Sensor Placement in Municipal Water Networks with Temporal Integer Programming Models

Jonathan Berry* William E. Hart† Cynthia A. Phillips‡
James G. Uber§ Jean-Paul Watson¶

Abstract

We present a mixed-integer programming (MIP) formulation for sensor placement optimization in municipal water distribution systems that includes the temporal characteristics of contamination events and their impacts. Typical network water quality simulations track contaminant concentration and movement over time, computing contaminant concentration time-series for each junction. Given this information, we can compute the impact of a contamination event over time and determine affected locations. This process quantifies the benefits of sensing contamination at different junctions in the network. Ours is the first MIP model to base sensor placement decisions on such data, compromising over many individual contamination events. The MIP formulation is mathematically equivalent to the well-known p-median facility location

*Discrete Algorithms and Math Dept, Sandia National Laboratories, Mail Stop 1110, P.O. Box 5800, Albuquerque, NM, jberry@sandia.gov

†Discrete Algorithms and Math Dept, Sandia National Laboratories, Mail Stop 1110, P.O. Box 5800, Albuquerque, NM; Ph: 505-844-2217; wehart@sandia.gov

‡Discrete Algorithms and Math Dept, Sandia National Laboratories, Mail Stop 1110, P.O. Box 5800, Albuquerque, NM, caphill@sandia.gov

§USEPA, Cincinnati, OH and Dept. of Civil Engineering, University of Cincinnati, Cincinnati, OH; PH (513)569-7974 uber.jim@epa.gov.

¶Discrete Algorithms and Math Dept, Sandia National Laboratories, Mail Stop 1110, P.O. Box 5800, Albuquerque, NM, jwatson@sandia.gov

problem. We can exploit this structure to solve the MIP exactly or to approximately solve the problem with provable quality for large-scale problems.

Keywords: Safety, Terrorism, Optimization, Optimization models, Municipal water, Water management

Introduction

Public water distribution systems are inherently vulnerable to accidental or intentional contamination because of their distributed geography. The use of on-line, real-time early warning systems (EWSs) is a promising strategy for mitigating these risks. The general goal of an EWS is to identify a low probability and high impact contamination incident while allowing sufficient time for an appropriate response that mitigates any adverse impacts. An EWS may complement 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 – and the deployment of robust EWSs to monitor drinking water in the distribution system remains a future goal.

A key element of the design of an effective EWS is the strategic placement of sensors throughout the distribution network. A variety of technical approaches have been developed to formulate and solve sensor placement problems in water networks, including mixed-integer programming (MIP) models (Berry et al., 2005; Lee et al., 1991; Lee and Deininger, 1992; Propato et al., 2005; Watson et al., 2004), combinatorial heuristics (Kessler et al., 1998; Kumar et al., 1999; Ostfeld and Salomons, 2004), and general-purpose metaheuristics (e.g., Ostfeld and Salomons (2004)). MIPs can often be solved to optimality in practice.

In this paper, we describe a MIP formulation for sensor placement optimization in water distribution networks that incorporates information about the temporal characteristics of a contamination event, as obtained from standard network simulation models. The water quality simulations compute contaminant concentration time-series for each junction in

a network. These time-series are then used to estimate the impact of the contamination event, including the impact of detection at different junctions in the network. Based on contaminant movement data, the MIP estimates whether contaminant arrives at a population center before the contaminant is detected. Moreover, the fidelity of the water quality simulations and corresponding contaminant impact calculations is independent from the MIP model. Thus our MIP model can be used with a diverse set of sensor placement objectives, and improvements in water quality simulations can be incorporated – without changing the underlying sensor placement solution strategy.

We show how to solve this MIP model with both heuristic and exact combinatorial solvers for large-scale, real-world sensor placement problems. Our computational results show GRASP heuristic is robust and scalable. Thus it is practical in contexts where many sensor placement analyses are required for trade-off studies. Furthermore, since this heuristic is solving a MIP model, we can bound the quality of heuristic solutions using the optimal value of linear programming relaxations of the MIP model. Thus, the structure of our modeling approach can be leveraged in many different ways.

