Optimal Sensor Placement for Water Distribution Network Security

Y. Shastri
Department of Bioengineering, University of Illinois at Chicago and
Vishwamitra Research Institute, 34, N. Cass Avn., Westmont, IL 60559
U. Diwekar
Vishwamitra Research Institute,
34, N. Cass Avn., Westmont, IL 60559

Abstract

Placement of sensors in water distribution networks helps timely detection of contamination and reduces risk to the population. Identifying the optimal locations of these sensors is important from economic perspective and has been previously attempted using the theory of optimization. This work extends that formulation by considering uncertainty in the network and describes a stochastic programming method that is capable of determining the optimal sensor location while accounting for demand uncertainties. The problem is formulated as a two stage stochastic programming problem with recourse. The solution of the problem is achieved by using a newly proposed algorithm aimed at efficiently solving stochastic nonlinear programming problems. This makes the problem solution computationally tractable as compared to the traditional stochastic programming methods. The proposed formulation and solution methodology are tested on an example network to perform a comparative study with other formulations. The results show the importance of uncertainty consideration in decision making and highlight the advantages of the proposed stochastic programming approach.

1 Introduction

The tragic events of September 11 have redefined the concept of risk and uncertainty. Water being important for all living creatures, water security has become a matter of utmost importance to national and international sustainability. Water utilities are feeling the growing need to detect and minimize the risk of malicious water contamination in distributed water networks to prepare for any catastrophic events. Minimization of risk by optimizing the water network design is perceived most promising. Application of optimization in water network design is a well researched area [1, 2, 3, 4]. While most of the articles discuss the structural and quantitative aspects of design such as network capacity, pipe diameter and length, component failures etc., work in the qualitative aspects such as chemical propagation, concentration of disinfectants, contamination minimization etc. has been minuscule in comparison. The goal to make water networks secured against contamination attack calls for such considerations at the design stage.

Integration of adequate number of sensors at appropriate places in distribution networks to detect contaminated water can provide an early detection system where appropriate control measures can be taken to minimize the risk. There are many possible objectives that can be formulated for optimal sensor placement, reflecting various costs and risks of an attack on a network [5, 6, 7, 8]. However it is necessary to include noise factors and consider uncertainties in these attack scenarios to obtain a robust solution in the face of risk. In its absence, the solutions of the sensor placement optimization problem do not give the true optimum locations. A methodology is therefore needed to identify optimum locations after accounting for possible uncertainties. This changes the deterministic optimization problem to an optimization under uncertainty problem. Optimization under uncertainty refers to the branch of optimization where there are uncertainties (or randomness) involved in the data or model, popularly known as stochastic programming problems [9]. Stochastic programming problems are computationally more intensive than their deterministic counterparts.

For the problem of sensor placement, uncertainty can manifest itself through changing population density/water demands at various junctions or through varying probability of contamination at a node. Population density or demand uncertainty can be of two different kinds, uncertainty with time (such as morning and evening) and uncertainty about the actual value at a particular time. The work presented in [5] considers the second type of population density uncertainty and models it by considering various noise levels along with uncertain attack (contamination) scenario affecting the objective function. But the problem is solved using deterministic optimization methods due to the restricted nature of uncertainty consideration. This work proposes a new stochastic programming based approach to solve the sensor placement problem. It models the demand uncertainty extensively, affecting the constraints along with the objective function, as a result of which the problem is transformed into a two stage stochastic programming problem with recourse. The solution of this problem is achieved through a new L-shaped BONUS algorithm (Shastri and Diwekar, unpublished manuscript, 2005). The proposed problem formulation and solution method give results that are truly optimum in an actual water distribution network.

The next section gives the motivation for this work, starting with an earlier formulation. This is followed by sections discussing the modifications to the original problem formulation leading to the proposed stochastic programming formulation. Next three Sections explain the solution procedure for the proposed problem and briefly discuss the algorithm and some of its important features. The last two sections present the results for the proposed formulation and its comparison with other formulations using a case study network and draw conclusions.

2 Motivation

Before formulating a stochastic programming problem, it is important to understand the need for such a formulation. This section gives the motivation for the work by demonstrating the deficiencies of the previous deterministic formulation.

As stated earlier, water network security problem can be formulated in many different ways depending on the objective. One of the approaches minimizes the risk from contaminations (attacks) using sensors for timely detection. Authors in [5] propose the problem of optimal sensor placement in a water distribution network to perform this task at minimum cost for maximum benefits using the theory of optimization. The Integer Programming (IP) formulation in [5] models a water network as a graph $G = (V, E)$ where E is a set of edges representing pipes, and V is a set of vertices, or nodes, where pipes meet (e.g. reservoirs, tanks, consumption points etc.). An attack is modeled as the release of a large volume of a harmful contaminant at a single point in the network with a single injection. The water network simulator EPANET [10] is used to determine an acyclic water flow, given a set of available water sources, assuming each demand pattern holds steady for sufficiently long. The IP formulation from [5] is shown below.

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{p=1}^{P}\sum_{j=1}^{n} \alpha_{ip} C_{ipj} \delta_{jp} \tag{1}$$

where

$$C_{ipi} \; = \; 1 \qquad i = 1, \ldots, n; \; p = 1, \ldots, P \tag{2}$$

$$s_{ij} \; = \; s_{ji} \qquad i = 1, \ldots, n-1, \; i \; < \; j \tag{3}$$

$$C_{ipj} \; \geq \; C_{ipk} \; - \; s_{kj} \qquad (i,k,j) \in E; \; s.t. \; f_{kjp} \; = \; 1 \tag{4}$$

$$\sum_{(i,j) \in E, i<j} s_{ij} \; \leq \; S_{max} \qquad s_{ij} \in (0,1); (i,j) \in E \tag{5}$$

In the above objective function, α_{ip} is the probability of an attack at node v_i, during flow pattern p, conditional on exactly one attack on a node during some flow pattern, δ_{jp} is the population density at node v_j while flow pattern p is active, and C_{ipj} is the contamination indicator. $C_{ipj} = 1$ if node v_j is contaminated by an attack at node v_i during pattern p, and 0 otherwise. Decision variable s_{ij} is 1 if a sensor is placed on (undirected) edge (v_i, v_j) and 0 otherwise. Risk is defined under a fixed number of flow patterns represented by binary parameters f_{ijp}. $f_{ijp} = 1$ if there is positive flow along (directed) edge $e = (v_i, v_j)$ during flow pattern p and 0 otherwise. Gross shifts in demand at various nodes with time are expected to contribute different flow patterns in the problem. The problem considers noise (uncertainty) through variable population density δ_{jp} and attack probability α_{ip} at different nodes with a known probability distribution (uniform or normal). The uncertain δ_{jp} and α_{ip} values are either sampled or specified by scenarios, C_{ipj} and f_{ijp} values are obtained from EPANET simulator while the other parameters are known constants.

The first set of constraints ensures that when a node is directly attacked, it is contaminated. The second set indicates that a single sensor covers a pipe for flow in both directions. The third set propagates contamination from node v_k to node v_j if node v_k is contaminated, there is positive flow along a directed edge from v_k to v_j and there is no sensor on that edge. The next constraint enforces a limit on the total number of sensors, S_{max}. The final set of constraints forces integrality of sensor placement decisions. The first and third sets of constraints propagate values of 1 whenever there are no sensors to prevent the propagation (otherwise the contamination variables are 0). The population at any node is considered at risk if it consumes contaminated water.

The goal of the above IP problem is to find the optimal sensor configuration so as to minimize the expected fraction of population at risk from an attack. The model assumes fixed demand patterns and makes no assumption of how long each pattern holds, how often it appears, or the order in which the patterns appear. In objective function (1), each node is weighted by the number of people potentially consuming water at that node. The population density, not necessarily the demand, affects the observed network flow patterns.

