SAND2011-0444C

Minimize Impact or Maximize Benefit: the Role of Objective Function in Approximately Optimizing Sensor Placement for Municipal Water Distribution Networks

William E. Hart
Sandia National Laboratories
Albuquerque New Mexico, USA
{wehart}@sandia.gov

Regan Murray
US Environmental Protection Agency
Cincinnati, OH USA
Murray.Regan@epa.gov

Cynthia A. Phillips
Sandia National Laboratories
Albuquerque, NM USA
caphill@sandia.gov

Abstract

We consider the design of a sensor network to serve as an early warning system against a potential suite of contamination incidents. Given any measure for evaluating the quality of a sensor placement, there are two ways to model the objective. One is to minimize the impact or damage to the network, the other is to maximize the reduction in impact compared to the network without sensors. These objectives are the same when the problem is solved optimally. But when given equally-good approximation algorithms for each of this pair of complementary objectives, either one might be a better choice. The choice generally depends upon the quality of the approximation algorithms, the impact when there are no sensors, and the number of sensors available. We examine when each objective is better than the other by examining multiple real world networks. When assuming perfect sensors, minimizing impact is frequently superior for virulent contaminants. But when there are long response delays, or it is very difficult to reduce impact, maximizing impact reduction may be better.

Key words: sensor placement, approximation algorithms

1 Introduction

Optimization-based methods for determining placements for sensors in municipal water distribution networks depend critically on the precise objective function, i.e. how one measures the quality of a sensor placement. A typical single objective for the sensor placement problem attempts to minimize the impact, or damage, caused by a suite of potential contamination events. The way impact is measured can determine the preferred or feasible solution methods and the nature and quality of the resulting sensor placements.

Previous studies have considered multiple multiple ways to measure the impact from a single contamination incident. These usually quantify human health affects or damage/clean-up expense to the network itself. For example, the EPA's TEVA-SPOT (Threat Ensemble Vulnerability Analysis Sensor Placement Optimization Toolkit) can measure a single contamination impact using the number of people exposed to a dangerous level of contamination, number of people killed, the volume of contamination released from the network, the pipe-feet of network contaminated, etc. Given impacts from a single contamination incident, the objective also specifies how to combine these into a single incident. TEVA-SPOT includes average, worst-case, and various tail-bounding metrics. Models of sensor performance such as sensor failures or uncertainties in parameters affect the timing and quality of detection and hence the objective function. In this paper we explore the effect of a different aspect of the objective: complementarity.

The goal of sensor placement is to minimize impact. The complement of this objective is to maximize impact reduction. So, for example, for the objective "minimize average number of people exposed" the complement is "maximize number of people saved from exposure," where the baseline is the number of people exposed when there are no sensors installed. The two objectives are equivalent when solved to optimality. However, they are not the same when solving sensor placement problems approximately where quality is measured by relative error. The two types of objectives are generally incomparable: there will be some settings where maximization is better and other settings where minimization is better. For any way to pick a single objective function from the above choices, the optimizer must also select an optimization direction, one of the two complementary objectives.

An Example To illustrate the differences, consider a simple example. Suppose our impact measure is number of people killed on average taken over a suite of potential contamination incidents. Suppose we have an intelligent enumerative method that can eventually find the optimal placement of p sensors, but we allow the procedure to stop as soon as it has a solution that is provably within 10% of the optimal. Although in general the procedure may do better than 10% error, for this discussion, assume it returns an answer that has precisely 10% error. That is, we are analyzing the guarantee. Suppose that for a given instance, there are 1000 people killed (on average) if the network has no sensors, and the optimal sensor placement reduces the average number of deaths to 100. A search that is minimizing the number of people killed will return a solution in which 110 people are killed. That is, there are 10% more than the fewest possible number of deaths. A search that is maximizing the number of people saved will return a solution in which 190 people are killed. That is, the solution fails to save 10% of the possible 900 that could be saved by an optimal sensor placement. Thus the solution found using the min-impact objective was better than the solution found by the max-impact-reduction objective. It was better when measured by either objective.