The next section reviews the previous literature on combinatorial sensor placement formulations for water distribution networks. Section describes our MIP formulation, as well as a revision of this formulation that facilitates its application to large-scale problems. Section describes the application of exact and heuristic solvers to large sensor placement problems. Finally, we discuss the significance of these results in Section .

Background

Sensor placement problems can be naturally formulated as optimization problems. Although our focus is on detecting contamination events within an EWS, methodologies for placing water quality monitoring stations are related to sensor placement problems for EWS design. Consequently, we include them in our comparison of modeling approaches, and for simplicity of presentation we refer to the placement of water quality monitoring stations as a sensor placement problem.

For EWS design, the general goal of sensor placement optimization is to place a limited number of sensors in a water distribution network such that the impact to public health of an accidental or intentional injection of contaminant is minimized. However, there is no specific formulation of the problem that is widely accepted by the water resources management community. A major contributing factor is the wide range of design objectives that are important when considering sensor placements, e.g., minimizing the cost of sensor installation and maintenance, the response time to a contamination event, and the extent of contamination – which impacts recovery costs. Additionally, it is difficult to quantify precisely the health impact of a contamination event; human water usage is often poorly characterized, in terms of both water consumption patterns and how water consumption impacts population health. Consequently, surrogate measures like the total volume of contaminated water demanded have been used to model health impacts; this measure assumes that human water consumption is proportional to water demand.

One common feature of sensor placement formulations is the simplifying assumption that sensors can accurately measure water quality and/or the presence of contaminants. Although this may be reasonable for water quality measurements, it remains unclear how well this assumption will apply to EWS design activities. New sensor technologies are needed to detect contaminant threats, but the robustness and accuracy of these contaminant-specific sensors remains unclear.

For the computational studies in this paper, uncertainty in our solutions is largely caused by uncertain simulator input data and any inaccuracies of contaminant fate and transport modeling, not by limitations in solution technology for the mathematical optimization problem. Future improvements in data collection or simulation will automatically improve the quality of our sensor placement.

Given a particular contamination scenario, a given sensor placement will either detect contamination as it is transported through the network, or not. Contamination scenarios are drawn from a theoretically infinite combination of contaminant type (including fate and transport characteristics); contaminant source location, mass flow rate, time of day, and

duration; and network operating condition. All sensor-placement work to date assumes a particular finite set of scenarios, upon which the solution is conditioned. We have identified three properties that can distinguish such scenario-based sensor placement models:

Transport (external vs. internal) In an *internal-transport* model, the impact of a contamination event is embedded in the formulation through explicit constraints incorporating pipe network topology, flow directions, and travel times. In an *external-transport* model, a pre-processing step uses standard water quality simulation models to predict contaminant concentration time-series resulting from particular contamination scenarios. The formulation thus implicitly incorporates the physical and chemical principles of the simulation model, by incorporating the calculated impact of a contamination event given detection by a sensor. The implicit external-transport formulations can more easily capture realistic temporal evolution of contamination concentrations, because the formulation size does not depend directly on water quality simulation details (e.g., number of time steps or demand changes). In contrast, for computational reasons the explicit internal-transport formulations may assume one or more distinct patterns of network flows, ignoring the effect of temporal flow transitions on the evolution of contaminant concentration.

Scenario Handling (raw vs. summary) Sensor placement models accept as input either raw information about each contamination event or summary information (such as averages) derived from the full set of events.

Temporal (vs. nontemporal) Nontemporal models do not explicitly represent time. Instead, the concept of node protection is structural; nodes are protected only by sensors that are physically between the injection point and themselves.

Almost all of the research on sensor placement optimization has considered internal-transport formulations. These differ from each other by (1) the design and/or performance objective considered and (2) how network flows are modeled. Lee et al. (1991; 1992) developed a formulation related to a set covering problem. Subsequently several researchers

refined the model (see Ostfeld and Salomons (2004) for a review). These researchers used network flow information to compute the fraction of water flow that passes between any pair of junctions, and then placed sensors to maximize the coverage of water flow.

