Scalable Water Network Sensor Placement via Aggregation

Jonathan W. Berry* Robert D. Carr* William E. Hart*
Cynthia A. Phillips*

Abstract

The quality of a sensor placement for a municipal water distribution network is usually judged by its performance across a set of contamination scenarios. As networks grow, the number of scenarios needed to accurately model a full set of possible events based on season, special events, and type of contamination can grow even more rapidly. We introduce two new methods for reducing the problem size. Scenario aggregation introduces no new approximations and can reduce problems size and reduce running time regardless of the solution method. Witness aggregation is a technique well suited for integer-programming-based solution methods. We give two variants of witness aggregation. We present preliminary experimental results for a moderate-sized network and enriched set of scenarios. Applying both scenario and witness aggregation gave a solution within 1% of optimal in two orders of magnitude less time than not using aggregation.

1 Introduction

A key shortfall of most published water sensor placement methods to date is their inability to handle truly rich ensembles of injection events, encompassing many different contaminants, different seasonal flow patterns, etc. When such large ensembles are considered, the problems grow too large to handle, even on the most powerful workstations. We address this issue via new aggregation techniques that are quite different from skeletonization.

Sensor placement problems for municipal water networks usually involve the placement of a fixed number of sensors to minimize the expected impact of a set of contamination event scenarios. The quality of the sensor placement depends upon how well the set of scenarios represents the set of possible contamination events. To be representative, a full scenario set may be very large. Demand patterns, and hence contamination transport, can vary considerably based on the season or day of the week. Event impact depends upon the nature

*Discrete Algorithms and Math Department, Sandia National Laboratories, Mail stop 1318, P.O. Box 5800, Albuquerque, NM, 87185-1318, USA. {jberry, rdcarr, wehart, caphill}@sandia.gov. Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy's National Nuclear Security Admininstration under contract DE-AC04-94AL85000.

of the contaminant. Thus a full scenario set considers season, type of contaminant, time of day, location, etc.

The size and difficulty of sensor-placement instances depend upon total number of witness events, that is, the number of possible network locations where a sensor could observe an event, summed over all events. Thus the size and difficulty grows with the number of scenarios and the size of the network, and is heavily influenced by demand patterns. For example, when many events can spread through a significant portion of a network, they can dramatically increase problem size.

One general way to reduce the size of an instance is through aggregation. Researchers have previously considered skeletonization[7]. Skeletonization is available in commericial codes such as the H2OMAP Skeletonizer [3] and Skelebrator by Bentley Systems. Skeletonization typically has four operations. The first operation merges dead-end branches into a main branch. If one thinks of the network as a graph, this merges leaves into their parents. A second drops all pipes whose diameter is below a threshold. The third operation merges serial pipes (removes degree-two nodes). And the last merges parallel pipes. One can adjust parameters (such as diameter or roughness) of the new pipes that result from serial or parallel merges so that the new system is hydraulically equivalent to the first. However, the second operation can alter flows in the network and the first operation merges water demands, and hence can change the impact of an event. Skeletonization can decrease the size of the network, making simulations more manageable and making sensor location problems smaller. However, the smaller system is an approximation of the original, since some flow patterns may be altered. Also, it's not clear how to map a sensor placement on the skeletonized network back to locations in the more refined network. There is likely no single location that will perform as well as a single "aggregated" sensor covering the whole region.

We consider two types of aggregation that work with an all-pipes model, or the maximum refinement available for the network. Witness aggregation combines possible witness locations for a single scenario based upon their witness quality, not based on geography. Scenario aggregation replaces a set of scenarios with a single representative scenario. Because the number of scenarios could be much larger than the number of network nodes, reducing the number of scenarios while approximately maintaining scenario coverage will likely be critical for scalability.

This paper considers the value of these two types of aggregation for *integer programs* (IPs). An integer program is the optimization (maximization or minimization) of a linear objective function subject to linear constraints and nonlinear integrality constraints on some or all of the variables. Integer programming solvers can give solutions to IPs with instance-specific provable bounds. So IPs add no more error than that already present in modeling approximations. However, IPs are computationally demanding. Therefore, size reduction is particularly important for IP-based approximation methods. Even pure heuristic methods will have size problems for rich events sets, especially on a standard PC.