Now suppose instead that the optimal sensor placement only reduces the number of fatalities to 900. A search based on minimizing impact with a 10% error will return a placement where 990 people are killed. A search based on maximizing lives saved will return a placement where 910 people are killed. In this case, the complementary (maximize savings) objective solution is better when measured by either objective.

Approximation Algorithms In practice, many difficult optimization problems are solved in practice with *heuristics*. These are methods that run sufficiently quickly and seem to give good answers. But there is no performance guarantee for heuristics. An *approximation algorithm* runs quickly in a formal sense. Specifically, its running time is a polynomial function of the input size. For the sensor placement problem, the input size is the number of nodes in the network, the number of contamination incidents in the suite, and the number of sensors p (which is bounded by the number of network nodes). An approximation algorithm also guarantees performance. An α-approximation algorithm for $\alpha \geq 1$ provably returns a solution that is no more than α times the optimal for a minimization problem. For a maximization problem, it returns a solution that is at least $1/\alpha$ times the optimal (maximum) solution. In the example above, we described 1.1-approximation algorithms. A 2-approximation algorithm for the min-impact objective always returns a solution with impact no more than twice the optimal and a 2-approximation for the maximize-impact-reduction objective always returns a solution that eliminates at least half of the impact eliminated by an optimal solution.

We are aware of only one approximation algorithm for sensor placement in water networks, due to Krause et. al. [5, 4]. Krause et. al. consider the model introduced by Berry et al. [1]. In this model, an EPANET [7] simulation of a contamination incident tracks the evolution of the contamination plume through time, assuming "normal" network demand and control patterns. This shows where and when a sensor could detect that contamination incident. It also allows computation of the impact over the whole network as a function of time since the beginning of the contamination incident. If sensors are perfect, they raise an alarm precisely when there is contamination at their location (no false positives or false negatives). For perfect sensors, the alarm sounds at the first moment an indwelling sensor could observe contamination. The utility then issues a general alarm that will stop further impact, perhaps after a suitable response delay.

Berry et al showed that the resulting optimization problem, which places sensors to minimize the average impact over all contaminiation incidents, is a p-median problem. This problem is NP-hard and there are no known approximation algorithms when the impact values do not satisfy the triangle inequality. Krause et al, however, proved that the greedy algorithm is an $\frac{e-1}{e}$-algorithm (approximately a 1.58-approximation) for the complementary objective: maximizing the impact reduction compared to having no sensors. This algorithm will always guarantee some benefit even when it's difficult to save anyone. However, it is not an α-approximation for the problem of minimizing impact for any constant α. To see this, consider a case where there is an arbitrarily large impact D when there are no sensors, but p sensors can reduce the impact to 1. An α-approximation for impact minimization must return a solution with at most α impact, but the Krause-et-al algorithm only guarantees an impact of at most $D - \frac{D-1}{\alpha}$, which is greater than α for sufficiently large D.

In this paper we focus on the p-median formulation and its complement. Because even large-scale problems can be solved in practice, it provides an excellent test case for exploring when to use each optimization direction in approximately optimizing sensor placement.

We show that for each problem there is some number of sensors p_c where approximation algorithms of equal quality, one for each objective direction, are equivalent. This is the *crossover* point. If the utility can place more than p_c sensors, then it is best to minimize impact and if has fewer than p_c sensors, it is better to maximize impact reduction. The crossover point depends upon the quality of the approximation algorithm α and on the (worst-case) impact when there are no sensors, which we denote by w.

We show how to identify the crossover point and use that to provide guidance on selecting an objective direction when approximately optimizing a sensor placement. In section 2 we discuss crossover points and their computation in more detail. In section 3 we describe our experiments on seven real networks, in section 4 we give the results of the experiments, and in section 5 we discuss the implications of the experiments for sensor-placement optimization.