Kessler et al. (1998) and Ostfeld and Kessler (2001) introduced an internal-transport, summary-scenario formulation in which the objective is to ensure a pre-specified maximum volume of contaminated water consumed prior to detection. This formulation is based on a set covering problem. For each contamination event, there must be a sensor on some junction that will detect the event early enough to meet a pre-specified "level of service." Kessler et. al. estimate contaminant consumption using an auxiliary network determined via analysis of hydraulic simulation outputs. The network has a directed edge from junction v_i to v_j if there is flow from v_i to v_j at any point in the simulation. These directed edges are weighted by the *average* velocity from v_i to v_j over the course of simulation. Kessler et. al. use this auxiliary graph along with network pipe lengths to estimate the shortest travel time between all pairs of vertices in the original water network.

Berry et al. (2003; 2005) and Watson et al. (2004) introduce a variety of internal-transport formulations expressed as MIPs. In Berry et al. (2003; 2005), the objective is to minimize the expected fraction of the population exposed to a contamination event. They use hydraulic simulation results to compute a fixed flow orientation for each pipe in the network for each of p distinct non-overlapping time intervals, referred to as patterns. A population consuming water at a junction v_j is considered exposed to contaminant injected at a junction v_i if and only if there exists a flow path from v_i to v_j along which there is no sensor. Thus this is a nontemporal model. Watson et al. (2004) generalize this formulation to consider a range of optimization objectives, some of which account for travel times by considering contaminant propagation rates within each flow pattern separately, as opposed to aggregating these into a single auxiliary network as is done by Kessler et al. (1998).

Ostfeld and Salomons (2004) propose a model most similar to ours. It is an external-transport raw-scenario model. Their objective is to ensure that the expected impact of a contamination event is within a pre-specified "level of service", defined as the maximum vol-

ume of sufficiently contaminated water consumed prior to detection. Mirroring the earlier approach of Kessler et al., Ostfeld and Salomons introduce a formulation based on set covering, in which sensors are allowed to cover only those junctions for which detection can be guaranteed within the pre-specified level of service. They approximately solve the problem using a genetic algorithm, a method that provides no solution quality guarantees.

The sensor placement model described in this article was first presented by Berry et al. (2004). This model was recently refined by Propato et al. (2005), who consider an alternate formulation of the underlying combinatorial structure of the sensor placement problem. They argue that the structure of this formulation can be exploited to limit the number of water quality simulations that are needed for sensor placement when contaminant dynamics are conservative or first-order.

The MIP Model

We now introduce our MIP 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; installation of sensors on pipes is disallowed because we rely on water quality simulations that cannot provide this information. We assumed 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 quadruples 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. For a given contamination scenario, we use water quality anlysis 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.

Let $d_a(t)$ be the total network-wide impact of a contamination scenario a at any given point in time $t \geq t_s^a$. We defer precise specification of an "impact" to Section ; a key characteristic of our formulation is that it captures a wide range of possible definitions. 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. Let $d_{aj} = d_a(\gamma_{aj})$ be the total impact of 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. Thus d_{aq} is 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. Each

contamination scenario $a \in A$ has a likelihood α_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.

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

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

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

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 (on average) is credited with raising the alarm for each contamination scenario. The second set forbids a location from raising an alarm if there is no sensor installed there. The last constraint enforces the limit on the total number of sensors. Consider an optimal solution to DSP (binary choices for s_i). If the impacts are all non-negative, then for scenario a, the set of locations i such that $x_{ai} > 0$ all have the same (minimum) impact.

A special case of DSP was first described by Berry et al. (2004). Remarkably, the DSP 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 comparing the DSP 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 DSP 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.

We used a slightly revised formulation of DSP in 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 a coarse reporting time-step is used with the water quality simulation. This observation has led us to consider the following reformulation of DSP:

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

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

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, every feasible solution for DSP has a corresponding solution in $cDSP$ with the same sensor placement. The only difference is that if sensor s_i is the witness for attack a, the IP might "credit" the observation to a non-existant sensor s_j with the same impact value (setting $x_{aj} = 1$ rather than $x_{ai} = 1$). We can always map the selected observation variable to a real sensor with the same impact. Because the impact for each attack is the same, the objective value is the same, so we can use $cDSP$ to find optimal sensor placements. In preliminary experiments, $cDSP$ was often ten times smaller than DSP, and we have observed corresponding reductions in optimization runtime.