Witness aggregation changes the structure of the underlying sensor placement problem by creating groups of equivalent network locations. It is particularly valuable for integer-programming-based methods for sensor placement, since integer programming can easily handle the extra complexity. Scenario aggregation should benefit any optimzation method, since it makes the problem smaller without changing its fundamental structure.

These aggregation methods generally have a tradeoff between problem size (and runtime) vs the quality of the solution. We measure solution quality relative to the value of the optimal solution to the non-aggregated IP problem. We describe two types of witness aggregation. Percent-of-maximum-range (PMR) aggregation can guarantee a maximum formulation size, but does not provide any guarantee on the quality of the approximation, except in a special no-error case. *Ratio* aggregation can guarantee a bound on the solution quality, but generally does not provide any guarantee on the problem size. In this paper we consider only a special case of scenario aggregation that guarantees no error. In particular, it should work well for events that differ only in contamination properties.

We present preliminary computational results for a moderate-sized network using a suite of scenarios that includes multiple contaminants, a synthetic set of demands to model multiple "seasons," and multiple times of day and injection points. We show that even when we allow no additional error for the aggregation, that is, even if the optimal value of the IP is not changed, witness aggregation and scenario aggregation can significantly reduce problem size and runtime. For this instance, we show that witness aggregation that provably bounds solution quality is significantly better than witness aggregation that bounds instance size. Combining scenario aggregation with ratio witness aggregation can give a solution within 1% of optimal in two orders of magnitude less time than using the full unaggregated data set.

The remainder of the paper is organized as follows. In Section 2, we give the basic, non-aggregated IP for sensor placement and describe our modeling assumptions. In Section 3, we describe PMR and ratio witness aggregation. In Section 4, we describe an error-free form of witness aggregation. In Section 5 we describe our experiments and our data. In Section 6, we present and discuss our experimental results. Finally, in Section 7, we discuss future work and related witness aggregation work.

2 The Basic Model

In this section we describe the model of the sensor placement problem presented in [2, 1]. This is the (non-aggregated) problem we wish to solve as quickly and as well as possible. We describe the model for completeness, and discuss particular modeling choices for this paper.

We assume a fixed budget of p sensors. We can place sensors at any *feasible* junction in a distribution network. Some junctions are infeasible due to cost, accessibility issues, etc. We do not consider installation of sensors on pipes because we rely on water quality simulations that cannot provide information on contaminant concentrations along pipes. We assume that sensors are capable of detecting contaminants at any concentration level. We assume that a general alarm is raised when contaminant is first detected by a sensor, and this alarm prevents any further consumption of contaminated water.

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 holding per hour throughout the day. Each pattern represents the demand during a particular time interval on a "typical" day.

Because we do not know a priori where a contamination event will occur, we place sensors to provide a compromise solution across a set of weighted event scenarios. Each contamination scenario is defined by an origin node, a start and stop time, type of contaminant, concentration, and injection rate. For a given contamination scenario, we use water quality anlysis software (e.g., EPANET [6]) to compute the contaminant concentration at each junction in the network for a regularly spaced set of time intervals during a simulation horizon. For example, we may ask EPANET to simulate the movement of contaminant for each scenario for a typical 3-day period, reporting the contaminant concentrations every 5 minutes. For this paper, we consider only one injection site and one type of contaminant per event, though all our methods extend to any scenario that can be simulated by EPANET or an equivalent system.

One can measure damage from water contamination in a number of ways: mass of contaminant released to the network, length of pipes contaminated, etc. Watson et al [8] describe a number of such measures. Because we are considering a scenario set that includes different types of contaminants, we wish to use a measure suitable for any contamination type. Thus for this paper, we measure impact by population exposed. This is the number of people exposed to a potentially-harmful dose of contamination.

For each scenario, we use EPANet to compute the time at which a contaminant plume first arrives at each location. We also compute, given information about the nature of the contaminant, population distribution, consumption patterns, etc, the total network-wide impact of each contamination scenario at each time. Given a specific sensor placement, we assume that event a is detected at the first time that its plume arrives at some node i occupied by a sensor. We say that node i *witnesses* event a, since it is the first sensor to observe it and raise the alarm. We also add a single *dummy* location, which can "witness" any event with impact equal to the maximum possible impact. The dummy location represents a failure to detect an event with a real sensor within the simulation horizon.

