Incremental Sensor Placement Optimization on Water Network*

Xiaomin Xu[1], Yiqi Lu[1], Sheng Huang[2], Yanghua Xiao[1,**], and Wei Wang[1]

[1] School of computer science, Fudan University, Shanghai, China
{10210240041,10210240030,shawyh,weiwang1}@fudan.edu.cn
[2] IBM China Research Lab, Shanghai, China
huangssh@cn.ibm.com

Abstract. Sensor placement on water networks is critical for the detection of accidental or intentional contamination event. With the development and expansion of cities, the public water distribution systems in cities are continuously growing. As a result, the current sensor placement will lose its effectiveness in detecting contamination event. Hence, in many real applications, we need to solve the *incremental sensor placement* (ISP) problem. We expect to find a sensor placement solution that reuses existing sensor deployments as much as possible to reduce cost, while ensuring the effectiveness of contamination detection. In this paper, we propose scenario-cover model to formalize ISP and prove that ISP is NP-hard and propose our greedy approaches with provable quality bound. Extensive experiments show the effectiveness, robustness and efficiency of the proposed solutions.

1 Introduction

Monitoring water networks in cities for the safety of water quality is of critical importance for the living and development of societies. One of the efforts for water safety is building *early warning systems* (EWSs) with the aim to detect contamination event promptly by installing sensors in a water distribution network. When placing sensors in a water network, it is always desired to maximize the effectiveness of these sensors with minimal deployment cost. Hence, optimizing the sensor placement on water network so that the adverse impact of contaminant event on public health is minimized under the budget limit becomes one of the major concerns of researchers or engineers when building EWSs.

Problem Statement. With the development and expansion of cities, especially those cities in developing countries like China, India, the public municipal water distribution systems are ever-expanding. As a consequence, the current sensor placement of EWS generally will lose its effectiveness in detecting contamination events. See Figure 1 as an example. The sensor deployment in the original water network is shown in Figure 1(a). Sometime later, the water network may significantly expand (the expanded network is shown in Figure 1(b)). Clearly, if we keep the sensor deployment

* Supported by the National NSFC (No. 61003001, 61170006, 61171132, 61033010); Specialized Research Fund for the Doctoral Program of Higher Education No. 20100071120032; NSF of Jiangsu Province (No. BK2010280).

** Correspondence author.

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part III, LNAI 8190, pp. 467–482, 2013.

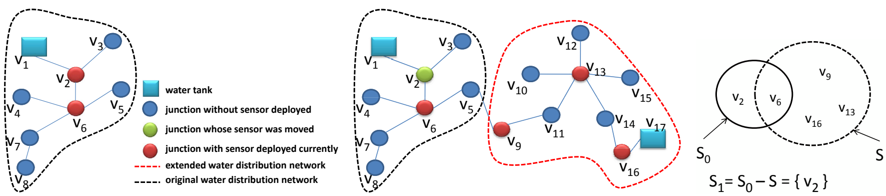
unchanged, it will be hard for the sensors in the original locations to detect the contamination events occurring in the expanded part of the network (marked by the red dotted line in Figure 1(b)).

To keep the effectiveness of EWS in detecting such contamination events, in most cases people needs to add more sensors or reinstall existing sensors. In the previous example, to ensure the effectiveness of contamination detection on the expanded network, we may move one sensor deployed in v_2 (the green node in Figure 1(a)) from the original network and reinstall it on a location (say v_9) of the expanded network. We may also need to buy two new sensors and deploy them on v_{13} and v_{16}.

Effectiveness of above redeployment comes at the cost of reinstalling and adding new sensors. However, in many cases, the budget is limited. To reduce the cost, the newly installed sensors and reinstalled sensors are always expected to be minimized. Consequently, in real applications of water network, an *incremental sensor placement* (ISP) problem usually arises, in which *we need to find a sensor placement that reuses existing sensor placement as much as possible, and meanwhile guarantees the effectiveness to detect contaminations.*

Despite of the importance of ISP, rare efforts can be found to solve it efficiently and effectively. Most of existing related works focus on sensor placement on a static water network. These solutions on static water networks can not be straightforwardly extended to solve ISP since reusing current sensor placement is a new objective of the problem. If we directly recalculate a sensor placement on the current water networks, the resulting sensor deployment may not have sufficient overlap with the current deployment, which will lead to high cost of redeployment. Furthermore, the water networks in real life are generally evolving in a complicated way, which also poses new challenges to solve ISP.

In summary, it is a great challenge to design a strategy to deploy sensors incrementally under limited budget without losing the effectiveness of contamination detection.



(a) Original Network (b) Current Network (c)

Fig. 1. Demonstration of a possible incremental sensor deployment solution. Originally, 2 sensors were deployed. When the network was expanded, 2 additional sensors would be deployed and one original sensor (marked by green node) was moved to the expanded part of the network.

Contribution and Organization. To overcome the difficulties stated above, we reduce the ISP problem to *maximal coverage* (MC) problem. Based on this model, we develop several heuristic algorithms to solve ISP. The contribution of this paper can be summarized as follows:

1. We reduce sensor placement optimization to *maximal coverage* problem.

2. We propose the problem of ISP. We show that ISP is NP-hard, and the objective function of ISP is *submodular*.
3. We propose a series of greedy algorithms with provable bound on the solution quality to solve static sensor placement and then extend them to solve ISP.
4. We conduct extensive experiments to show the effectiveness, robustness and efficiency of our proposed solutions.

The remainder is organized as follows. Sec 2 briefly reviews the related works. Sec 3 formalizes the ISP problem and builds the theory for this problem. In Sec 4, we present the solutions for the proposed problems. Sec 5 shows the experimental results. We close the paper with a brief conclusion in Sec 6.