For simplicity of presentation, our subsequent discussion will refer to DSP when describing MIP formulations. However, the $cDSP$ is the actual MIP model used in our experiments.

Empirical Results

We now describe the application of MIP and heuristic solvers for DSP on a number of large-scale, real-world water distribution networks. We describe a heuristic search strategy in Section , our methodology and test networks in Section , and results for MIP and heuristic approaches in Section and Section , respectively.

Solution Methods

Equivalence with the p-median problem has an immediate bearing on our approach to solving DSP, because we can directly leverage the extensive literature on algorithms for solving the p-median problem. The p-median problem can in principle be solved with a MIP solver. Further, optimal integer solutions frequently result by relaxing the integral constraints and solving the corresponding pure linear program (LP) (ReVelle and Swain, 1970). However, heuristics are often used in practice when dealing with large problem instances due to the rapid growth in the number of constraints and variables as problem size increases.

The current state-of-the-art heuristic for the p-median problem is a hybrid approach recently introduced by Resende and Werneck, which we denote RW. The core mechanism underlying RW is a Greedy Randomized Adaptive Search Procedure (GRASP), which is used to generate a set of high-quality solutions using biased greedy construction techniques. Steepest-descent hill-climbing is then used to move from each of the resulting solutions to a local optimum. Finally, path relinking is used to further explore the set of solutions lying at the intersection of the resulting local optima. For a complete description of RW, we refer the reader to (Resende and Werneck, 2004).

Methodology and Test Problems

Each contamination scenario consists of a single EPANET mass injection event of rate 5.7E10 and duration 12 hours. The duration of the entire simulation is 96 hours. EPANET (Rossman, 1999) is used to perform water quality simulations for each contamination scenario, and the resulting concentration time-series τ_j are used to compute the impact factors d_{aj} for each combination of $a \in \mathcal{A}$ and $v_j \in V$. Simulations begin at time $t_s^a = 0$ and the 96 hour duration covers multiple iterations of a daily demand cycle.

Although the objective function of DSP allows for meaningful contamination probabilities α_a, for simplicity our experiments address only the case in which $\alpha_a = 1/|\mathcal{A}|$. The impact values d_{ai} are obtained from water quality simulations performed by the EPANET toolkit, which has been instrumented for the objective of contaminant mass consumed. Other objectives, such as population exposed, number of failed detections, etc., are addressed by generating different impact numbers, not by changing the MIP structure.

We consider three real-world test networks, which we denote SNL-1, SNL-2, and SNL-3. These networks respectively contain roughly 400, 3000, and 12000 junctions, and 450, 4000, and 14000 pipes. The actual identities, exact dimensions, and pump/valve/tank/reservoir/well counts of these networks are withheld for security purposes. We observe that these models are *not* all-pipes models; the complexity is strictly due to size of the region served by the particular utilities from which the models were obtained. The numbers of nonzero demand junctions for these three networks are 105, 1621 and 9705, respectively. For each network, contamination event start times were set to $t_s = 0$, i.e., there is a single attack scenario per non-zero demand node. The EPANET water quality reporting step for all networks equaled 5 minutes.

SNL-3 is an order of magnitude larger than any network previously considered in the sensor placement optimization literature, and SNL-1 is an order of magnitude larger than that typically investigated. The largest network considered in most analyses (e.g., see Kessler et al. (1998) and Ostfeld and Salomons (2004)) is the "Anytown U.S.A." network (Walski et al., 1987), which consists of 34 pipes, 16 junctions, two tanks, one pump, and one well. Berry

et al. (2004) solve DSP with a MIP solver for on a network containing roughly 450 junctions and 600 pipes. Watson et al. (2004) solve internal-transport (sometimes nontemporal) MIP models with MIP solvers, using both the smaller 450 junction network in addition to a larger network with roughly 3500 junctions.

