perfect-sensor problem instances on many different networks, we have not seen a case in which our MIP formulations have found a solution any better than the heuristic's solution. To phrase this differently, whenever the MIP has solved, it has proven the heuristic solution to be optimal. This applies to networks with thousands of nodes.

It is a straightforward matter to adapt the GRASP heuristic to handle the imperfect sensor case. The only difference is that the objective value of each neighboring solution is now computed using the imperfect sensor objective. Efficient computation of these objective values is possible, but the data structures and algorithms are beyond the scope of this conference paper.

As we shall see, the remarkable success of this local search heuristic continues with the new imperfect sensor model.

6 Experiments

We have implemented our imperfect sensor optimization models in three different ways:

- As a non-linear mathematical program
- As a MIP
- As a local search heuristic

In each of these implementations, we have omitted the logic associated with false positive detection. This logic could be included easily, but recall that we assume that the specificity imposed by human interpreters of ROC curves will be extremely high, making false positives extremely unlikely. In this context, we chose for our experiments to handle false negatives only.

Details of our non-linear mathematical program are omitted, as are results. Open-source non-linear solvers with default settings generally were not able to solve this formulation. Commercial solvers were able to solve the "small" and "medium" instances described in Section 6.2 below, but the resulting fractional solutions are not globally optimal, and therefore are not lower bounds on the optimal solution. With improved non-linear global optimizers, this non-linear formulation may become a powerful tool for bounding.

We had more immediate success in finding bounds with the linearized MIP formulation described in Section 4, though this success was limited to networks with hundreds of nodes rather than thousands. We were able to use this model to find optimal solutions. These solutions matched those of the revised local search heuristic, proving its output to be optimal for these instances.

6.1 Evaluating Solutions

Recall the formulation of the objective function in our models:

Witness	1	2	3	none
Probability (x_{ai})	0.3	0.2	0.15	0.35
Impact	100	200	300	5000

Table 1: Example computation of the objective function with imperfect sensors. The expected impact of this injection is $0.3(100) + 0.2(200) + 0.15(300) + 0.35(5000)$.

$$\sum_{a \in \mathcal{A}} \alpha_a \sum_{i \in \mathcal{L}_a} d_{ai} x_{ai}$$

In the context of perfect sensors, this says that the cost of a solution is the sum over all possible injections of the contaminant at each junction. Given a, exactly one of the witness variables x_{ai} is 1. In other words, a perfect sensor at location i witnesses the injection, and we are therefore certain (up to the quality of the simulations) that the impact of that injection is d_{ai}. Note that d_{ai} may be an impact incurred after some assumed response delay. Still, knowing that the sensor at location i is perfect, we know when to start the response timer.

When we have imperfect sensors, this objective function is interpreted differently. In particular, the witness variables x_{ai} are now witness probabilities. For a trivial example, consider Table 1. A hypothetical injection hits three nodes at locations that could host a sensor. After hitting these three, it may continue to impact them for some time, and may hit other nodes that can't host sensors. In the table, we see that a hypothetical solver has found that location 1 has a 30% chance of being the first witness of the contaminant, location 2 has a 20% chance of being the first witness, location 3 has a 15% chance, and there is a 35% chance that all three will fail. The expected impact of this injection, therefore, is a discrete random variable, and its expected value is

$$0.3(100) + 0.2(200) + 0.15(300) + 0.35(5000) = 1865.$$

The imperfect sensor solution, therefore, is an expected value of the expected values of the impacts of the individual injections.