2 Related Work

A large number of approaches have been proposed for optimizing water sensor networks. They differ from each other in the design and/or performance objective. [1] developed a formulation related to a set covering problem with an premise that sampling at a location supplied by upstream nodes provides information about water at the upstream nodes. Subsequently several researchers refined the model by greedy heuristic-based algorithm [2] and genetic algorithm [3]. [4,5] introduced a scenario in which the objective is to ensure a pre-specified maximum volume of contaminated water consumed prior to detection and also reduced this problem to set cover problem. [6,7] introduced an MIP(mixed integer programming) solution for the objective to minimize the expected fraction of population exposed to a contamination. The objective of [8,9] is to ensure that the expected impact of a contamination event is within a pre-specified level, and [8] introduced a formulation based on set cover and solved the problem using genetic algorithm while [9] use a MIP based solution. In order to achieve the objective defined in [10], [11,12,13,14,15,16,17,18,19,20,21] adopted multi-objective optimization by different methods such as heuristic, predator-prey model or local search method and [22] used the submodular property to achieve an approximation guarantee.

In our paper, we use the concept of submodularity [23,24] to solve the problem of sensor placement on dynamic water network. Submodularity was widely used in sensor placement optimization [22,25]. But these solutions are mostlybuilt for static water network. Besides sensor placement, submodularity has also been widely adopted in finding influencers [26], influence maximization [27] and network structural learning [28]. All the solutions mentioned above are not designed for incremental sensor placement, which is a more realistic problem in real-world. To the best of our knowledge, this paper is the first one addressing the incremental sensor placement on dynamic water networks.

3 Incremental Sensor Placement

In this section, we formalize ISP. We start this section with the introduction of preliminary concepts in Sec 3.1. Please refer to [6] for more background knowledge. Then, we propose a scenario-cover based model in Sec 3.3, upon which ISP is defined in Sec 3.2. Finally, we show that ISP is NP-hard in Sec 3.4 and the submodularity of the objective function used in ISP in Sec 3.5.

3.1 Preliminaries

A water distribution system is modeled as an undirected graph $G = (V, E)$, with vertices in V representing junctions, tanks, and other resources in the systems; edges in E representing pipes, pumps, and valves. Suppose we need to monitor a set of *contamination scenarios* $\mathcal{A}$. Each *contamination scenario* $c \in \mathcal{A}$ can be characterized by a quadruple of the form (v_x, t_s, t_f, X), where $v_x \in V$ is the origin of the contamination event, t_s and t_f are the start and stop times of a contamination event, respectively, and X is the contamination event profile (which describes the contamination material injected at a particular concentration or at a given rate).

Let $L \subseteq V$ be the set of all possible *sensor locations* (*e.g.* junctions in the water network). A placement of p sensors on $L \subseteq V$ is called a *sensor placement*. In general, we do not distinguish sensors placed at different locations. Thus, a subset of L holding the sensors can uniquely determine a sensor placement. In this paper, we always use the subset of L to describe a sensor placement.

One of key issues in water sensor placement is the measurement of the total impact of contamination scenario c when c is detected by a sensor deployed at vertex v in *a sensor placement* S. We use $d_{c,v}$ to denote such an impact. More specifically, $d_{c,v}$ represents the total damaging influence caused by contamination scenario c during the time period from the beginning of c to the time point at which c was detected by a sensor deployed at vertex v. $d_{c,v}$ can be defined from various aspects, including *volume of contaminated water consumed* [4], *population affected* [6], and *the time until detection* [2]. In this paper, we use *the time until detection* as the quantitative criteria to evaluate the adverse impact of each contamination scenario. Solutions proposed in this paper can be directly extended on other adverse impact measures.

The contamination scenarios can be simulated by water quality analysis software (*e.g.* EPANET [1]). Based on the simulation data, $d_{c,v}$ can be computed accordingly. In this paper, each $d_{c,v}$ can be considered as the input of the major problem that will be addressed. Given a set of contamination scenarios $\mathcal{A}$ and sensor locations L, we have a $|\mathcal{A}| \times |L|$ matrix with each element representing $d_{c,v}$. We call this matrix as *contamination scenario matrix* (CS for short) and denote it by $D_{\mathcal{A},L}$. We use Example 1 to illustrate above basic concepts.

Example 1 (Basic Concepts). We give in Table 1(a) a scenario set (i.e., $\mathcal{A}$) for water distribution network described in Figure 1(a). A matrix $D_{\mathcal{A},L}$ provided by contamination simulation is given in Table1(b), in which *the time until detection* is adopted as the measure of the adverse impact of each contamination scenario. As an example, $d_{c_1,v_1} = 7$ implies that placing a sensor on v_1 can detect contamination event c_1 within 7 minutes.

3.2 Scenario-Cover Based Modeling

In this subsection, we will reduce sensor placement optimization to maximal coverage problem. Based on this model, we will formalize the static sensor placement problem

[1] http://www.epa.gov/nrmrl/wswrd/dw/epanet.html

Table 1. Table 1(a): Scenario set $\mathcal{A}$. Table 1(b): Contamination scenario matrix $D_{\mathcal{A},L}$. Each element in the matrix denotes the the time (in minutes). Table 1(c): $R_{v,M}$ for each vertex shown in Figure 1(a).

(a)

scenarios	v_x	t_s	t_f	***X(mg/min)***
c_1	v_1	0:00	1:00	1000
c_2	v_2	0:20	1:00	1000
c_3	v_7	0:15	1:00	1000
c_4	v_5	0:12	1:00	1000

(b)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
c_1	7	9	12	18	14	13	23	14
c_2	12	5	8	16	12	12	15	17
c_3	14	12	16	15	12	7	5	11
c_4	26	18	17	13	5	7	14	15

(c)

V	$R_{v,M}$	V	$R_{v,M}$
v_1	$\{c_1\}$	v_5	$\{c_4\}$
v_2	$\{c_1,c_2\}$	v_6	$\{c_3,c_4\}$
v_3	$\{c_2\}$	v_7	$\{c_3\}$
v_4	$\emptyset$	v_8	$\emptyset$

and incremental sensor placement problem. After that, in subsection 3.4, we will show that problem ISP (Definition 4) is NP-hard.