2 Crossover Points

In this section we discuss what a crossover point is and how to compute them. Assume that we have two approximation algorithms, one, A, that attempts to minimize impact and another, B that attempts to maximize impact reduction. Assume that both offer the same performance guarantee $\alpha \geq 1$. This is appropriate when using, for example, a branch-and-bound solver with the gap set to α. In general, for algorithms that have a specific error guarantee, rather than accepting an error parameter, the approximation ratios for algorithms A and B will not be the same. This is the case for the p-median formulation, for example, where B has an approximation guarantee of about 1.58 and there is no polynomial-time algorithm known that gives an approximation bound for A. The following derivation can be generalized to the case where the approximation ratios for A and B differ. Figure 2 illustrates the discussion in this section.

Given an instance of the p-median sensor placement problem, let w be the impact when there are no sensors. Let b be the impact for p optimally-placed sensors. Then Algorithm A guarantees a solution with impact no more than αb. For this instance, the optimal impact reduction is $w - b$. Algorithm B guarantees a solution with impact reduction at least $\frac{(w-b)}{\alpha}$, which will therefore have impact no more than $w - \frac{(w-b)}{\alpha}$. Either algorithm could do better than its respective guarantee, but we are analyzing only the guarantee, when each algorithm returns a solution of precisely the quality promised.

The two algorithms promise the same quality solution when

$$\begin{aligned}
\alpha b &= w - \frac{(w-b)}{\alpha} \\
\alpha^2 b &= \alpha w - w + b \\
(\alpha^2 - 1)b &= (\alpha - 1)w \\
b &= \frac{\alpha - 1}{\alpha^2 - 1} w.
\end{aligned}$$

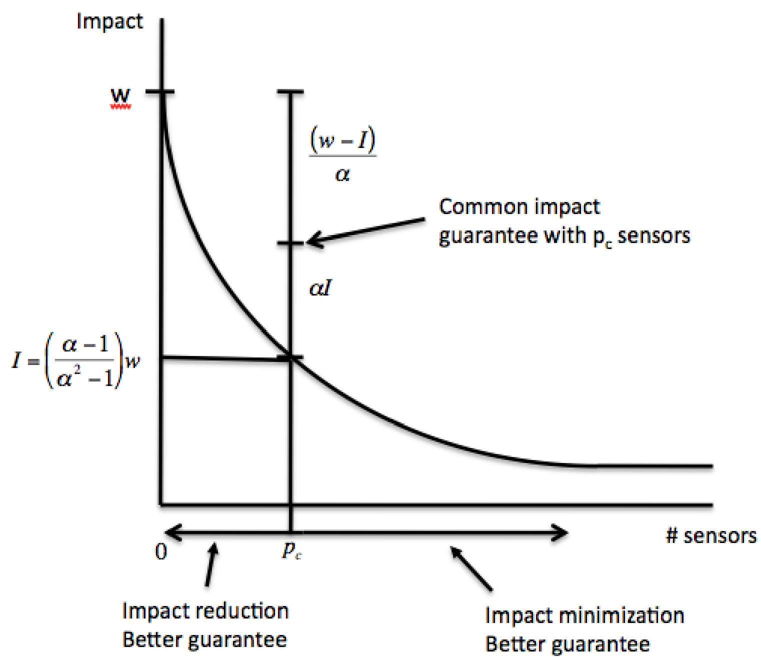

Figure 1: The crossover point p_c is the minimum number of sensors necessary to reduce the impact to the crossover optimum, shown as I. With p_c sensors an α-approximation algorithm to minimize impact and an α-approximation algorithm to maximize impact reduction offer the same guarantee. If there are more than p_c sensors available, the min-impact algorithm gives a better guarantee. If there are fewer, the maximum-reduction algorithm gives a better guarantee.