The MIP formuation uses the following input parameters:

- $\mathcal{A}$, the set of event scenarios,
- p, the number of sensors,
- $L \subseteq V$, a set of feasible sensor placement locations,
- α_a, the probability (or weight) of scenario $a \in \mathcal{A}$,
- $\mathcal{L}_a$, the subset of vertices in $L \cup \{q\}$ hit (and possibly contaminated by) scenario a. The dummy location q is in $\mathcal{L}_a$ for all events a.
- d_{ai}, the impact of event a if it is witnessed by a sensor at location i.

The MIP uses two types of variables:

For each $i \in L$, we have a binary variable:

$$s_i = \begin{cases} 1 & \text{if we place a sensor on location } i; \\ 0 & \text{otherwise.} \end{cases}$$

For each $a \in \mathcal{A}$ and $i \in L$, we have

$$x_{ai} = \begin{cases} 1 & \text{if a sensor at location } i \text{ witnesses event } a; \\ 0 & \text{otherwise.} \end{cases}$$

The x_{ai} variables are always binary if the s_i are, so in practice, one need not explicitly designate these as integer variables.

The sensor placement problem, called BSP, for Basic Sensor Placement, is as follows:

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

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

The objective minimizes the average (weighted) impact of the set of scenarios (event). The first set of constraints assures that exactly one sensor is credited with raising the alarm for each contamination event. This might be the dummy sensor if no real sensor witnesses the event. The second set forbids a real (non-dummy) 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 BSP (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.

The BSP model is identical to the well-known p-median facility location problem [5]. 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. There is an equivalence between (1) sensors and facilities, (2) contamination scenarios and customers, and (3) contamination impacts and distances. There has been considerable work on p-median computations in the literature. The structure of the data in facility location problems can differ in practice from that in sensor placement problems For example, frequently distances satisfy the triangle inequality and any customer can use any facility. However, many of the general ideas for the p-median problem can apply to this context.

3 Witness Aggregation

In this section we give an expanded notion of witness aggregation first used in [1]. We first describe the basic notion of witness aggregation and then give two specific ways to select which witnesses can be grouped to reduce problem size, sometimes at the expense of solution quality.

Berry et. al. [1] 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 using a coarse reporting time-step with the water quality simulation. All locations that have the same impact value for event a are equivalent in their quality as witnesses for event a.

Motivated by this observation, Berry et. al modified the BSP formulation so witness variables are associated with an impact value for a particular event rather than a location.

In [1], they consider locations equivalent only if they have precisely equal impact values. We generalize this to allow merging of locations that are not precisely equal. Specifically, let $\hat{\mathcal{L}}_{ai} \subseteq \mathcal{L}_a$ be the ith set of grouped locations for event a. We call this the ith *superlocation* for event a. We denote the set of superlocations for event a by $\hat{\mathcal{L}}_a$. Let $\hat{d}_{ai}$ be the largest impact value for event a if witnessed by any location in $\hat{\mathcal{L}}_{ai}$ (that is, $\hat{d}_{ai} = \max_{i \in \hat{\mathcal{L}}_{ai}} d_{ai}$). And let x_{ai} be a binary variable that is 1 if event a is witnessed by some location in $\hat{\mathcal{L}}_{ai}$.

Then the IP for the witness-aggregated sensor placement problem is as follows:

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

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

This is related to the BSP, but now in addition to selecting a superlocation to witness an event, the IP must also select an actual sensor from the superlocation. The locations grouped in a superlocation for an event are not necessarily located physically close in the network even though the contamination for event a reaches them at approximately the same time.

For each event, we consider a list of locations in $\mathcal{L}_a$ sorted by impact. We consider superlocations that are contiguous sublists in this sorted list. Generally, we group two witnesses if the difference in their impact values meets a given threshold. In general, it is hard for a user to determine a good threshold without carefully exploring the data. We now describe two ways to create superlocations. Each method accepts a parameter valued between 0 and 1 inclusive. Depending on the value of this parameter, the groups vary from totally aggregated (one witness per event) to the no-error aggregation used in [1].