In real applications, it is a reasonable objective to limit the worst damage caused by a potential contamination event under a certain level. For example, we usually expect that the contaminated population can not exceed a certain threshold when we choose a rational water sensor placement strategy. We use *credit* M to capture the worst damage that we can afford.

Definition 1 (Covered Scenario with Credit M). *Given a contamination scenario matrix $D_{\mathcal{A},L}$ and a sensor placement S, if a contamination scenario c can be detected by a sensor deployed at vertex $v \in S$ such that $d_{c,v} \leq M$, then we say that the contamination scenario c is* covered by S with credit M.

Given the contamination scenario matrix $D_{\mathcal{A},L}$, the set of scenarios in $\mathcal{A}$ that is covered by v with credit M can be uniquely determined. This set can be formally defined as:

$$R_{v,M} = \{c | d_{c,v} \in D_{\mathcal{A},L}, d_{c,v} \leq M\} \tag{1}$$

$R_{v,M}$ is the set of scenarios in $\mathcal{A}$ whose total harmful impact is within credit M if we place a sensor on vertex v. If we add vertex v into sensor placement S, then each contamination scenario in $R_{v,M}$ would be covered by S. In other words, $R_{v,M}$ represents the contamination detection performance when v is added into sensor placement.

Given the water network, $D_{\mathcal{A},L}$ and credit M, we hope that *the sensor placement can cover contamination scenarios with credit M as many as possible*. For this purpose, we use $F(S)$, an evaluation function defined on sensor placement, formally defined in Definition 2, to precisely quantify the *the number of scenarios* covered *by S with credit M*. Thus, our design objective is to maximize $F(S)$.

Definition 2 (Evaluation Function). *Given a credit M and a contamination scenario matrix $D_{\mathcal{A},L}$, the evaluation function $F(S)$ of a sensor placement $S \subseteq L$ is defined as:*

$$\text{F(S)} = |\bigcup_{v \in S} R_{v,M}| \tag{2}$$

Now the *static sensor placement optimization* on a water network can be rewritten in the scenario-cover based model. We illustrate key concepts in scenario-cover based modeling and the SP problem based on this modeling in Example 2.

Definition 3 (Sensor Placement (SP)). *Given a water network G with sensor locations L, scenario set $\mathcal{A}$, CS matrix $D_{\mathcal{A},L}$, a credit M and an integer p, finding a sensor placement $S \subseteq L$ such that $|S| \leq p$ and $F(S)$ is maximized.*

Example 2 (Scenario-cover Based Modeling). Continue the previous example and consider the water network shown in Figure 1(a). Suppose the detection time credit $M = 10$ minutes and contamination scenario set is $\mathcal{A}$ shown in Table 1(a). Table 1(c) shows the covered scenario set $R_{v,M}$ for each v. We find that $S = \{v_2, v_6\}$ can cover all the scenarios in $\mathcal{A}$. Hence, S is a solution of SP.

Note that sensor placement optimization can be also reduced to *integer linear programming* (ILP), which could be solved by the state-of-the-art *mixed integer programming* (MIP) solver such as ILOG's AMPL/CPLEX9.1 MIP solver that was widely used in the previous works [9]. However, in general, these solvers are less efficient and can not scale up to large water distribution networks.

3.3 Incremental Sensor Placement

Now, we are ready to formalize *incremental sensor placement* (ISP). In ISP, beside quality, *cost* is another major concern. Following the scenario-cover based modeling, the quality of a sensor placement can also be measured by $F(S)$. Next, we first give the cost constraint in ISP then give the formal definition of ISP.

Cost Constraint. Let S_0, S be the sensor placement in the original water network and the adjusted sensor placement in the expanded network, respectively. In general, when the network grows larger, we need more sensors. Hence, $|S| \geq |S_0|$. To reduce the cost of S, we may reuse S_0. There are two kinds of reuse:

1. First, the sensor placement on some location may be completely preserved. $S \cap S_0$ contains such sensor locations. No cost will be paid for this kind of reuse.
2. Second, we may move sensors to other locations. For such kind of reuse, we need to pay the uninstall and reinstall cost.

Let $S_1 \subseteq S$ be the set of sensor locations in which sensors are reused in the second case. The relationships among S_0, S_1 and S for the running example is illustrated in Figure 1(c).

Let C_1, C_2 be the cost of installing and uninstalling a sensor, respectively. The cost of replacing an existing sensor to a different location can be approximated by C_1 and C_2. Suppose the overall cost budget C is given. Then, we expect $|S_1|(C_1 + C_2) + (|S| - |S_0|)C_1 \leq C$. In general, C_1, C_2 can be considered as constants in a typical water network. Hence, limiting the cost is equivalent to limiting $|S_1|$ and $|S| - |S_0|$. Thus, we can use two input parameters k_1, k_2 to bound $|S_1|$ and $|S| - |S_0|$, respectively, to precisely model the cost constraint.

Problem Definition. Now, we can define the Incremental Sensor Placement (ISP) problem in Definition 4. We use Example 3 to illustrate ISP.

Definition 4 (Incremental Sensor Placement(ISP)). *Given a water network G with sensor locations L, scenario set $\mathcal{A}$, CS matrix $D_{\mathcal{A},L}$, a credit M and a sensor placement $S_0 \in L$ and two integers k_1, k_2, find a new sensor placement $S \subseteq L$ such that (1) $|S_1| \leq k_1$, $|S| - |S_0| = k_2$, where $S_1 = S_0 - S$ is the set of places at which sensors have been uninstalled from S_0 and replaced on a sensor location outside of S_0; and (2) $F(S)$ is maximized.*

Example 3 (ISP). Continue the previous example shown in Figure 1, where $S_0 = \{v_2, v_6\}$. Suppose $k_1 = 1$,$k_2 = 2$, which means that we could modify the location of at most 1 sensor deployed in S_0 and 2 additional sensors would be deployed on the network. Our goal is to find a sensor placement solution S such that $|S_0 - S| \leq 1$, $|S| - |S_0| = 2$ and $F(S)$ is maximized.