While the idea of uncertainty introduction in the problem is appropriate, its effect needs to be carefully assessed. A careful analysis suggests that, along with the objective, constraints are also affected by uncertainty. This is because nodal demand is expected to be correlated with the uncertain population density, particularly in residential networks, where consumption will be directly correlated with population density. Once such a relationship is considered, the problem is complicated. Flow characteristics of a network such as flow rate and direction depend on nodal demands. Change in those demands due to uncertainty is expected to change the flow directions. Therefore the assumption of unchanged flow pattern is no more valid. An approach making such an assumption will give sub-optimal results.

To ascertain this point, simulations were performed on “Example Network 2” of EPANET [10] considering demand variations in the range of $\pm 25\%$ for all nodes around a mean. The network is shown in figure 1. A network can mathematically be represented by the C_{ipj} values without any sensors. Figure 2 shows such a representation for this network. The values in bold and shaded background are the ones that change after considering uncertain demands resulting in a second flow pattern. It is observed that for 100 samples, the distribution of patterns is 85/15 for pattern 1 to pattern 2. This analysis shows that flow patterns change due to uncertainty and hence the effect of uncertainty on constraints is important for this problem. Next section discusses the modified problem formulation.

3 Modified Problem Formulation

3.1 Problem formulation

The modified problem formulation, transforming the problem into the realm of stochastic programming, is given as follows.

$$\text{Minimize} \quad \sum_{l=1}^{N_{samp}} \sum_{i=1}^{n} \sum_{p=1}^{P} \sum_{j=1}^{n} \alpha_{ip}(l)\ \delta_{jp}(l)\ C_{ipj} \tag{6}$$

$$where$$

$$C_{ipi} \ = \ 1 \qquad i = 1, \ldots, n;\ p = 1, \ldots, P \tag{7}$$

$$s_{ij} \ = \ s_{ji} \qquad i = 1, \ldots, n-1,\ i \ < \ j \tag{8}$$

$$C_{ipj} \ \geq \ C_{ipk} \ - \ s_{kj} \qquad (i,k,j) \in E;\ s.t.\ f_{kjp} \ = \ 1 \tag{9}$$

$$\sum_{(i,j)\in E, i<j} s_{ij} \ \leq \ S_{max} \qquad s_{ij} \in (0,1); (i,j) \in E \tag{10}$$

Stochastic programming problems are often solved using sampling based techniques where the uncertain distribution is approximated by a sample set [11]. The modified formulation is to be solved using such an approach for which the uncertain parameters are sampled N_{samp} times.

3.2 Discussion of the modified formulation

The modified formulation is based on the previous formulation (1)-(5). An important difference is in the effect of uncertainty on constraints. In the original formulation, flow patterns do not change due to uncertain population density. So the uncertainty essentially acts as weighting on the objective function. In formulation (6)-(10), flow demands at various nodes are taken to be directly proportional to the population density. As mentioned previously, the uncertain space is discretized through N_{samp} samples of population density. Using its relationship with nodal demands, the network is simulated in EPANET for each sample and corresponding flow patterns along with their frequency

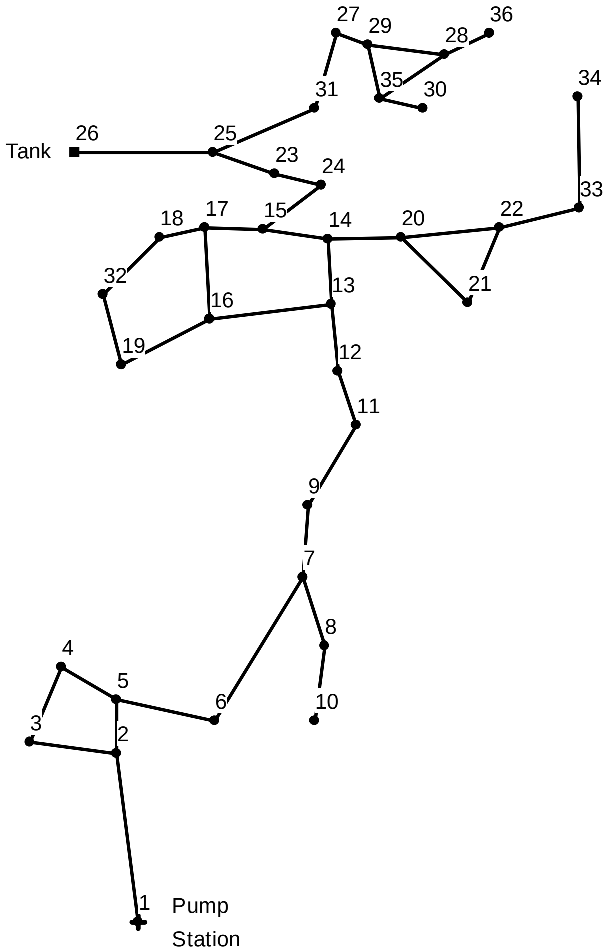

Figure 1: Example Network 2 from EPANET

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	***1***	***1***	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	1	1	1	1	0	1	0	1	1	1	***0***	***0***	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	***1***	0	0	0	0	0	0	0	0	0	0	0	0	***1***	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	1	***1***	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	***0***	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	***1***	***1***	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	***1***	***1***	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	***0***	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
33	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	***0***	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
34	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	***0***	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

Figure 2: Flow pattern 1 for “Example Network 2” of EPANET without sensor placement

of occurrence are recorded, which comprise the set P of possible flow patterns. In the original formulation, provision is made to consider different flow patterns corresponding to gross shifts in demands (such as those observed in the morning and in the afternoon in a network). Note must be made that the proposed additional flow patterns in the new formulation are solely due to demand uncertainties and are an addition to the basic patterns considered in formulation (1)-(5), thereby extending the set of possible flow patterns. Another difference is the incorporation of frequency of pattern occurrence in the modified problem which is disregarded in the original problem.

3.3 Comparison of original and modified formulation

The results for the original and modified formulations are compared for an example water network. Figure 3 shows the water network which is based on the “Example Network 1” of EPANET. There are 12 nodes in the network, comprising of two pumping stations, one storage tank and nine consumptions points. Four nodes, 12, 13, 22 and 23, have uncertain demands while the attack probability is considered to be fixed and equal at all the nodes. One basic demand pattern is considered. It is assumed that demand variation will be the result of population density uncertainty. The nodes with uncertain demands have a variation of $\pm 25\%$ with normal distribution around a mean. As a result, there is one basic pattern and possibility of additional patterns due to uncertain demands. EPANET is used to perform all hydraulic simulations. In the original formulation, only the basic pattern is considered while during simulations it was observed that eight more patterns appear due to uncertain demands which makes the size of P to be nine.

The problem is solved for both formulations using standard sampling based optimization procedure. The comparison shows that the results differ in two aspects, final sensor locations and objective function (risk) value. For example, when the network is allowed to have maximum three sensors, the placement of sensors is different for the two cases. The comparison of estimated and actual objective (risks) is shown in figure 4. It should be noted that the estimated (model) risk for the original formulation is always less than the estimated risk for the modified formulation. This is because the original formulation does not consider all possible flow patterns. The actual risk for the original formulation is calculated by performing stochastic simulations (simulations considering all flow patterns) for the identified sensor locations. For the modified formulation, the estimated risk corresponds with the actual risk since it is computed using stochastic simulations. These actual risks are also plotted in figure 4. It is observed that the actual risk for original problem solution is always higher than estimated risk. However the actual risk for the modified formulation solution, by considering flow pattern uncertainty at the decision making stage, is never higher than the actual risk for the original formulation. These results indicate that the original problem formulation, by ignoring additional flow patterns, gives solutions that are not truly optimum in actual networks. On the contrary, the modified formulation quantifies the actual risk more accurately and hence results in optimum solutions.

4 Proposed extensive formulation

One of the major problems facing water utilities is the cost of sensors. Sensors are available in a broad range of resolution, accuracy and costs. However, the previous formulations did not consider sensor costs. In the following formulation, costs of sensors, β_{ij}, are considered along with uncertainties u in population density and probability of attack, converting it into a two stage stochastic programming problem with recourse and a nonlinear model for water network.