We call $\frac{\alpha-1}{\alpha^2-1}w$, the *crossover optimum*. Let p_c be the minimum number of sensors necessary to reduce the impact to at most the crossover optimum $\frac{\alpha-1}{\alpha^2-1}w$. For ease of discussion, assume the optimal impact with p_c sensors is precisely the crossover optimum. When a utility can place p_c sensors, either algorithm gives the same guarantee, namely

$$\frac{\alpha(\alpha-1)w}{\alpha^2-1}.$$

We call p_c the crossover point. If we repeat the same algebraic derivation above with inequality instead of equality, we see that whenever the optimal impact for p sensors is less than the crossover optimum, algorithm A (min impact) gives a better guarantee and whenever the optimal impact for p sensors is larger than the crossover optimum, algorithm B (max impact reduction) gives a better guarantee. Since the optimal impact can only decrease with more sensors, that means whenever a utility has at least p_c sensors to place, Algorithm A gives a stronger guarantee, and whenever a utility has fewer than p_c sensors to place, Algorithm B gives a stronger guarantee.

When $\alpha = 2$, the crossover optimum is $w/3$, where both algorithms guarantee an impact no more than $2w/3$. For the Krause et. al. guarantee on the complementary objective

Name	# nodes	# contamination events	response delay (hours)
Net1	3196	3020	0
Net2	3358	1621	0
Net3	8139	6766	6
Net4	12587	10552	6
Net5	12624	64554	0
Net6	48164	9162	0
Net7	55144	25720	0

Table 1: Some statistics on our real-world examples.

($\alpha = 1.58$), the crossover optimum is 0.39. As α gets smaller (the approximation algorithms get better), the crossover optimum approaches $w/2$. For example, when $\alpha = 3/2$, that is, a 50% error, the crossover optimum is $2b/5$. When $\alpha = 1.1$, a 10% error, the crossover optimum is $0.48b$. Thus $W/2$ is approximately the crossover optimum for $\alpha = 1 + \epsilon$ for arbitrarily small $\epsilon > 0$.

3 Experimental Design

We computed the crossover point for seven real-world networks which we call networks 1 to 7. The network names are sorted by the number of nodes. Table 1 gives some information about the example networks and the nature of the contamination incidents. There is one contamination incident at each non-zero-demand node at a single time of day. In all cases, we use a detection threshold of zero, an injection duration of one hour, and run the simulation for 168 hours. We use the population-dosed objective measure, for several dosage levels on most networks. This is the number of people in the network who received and LD50-level of exposure. This is a level that has a 50% probability of killing the victim. Davis and Janke [2] describe this impact measure in more detail. For most cases, we use place the population on network nodes probabilistically based on the demands at the nodes. However, for Network 3, we spread the population evenly.

Several of these examples are networks Davis and Janke used in their study [2]. Our networks $1, 2, 3$ and ,5 correspond to their network $2, 4, 9$, and 6 respectively. Davis and Janke give additional information on the population and other network features such as the number of sources. Network 5 is BWSN-2 from the Battle of the Water Sensor Networks [6]. Network 4 is a modified version of BWSN with lower connectivity.

For each example (network and objective), we computed the impact with no sensors and used that to determine the crossover optimum for three different levels of approximation ratio α. The first is $1+\epsilon$, which is half the no-sensors impact. This is the crossover for arbitrarily good approximation algorithms. The second is $\alpha = 1.58$, the approximation guarantee for the maximize-impact-reduction objective for the greedy algorithm as proved by Krause et. al.

The third is $\alpha = 2$. Given the goal impact value, we can calculate the minimum number of sensors using integer programming or by using an algorithm (or heuristic) for either objective and doing binary search over the number of sensors.

4 Results

Table 2 shows the results of our computation. For Networks 1 through 5 these results come from integer programming calculations and therefore have a computational proof of optimality. Networks 6 and 7 required too much memory for an integer program. The crossover point (c_p) values come from heuristic local search (GRASP) computations using the heuristic available in TEVA-SPOT[3]. In almost all cases, there is a matching lower bound from a TEVA-SPOT Lagrangian calculation. This proves the values are optimal. For Network 6, when c_p is very large (e.g. 176 or 639), the lower bound was not high enough to prove optimality. It is likely that the lower bound is weak because the GRASP heuristic has consistently found optimal solutions in practice and the Lagrangian computation has not been extensively tested at high sensor counts.