Notice that in our problem Definition 4(ISP), the sensor placement S cannot guarantee that all the contamination scenarios in scenario set $\mathcal{A}$ are covered. In fact, according to previous works [25,16,17], it requires an exponential number of sensors to detect all scenarios in $\mathcal{A}$, which is unaffordable in real world due to the limited budget. Hence, covering scenarios as many as possible under the budget limit is a more realistic objective for real applications.

3.4 NP-Hardness of ISP

In this subsection, we will show that ISP is NP-hard. Our proof consists of two steps. We first show that SP in our model is NP-hard by the reduction from the NP-Complete problem Maximal Coverage Problem (Definition 5) to SP. Then, we prove the NP-hardness of ISP by showing that SP is a special case of ISP.

Definition 5 (Maximal Coverage Problem (MC)). *Given a number k, universe V with n elements and a collection of sets $\mathcal{C} = \{C_1...C_m\}$ such that each $C_i \subseteq V$, find a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| \leq k$ and $|\bigcup_{C_i \in \mathcal{C}'} C_i|$ is maximized.*

Theorem 1 (NP-hardness of SP). *SP is NP-hard.*

Proof. It can be shown that for any instance of MC, i.e. $< V, \mathcal{C} = \{C_1, ..., C_m\}, k >$ we can construct an instance of SP accordingly such that there exists solution for this instance of MC if and only if the corresponding SP instance could be solved. For this purpose, let $V = \mathcal{A}$, $k = p$, and for each C_i we define a sensor location v_i such that $|L| = m$. Then, we just need to show that given $\mathcal{C} = \{C_1, ..., C_m\}$, we can always construct a pair $< M, D_{\mathcal{A},L} >$ such that each $R_{v_i,M} = C_i$. Clearly, it can be easily constructed. For any M (without loss of generality, we may assume it as an integer), we can create matrix $D_{\mathcal{A},L}$ as follows. For each $v_i \in L$ and each scenario $a \in \mathcal{A}$, we set $d_{a,v_i} = \infty$ at first. Then, for each $c \in C_i$, $i = 1, ..., m$, we set d_{c,v_i} to $M - 1$. As a result, we surely have $R_{v_i,M} = C_i$. ∎

Theorem 2 (NP-hardness of ISP). *ISP is NP-hard.*

Proof. Let k be the number of senors to be placed in problem SP. We can simply set $S_0 = \emptyset$, and $k_1 = 0, k_2 = k$ to transform an instance of SP into a problem instance of ISP. Since SP is NP-hard, ISP is also NP-hard. ∎

3.5 Submodularity of Evaluation Function

In this subsection, we will present the major properties of evaluation function $F(\cdot)$, especially the submodularity of this function, which underlies our major solution for ISP.

$F(S)$ has some obvious properties. First, it is *nonnegative*. It is obvious by definition. $F(\emptyset) = 0$, i.e., if we place no sensors, the evaluation function is 0. It is also *nondecreasing*, i.e., for placement placements $A \subseteq B \subseteq L$, it holds that $F(A) \leq F(B)$. The last but also the most important property is *submodularity*. Intuitively, adding a sensor to a large deployment brings less benefits than adding it to a small deployment. The diminishing returns are formalized by the combinatorial concept of *submodularity* [23] (given in Definition 6). In other words, adding sensor s to the smaller set A helps more than adding it to the larger set B. Theorem 3 gives the results.

Definition 6 (Submodularity). *Given a universe set S, a set function F is called* submodular *if for all subsets $A \subseteq B \subseteq S$ and an element $s \in S$, it holds that $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$.*

Theorem 3 (Submodularity of $F(\cdot)$). *Sensor placement evaluation function $F(\cdot)$ given in Definition 2 is submodular.*

Proof. For sensor placement $A \subseteq L$, $F(A \cup \{i\}) = |R_{i,M} \cup (\bigcup_{v \in A} R_{v,M})|$. Then, we have $F(A \cup \{i\}) - F(A) = |R_{i,M} \cup \bigcup_{v \in A} R_{v,M}| - |\bigcup_{v \in A} R_{v,M}| = |R_{i,M} - \bigcup_{v \in A} R_{v,M}|$. Given two sensor placements $A \subseteq B \subseteq L$. We have $\bigcup_{v \in A} R_{v,M} \subseteq \bigcup_{v \in B} R_{v,M}$. Thus, $R_{i,M} - (\bigcup_{v \in A} R_{v,M}) \supseteq R_{i,M} - (\bigcup_{v \in B} R_{v,M})$. Consequently, we have $F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$. ■

Thus, the SP problem on water distribution networks can be cast as a submodular optimization problem, which has been proven to be NP-hard and can be solved with bounded quality [22]. Next, we will present the detail of solutions based on submodularity of the evaluation function.

4 Algorithm Solutions

Let L be the set of locations that can hold sensors and q be the number of sensors to be installed. To solve SP, a brute-force solution needs to enumerate all $C^q_{|L|}$ sensor deployments, where C represents the binomial coefficient. For ISP, we should enumerate $\sum_{0 \leq i \leq k_1} C^i_{|S_0|} C^{i+k_2}_{|V|-|S_0|}$ possible deployments, where S_0 is the original sensor placement. In both cases, the search space is exponential. Hence, it is computationally prohibitive to find the global-optimal deployment for large water distribution network by an exhaustive enumeration approach. To overcome the complexity, we will first present a greedy approach to solve SP. Then, we will further extend it to solve ISP.