We conducted all experiments on a dual-processor 64-bit 2.2GHz AMD Opteron Linux workstation with 20 GB of RAM and 60GB of total (RAM plus swap) memory. The execution of the water quality simulations to compute impact values required non-trivial amounts of computation. For SNL-1, SNL-2, and SNL-3, the respective mean times required to perform a water quality analysis for a single contamination scenario are approximately 0.75, 1.25, and 4 seconds using EPANET. The run-times required to obtain the full suite of water quality simulations range from under an hour for SNL-1 to over 2 days for SNL-3. However, this computation could be easily parallelized across a set of standard workstations.

Solution via Mixed-Integer Programming

We solved DSP problems with ILOG's AMPL/CPLEX 9.1 MIP solver, which is a state-of-the-art MIP solver. We computed optimal solutions to cDSP for each of our test networks for a range of sensor budgets, which were selected to be realistic examples of what might be used in practice. The computational results for 20 sensors are shown in Table 1. In this table, the linear programming statistics describe the constraint matrix: the number of rows, columns and non-zero coefficients in the constraint matrix. These results are typical of those for other sensor budgets. All MIPs for SNL-1 and SNL-2 solved without branching.

Cplex could not solve the cDSP formulation for largest problem (SNL-3) on a standard (32-bit) workstation since it required 2.5Gb of memory. It could not solve the larger DSP formulation even on our high performance 64-bit machine. The solution of cDSP for SNL-3 with 4 contamination times per junction required roughly 2 hours of running time and 9 gigabytes of RAM. Exploring another scalability dimension, allowing 96 different contamination times per junction on a network with roughly 3500 junctions required roughly 4 days of CPU time and more than 20 gigabytes of RAM.

Another scalability issue concerns the fidelity of the data used for cDSP. In order to make the water quality simulations more generally useful, our data collection process involves an intermediate step in which the concentrations at each junction at each reporting step are saved. These files become prohibitively large when small reporting steps are used in conjunction with large models. In particular, in our experiments with SNL-3, our choice of a 60 minute reporting step reduced the space requirements for concentration data, but also increased the number of indistinguishable "first hits" at each reporting step.

Solution via the *RW* Heuristic

Next, we consider the performance of the *RW* heuristic on each of our test networks; the results are reported in Table 2. On both SNL-1 and SNL-2, *RW* executes in negligible runtimes and requires at most modest amounts of memory. Further, the solutions generated by the heuristic are provably *optimal*; the impact is equivalent to that yielded by the exact MIP solvers, as obtained during the course of the experiments described in Section . Although not reported, we observe identical behavior on a range of sensor budgets. Relative to the MIP solver, results can be an order of magnitude or more faster, and require no more total memory. However, it is important to note that the heuristic cannot in isolation *prove* the optimality of its result.

On SNL-3, the *RW* heuristic generates a final solution in roughly 2.5 minutes, while requiring 2.5 GB of RAM. In contrast, the MIP approach to solving the same test network consumed roughly the same amount of total memory in 13.5 minutes. The heuristic's advantages are more pronounced on experiments with larger numbers of attacks. Furthermore, *RW* has always generated provably optimal solutions for our experiments on real networks.

Conclusions

The DSP/cDSP MIP formulation represents a natural evolution of combinatorial modeling for contaminant sensor placement. DSP is a particularly interesting MIP formulation because

the value of solutions to this MIP are exactly the same as if the solution was evaluated independently. Consequently, with DSP we can disassociate issues related to combinatorial modeling and water quality analysis. For example, current modeling limitations with DSP are now principally due to the fidelity of the water quality simulations or invalid assumptions relating to the attack scenario, sensor behavior, or emergency response protocols.

Our experiments demonstrate that exact and heuristic solvers can be effectively applied to DSP instances that are at least an order of magnitude larger than problems commonly used in the water distribution community literature. Most research on sensor placement methods has focused on small-scale water distribution networks with at most 200 junctions and pipes. We have demonstrated that exact and heuristic solvers can compute optimal solutions to cDSP for networks with ten thousand junctions, with reasonable computational effort.