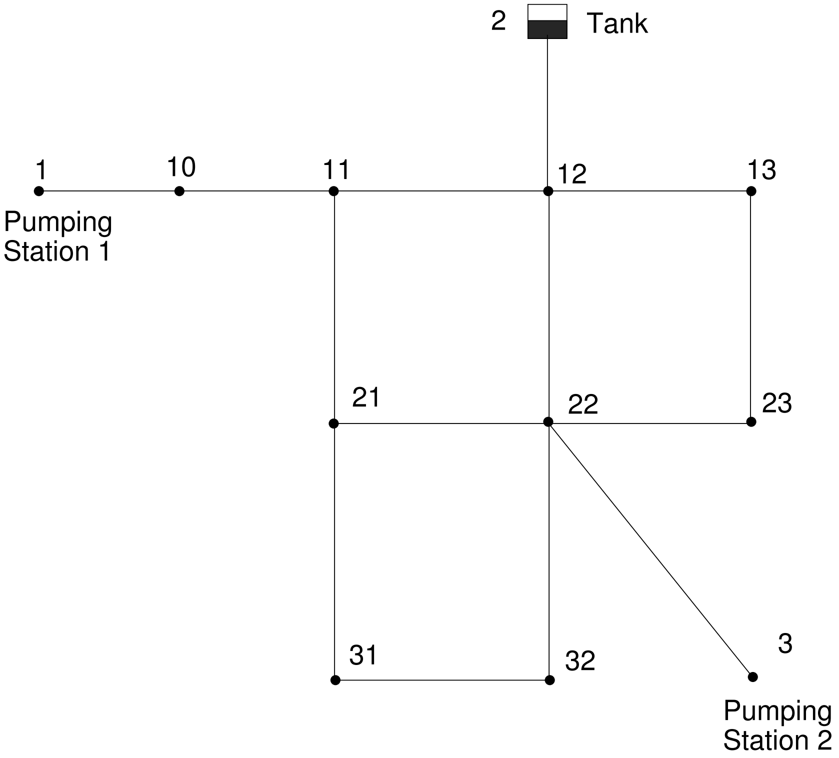

Figure 3: Example Water Network

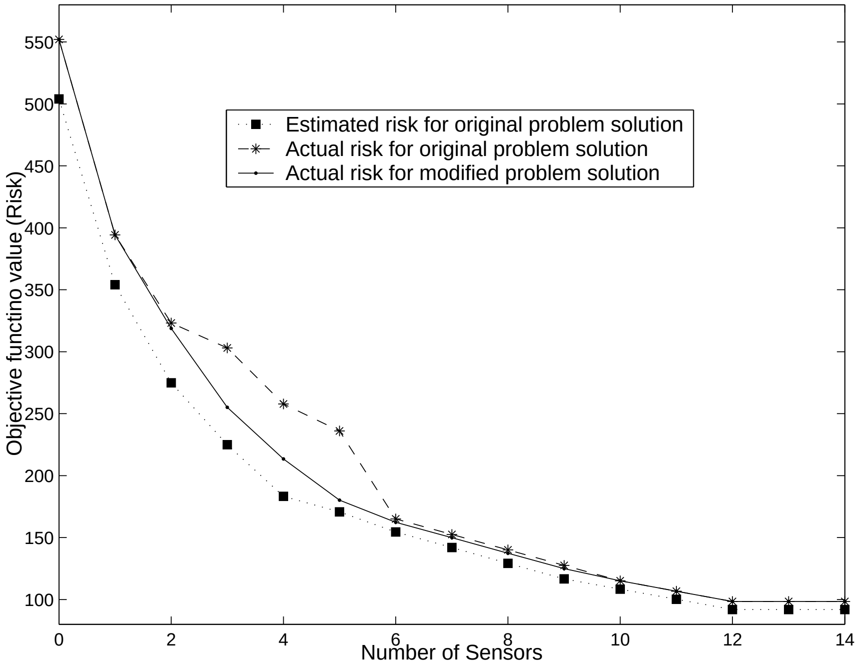

Figure 4: Comparison of the objective function values for two formulations

First Stage Problem

$$\text{Minimize} \quad \sum_{(i,k)\in E} \beta_{ij} s_{ij} \; + \; E[R(C,s,u)] \tag{11}$$

$$where$$

$$s_{ij} \; = \; s_{ji} \qquad\qquad i = 1,\ldots,n-1, \; i \; < \; j \tag{12}$$

$$\sum_{(i,j)\in E, i<j} s_{ij} \; \leq \; S_{max} \qquad\qquad s_{ij} \in (0,1); (i,j) \in E \tag{13}$$

Second Stage Problem

$$\text{Minimize} \quad E[R(C,s,u)] \; = \; \sum_{l=1}^{N_{samp}} \sum_{i=1}^{n} \sum_{p=1}^{P} \sum_{j=1}^{n} S \; \alpha_{ip}(l) \delta_{jp}(l) C_{ipj} \tag{14}$$

$$where$$

$$C_{ipi} \; = \; 1 \qquad\qquad i = 1,\ldots,n; \; p = 1,\ldots,P \tag{15}$$

$$C_{ipj} \; \geq \; C_{ipk} \; - \; s_{kj} \qquad\qquad (i,k,j) \in E; \; s.t. \; f_{kjp} \; = \; 1 \tag{16}$$

In the above formulation, the first stage decisions are s_{ij} variables and recourse variables are the contamination indicators C_{ipj}. S is the cost associated with each person affected by the contamination (e.g. treatment cost). Rest of the notations and their explanations are same as those for formulations (1)-(5) and (6)-(10). The modified formulation previously explained constitutes the recourse function R. The uncertain space is again discretized through N_{samp} samples. Next section discusses the general solution procedure of the two stage problem.

5 Solution Procedure

The problem formulation given by equations (11) to (16) is a two stage stochastic programming problem with recourse and uses sampling to approximate the continuous uncertain space. This calls for the application of sampling based optimization methods where the sampling technique constitutes an important aspect. This work uses Hammersley Sequence Sampling (HSS) technique which has been recently proposed [12, 13] and is shown to have k dimensional uniformity property. The quasi-Monte Carlo technique uses low discrepancy Hammersley sequence to generate Hammersley points. The technique uses these points to uniformly sample a unit hypercube and inverts these points over the joint cumulative probability distribution to provide a sample set for the variables of interest. Two sampling based solution methods are the L-shaped method embedded with sampling [14, 15, 16], and Stochastic Decomposition method [17]. Stochastic Decomposition is suitable for continuous uncertain variables, but generates many cuts with small number of additional samples on each cut which can distort the uniformity of the sample. This work therefore considers the sampling based L-shaped method which uses a fixed number of samples and so can exploit the k dimensional uniformity property of HSS.

L-shaped method is a scenario based method applicable for discrete distributions to solve two or multi stage stochastic programming problems [18]. The first stage problem (master problem) uses a linear approximation of the second stage (nonlinear) recourse function and additional constraints (sequentially generated in second stage) to fix the first stage decision variables. These first stage decision variables are passed on to the second stage, where the task is to compute the exact value of the recourse function using the uncertainty realizations i.e. scenarios. This is done by solving the dual of the second stage problem for every scenario. Solution of all dual problems is used to compute the expected value of the recourse function which is subjected to two tests, the feasibility test and the optimality test. This results in possible generation of two kinds of cuts, the feasibility cut and optimality cut. These cuts are added to the master problem and lead to a better approximation of the recourse function in the master problem solution during the next iteration. The new first stage decisions are again passed on to the second stage where the dual problems are again solved for each scenario. This process of first and second stage problem solution and cut generation is continued iteratively and the objective function value improves with each iteration. The termination criterion is based on the accuracy of the linear approximation of the exact nonlinear recourse function. For sampling based method, scenarios in the second stage are replaced by samples and the termination criterion is based on the statistical property of distributions. New set of samples are taken at each iteration which makes this an internal sampling method in the language of stochastic programming. Please refer to [11] for a detailed explanation of the algorithm.

If the second stage problem solution depends on the value of a model variable, then the model needs to be simulated for each sample in an iteration and at each such iteration which can be a computational bottleneck. For this particular problem EPANET based water network will need to be simulated for each sample of δ_{jp} to generate the C_{ipj} and f_{ijp} values of the second stage dual problem solution. This calls for high computational requirements. The problem is instead solved by implementing a new algorithm proposed by this group to solve stochastic nonlinear programming problems with high computational efficiency. The algorithm will be briefly explained in the next section.