4.1 Greedy Algorithms for SP

The basic greedy heuristic algorithm starts from the empty placement $S = \emptyset$ and proceeds iteratively. In each iteration, a new place $v \in V$ which leads to the most increase of F, *i.e.,*

$$\delta_v = F(S \cup \{v\}) - F(S), \tag{3}$$

$v_c = \arg\max_{v \notin S}(\delta_v)$ would be added to S. In other words, at each iteration, we always select the place covering the largest number of uncovered scenarios. A fundamental result in [23] shows that the above greedy procedure will produce a *near optimal* solution for the class of *nondecreasing submodular functions*. More specifically, for any instance of SP, the greedy algorithm always return a sensor placement S such that $F(S) \geq (1 - \frac{1}{e})F(S^*)$, where S^* is the global optimal solution to this instance. Hence, the greedy solutions achieve an approximation ratio at least $1 - \frac{1}{e} \approx 63\%$ compared to the global optimal solution.

We present the greedy procedure in Algorithm 1. After the selection of v_c, we update $R_{v,M}$ by excluding the contamination scenarios that have been covered by v_c (line 6-8). Thus, when the procedure proceeds, the size of $R_{v,M}$ becomes progressively smaller. Note that the size of $R_{v,M}$ for each candidate vertex in the beginning of each iteration is equal to δ_v as defined in Equation 3.

The running time of the algorithm is proportional to the number of sensor locations $|L| = n$ of the water network, the number of sensors to be deployed p, the size of contamination scenarios $|\mathcal{A}| = m$ and the time taken to calculate the size of remaining uncovered scenarios for each $v \in L - S_G$. The update of $R_{v,M}$ needs set union operation, whose complexity is $O(m)$. In each iteration, $O(n)$ vertices need to be evaluated on its quality function. Hence, the total running time is $O(pnm)$.

Algorithm 1. GreedySP

Input: p, M, $R_{v,M}$ for each $v \in L$
Output: S_G
1: $iter \leftarrow 1$
2: $S_G \leftarrow \emptyset$
3: **while** $iter \leq p$ **do**
4: $\quad v_c \leftarrow \arg\max_{v \in L - S_G} |R_{v,M}|$
5: $\quad S_G \leftarrow \{v_c\} \bigcup S_G$
6: $\quad$ **for** each $v \in L - S_G$ **do**
7: $\quad\quad R_{v,M} \leftarrow R_{v,M} - R_{v_c,M}$
8: $\quad$ **end for**
9: $\quad iter \leftarrow iter + 1$
10: **end while**
11: **return** S_G

4.2 Algorithms for ISP

In this section, we will further apply the above greedy heuristic to solve ISP. Compared to SP, effective reuse of the original sensor placement S_0 is one of the distinctive concern of ISP. We use a function $Select$ to decide the subset of the original sensor placement to be reused. And we will discuss different selection strategies used in $Select$ and their effectiveness in detail later. According to our definition of ISP (see Definition 4), there are at least $|S_0| - k_1$ sensors remain unchanged in the new sensor placement.

The greedy algorithm to solve ISP is presented in Algorithm 2, which consists of three major steps. In the first step (line 1), we use $Select$ to choose $|S_0| - k_1$ sensors in S_0 to be preserved (denoted by S_r). In the second step (line 2-5), the algorithm updates $R_{v,M}$ for every sensor location $v \in L - S_r$ at which no sensor is deployed by eliminating the scenarios *covered* by the $|S_0| - k_1$ sensors. In the last step (line 7), we directly call Algorithm 1 to calculate a solution for deploying k_1+k_2 sensors on $L-S_r$. The union of the result of the last step and S_r will be returned as the final answer.

Obviously, the selection strategy used in $Select$ function has a significant impact on the quality of the final sensor placement. We investigate three candidate strategies: *randomized, greedy* and *simulated annealing*. The effectiveness of three strategies will be tested in the experimental section.

Randomized Heuristic (RH). The straightforward strategy is choosing $|S_0|-k_1$ sensors from the original sensor placement S_0 randomly such that each sensor has the same probability to be selected.

Greedy Heuristic (GH). The greedy heuristic is identical to that used in Algorithm 1. Start with the empty placement, $S=\emptyset$ and proceeds $|S_0|-k_1$ times iteratively. In each iteration, select vertex $v \in S_0$ such that $\delta(v)$ is maximal from the remaining vertices in S_0.

Simulated Annealing (SA). First, we use Randomized Heuristic to choose $|S_0|-k_1$ sensors denoted as S_{RH}. Then, we start from S_{RH}, and perform a local search based approach called Simulated Annealing to get a local optimal solution. Simulated Annealing proceeds iteratively. Let S_{cur} be the solution to be optimized in current iteration. The simulated annealing proceeds as follows. At each round, it proposes an exchange of a selected vertex $s \in S_{cur}$ and an unselected vertex $s' \in S_0 - S_{cur}$ randomly, then computes the quality gain function for the exchange of s, s' by

$$\alpha(s,s') = F(S_{cur} \cup \{s'\} - \{s\}) - F(S_{cur}) \tag{4}$$

If $\alpha(s,s')$ is positive, i.e. the exchange operation improves the current solution, the proposal is accepted. Otherwise, the proposal is accepted with probability $\exp(\frac{\alpha(s,s')}{\vartheta_t})$, where ϑ_t is the *annealing temperature* at round t, and $\vartheta_t = Cq^t$ for some large constant C and small constant q $(0 < q < 1)$. We use exponential decay scheme for $annealing$ $temperature$. Such exchanges are repeated until the number of iterations reach to the user-specified upper limit.

Algorithm 2. Greedy Algorithm for ISP

Input: p, M, $R_{v,M}$ for each $v \in V$, S_0, k_1
Output: S_G

$S_r \leftarrow Select(S_0, |S_0|-k_1)$
for each $u \in S_r$ **do**
  for each $v \in L - S_r$ **do**
    $R_{v,M} \leftarrow R_{v,M} - R_{u,M}$
  end for
end for
$S' \leftarrow GreedySP(k_1+k_2, M, \{R_{v,m} | v \in L - S_r\})$
return $S_r \cup S'$

5 Experimental Study

In this section, we present our experimental study results. We show in Sec 5.1 the experiment setup. Then, in Sec 5.2 we justify our scenario-cover based modeling through comparisons of different solutions to SP. Sec 5.3 5.4 5.5 5.6 will study our solution to ISP, including the influence of parameters k_1, k_2, robustness and performance.