As the dosage level required for exposure increases, the worst-case impact (with no sensors) goes down and a relatively larger number of sensors are required to drive the impact lower. For all these networks, dosage levels 0.01 or lower had low crossover points, well within the limits one might expect a utility to deploy. The c_p for dosage level of 0.1 varies considerably with network. Networks 1 and 6 are still low. Networks 2 and 4 range from moderate to high. Network 3 already requires more sensors at dosage level 0.1 than most utilities will deploy. This may be due in part to the 6-hour response delay for Network 3. Delay adds unavoidable impact. By dosage level 1.0 almost all the networks require far more sensors than a utility will likely deploy.

5 Discussion

Our examples show that in general there are some cases where, given approximation algorithms of equal quality for complementary objectives, it is sometimes best to use one and sometimes best to use the other. When the approximation algorithms are particularly good, the crossover optimum moves toward $w/2$, which favors the min-impact metric. In practice for p-median, perfect-sensor formulations, algorithms are sufficiently good in practice to use the min-impact objective. Though both objective directions are natural, the min-impact better reflects the way government and the media evaluate disasters (number of casualties, not number unaffected). However, there may be stronger reasons to consider the complementary objective in the future. Poor approximation ratios, as may be the case with more complex nonlinear objectives including uncertainty, favor the max-reduction method. When the optimal value is very large compared to w, it is best to use the maximize-impact-reduction objective. In some cases, there may be no number of sensors such that it's best to use the min-impact objective. This happens, for example, when it is impossible to reduce the

Network	dosage (mg)	α	c_p	Network	dosage (mg)	α	c_p
Net1	0.0001	$1+\epsilon$	1	Net4	0.0001	$1+\epsilon$	2
Net1	0.0001	1.58	2	Net4	0.0001	1.58	2
Net1	0.0001	2	2	Net4	0.0001	2	2
Net1	0.1	$1+\epsilon$	2	Net4	0.001	$1+\epsilon$	2
Net1	0.1	1.58	2	Net4	0.001	1.58	2
Net1	0.1	2	2	Net4	0.001	2	3
Net1	1.0	$1+\epsilon$	2	Net4	0.01	$1+\epsilon$	3
Net1	1.0	1.58	2	Net4	0.01	1.58	4
Net1	1.0	2	3	Net4	0.01	2	6
Net2	0.0001	$1+\epsilon$	1	Net4	0.1	$1+\epsilon$	7
Net2	0.0001	1.58	1	Net4	0.1	1.58	34
Net2	0.0001	2	1	Net4	0.1	2	83
Net2	0.001	$1+\epsilon$	1	Net4	1.0	$1+\epsilon$	251
Net2	0.001	1.58	1	Net4	1.0	1.58	606
Net2	0.001	2	1	Net4	1.0	2	919
Net2	0.01	$1+\epsilon$	1	Net5	pe	$1+\epsilon$	3
Net2	0.01	1.58	2	Net5	pe	1.58	5
Net2	0.01	2	3	Net5	pe	2	7
Net2	0.1	$1+\epsilon$	11	Net6	0.0001	$1+\epsilon$	2
Net2	0.1	1.58	33	Net6	0.0001	1.58	2
Net2	0.1	2	57	Net6	0.0001	2	3
Net2	1.0	$1+\epsilon$	146	Net6	0.001	$1+\epsilon$	2
Net2	1.0	1.58	236	Net6	0.001	1.58	3
Net2	1.0	2	293	Net6	0.001	2	3
Net3	0.0001	$1+\epsilon$	1	Net6	0.01	$1+\epsilon$	3
Net3	0.0001	1.58	2	Net6	0.01	1.58	3
Net3	0.0001	2	2	Net6	0.01	2	4
Net3	0.001	$1+\epsilon$	2	Net6	0.1	$1+\epsilon$	11
Net3	0.001	1.58	2	Net6	0.1	1.58	23
Net3	0.001	2	3	Net6	0.1	2	36
Net3	0.1	$1+\epsilon$	61	Net6	1.0	$1+\epsilon$	176
Net3	0.1	1.58	184	Net6	1.0	1.58	424
Net3	0.1	2	303	Net6	1.0	2	639
Net3	1.0	$1+\epsilon$	267	Net7	x	$1+\epsilon$	1
Net3	1.0	1.58	536	Net7	x	1.58	1
Net3	1.0	2	758	Net7	x	2	1
Net7	y	$1+\epsilon$	2	Net7	z	$1+\epsilon$	6
Net7	y	1.58	2	Net7	z	1.58	10
Net7	y	2	3	Net7	z	2	14