6 Algorithm: L-shaped BONUS

The algorithm structure used in this work is an integration of sampling based L-shaped method and BONUS (Better Optimization of Nonlinear Uncertain Systems), an algorithm recently proposed by Sahin and Diwekar [19]. While the basic structure is that of the L-shaped method described before, BONUS helps reduce the computational burden at the second stage solution. The following sections explain BONUS algorithm in brief and its incorporation into the L-shaped method.

6.1 BONUS

BONUS is a new algorithm proposed to solve Stochastic nonlinear programming (SNLP) problems [19]. Traditional SNLP methods rely on improving the probabilistic objective function by repeated evaluation for each sample in every iteration which is computationally expensive. But in BONUS, as shown in figure 5, instead of running the model for given samples in every iteration, reweighting approach is used to bypass repeated model simulations. The reweighting approach, based on the various reweighting schemes proposed by Hesterberg [20] and as used in BONUS algorithm, is shown in figure 6. An initial, uniform base distribution of the uncertain space is generated and the model is run for each sample to determine the output distribution. At subsequent iterations for the new sample set, normalized weights are generated using the base and new sample sets. These weights along with the known output distribution for the base sample set are used to approximate the probabilistic behavior of the new output distribution. The model is thus not re-run. Appendix A gives the theory behind reweighting approach. A detailed explanation on use of reweighting in optimization algorithm is given in [19]. This concept of reweighting is central to BONUS and has been incorporated in the proposed algorithm.

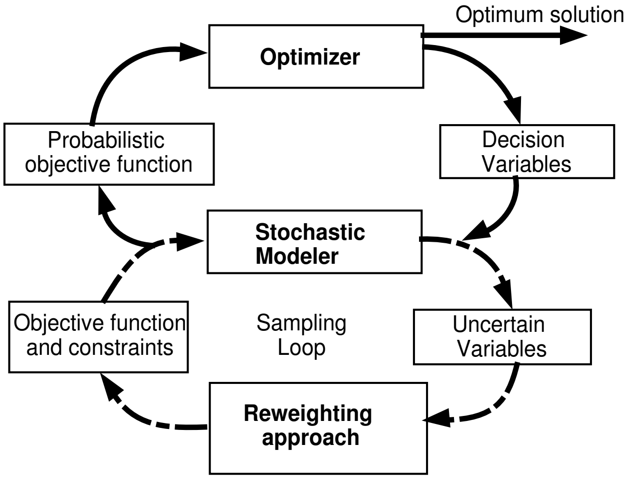

Figure 5: BONUS algorithm

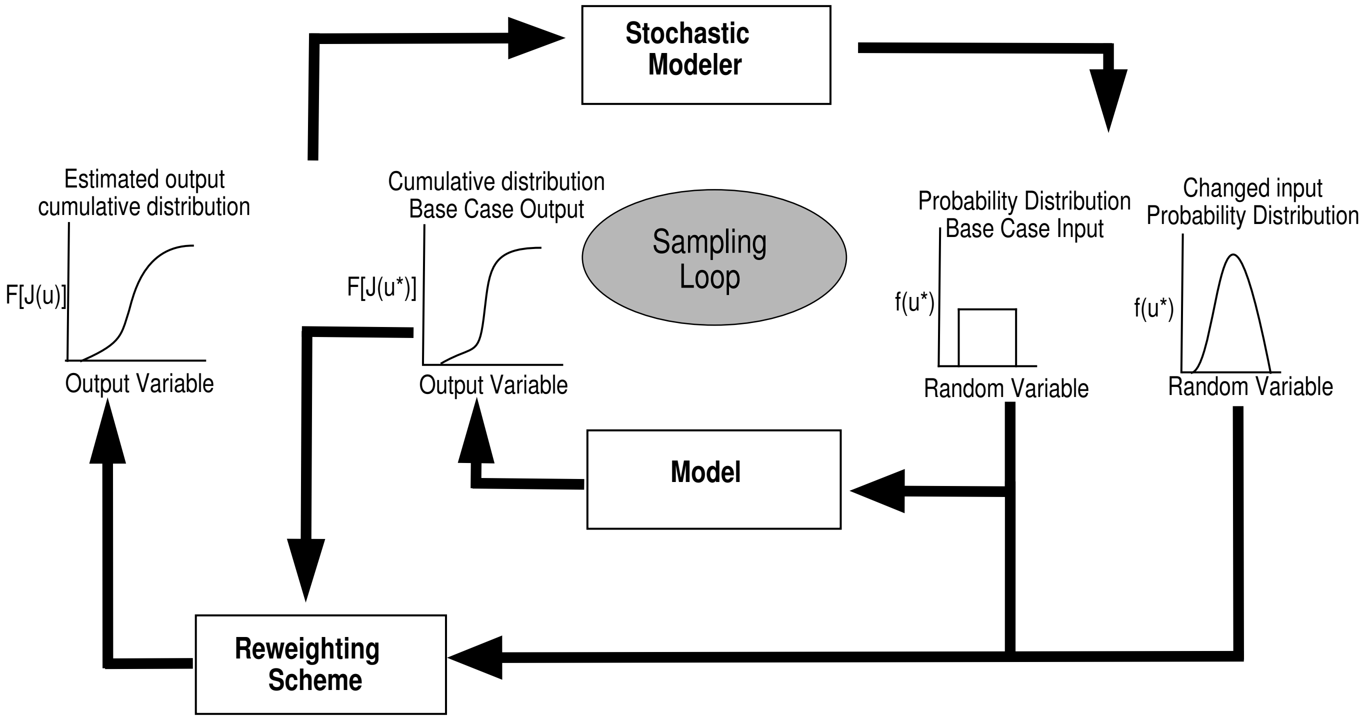
Stochastic Modeler
Estimated output cumulative distribution
F[J(u)]
Output Variable
Cumulative distribution Base Case Output
F[J(u*)]
Output Variable
Sampling Loop
Probability Distribution Base Case Input
f(u*)
Random Variable
Changed input Probability Distribution
f(u*)
Random Variable
Model
Reweighting Scheme

Figure 6: Reweighting scheme

6.2 L-shaped with sampling integrated with BONUS

The new L-shaped BONUS algorithm is schematically shown in figure 7 (Shastri and Diwekar, unpublished manuscript, 2004). First stage decisions are made using linearized approximation of the second stage nonlinear recourse function and utilizing the feasibility and optimality cuts, if generated. This also determines the lower bound for the objective function. The second stage objective is the expected value of recourse function, which depends on first as well as second stage decision (recourse) variables. Following the sampling based L-shaped method, first stage decisions are passed on to the second stage where the sub-problem is solved for each uncertainty realization. The idea of the proposed algorithm is to reduce computations at the sub-problem solution stage by using reweighting scheme to bypass nonlinear model computations. Reweighting scheme, as shown in figure 6 and explained in appendix A, needs the model output distribution for base case uniform input distribution. For this purpose, during the first optimization iteration, the nonlinear model is simulated for each sample generating base case output distribution. Sub-problem solution for each sample is then used to derive optimality cut for the master problem and generate upper bound for the objective function. Second optimization iteration solves the first stage master problem using the cuts and first stage decisions along with an updated lower bound are passed on to the second stage problem. During this iteration, when new set of samples are taken by the stochastic modeler, model simulation is not performed for each sample. The reweighting scheme is used to predict the probabilistic values (expectation) of the model output. The base case output distribution along with the two sample sets are used for this prediction. The expected value of model output is used to solve the second stage dual sub-problem to generate cuts and update the objective function upper bound. This procedure of reweighting based estimation is continued in every subsequent iteration till the L-shaped method based termination criteria is encountered.

For the problem of sensor placement, second stage dual problem needs C_{ipj} values which depend on the flow patterns and their frequency of occurrence, and which require EPANET simulation for each sample. According to the proposed algorithm, EPANET simulations will be performed only for one set of uniform samples. For the subsequent iterations, reweighting scheme will be used to estimate the flow patterns for the new sample set. Since reweighting forms the core of this algorithm, its application to the sensor placement problem will be elaborated in the next section.