PMR Witness aggregation: PMR aggregation accepts a parameter $0 \leq \rho \leq 1$. Let D be the maximum value, taken over all events, of the difference in impact between the best possible witness and the dummy (failure to detect). For every superlocation, the difference between the best witness (lowest impact) in a superlocation and the worst witness (highest impact) in a superlocation is no more than ρD. For PMR witness aggregation, we passed through the list of witnesses for an event starting with the highest-impact event (dummy), adding locations to the current location as long as the threshold is obeyed. Because D is set according to the maximum impact difference over all events, every event will have no more than $\lceil 1/\rho \rceil$ superlocations. Thus if $\rho = 1$, there is only 1 superlocation and the error for each witness event can be as bad as D. If $\rho = 0$, then there is no error, but there are as many superlocations for an event as there are possible impact values for it.

Ratio: Ratio aggregation accepts a parameter $0 \leq r \leq 1$. For every superlocation, the ratio of highest impact to lowest impact is no more than $1/r$. If $r = 1$, then the impacts have to be equal, and we again have the no-error case from [1]. If $r = 0$, then the ratio is infinite and all possible witnesses are merged to a single superlocation for each event. The optimal solution to a problem with level-r ratio aggregation is guaranteed to be an r-approximation for the original problem. That is, the optimal sensor placement for the aggregated problem will have an true impact provably at most r times larger than the true optimal value.

We also consider aggregation where we force at least 2 witnesses per event. We call this a *distinguish-detection* option. When the thresholds or ratios become large, some smaller events may drop out (drop to one witness). If there are many such events, this can lead to a lot of error. Forcing the dummy to be in a superlocation by itself encourages the IP to detect as many events as possible.

4 Scenario Aggregation

In this section, we consider scenario aggregation: replacing a pair or a group of scenarios with a single new scenario that is equivalent.

Consider two events. Suppose event 1 has weight α_1 and k possible witnesses: $\ell_{11}, \ell_{12}, \ldots, \ell_{1k}$ with impacts $d_{11}, d_{12}, \ldots, d_{1k}$ respectively. Note that d_{1k} is the impact of a failed detection. Similarly suppose event 2 has weight α_2 and $j \leq k$ possible witnesses $\ell_{21}, \ell_{22}, \ldots, \ell_{2j}$ with impacts $d_{21}, d_{22}, \ldots, d_{2j}$, respectively. We assume that if two witnesses have the same impact for an event, they are sorted within the list for that event by some unique ID. Suppose $\ell_{1i} = \ell_{2i}$ for all $i = 1, \ldots, j-1$. That is, suppose the ordered witness list for event 1 is a prefix of the ordered list for event 2, excluding the dummy. For any given sensor placement, events 1 and 2 always have the same best witness. So we can create a new event (event 3) with weight $\alpha_3 = (\alpha_1 + \alpha_2)$ and the same witness list as event 1. Impact $d_{3i} = (\alpha_1 d_{1i} + \alpha_2 d_{2i})/(\alpha_1 + \alpha_2)$ for $i = 1 \ldots j-1$ and $d_{3i} = (\alpha_{1i} + \alpha_{2j})/(\alpha_1 + \alpha_2)$ for $i \geq j$. Essentially, we pad out the witness list for event 2 with the dummy (failed detection) witness. Then event 3 is the average of the two events. This combined event has the (common) best witness for the two previous events with the same effect on the objective function.

We expect scenario aggregation will be particularly powerful for pairs of contamination events that differ only in the nature of the contaminant. The contaminants should travel in the same pattern. If they differ only in decay, so that after a specific time, one contaminant is gone, then we expect the witness lists to agree. This is particularly true of events that do not travel through a large number of network locations.

5 Experimental Design/Data

In this section, we describe our preliminary experiments. We used a single real-world network with 3358 nodes. We consider an event at 3am, 9am, 3pm, and 9pm for each of the 1621 non-zero demand nodes. We consider 2 types of contaminants, a biological contaminant and a chemical contaminant. Finally, we have a real set of demands, which we associate with a "normal," winter pattern. We created a plausible set of demands for a "summer" pattern where water use is generally higher. In particular, water use at night increases as residents water their lawns (e.g. in desert climates). For our data set, demands ranged from near zero to about 350. Our normal patterns varied by the hour. We created a new pattern of EPANET multipliers for all nodes with demands between 5 and 30. Specifically, we increased the 3 lowest-demand hours by a factor 5, to represent night-time watering. For each of the two peaks of demand during the day, we increased the multiplier for the top (local mode) by an additive 0.1 and each of the time periods on either side of the mode by an additive 0.05.