5.1 Experiment Setup

We use two real water networks provided by BWSN challenge [10] to test our algorithms. The first one is *BWSN1* containing 129 nodes. The second one is *BWSN2* containing 12,527 nodes. We run EPANET [1] hydraulic simulation and water quality simulation for the two water distribution networks and use the *time until detection* [10] as the criteria to evaluate contamination impact. For network *BWSN1*, 516 contamination scenarios were generated for each of the vertices in the water distribution network at 4 different attack time (*i.e. the start time of contamination* t_s): 6 *A.M.*, 12 *A.M.*, 6 *P.M.*, 12 *P.M.*. The reason why we vary the start time is that during simulation of water network, parameters of resources in the water systems such as junction's water pressure or pipe's flow velocity would change over time, which will influence the propagation of contamination events. Each contamination scenario features 96-hour injection of a fictional contaminant (*i.e. stop time point* t_f *of each scenario is set to be the end time point of contamination simulation*) at strength $1000mg/min$ (using EPANET's 'MASS' injection type). We record the contaminant concentration at intervals of 5 *minutes* and use these time points as the time series.

For network *BWSN2*, we generate 4000 contamination scenarios originating from 1000 randomly selected vertices in the water distribution network with settings identical to that of *BWSN1*. In the following experiments, we fix M as 120 *min* and 150 *min* for *BWSN1* and *BWSN2*, respectively. For the water quality simulation on the two networks, we assume that a deployed sensor would alarm when the concentration of contaminant surpasses $10mg/L$. To compare the effectiveness of our solution, we use *detect ratio*, defined as $\frac{F(S)}{|\mathcal{A}|}$, $i.e$, the proportion of covered contamination scenario, to measure the quality of the sensor placement S, where $|\mathcal{A}|$ denotes the total number of contamination scenarios considered. We run all algorithms on a machine with 2G memory and 2.2GHZ AMD processor. All algorithms are implemented in C++.

5.2 Effectiveness and Efficiency of Solutions to SP

In this subsection, we will show the effectiveness and efficiency of our SP solution. The purpose is to justify the scenario-cover based model. Through the comparisons to other solutions, we will show that by modeling sensor placement optimization in the form of scenario-cover, our solution achieves good solution quality (comparable to the state-of-the-art solution) but consumes significantly less running time. With the growth of real water network, improving the scalability without sacrificing the quality will be a more and more critical concern.

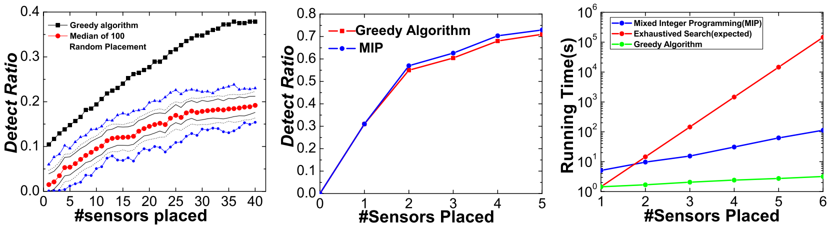
We compare to the following approaches:

1. *Random placement*. In a random placement, we randomly select p sensor locations as a solution. We repeat it for 100 times. For each random placement, we calculate its *detect ratios*. Then, we summarize the the median, minimum, maximum, 10th, 25th, 75th, and 90th-percentiles over 100 random solutions in our experiment.

2. *Exhaustive search*. It's a brute-force solution by enumerating all possible sensor placements. Clearly it gets the optimal result but consumes the most time. Since the time cost is unaffordable, we estimate the entire running time by multiplying the time of enumerating one placement and the number of possible enumerations .
3. *MIP(mixed-integer programming)*. The state-of-the-art approach for SP uses mixed-integer programming modeling. We use LINDO [2], a state-of-the-art MIP solvers, to solve SP.

On large network We first compare random placement to our greedy approach on the large network *BWSN2*. MIP and exhaustive search can not scale to large networks, hence are omitted here. The result is shown in Figure 2(a). We can see that our method can detect significantly more scenarios than random placement. Even the optimal one in 100 random placements is worse than our greedy solution. Hence, in the following test, random placement will not compared.

On small network We also compared different approaches on the small network: *BWSN1*. The result is given in Figure 2(b) and Figure 2(c). From Figure 2(b),we can see that our greedy solution's quality is comparable to that produced by MIP. Their difference is less than 3.7%. However, MIP costs almost one order of magnitude more time than greedy solution. This can be observed from Figure 2(c).



(a) (b) (c)

Fig. 2. Comparison of effectiveness and time. Figure 2(a) compares the results of greedy solution and random placement strategy on *BWSN2*. Figure 2(b) compares solution quality(*detect ratio*) of our greedy algorithm and MIP. Figure 2(c) compares running time of our greedy algorithm, exhaustive search and MIP.

5.3 Influence of k_1

In this experiment, we explore the influence of k_1, i.e., the maximal number of redeployed sensors, on the solution quality of ISP. By this experiment, we justify the motivation of ISP. We show that we only need to modify a relatively small part of the original sensor placement to keep the effectiveness of sensors while significantly reducing the deployment cost.

We first need to simulate the growth of a water network since no real evolving water network data is available. For the simulation, we first find a subregion of the entire water network and consider it as the network at the earlier time. We solve the SP problem on

[2] http://www.lindo.com/

this subregion and obtain a sensor placement S_0 on this subregion. Then, the entire water network can be regarded as the network after growth with S_0 as the original sensor placement.