Our successful application of cDSP and p-median heuristics to DSP takes advantage of the mathematical structure in this problem to (a) significantly reduce the cost of evaluating sensor placement ensembles, (b) tailor the heuristic to the particular structure of this application, and (c) rigorously assess whether the value of the final solution is close to the value of an optimal solution. This heuristic is qualitatively different from general-purpose simulation-based optimization methods that have been previously considered for sensor placement (e.g. Ostfeld and Salomons (2004)) by not treating the search strategy as an outer loop around the water quality analysis. Instead, we precompute the impact of contamination scenarios on the entire network, and then re-use these values to assess the quality of any ensemble of sensors. Additionally, our heuristic optimizer is specifically tailored to the p-median structure of DSP. Thus we can reasonably expect that it will outperform general-purpose heuristics, particularly because the p-median problem is well-studied. Finally, we have leveraged the fact that the LP-relaxation of the DSP MIP formulation can provide performance bounds for the final solution provided by the heuristic. Although many authors have claimed that their sensor placement heuristics are capable of locating optimal solutions, we can quantify how close to optimality our heuristic has achieved in a rigorous fashion, either by comparing

with an optimal solution provided by a MIP solver, or by comparing with an LP-bound. Taken together, these observations have enabled the effective application of the p-median heuristic to large-scale water networks that are 500 times larger than the largest water network considered by Ostfeld and Salomons (2004), who consider the application of a genetic algorithm to a similar sensor placement formulation.

Scalability challenges remain a critical research focus, even for these methods. To adequately account for important temporal effects, we may need to (1) use a large number of attack scenarios and (2) use a small water quality reporting step. These factors dramatically increase the size and difficulty of DSP. The number of attack scenarios determines the number of water quality simulations used to define a DSP instance. Furthermore, the water quality reporting step impacts the number of variables and constraints in the DSP formulation. Addressing these scalability issues will require the development of techniques to perform parallel simulations and to solve large instances of DSP. For example, we expect the methodology used to formulation cDSP could be generalized to formulate reduced-fidelity MIP models that can be solved much more efficiently.

Our experience with the *RW* algorithm suggests that this heuristic method is not as sensitive to these scalability challenges as the MIP solvers for DSP. Specifically, *RW* appears less sensitive to the number of attack scenarios used in a DSP formulation. We expect *RW* to be able to solve most large-scale problems using high-performance 64-bit workstations. Even in cases where the memory requirements are large, this methodology can trade-off runtime for maximum memory utilization. As detailed in (Resende and Werneck, 2004), the large memory requirements are due to pre-computations that yield significant run-time improvements. Consequently, it is therefore possible to take the complementary approach and sacrifice run-time for reduced memory requirements.

Finally, related research on more fundamental modeling issues is also needed to make the application of DSP effective in practice. For example, methods to address solution robustness (Carr et al.), worst-case optimization objectives, and multiple-objective analysis (Watson et al., 2004) are needed. Practical assessment of sensor placement configurations will

require the analysis of trade-offs between sensor placement objectives, as well as assessments of solution robustness to data variabilities.

Acknowledgements

We thank Phil Meyers at Pacific Northwest National Laboratory for noting that DSP is equivalent to the p-median facility location problem. 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.

References

Berry J, Fleischer L, Hart WE, and Phillips CA. Sensor Placement in Municipal Water Networks. In Bizier P and DeBarry P, editors, *Proc. World Water and Environmental Resources Congress*. American Society of Civil Engineers, 2003.

Berry J, Fleischer L, Hart WE, Phillips CA, and Watson JP. Sensor Placement in Municipal Water Networks. *J. Water Planning and Resources Management*, 131(3):237–243, 2005.

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

Carr R, Greenberg HJ, Hart WE, Konjevod G, Lauer E, Lin H, Morrison T, and Phillips CA. Robust optimization of contaminant sensor placement for community water systems. *Mathematical Programming B*. (to appear).

Drage BE, Upton JE, and Purvis M. 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.

Kessler A, Ostfeld A, and Sinai G. Detecting Accidental Contaminations in Municipal Water Networks. *Journal of Water Resources Planning and Management*, 124(4):192–198, 1998.