Thus our final data set had 4 event times per day for each of 2 contaminants for each of 2 seasons for each non-zero demand node.

We expected that the witness lists of chem and bio events for the same (time, location, season) triple would generally fit the requirements for zero-error scenario aggregation. However, for events that traveled through a large number of network locations, this was not the case. This did not meet our expectations for the perfect simulator world, probably due to numerical issues in EPANET (thresholds for propagating through pipes). However, it does provide a richer data set that plausibly models events where sensor performance is concentration-specific. Perhaps the sensors for one type of contaminant are more sensitive than that for the other. Even though we did not have this idealized data set, scenario aggregation still reduced the size of many problems as detailed in the next section.

6 Results and Discussion

In this section, we present the results of witness and scenario aggregation for the data set presented in Section 5. We ran the non-scenario-aggregated cases on a system with four 2Ghz AMD processors, Fedora core 3 64bit operating system, and 64Gb of RAM. We used cplex 10.0, a commercial integer programming code which did not take advantage of the extra processors. We ran the scenario-aggregated cases using the same version of cplex on a 2-processor dual-core machine with 3.6Ghz Xeon processors, 8Gb of RAM, and 20Gb of swap space.

Table 1 gives the results of PMR aggregation with no scenario aggregation for various values of ρ with and without the distinguish-detection option. It also gives the sizes and runtime information for the no-error witness aggregation case ($\rho = 0$) and for the case where we do no aggregation. We repeat these two baselines in subsequent tables for ease of comparison. Though the problem sizes and runtimes decrease with increasing value of ρ (aggregation threshold relative to the maximum range), the solutions from these aggregated IPs are poor, becoming a factor of two off when the threshold is 0.25. Distinguishing detection gives better approximation values, but at the cost of larger sizes and runtimes.

Table 2 shows the results for several values of ratio-based witness aggregation. The approximation values are much better than those for PMR-based witness aggregation. In fact, when the worst impact in a superlocation can be 50% greater than the best, the solution returned by the IP is optimal, and when the ratio can be as large as 2, the solution to the aggregated IP had less than 1% error.

Tables 3 and 4 give similar results when we add no-error scenario aggregation. The number of constraints is reduced via sceneario aggregation, usually by 25–30%. Scenario aggregation did not reduce the number of nonzeros in the constraint matrix by a significant amount. This implies that the aggregated scenarios generally had short witness lists. The running time is usually significantly smaller. However, because we ran these tests on somewhat different machines, we will need to do further experimentation to determine the precise runtime benefit of scenario aggregation. Since it costs nothing in solution quality, it seems

ρ	max # witness	DD	# variables	# constraints	# nonzeros	runtime (seconds)	IP value	true value	gap (%)
none	N/A	N/A	16854011	16850654	67334870	79504	1186	1186	0
0	N/A	N/A	2506339	2502982	23770968	22415	1186	1186	0
0.125	8	no	31323	27966	12169827	722	24.78	2060	73.7
0.125	8	yes	75151	71794	16477404	1262	191	1665	40.4
0.25	4	no	18025	14668	9842434	322	6.19	2743	131
0.25	4	yes	63460	60103	16442331	1417	140	2382	101
0.5	2	no	7179	3822	3416662	17	0.166	9302	684
0.5	2	yes	57035	53678	16423056	1157	97.6	2411	103

Table 1: Results using PMR aggregation. No scenario aggregation. DD = distinguish detection. IP value is the optimal value for the aggregated integer program. True value is the actual average impact for the full set of scenarios for the sensor placement returned by the IP. The gap is a relative error between the true value of the solution to the aggregated IP and the value of the optimal sensor placement.

r	max ratio high/low	# variables	# constraints	# nonzeros	runtime (seconds)	IP value	true value	gap (%)
none	N/A	16854011	16850654	67334870	79504	1186	1186	0
0	N/A	2506339	2502982	23770968	22415	1186	1186	0
0.66	1.5	208659	205302	14397000	4083	979	1186	0
0.5	2	153410	150053	13123912	1850	845	1195	.76
0.25	4	104713	101356	10056556	905	597	1242	4.7
0.125	8	87057	83700	6622042	404	440	1477	24.5

Table 2: Results using ratio aggregation with no scenario aggregation. We did not use the distinguish detection for this test set. IP value is the optimal value for the aggregated integer program. True value is the actual average impact for the full set of scenarios for the sensor placement returned by the IP. The gap is a relative error between the true value of the solution to the aggregated IP and the value of the optimal sensor placement.