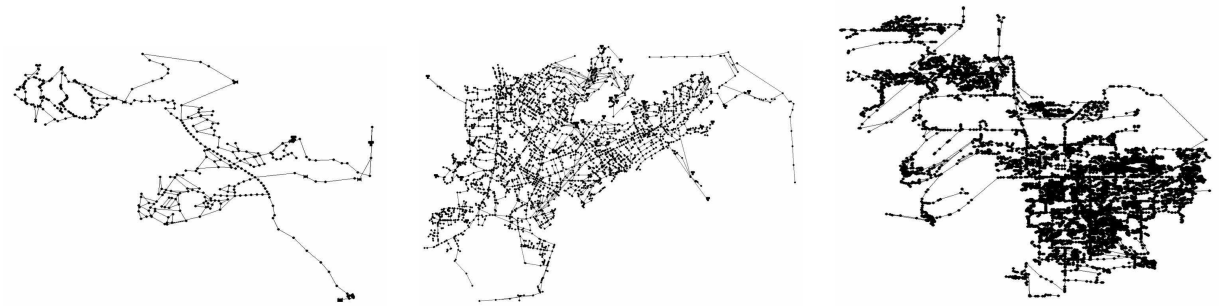
A sensor placement found by any of our models (perfect or imperfect sensors) may be evaluated in the context of either perfect or imperfect sensors. Our methodology for measuring the value of the imperfect sensor model will leverage this property.

6.2 Test Instances

We evaluate our models on three different water networks: a "small" instance with about 400 nodes, a "medium" instance with about 3000 nodes, and a "large" instance with about 11000 nodes. Figure 1 shows morphed pictures of these networks to give some idea of their topology without revealing true coordinates. The latter are significantly different from those shown in the pictures. These are the same three instances used in Berry et al. (2006). As in that work, for each instance we create injection ensembles containing injections at each non-zero demand node, for a single start time. The injection duration is 24 hours, and the strength is a large number of cells per liter (which we omit intentionally).

For each instance, we consider two different sets of three detection classes:

- false negative probabilities 0.25, 0.5, and 0.75.
- false negative probabilities 0.7, 0.75, and 0.80.



(a) (b) (c)

Figure 1: Three network instances: (a) a “small” instance with about 400 nodes, (b) a “medium” instance with about 3000 nodes, and (c) a “large” instance with about 11000 nodes.

Network	Perf. Sens. Sol.(PSS)	PSS/Imp. Obj.	Imperf. Sens. Sol.	%of PSS impact.
small	63.25*	298.55	182.35*	61%
medium	1061.09*	2869.62	2288.43	80%
large	518.29	1070.62	720.82	67%

Table 2: Results for the number of people sickened during a biological event under sensor placement found by perfect-sensor and imperfect sensor models. The third column gives impacts with sensors placed according to the perfect sensor solution. The fourth column lists the result of evaluating the perfect sensor solution in the context of sensor failures. The fifth column shows the evaluation of the imperfect sensor solution, and the fifth shows the benefit of using the imperfect sensor formulation. The asterisks indicate solutions that have been proven to be optimal by our MIP formulations.

The network nodes were partitioned into three nearly equal-sized groups in an arbitrary way (by node label order).

The results presented in Section 6.3 are compiled under the admittedly unrealistic assumption that there is zero response delay. In other words, the instant that a sensor happens to detect the injection all impacts stop accumulating. We discuss the implications of this assumption and removing it below.

We consider population to be highly correlated to demand. The “people sickened” figures below make use of this assumption rather than census or billing data.

6.3 Results

We find that the ROC-based imperfect sensor MIP is scalable up to the small instance of a few hundred nodes, but not beyond that without further work. However, the local search heuristic can find provably optimal solutions for these small instances. Further, it can find solutions for the medium and large instances that are likely to be optimal or nearly optimal.

The key question is whether the extra effort involved in modeling imperfect sensors is justified. Are sensor placements found by the perfect sensor models poor enough, when evaluated in the context of sensor failures, to make the added complexity of the imperfect sensor model worthwhile? We explore this question presently by discussing the results shown in Table 2. The numbers presented in this table represent the aggregate population sickened (but not necessary killed) by the contaminant, in this case a biological agent. We use the Murray et al. (2006) health impacts model to calculate the expected number of people exposed to and sickened by the agent.