Kumar A, Kansal ML, and Arora G. Discussion of 'Detecting Accidental Contaminations in Municipal Water Networks'. *Journal of Water Resources Planning and Management*, 125(4):308–310, 1999.

Lee BH and Deininger RA. Optimal Locations of Monitoring Stations in Water Distribution System. *Journal of Environmental Engineering*, 118(1):4–16, 1992.

Lee BH, Deininger RA, and Clark RM. Locating Monitoring Stations in Water Distribution Systems. *Journal, Am. Water Works Assoc.*, pages 60–66, 1991.

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

Ostfeld A and Kessler A. Protecting urban water distribution systems against accidental hazards intrusions. In *Proceedings IWA Second Conference*. IWA, 2001. CD-ROM.

Ostfeld A and Salomons E. Optimal Layout of Early Warning Detection Stations for Water Distribution Systems Security. *Journal of Water Resources Planning and Management*, 130(5):377–385, 2004.

Propato M, Piller O, and Uber J. A Sensor Location Model to Detect Contaminations in Water Distribution Networks. In *Proc. World Water and Environmental Resources Congress*. American Society of Civil Engineers, 2005.

Resende M and Werneck R. A Hybrid Heuristic for the p-Median Problem. *Journal of Heuristics*, 10(1):59–88, 2004.

ReVelle C and Swain R. Central Facilities Location. *Geographical Analysis*, 2:30–42, 1970.

Rossman LA. 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`.

Schmitz P, Krebs F, and Irmer U. 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.

Stoks PG. Water Quality Control in the Production of Drinking Water from River Water. In Adriaanse M, van der Kraats J, Stoks P, and Ward R, editors, *Proceedings: Monitoring Tailor-made*. RIZA, Lelystad, The Netherlands, 1994. (ISBN 9036945429).

Walski T, Brill E, Gessler J, Goulter I, Jeppson R, Lansey K, Han-Lin L, Liebman J, Mays L, Morgan D, and Ormsbee L. Battle of the Network Models: Epilogue. *Journal of Water Resources Planning and Management*, 113(2):191–203, 1987.

Watson JP, Greenberg HJ, and Hart WE. A multiple-objective analysis of sensor placement optimization in water networks. In *Proc. World Water and Environment Resources Conference*. 2004.

Case Study

We illustrate our techniques by applying them to the familar "EPANET Example 3." This is a network with 97 nodes and 116 pipes. In order to prepare for a run of cDSP, we run an EPANET hydraulic simulation, then 236 EPANET water quality simulations: one for each of the 59 non-zero demand nodes, for each of four different attack times: 12 A.M., 6 A.M., 12 P.M., and 6 P.M. Each simulation features a 24-hour injection of a fictional contaminant at strength 100 mg/min (using EPANET's "MASS" injection type), and has a total duration of 48 hours.

Sample cDSP run

At each EPANET reporting step (every 5 minutes), we compute for each of the 236 simulated attacks the expected total mass of contaminant consumed since time 0. We also note the set

of nodes that first experienced contaminant concentration greater than 1e-7 at that reporting step. These data become the primary input values for cDSP.

Solving cDSP with a budget of 5 sensors and a response delay of zero hours determines the optimal expected contaminant mass consumed (over all 236 attacks) to be roughly 22 kg. The median impact is 1.3 kg, indicating that some attacks, if undetected, cause significant impact. A sensor placement found by cDSP that achieves these results is shown in Figure 1. The enlarged nodes are those with sensors, and they are sized proportionally according to the sum of the impacts of the attacks first witnessed by each sensor. The sensors tend to be placed at leaf nodes because of the long attack duration. There is a heavy penalty for failing to detect an attack at a large-demand leaf node. When modeling shorter injections, the sensors tend to be placed on internal nodes. A real application of sensor placement would involve running many instances of these models under varying attack assumptions and identifying areas of the network that usually receive sensors, regardless of input parameters.