7 Use of reweighting for pattern estimation

7.1 The reweighting approach

A particular flow pattern is mathematically identified by the various f_{ijp} values in the network. To check the validity of the reweighting scheme, samples of a fixed sample size are taken for two kinds of distributions, uniform and normal. For these distributions, simulations are performed using EPANET and the value of every f_{ijp} is noted and summed up. For example, if f_{142} has a value of 1 56 times and that of 0 the remaining 44 times in 100 simulations, the number 56 in noted against f_{142}. The cumulative values of all f_{ijp} for the normally distributed samples are then estimated using reweighting scheme which uses f_{ijp} values for the uniformly distributed samples along with samples for uniform and normal distribution.

To estimate the accuracy/error of prediction, expected values of all f_{ijp} for normal distribution, computed through reweighting, are compared with the expected values of f_{ijp} determined by simulations. For example, the actual expected value of f_{142} is 0.56 (56/100). If the estimated expected value is 0.52 then the error of estimation is 0.04. This kind of data is recorded for all nodes in the network and the standard deviation of this error of estimation is used as an indicator of the overall estimation quality. The error of estimation is a function of sample size and number of uncertain variables. These relationships are reported below.

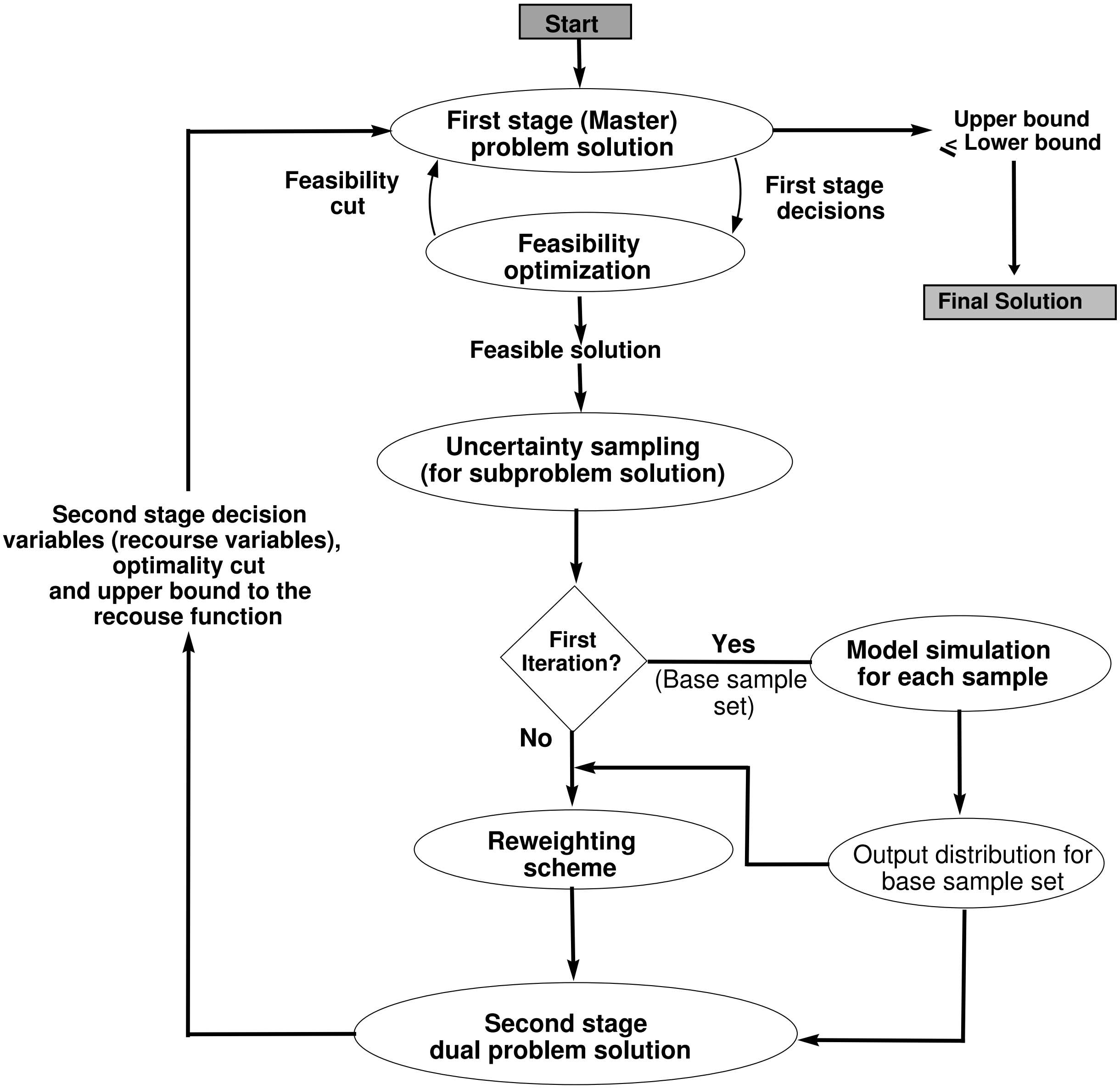
Start
First stage (Master) problem solution
Upper bound ≤ Lower bound
Final Solution
Feasibility cut
First stage decisions
Feasibility optimization
Feasible solution
Uncertainty sampling (for subproblem solution)
Second stage decision variables (recourse variables), optimality cut and upper bound to the recouse function
First Iteration?
Yes
(Base sample set)
Model simulation for each sample
No
Reweighting scheme
Output distribution for base sample set
Second stage dual problem solution

Figure 7: L-shaped BONUS algorithm

7.2 Effect of sample size

The estimation error analysis is carried out for various sample sizes, ranging from 100 to 700 and the variation is shown in figure 8. The estimation error parameter for 700 samples reduces to about 15% of the value for 100 samples. It can therefore be concluded that a larger sample size tends to produce better results.

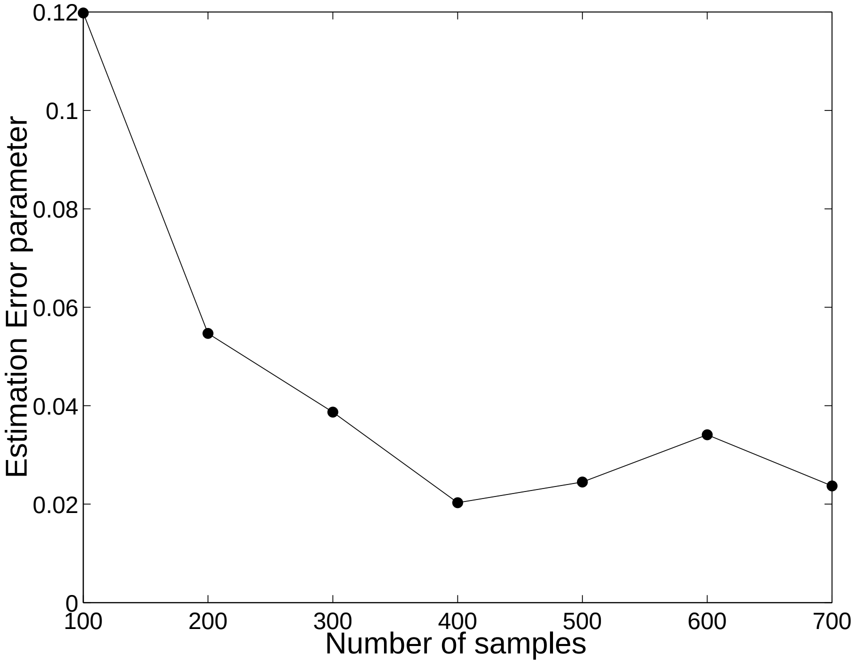

Figure 8: Estimation error dependence on number of samples

7.3 Effect of number of uncertain variables

The number of uncertain nodes is progressively increased and corresponding error of estimation is calculated. The prediction error for one uncertain node is 0.13 while that for eight uncertain nodes is 13.31. This indicates that estimation quality degrades with increasing number of uncertain nodes in the network

7.4 Back estimation of flow patterns

The reweighting approach gives the expected values of all f_{ijp} in the given network for normal distribution. But the problem solution needs the frequency of occurrence of various flow patterns which is to be calculated using the estimated f_{ijp} values. This is achieved by solving an optimization problem. Given a particular number of simulations n (typically 100 in this work), the task is to find out how many times out of n does a particular pattern appear, using the estimated expected values.