Network	num samples	Perf. Sens. Sol.(PSS)	Imperf. Sens. Sol.	ROC-based MIP bound
small	5	0.14s	0.39s	13070s
medium	5	8.74s	247s	NA
large	1	36.36s	2155s	NA

Table 3: Running times for the local search heuristic in its perfect sensor and imperfect sensor forms. Note that although the times increase for the imperfect sensor form, they are still easily tractable even for large networks. The number for the ROC-based MIP is the time until its lower bound equals the heuristic solution.

Consider the small network. The expected impact over all injections is about 63 people sickened. Taking this same sensor placement, and calculating the expected impact when sensors can fail, we find a more realistic expected impact of about 298 people sickened. However, the optimal solution, when failures are allowed, is about 182 people sickened. Thus, placing sensors with the imperfect sensor model saves an expected 116 people from exposure over the perfect sensor model. We obtain similar results for the medium and large instances. Significant fractions of the people who would have been exposed under the perfect sensor model are spared under the imperfect sensor model.

Table 3 shows timing results for the local search heuristic and ROC-based MIP on these problems. Note that the MIP takes many hours to establish a lower bound that matches the heuristic solution, thus proving it optimal. It becomes too large to solve at all for the medium and large instances. The heuristic, on the other hand, easily handles even the largest of these instances in less than one hour.

We give three caveats concerning these results:

- Large numbers of sensors to be placed will decrease the apparent advantage of the imperfect sensor model.
- Long response delays will also decrease this advantage.
- Other objectives such as mass of contaminant consumed or extent of contamination may have smaller relative advantages.

However, considering that sensor installations are very expensive, we don't expect to be granted budgets for hundreds of fixed sensors. As for the second point, a long response delay affects more than the relative advantage of imperfect sensor models. If the delay extends beyond a day or two, the whole enterprise of placing sensors becomes questionable. Much of the potential impact mitigated by sensors disappears with long response delays. We plan to continue our comparisons between perfect and imperfect sensor models in the presence of moderate response delays in the full paper.

We mention other objectives simply because we have anecdotal evidence that the population exposed numbers we have given show a greater relative advantage for the imperfect sensor model than we see in preliminary trials with other objectives. However, those trials were with larger numbers of sensors. We will explore this issue further in the full paper.

7 Summary

We have presented a naturally non-linear formulation of the problem of optimally placing water sensors when sensors may fail. We linearized this model to produce a MIP and used that formulation to prove the optimality of heuristic solutions generated for a small instance with several hundred nodes. We have also demonstrated that the heuristic solutions can be generated efficiency and can add significant value in terms of reduced expected impacts.

Acknowledgments

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. This work was funded through an Interagency Agreement with the United States Environmental Protection Agency.

References

Berry, J., Fleischer, L., Hart, W. E., Phillips, C. A., and Watson, J.-P. (2005). Sensor placement in municipal water networks. *J. Water Planning and Resources Management*, 131(3):237–243.

Berry, J., Hart, W., Phillips, C., Uber, J., and Watson, J. (2006). Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management: Special Issue on Drinking Water Distribution Systems Security (To Appear)*.

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

Klise, K. and McKenna, S. (2005). Water quality change detection; multivariate algorithms. In *Proc. SPIE Conference on Defense and Homeland Security*, volume 6203.

McKenna, S., Wilson, M., and Klise, K. (2006). Detecting changes in water quality data. *AWWA Journal, submitted*.

Mirchandani, P. and Francis, R., editors (1990). *Discrete Location Theory*. John Wiley and Sons.

Murray, R., Uber, J., and Janke, R. (2006). Modeling acute health impacts resulting from ingestion of contaminated drinking water. *Journal of Water Resources Planning and Management: Special Issue on Drinking Water Distribution Systems Security (To Appear)*.

Resende, M. and Werneck, R. (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1):59–88.

Rossman, L. A. (1999). The EPANET programmer's toolkit for analysis of water distribution systems. In *Proceedings of the Annual Water Resources Planning and Management Conference*.

Watson, J.-P., Hart, W. E., and Berry, J. (2005). Scalable high-performance heuristics for sensor placement in water distribution networks. In *Proc. World Water and Environment Resources Conference*.