Sensitivity analysis

Of course, the base demands in the EPANET input files are merely estimates. Let us consider the effect of variations in those demands on the sensor placements identified by cDSP. Specifically, using the EPANET toolkit, we modify the demands as follows. For each node, we compute a randomized demand in the interval $[q - 0.5q, q + 0.5q]$ for each water quality time step, where q is the EPANET-computed demand for that node at that time step. Then, for each node, we normalize this set of randomized demands so that their sum over all time steps is the same as the sum of the original demands.

With these randomized demands, we now proceed as above and solve cDSP. The *consensus* of two sensor placements is the size of their intersection divided by the number of sensors in each. Our results indicate not only a small variation in objective, but a strong consensus among sensor placements. This holds not only for EPANET Example 3, but for SNL-1 and SNL-2 as well. Table 3 contains our results. The last column of this table expresses the standard deviation of the objective values as a percentage of the mean objective.

The average pairwise consensus figure of 86.5% for SNL-2 is slightly deceiving since only exact matches are counted. The non-matching sensors are typically very close to one another or adjacent, so the effective consensus in a skeletonized model is close to 100%. There are, of course, many other parameters that might be varied in further sensitivity analyses. Real sensor placement efforts will involve implementing these studies. They will feature the combination of an extensive number of required runs and large suites of attack scenarios. Such a study would be completed by using the RW heuristic for most of the runs, then sampling from these to verify optimality via cDSP.

List of Figures

1 Optimal sensor placement for EPANET Example 3 under the conditions described in the demand sensitivity analysis (24 hour injection, 48 hour simulation). The sensor vertices are oversized, and the larger of these witness a larger share of the attack impacts. 24

List of Tables

1 Computational results for MIP solutions. 25
2 Computational results for the *RW* heuristic. 26
3 Sensitivity results based on randomized demands. 27

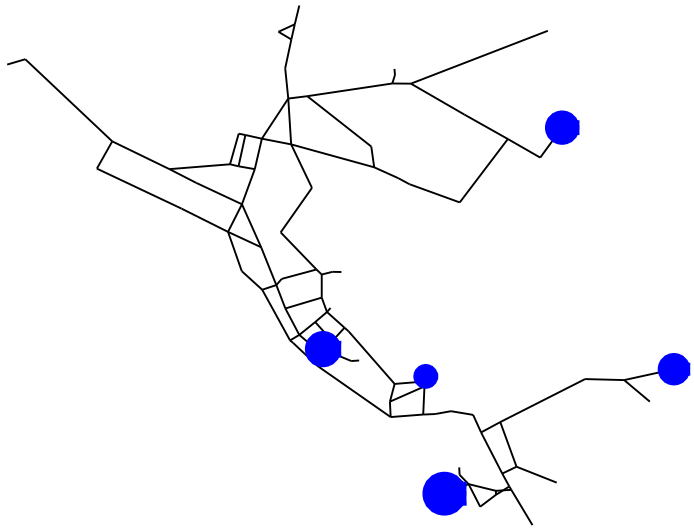

Figure 1: Optimal sensor placement for EPANET Example 3 under the conditions described in the demand sensitivity analysis (24 hour injection, 48 hour simulation). The sensor vertices are oversized, and the larger of these witness a larger share of the attack impacts.

Table 1: Computational results for MIP solutions.

			Linear Program Statistics			Performance Statistics	
Test Instance	#Attacks	p	#Rows	#Columns	#Non-Zeros	Memory	Run-Time
SNL-1	105	20	8156	8566	38372	18Mb	0.58s.
SNL-2	1621	20	263K	267K	1.7M	484Mb	87.2s.
SNL-3	9705	20	1.2M	1.3M	6.5M	2.5Gb	912.62s.

Table 2: Computational results for the RW heuristic.

		Performance Statistics	
Test Instance	p	Memory	Run-Time
SNL-1	20	8Mb	0.2s.
SNL-2	20	230Mb	12.5s.
SNL-3	20	2.8Gb	153.7s.

Table 3: Sensitivity results based on randomized demands.

Network	#Sensors	#Placements	Ave. Consensus	Std. Dev. of Obj.
EPANET Example 3	5	20	100%	0.3%
SNL-1	20	20	97.75%	1.3%
SNL-2	20	5	86.5%	0.3%