Due to the expansion of water network, S_0 may fail to detect contamination events. In this experiment, we exclude the influence caused by deploying new sensors through setting k_2 as 0. Note that by setting $k_2 = 0$, no new sensors will be added in our solution. Hence, the *detect ratio* may be quite small for a large network (as indicated in Figure 3(c)). Then, we vary the upper bound of reinstalled sensors k_1 from 0 to $|S_0|$ and observe the evolution of *detect ratio* varying with k_1. For each network, we set $|S_0| = 5$ and $|S_0| = 10$ for *BWSN1* and *BWSN2*, respectively. We define *increase rate* of *detect ratio* as the difference of *detect ratio* between $k_1 = d$ and $k_1 = d + 1$.

The results are shown in Figure 3(a) and 3(c). It is clear that *detect ratio* increases with the growth of k_1, indicating that if we allow more sensors to be redeployed, we can cover more contamination scenarios. However, the *increase rate* gradually decreases when k_1 increases. It is interesting to see that there exist critical points for both two networks ($k_1 = 2$ for BWSN1 and $k_1 = 4$ for BWSN2, respectively), after which the *detect ratio* will increase very slowly.

Considering results given above, we find a good trade-off between *detect ratio* and deployment cost. Since after the critical point, the improvement of *detect ratio* is slower than the increase of cost, generally we can set k_1 at the critical point to trade quality for cost.

Note that k_1 is the upper limit of the actual redeployed sensors. We further summarize the actual number of redeployed sensors in Figure 3(b) and 3(d). We can see that the actual number of reinstalled sensors is always equal to k_1, which implies that the original sensor placement S_0 needs to be redeployed completely to enhance the *detect ratio* on current water network.

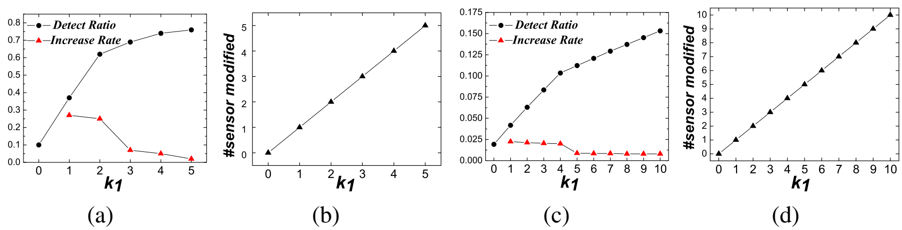
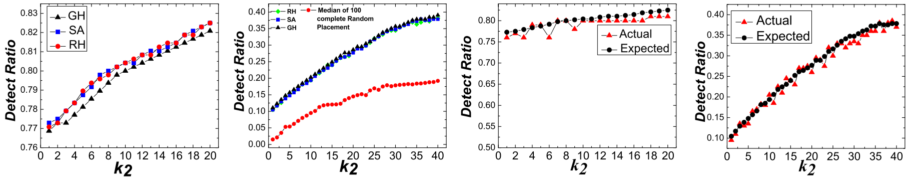


Fig. 3. (a)(c):*detect ratio* for *BWSN1* and *BWSN2* (b)(d): the actual number of redeployed sensors in *BWSN1* and *BWSN2*. The results show that solution quality generally increases with the growth of k_1 and there exists some critical point at which we can seek for a good tradeoff between solution quality and deployment cost.

5.4 Influence of k_2 and the Selection Strategies

In this experiment, we explore the influence of k_2 and compare the effectiveness of three strategies used in $Select$ function to solve ISP. We set k_1 as 2 and 5 for *BWSN1* and *BWSN2*, respectively. Other parameters are the same as the previous experiment. We set iteration number to be 1000 for the simulated annealing strategy.

From Figure 4(a), i.e., the result on *BWSN1*, we can see that simulated annealing and random heuristic strategy show only minor priority in *detect ratio* over the greedy heuristic. However, on *BWSN2* (shown in Figure 4(b)), the performance of the three strategies are quite close to each other. Such results imply that our solution is generally independent on the selection strategy. For comparison, we also present median of *detect ratio* of 100 complete random placements in Figure 4(b).



(a) *BWSN1* (b) *BWSN2* (c) *BWSN1* (d) *BWSN2*

Fig. 4. Figure 4(a),4(b) show the influence of k_2 and compare different selection strategies used in $Select$ function. It shows that solution quality increases with the growth of k_2, and three strategies shows similar effectiveness. Figure 4(c),4(d) present the robustness of proposed solutions. *Expected* represents the *detect ratio* on the training scenario set; *Actual* represents the average of $detect\ ratio$ on test scenario sets. These figures show that our sensor placement solution is *robust* against newly introduced contamination scenarios.

5.5 Robustness of Solutions

In general, there may exist potentially infinite number of possible contamination scenarios. A placement strategy with high *detect ratio* on the training contamination scenario set may be ineffective to detect contamination scenarios not belonging in the training set. An ideal solution is expected to be *robust* against the newly introduced contamination scenarios. In this section, we will show the robustness of our solution.

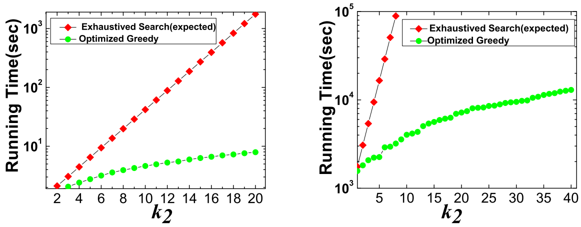
In our experiments, we set $|S_0| = 5, 10$, $k_1 = 2, 5$ for *BWSN1* and *BWSN2*, respectively. We use simulated annealing with 1000 iterations for the $Select$ function. We randomly generate four scenario sets for *BWSN1* and *BWSN2* as test sets, respectively. Each test set contains 100 scenarios. We choose one set as $\mathcal{A}$ and get solution S using Algorithm 2 with simulated annealing heuristic. Then, we compare *detect ratio* of S on the training set $\mathcal{A}$ and the average *detect ratio* of S on the other three sets.

The result is shown in Figure 4(c) and 4(d), where we vary the number of new sensors (k_2) and observe the evolution of $detect\ ratio$. It can be observed that when k_2 increases, $detect\ ratio$ of test case on average is quite close to that of the training sets, implying that our sensor placement which is effective on training scenario set $\mathcal{A}$ works well on other scenario sets as well. Hence, our solution to ISP is *robust* against unknown contamination scenarios.