The optimization problem mainly works using constraints i.e. constraints are the most important part of it. Let n_p represent the number of times pattern p appears and let P be the total number

Table 1: Demand patterns for example network

	1	2	3	10	11	12	13	21	22	23	31	32
Pattern 1	0	0	0	0	150	150	100	150	200	150	100	100
Pattern 2	0	0	0	0	200	200	50	150	200	100	100	100

of different patterns observed for the uniform sample set. The optimization problem is formulated as:

$$\text{Objective :} \qquad \text{Minimize } (\prod_{p=1}^{P} n_p) \tag{17}$$

$$\begin{aligned} \text{subject to} \qquad \sum_{p=1}^{P} n_p &= 100 \\ \sum_{p=1}^{P} n_p f_{ijp} &= \sum_{m=1}^{N_{samp}} f_{ij(\text{calculated})}(m), \qquad \text{for all } (i,j) \end{aligned} \tag{18}$$

The constraints thus try to achieve the estimated value of summation of f_{ij} using the f_{ijp} values of various known patterns. The optimization problem solution gives the values of n_p, the frequency of occurrence of various patterns. This result can be directly incorporated into the stochastic problem formulation mentioned. For the considered problem, the estimation was found to match the actual pattern distribution to a good extent.

8 Sensor placement problem: results

The previous sections explained the new stochastic programming based formulation for sensor placement problem solution and the new solution algorithm. This section applies the proposed formulation and algorithm on an example water network and compares the results of the proposed formulation with two other formulations.

The same network, used for the comparison of the modified and original formulation (figure 3), is considered for this analysis. The network details can be found in the corresponding section. The attack probability at various nodes is considered to be fixed and equal during any pattern for simplicity. In this analysis, two demand patterns, instead of one, are considered with gross shifts in demands (which might imitate the demand patterns during morning and afternoon hours in an actual network). Mean nodal demands for different patters are given in table 1. The two patterns are referred to as the basis demand patterns. Nodes 12, 13, 22 and 23 have uncertain demands, a variation of $\pm 25\%$ with normal distribution around the corresponding mean reported in table 1. As a result, there are now two basic patterns and possibility of additional patterns due to uncertain demands. EPANET is used to perform all hydraulic simulations. The sensor placement problem is solved using three different formulations to make a comparative study. These are:

1. Deterministic formulation (Method A): Considering the mean demands and two basic flow patterns (no uncertainty)

2. Formulation with noise consideration (Method B): Considering uncertain demands affecting only the objective function i.e. formulation (1)-(5) (original formulation) modified to include sensor costs and cost for affected person (S)

3. Stochastic formulation (Method C): Considering uncertain demands also affecting the constraints i.e. formulation (11) to (16) and solving the problem by L-shaped BONUS algorithm

Sampling for methods B and C is performed using the HSS technique and sample size is 100 for the reported results. The sample size is a compromise between desired accuracy and computational load and is not network specific. For method C, various flow patterns for the uncertain demands are identified for a base case uniform sample set via off-line simulations. The consideration of uncertainty generates 8 more flow patterns apart from the two basic patterns considered in methods A and B.

Sensor costs can vary depending on the type of sensor. A simple chlorine sensor for water can cost around $5000 while a real time multi-component sensor with a real time controller can cost much more. Two types of sensors are considered, those with low cost of $1500000 per sensor and those with high cost of $4500000 per sensor. The cost per affected person (S) is $30000. It is difficult to estimate the value of S since it will depend on the chemical as well as treatment costs at a particular place. The sensor costs and cost per affected person S are assumed. They are decided so as to show a tradeoff between different parts of the objective function in equation (11). The assumed costs also avoid possible numerical problems by adjusting both parts of equation (11) in the same numerical range.

The problem is solved with the maximum number of allowed sensors varying from 1 to 14. The estimated (model) and *actual* values of the objective function and percentage population at risk for some selected results are compared in tables 2 and 3, respectively. In table 2, the estimated values for methods A and B are very close to each other and do not appear to have any particular trend. The estimated values for the stochastic method C are always higher than those for methods A and B. However, these values should not be used for comparing different methods. More important are the *actual* objective and percentage population at risk values for the sensor locations identified by different methods. These values are reported in table 3. Probabilistic methods are generally used to estimate the actual objective function values in any problem. For methods A and B, the actual objective function and percentage population at risk values are calculated through stochastic simulations. Since method C performs such a probabilistic analysis in decision making, the estimated values reported in table 2 and the actual values are same for method C. The results show that in all cases, the actual objective function value for the stochastic method C is never higher than those for the other two methods. In many cases, the actual objective function values for methods A and B are higher than those for method C manifesting that the optimal solutions obtained by methods A and B are not truly optimum. This is due to the inability of formulations A and B to model uncertainty effects completely in decision making. Since the objective for optimization is total cost and not only the population at risk, some solutions for methods A and B result in lesser population at risk than method C but are still sub-optimal in terms of the total objective function. Such results are seen in table 3 for two and four maximum allowed sensors when high cost sensors are used.

The optimal locations identified by various methods are tabulated in table 4 in terms of branches with sensors, some of which are shown in figures 9 and 10 for low and high cost sensors, respectively. The branches with sensors are identified by the notations along these branches indicating the solution method and the maximum number of permitted sensors. For example, method B for 2 permitted sensors selects branches 10-11 and 12-22. Therefore notation ‘b2’ appears alongside these branches. The optimal locations for various methods also indicate some differences. Branch 10-11, being one of the entry points into the network, is identified as the most important one by all the methods. The results however differ when two low cost sensors are allowed. While method A and method C identify the second entry point 3-22 as the second most important branch, method B identifies branch 12-22 to be of greater importance (figure 9). This is because in the two basic flow patterns, the flow in branch 12-22 is always directed from 12 to 22. Therefore contamination at 12 is affecting nodes 21, 22 and 23 while high consumption node 12 is always immune to contamination at nodes 3 and 22. But, uncertainty considerations reveal the possibility of flow reversal in branch 12-22. This diminishes the effect of node 12 on 21, 22 and 23 while increases the combined relative effect of node 3 on nodes 12, 13, 21, 22 and 23. Method C therefore identifies branch 3-22 to be more important to reduce the overall risk. Similar reasons can be given to the observations with four low cost sensors. For high cost sensors, the results suggest that application of more than two sensors (one for method C) will not be cost effective when weighted against the advantages derived in terms of cost (figure 10). But this leaves greater population at risk of contamination. These results indicate a tradeoff between the desired safety and cost. It is also observed that increase in per person treatment cost S results in implementation of more sensors in the network since the importance of population at risk in the objective function increases. The results also suggest that location identification in the presence of uncertainty can be beyond the intuitive understanding of network hydraulics.

It is important to note that the differences in the optimal solutions are large enough to not ignore. The differences in the estimated objective and risk are as high as 30%. Since the estimated values for method C are the actual values, this difference emphasizes the extent of sub-optimality of results by methods A and B. Another important point to consider is the reduction in computational time with the proposed algorithm. The stochastic programming problem is solved for some selected cases using the standard L-shaped method and L-shaped BONUS algorithm. It should be noted that the solution using standard L-shaped method considers a fixed set of scenarios for uncertain demands and corresponding distribution of various patterns. When computational times of the two solution methods are compared for a sample size of 100, reduction by a factor of 5 is observed for the proposed L-shaped BONUS algorithm over the standard L-shaped method. It is also observed that the computational time increases exponentially with the sample size for the L-shaped method while it increases linearly for the proposed algorithm. The results also show that the average difference between the optimum cost and percentage population at risk is about 4% and 5.4%, respectively, which is within reasonable limits. The proposed L-shaped BONUS algorithm is thus not compromising the accuracy of result.