ρ	max # witness	DD	# variables	# constraints	# nonzeros	runtime (seconds)	IP value	true value	gap (%)
none	N/A	16854011	16850654	67334870	79504	1186	1186	0	
0	N/A	N/A	2456286	2452929	23549121	19530	1186	1186	0
0.125	8	no	31264	27907	12142241	375	24.78	2060	73.7
0.125	8	yes	53338	49981	16340277	598	188	1662	40.1
0.25	4	no	17996	14639	9825374	208	6.18	2743	131
0.25	4	yes	41669	38312	16305270	624	137	2367	99.6
0.5	2	no	7173	3816	3409850	14.6	0.17	9159	672
0.5	2	yes	35256	31899	16286031	583	94	2409	103

Table 3: Results using PMR aggregation with no-error scenario aggregation. DD = distinguish detection. IP value is the optimal value for the aggregated integer program. True value is the actual average impact for the full set of scenarios for the sensor placement returned by the IP. The gap is a relative error between the true value of the solution to the aggregated IP and the value of the optimal sensor placement.

a worthwhile optimization.

7 Future and Related Work

In this section, we outline some directions for future work. First, however, we will discuss some related witness aggregation work due to Church [4]. Church gave three methods of witness aggregation in the context of the p-median facility location problem. The first method does not apply to the sensor placement because it assumes that every facility can service every customer. This is equivalent to saying every location can witness every event, which is not true. The second method of witness aggregation aggregates all witnesses into a single supernode if their impact is sufficiently close to the dummy. Thus each event may have a superlocation, but only one at the end of its list. Finally, Church observed that if a single location can witness events 1 and 2, and the set of better witnesses (in any order) is identical for both events, then we can aggregate just that pair of witness variables. This is something one could do in addition to the aggregation techniques we describe in this paper.

There was considerable value to allowing witness aggregation that introduced error (approximation). In future work, we will consider scenario aggregation that introduces some error. We will consider how these methods interact with skeletonization.

References

[1] J. Berry, W. Hart, C. Phillips, J. Uber, and J-P. Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management*, 132:218–224, 2006.

[2] J. Berry, W. E. Hart, C. A. Phillips, and J. Uber. A general integer-programming-based

r	max ratio high/low	# variables	# constraints	# nonzeros	runtime (seconds)	IP value	true value	gap (%)
none	N/A	16854011	16850654	67334870	79504	1186	1186	0
0	N/A	2506339	2502982	23770968	22415	1186	1186	0
0.66	1.5	178887	175530	14257957	1832	978	1186	0
0.5	2	126185	122828	13003901	795	845	1195	.76
0.25	4	80519	77162	9965360	382	595	1242	4.7
0.125	8	63905	60548	6547852	188	440	1477	24.5

Table 4: Results using ratio aggregation and no-error scenario aggregation. IP value is the optimal value for the aggregated integer program. We did not use the distinguish detection for this test set. True value is the actual average impact for the full set of scenarios for the sensor placement returned by the IP. The gap is a relative error between the true value of the solution to the aggregated IP and the value of the optimal sensor placement.

framework for sensor placement in municipal water networks. In *Proc. World Water and Environment Resources Conference*, 2004.

[3] P. Boulos. *H2OMAP Skeletonizer Users' Guide*. MHW Soft, Inc, 300 North Lake Ave, Suite 1200, Pasadena, CA, 2005. First edition 1996.

[4] R. Church. COBRA: A new formulation of the classic p-median location problem. *Annals of Operations Research*, 122:103–120, 2003.

[5] P.S. Mirchandani and R.L. Francis, editors. *Discrete Location Theory*. John Wiley and Sons, 1990.

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

[7] T. Walski, J-L. Daviau, and S. Coran. Effect of skeletonization on transient analysis results. In *Proceedings of the World Water and Environmental Congress*, 2004.

[8] J.-P. Watson, H. J. Greenberg, and W. E. Hart. A multiple-objective analysis of sensor placement optimization in wateȓnetworks. In *Proc. World Water and Environment Resources Conference*, 2004.