Table 2: The crossover point for our example set. α is the approximation ratio for a hypothetical pair of approximation algorithms with complementary objectives. c_p is the crossover point. With more than this many sensors, it is best to use the min-impact approximation algorithm; with fewer it is better to use a maximum-impact-reduction approximation algorithm. For Net7 dosage $x < y < z$. Dosage information for Network 5 was not available.

average impact below $w/2$. This might be due to a large response delay, or because some network locations cannot host a sensor. Since real networks will likely have response delays and infeasible locations, this favors the maximize-impact-reduction metric, which can always achieve some benefit if there is any benefit to achieve.

Acknowledgements

Sandia is a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000. The U.S. Environmental Protection Agency through its Office of Research and Development partially funded and collaborated in the research described here under an Interagency Agreement with the Department of Energy. We thank Rob Janke of the EPA for his help with acquiring and understanding the examples for our experiments.

References

[1] J. Berry, W. E. Hart, C. E. Phillips, J. G. Uber, and J.-P. Watson, *Sensor placement in municiple water networks with temporal integer programming models*, J. Water Resources Planning and Management, 132 (2006), pp. 218–224.

[2] M. J. Davis and R. Janke, *Patterns in potential impacts associated with contamination events in water distribution systems*, J. Water Resources Planning and Management, (2010). 10.1061/(ASCE)WR.1943-5452.0000084.

[3] D. B. Hart and S. A. McKenna, *User's manual teva-spot toolkit*, Tech. Rep. 600/R-08/041A, United States Environmental Protection Agency, Oct 2009.

[4] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, *Efficient sensor placement optimization for securing large water distribution networks*, Journal of Water Resources Planning and Management, 134 (2008), pp. 516–526.

[5] A. Krause, J. Leskovec, S. Isovitsch, J. Xu, C. Guestrin, J. VanBriesen, M. Small, and P. Fischbeck, *Optimizing sensor placements in water distribution systems using submodular function maximization*, in Proc. 8th Annual Water Distribution System Analysis Symposium, 2006.

[6] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, F. di Pierro, S.-T. Khu, D. Savic, D. Eliades, M. Polycarpou, S. R. Ghimire, B. D. Barkdoll, R. Gueli, J. J. Huang, E. A. McBean, W. James, A. Krause, J. Leskovec, S. Isovitsch, J. Xu, C. Guestrin, J. VanBriesen, M. Small, P. Fischbeck, A. Preis, M. Propato, O. Piller, G. B. Trachtman, Z. Y.

WU, AND T. WALSKI, *The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms*, J Water Resource Planning and Management, 134 (2008), pp. 556–568.

[7] L. A. ROSSMAN, *The EPANET programmer's toolkit for analysis of water distribution systems*, in Proceedings of the Annual Water Resources Planning and Management Conference, ASCE, Reston, VA, 1999. Available at http://www.epanet.gov/ORD/NRMRL/wswrd/epanet.html.