The proposed stochastic programming formulation therefore gives a more realistic insight into the sensor placement problem by accounting for uncertainties and the proposed algorithm improves the computational efficiency. Moreover this was a small network. Bigger real life networks are expected to be more susceptible to uncertainties and advantages of the proposed L-shaped BONUS algorithm will become more prominent. This is because although estimation quality degrades with increasing number of uncertain nodes in the network, this effect can be countered using larger sample size. There can also be larger networks with same number of uncertain nodes. For example, if the example network is expanded to include 10 more nodes with deterministic demands, it will result in a larger network (22 nodes) with same number of uncertain nodes (four). Simulation load for traditional optimization methods will increase substantially for such network. In comparison, simulation load is considerably less for the proposed method due to reweighting approach while accuracy is unaffected since estimation quality is a function of number of *uncertain* nodes and not the total number of nodes.

Table 2: Comparison of estimated cost and risk for different solution methods

Maximum number of allowed sensors	Type of sensor	Method A		Method B		Method C	
		Cost($) $(\times 10^7)$	Percentage risk	Cost($) $(\times 10^7)$	Percentage risk	Cost($) $(\times 10^7)$	Percentage risk
1	Low Cost	2.1875	29.375	2.1067	29.647	2.2860	32.364
	High Cost	2.4875	29.375	2.4045	29.647	2.5860	32.364
2	Low Cost	1.8625	22.727	1.7954	22.658	1.9914	25.628
	High Cost	2.4625	22.727	2.3937	22.658	2.5860	32.364
4	Low Cost	1.7125	16.856	1.6964	18.885	1.8062	18.277
	High Cost	2.4625	22.727	2.3937	22.658	2.5860	32.364

Table 3: Comparison of actual cost and risk for different solution methods

Maximum number of allowed sensors	Type of sensor	Method A		Method B		Method C	
		Cost($) ($\times 10^7$)	Percentage risk	Cost($) ($\times 10^7$)	Percentage risk	Cost($) ($\times 10^7$)	Percentage risk
1	Low Cost	2.2860	32.364	2.2860	32.364	2.2860	32.364
	High Cost	2.5860	32.364	2.5860	32.364	2.5860	32.364
2	Low Cost	1.9914	25.628	2.1193	27.565	1.9914	25.628
	High Cost	2.5914	25.628	2.7193	27.565	2.5860	32.364
4	Low Cost	1.9738	20.816	1.9106	19.858	1.8062	18.277
	High Cost	2.5914	25.628	2.7193	27.565	2.5860	32.364

Table 4: Comparison of optimal sensor location results of different solution methods

Maximum number of allowed sensors	Type of sensor	Method A	Method B	Method C
1	Low Cost	10-11	10-11	10-11
	High Cost	10-11	10-11	10-11
2	Low Cost	10-11,3-22	10-11,12-22	10-11,3-22
	High Cost	10-11,3-22	10-11,12-22	10-11
4	Low Cost	10-11,3-22, 21-31,12-13	10-11,12-22, 3-22,12-13	10-11,3-22, 12-22,21-31
	High Cost	10-11,3-22	10-11,12-22	10-11

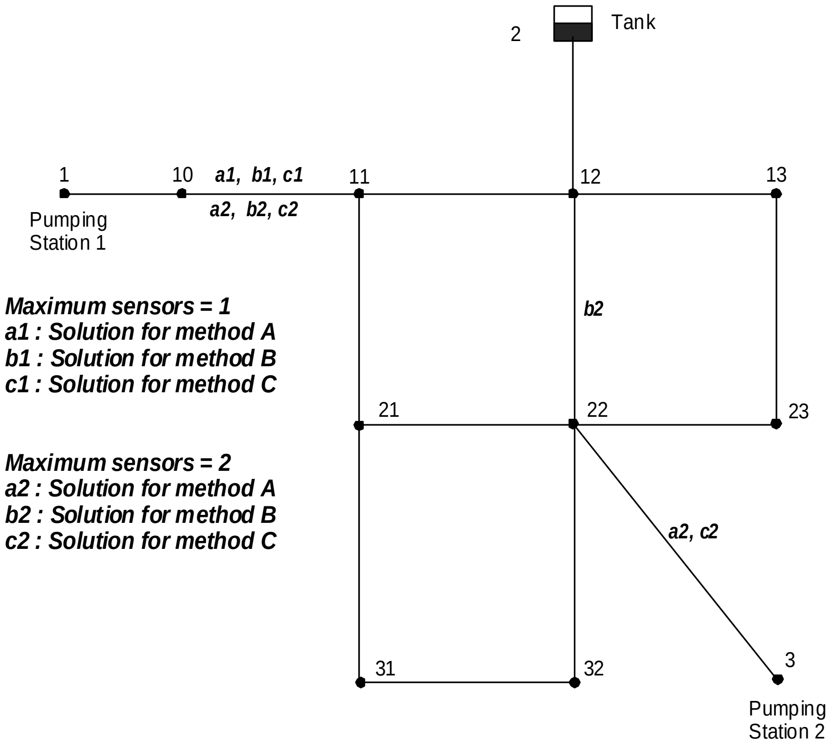
2
Tank
1
10
a1, b1, c1
11
12
13
a2, b2, c2
Pumping
Station 1
Maximum sensors = 1
a1 : Solution for method A
b1 : Solution for method B
c1 : Solution for method C
b2
21
22
23
Maximum sensors = 2
a2 : Solution for method A
b2 : Solution for method B
c2 : Solution for method C
a2, c2
31
32
3
Pumping
Station 2

Figure 9: Placement of sensor with different methods for low cost sensors

9 Conclusion

The problem of optimal sensor placement in a water distribution network is considered in this paper. Starting from an IP formulation proposed earlier, the work proposes a new stochastic programming formulation that is more robust in the face of uncertainties. It extends the effect of uncertainty to a greater level, including the effects on constraints along with the objective function by recognizing the possibility of changes in network flow patterns due to uncertainty. Also considered are the cost of sensors and cost for each affected person. This converts the problem into a two stage stochastic integer programming problem with nonlinear network models. The solution of this problem can be computationally taxing using traditional stochastic programming methods. A new algorithm, proposed by the authors to efficiently solve stochastic nonlinear programming problems, is used to solve the proposed formulation. Comparative results on the example network show that differences are appreciable enough to render this new approach important and valuable. It is also observed that the problem solution is computationally quite efficient. The results of the proposed formulation on bigger real life models are expected to throw more light on the efficiency and characteristics of the new formulation.

Acknowledgements

This work is funded by the National Science Foundation under the grant CTS-0406154.

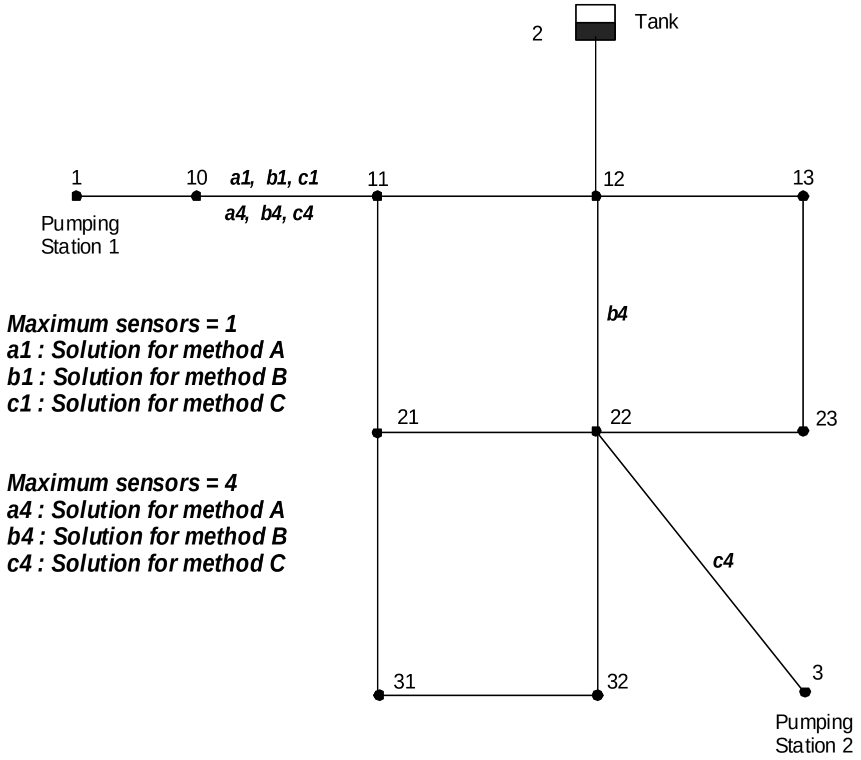
Tank
2
1
10
a1, b1, c1
a4, b4, c4
11
12
13
Pumping Station 1
b4
Maximum sensors = 1
a1 : Solution for method A
b1 : Solution for method B
c1 : Solution for method C
21
22
23
Maximum sensors = 4
a4 : Solution for method A
b4 : Solution for method B
c4 : Solution for method C
c4
31
32
3
Pumping Station 2