5.6 Performance of Our Solution for ISP

In this experiment, we test the performance of our solutions for ISP. MIP can not be easily extended on ISP problem. Random placement is certainly the fastest, but as shown in Sec 5.2, is of quite low quality. Hence, in this experiment, we only compare to exhaustive searching whose time is estimated as stated in Sec 5.2. We fix $|S_0| = 5$ and $|S_0| = 10$ for *BWSN1* and *BWSN2*, respectively.

The result is shown in Figure 5, from which we can see that our solution is faster than exhaustive search, and the speedup is almost two or three orders of magnitudes. Our algorithm generally linearly increases with the growth of k_2. Even on the large network with ten thousands of nodes, our solution finds a solution within 2-3 hours. The performance result implies that our approach can scale up to large water networks.



(a) *BWSN1* (b) *BWSN2*

Fig. 5. Running time comparison of solutions to ISP.

6 Conclusion

In this paper, we propose a new problem: incremental sensor placement optimization problem (ISP), in which we need to find an optimal sensor placement for the dynamic-evolving water network with the following two objectives: (1) keeping the deployment cost limited and (2) maximizing the effectiveness of the new sensor placement. We show this problem is NP-hard. We prove that the objective function used in the definition of ISP is submodular. Based on this property, we propose several greedy algorithms to solve this problem. Experimental results verify the effectiveness, robustness and efficiency of proposed solutions. We will further consider more realistic constraints on our problem to solve more specific real sensor optimization problems.

References

1. Lee, B.H., Deininger, R.: Optimal locations of monitoring stations inwater distribution system. J. Environ. Eng. 118(1), 4–16 (1992)
2. Kumar, A., et al.: Identification of monitoring stations in water distribution system. J. Envir. Engin. 123(8), 746–752 (1997)
3. Al-Zahrani, M.A., Moied, K.: Locating optimum water quality monitoring stations in water distribution system. In: Proc. World Water and Environmental Resources Congress (2001)
4. Kessler, A., et al.: Detecting accidental contaminations in municipal water networks. J. Water Resour. Plann. Manage 124(4), 192–198 (1998)
5. Ostfeld, A., Kessler, A.: Protecting urban water distribution systems against accidental hazards intrusions. In: IWA Second Conference (2001)

6. Berry, J.W., Fleischer, L., et al.: Sensor placement in municipal water networks. J. Water Resour. Plann. Manage 131(3), 237–243 (2005)
7. Jean-paul, W., et al.: A multiple-objective analysis of sensor placement optimization in water networks. In: Critical Transitions In Water And Environmental Resources Management, pp. 1–10 (2006)
8. Ostfeld, A., Salomons, E.: Optimal layout of early warning detection stations for water distribution systems security. J. Water Resour. Plann. Manage 130(5), 377–385 (2004)
9. Berry, J., Hart, W.E., et al.: Sensor placement in municipal water networks with temporal integer programming models. J. Water Resour. Plann. Manage 132(4), 218–224 (2006)
10. Ostfeld, A., et al.: Battle of water sensor networks: A design challenge for engineers and algorithms. J. Water Resour. Plann. Manage 134(6), 556 (2008)
11. Ostfeld, A., Salomons, E.: Sensor network design proposal for the battle of the water sensor networks (bwsn). J. Water Resour. Plann. Manage 134(6), 556–568 (2008)
12. ZhengYi, W., Tom, W.: Multi objective optimization of sensor placement in water distribution systems. In: WDSA 2006, pp. 1–11 (2006)
13. Dorini, G., et al.: An efficient algorithm for sensor placement in water distribution systems. In: WDSA 2006, pp. 1–13 (2006)
14. Guan, J., et al.: Optimization model and algorithms for design of water sensor placement in water distribution systems. In: WDSA 2006, pp. 1–16 (2006)
15. Berry, J., et al.: A facility location approach to sensor placement optimization. In: WDSA 2006, pp. 1–4 (2006)
16. Huang, J.J., et al.: Multiobjective optimization for monitoring sensor placement in water distribution systems. In: WDSA 2006, pp. 1–14 (2006)
17. Preis, A., Ostfeld, A.: Multiobjective sensor design for water distribution systems security. In: WDSA 2006, pp. 1–17 (2006)
18. Ghimire, S.R., Barkdoll, B.D.: Heuristic method for the battle of the water network sensors: Demand-based approach. In: WDSA 2006, pp. 1–10 (2006)
19. Gueli, R.: Predator-prey model for discrete sensor placement. In: WDSA 2006, pp. 1–9 (2006)
20. Ghimire, S.R., Barkdol, B.D.: A heuristic method for water quality sensor location in a municipal water distribution system:mass related based approach. In: WDSA 2006, pp. 1–11 (2006)
21. Eliades, D., Polycarpou, M.: Iterative deepening of pareto solutions in water sensor networks. In: WDSA 2006, pp. 1–19 (2006)
22. Leskovec, J., et al.: Cost-effective outbreak detection in networks. In: KDD 2007, pp. 420–429 (2007)
23. Nemhauser, G.L., et al.: An analysis of approximations for maximizing submodular set functions. Mathematical Programming 14, 265–294 (1978)
24. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Operations Research Letters 32, 41–43 (2004)
25. Krause, A., et al.: Efficient sensor placement optimization for securing large water distribution networks. J. Water Resour. Plann. Manage (2008)
26. Goyal, A., et al.: Learning influence probabilities in social networks. In: Goyal, A., et al. (eds.) WSDM 2010, pp. 241–250. ACM (2010)
27. Chen, W., et al.: Efficient influence maximization in social networks. In: KDD 2009, pp. 199–208. ACM (2009)
28. Gomez Rodriguez, M., et al.: Inferring networks of diffusion and influence. In: KDD 2010, pp. 1019–1028. ACM (2010)