Figure 10: Placement of sensor with different methods for high cost sensors

A Reweighting

Reweighting approach is based on various reweighting schemes proposed in [20]. It is an extension of the importance sampling concept of estimating something about a distribution (target distribution $f(x)$) using observations from a different distribution (design distribution $g(x)$), where these distributions are represented by respective probability density functions. Let X be a random variable with probability density function $f(x)$ and $Q(X)$ be a function of X. Then to estimate a certain property of $Q(X)$, such as the expected value $\mu = E_f[Q(X)]$, importance sampling solves a different problem of estimating $E_g[Y(X)]$, where

$$Y(x) = Q(x)\frac{f(x)}{g(x)} \tag{19}$$

and samples X_i are now drawn from $g(x)$. Distribution $g(x)$ can be designed to achieve desired results (e.g. reduced variance, better representation of rare events). A weight function $W(x)$ is defined as

$$W(x) = \frac{f(x)}{g(x)} \tag{20}$$

which gives the likelihood ratio between target and design distributions and weighs observations of $Q(x)$. To perform this estimation effectively, Hesterberg [20] proposed various design distributions $g(x)$ (e.g. defensive mixture distributions) and estimation schemes (integration estimate, ratio estimate). In ratio estimate, weights W_i are normalized to avoid problems when they do not sum to 1. The normalized weights V_i and estimate μ is given as

$$V_i = \frac{W_i}{\sum_{j=1}^{n} W_j} \tag{21}$$

$$\mu = \sum_{i=1}^{n} V_i Q(X_i) \tag{22}$$

where n is the sample size. Means, higher moments and percentiles can be computed using such relations. The reweighting scheme in the proposed algorithm is based on this ratio estimate.

The reweighting approach, as used in the BONUS algorithm, is schematically shown in figure 6. Suppose X represents the uncertain variable in stochastic programming problem and $Q(X)$ is the output of stochastic modeler. For the first iteration, base case samples X_i^* with uniform distribution ($g(x)$) are drawn and the model is simulated for each sample to get the complete model output distribution $Q(X_i^*)$. During the subsequent iterations, new samples X_i of required distribution ($f(x)$) are drawn. Having known the model response $Q(X_i^*)$ for sample set X_i^* from distribution $g(x)$, it is possible to use equation 22 to estimate the expected value of model response $Q(X_i)$ for new sample set X_i from distribution $f(x)$. The expected value of the stochastic model response $Q(X_i)$ for new sample set X_i is therefore given as

$$E_f[Q(X_i)] = \sum_{j}^{n} \frac{\frac{f(X_j)}{g(X_j^*)}}{\sum_{i=1}^{n} \frac{f(X_i)}{g(X_i^*)}} \, Q(X_j^*) \tag{23}$$

In a sampling based algorithm, use of this procedure calls for determining the probability density function from the available sample set. This is carried out using the Gaussian Kernel Density Estimation technique [21] which is a nonparametric density estimation technique. The basic idea behind this technique is to place a bin of certain width $2h$ around every sample X and weigh that sample by the number of other samples X_i in the same bin. If this bin is replaced by a kernel function such as normal density function, the density function for the sample set X_i is calculated using equation 24.

$$f(X) = \frac{1}{n.h} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{X-X_i}{h})^2} \tag{24}$$

where h is the window width, also called the smoothing parameter or bandwidth. Value of h decides the fineness of density estimation. For this work, it is taken as the standard deviation of the sample set.

Thus, given two sample sets, equation 24 is used to determine the density function at each sample point for both the distributions which are then used in equation 23 to find out the output distribution for the second sample set.

References

[1] G. Eiger, U. Shamir, and A. Ben-Tal. Optimal design of water distribution networks. *Water Resources Research*, 30(9):2637–2646, 1994.

[2] K.V.K. Varma, S. Narasimhan, and S.M. Bhallamudi. Optimal design of water distribution systems using an NLP method. *Journal of Envir. Engg.*, 123(4):381–388, 1997.

[3] D.A. Savic and G.A. Walters. Genetic algorithms for least cost design of water distribution networks. *Journal of Water Res. Planning and Management*, 123(2):67–77, 1997.

[4] C. Xu and I. Goulter. Reliability based optimal design of water distribution networks. *Journal of Water Res. Planning and Management*, 125(2):352–362, 1999.

[5] J. Berry, L. Fleischer, W. Hart, C. Phillips, and J. Watson. Sensor placement in municipal water networks. *To appear in special issue of Journal of Water Res. Planning and Management*, 2005.

[6] A. Kessler, A. Ostfeld, and G. Sinai. Detecting accidental contaminations in municipal water networks. *Journal of Water Res. Planning and Management*, 124:192–198, 1998.

[7] A. Kumar, M.L. Kansal, and G. Arora. Discussion of detecting accidental contaminations in municipal water networks. *Journal of Water Res. Planning and Management*, 125:308–310, 1999.

[8] M.E. Tryby, D.L. Boccelli, J.G. Uber, and L.A. Rossman. Facility location model for booster disinfection of water supply networks. *Journal of Water Res. Planning and Management*, 128:322–332, 2002.

[9] U.M. Diwekar. *Introduction to Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2003.

[10] L.A. Rossman. *EPANET users manual*. Risk reduction engg. lab, Environmental Protection Agency, Cincinnati, Ohio, 1993.

[11] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer series in Operations Research, 1997.

[12] J.R. Kalagnanam and U. Diwekar. An efficient sampling technique for off-line quality control. *Technometrics*, 39(3):308–319, 1997.

[13] R Wang, U. Diwekar, and E. Grégorie Padró. Efficient sampling techniques for uncertainties and risk analysis. *Environmental Progress*, 23(2):141–157, July 2004.

[14] G.B. Dantzig and P.W. Glynn. Parallel processors for planning under uncertainty. *Annals of Operations Research*, 22:1–21, 1990.

[15] G. Infanger. Monte carlo (importance) sampling within a benders decomposition algorithm for stochastic linear programs. *Annals of operations research*, 39:69–95, 1992.

[16] G.B. Dantzig and G. Infanger. *Computational and Applied Mathematics I*. Elsevier Science Publishers, B.V. (North Holland), 1992.

[17] J. Higle and S. Sen. Stochastic decomposition: an algorithm for two stage linear programs with recourse. *Annals of Operations Research*, 16:650–669, 1991.

[18] R. Van Slyke and R.J-B Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[19] K.H. Sahin and U.M. Diwekar. Better Optimization of Nonlinear Uncertain Systems (BONUS): A new algorithm for stochastic programming using reweighting through kernel denisty estimation. *Annals of Operations Research*, 132:47–68, 2004.

[20] T. Hesterberg. Weighted average importance sampling and defensive mixture distribution. *Technometrics*, 37:185–194, 1995.

[21] B.W. Silvermann. *Density estimation for statistics and data analysis*. Chapman and Hall (CRC reprint 1998), Boca Raton, USA, 1986.