Optimal placement of imperfect water quality sensors in water distribution networks

de Winter, Casper; Palleti, Venkata Reddy; Worm, Daniel; Kooij, Robert

DOI
10.1016/j.compchemeng.2018.10.021

Publication date
2019

Document Version
Final published version

Published in
Computers and Chemical Engineering

Citation (APA)
de Winter, C., Palleti, V. R., Worm, D., & Kooij, R. (2019). Optimal placement of imperfect water quality sensors in water distribution networks. *Computers and Chemical Engineering*, *121*, 200-211. https://doi.org/10.1016/j.compchemeng.2018.10.021

Computers and Chemical Engineering 121 (2019) 200–211

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Optimal placement of imperfect water quality sensors in water distribution networks

Casper de Winter[a,b], Venkata Reddy Palleti[b,*], Daniel Worm[c], Robert Kooij[b,d]

[a] Erasmus School of Economics, Erasmus University Rotterdam, The Netherlands
[b] iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design, Singapore
[c] Cyber Security and Robustness, TNO, The Hague, The Netherlands
[d] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Technology Delft, The Netherlands

ARTICLE INFO

Article history:
Received 7 March 2018
Revised 17 October 2018
Accepted 25 October 2018
Available online 26 October 2018

Keywords:
Water distribution networks
Sensor placements
Contaminant detection
Imperfect sensors
Greedy algorithm

ABSTRACT

Water Distribution Networks (WDNs) are often susceptible to either accidental or deliberate contamination which can lead to poisoned water, many fatalities and large economic consequences. In order to protect against these intrusions or attacks, an efficient sensor network with a limited number of sensors should be placed in a WDN. In this paper, we focus on optimal sensor placements by introducing two greedy-based algorithms in which the imperfection of sensors and multiple objectives can be taken into account. The algorithms were tested using a medium scale urban WDN. It is shown that our algorithms are able to find sensor placements in reasonable time and that its solutions are close to optimal. Furthermore, relaxing the often used assumption that sensors work perfectly results in different sensor placements than were found before, indicating the importance to take sensor imperfection into account when placing sensors.

1. Introduction

Water Distribution Networks (WDNs) form a crucial part in our life by providing clean, safe drinking water to billions of people around the world. A WDN supplies fresh water from water sources to households, companies e.g. using a large hydraulic system. This system or network consists of many elements such as reservoirs, tanks, treatment facilities, pumps, pipes, and valves. These networks are diverse and can be very large, consisting of hundreds or thousands of kilometers of underground pipes. An increasing number of people make use of water from such a system every day and rely on the safety and the quality of water in their lives or work. If a problem arises within the WDN, the impact on society can be enormous.

There are several threats to a WDN which can be divided into physical and chemical disruptions. Physical disruptions, such as leaking pipelines, failing pumps or intentional attacks on the network itself, will have a big economical impact but are not considered a serious risk to human beings. The biggest threat to a population comes from intentional or accidental chemical contamination within the water network. In order to protect the public from such intrusions it is necessary to incorporate an early warning system with a sensor network to monitor the quality of the drinking water effectively and efficiently in a WDN. Together with EPA (the United States Environmental Protection Agency), Murray et al. (2009) estimated that a contamination warning system could save half of the expected fatalities and over 19 billion dollars of associated economic impact on a water network of a large municipality.

Accidental contamination of a WDN may occur in many ways including breaking of pipelines. However, a much bigger and more lethal threat to a society happens with the intentional poisoning of drinking water by criminals or terrorists. In every volume of the report The World's Water Gleick and Heberger (2014), all known conflicts and threats which involve water resources or water systems are enumerated. These upcoming threats show the relevance of a good warning system. A sensor system should be able to quickly detect intrusions in the WDN and therefore reduce sickness, fatalities and the associated economic consequences. Due to cost and maintenance reasons, it is of course not possible to place sensors at every place in the network. Hence, a small number of sensors need to be placed efficiently to achieve an effective monitoring. Several objectives and different algorithms have been considered to achieve an optimal or sub-optimal placement of sensors. In previous works, it has mostly been assumed that the sensors detect every contamination (100% reliability). This is most likely an unrealistic assumption and with only a limited number of sensors

[*] Corresponding author.
E-mail address: venkata_palleti@sutd.edu.sg (V.R. Palleti).

https://doi.org/10.1016/j.compchemeng.2018.10.021
0098-1354/© 2018 Elsevier Ltd. All rights reserved.

to be placed in the WDN, the unreliability of one sensor could have severe consequences.

In this work, we introduce new heuristic algorithms for solving the sensor placement problem when sensors are not 100% reliable, making use of a maximum covering model with weighted edges. Further, we quantify the effect of the reliability of imperfect sensors on the optimal sensor placement and explore the effect of multiple objectives on the sensor placement. A basis of this research is a recent paper by Palleti et al. (2016) in which they design a perfect-sensor network using a greedy heuristic based on the set covering problem (SCP) to be able to detect the contamination and identify the source of the contamination. While maximizing the probability of detection has always been one of the main objectives when designing a sensor network, the objective of identifying the source of the attack is quite new in literature. When the point of intrusion is known, it is possible to take action instantly and to get some parts of the water network back to operation sooner. Besides these objectives of maximizing detection and identification, we introduce objectives to also include objectives to minimize the time to detection and the impact of a contamination as well.

The organization of the paper is as follows. In Section 2 an overview of the related work on sensor placement is given, and the novelty of our work compared to the literature is described. In Section 3, the main assumptions and the mathematical formulation of the sensor placement problem are described, as well as the greedy algorithms proposed to solve this problem. Section 4 contains the main results of applying the developed algorithms on a case study focusing on the Bangalore WDN. Finally, Section 5 describes the main conclusions and suggestions for future research.

2. Related work

2.1. Perfect sensor placement

Several researchers have addressed the sensor placement problem in WDNs considering different objectives assuming perfect sensors. It was first mentioned by Lee et al. (1991), and Lee and Deininger (1992) where they maximized the demand coverage by sensors. Kessler et al. (1998) considered level of service as an objective for sensor location in WDNs. Propato (2006) developed a mixed-integer linear program to identify optimal sensor locations for early warning against accidental and intentional contaminations in drinking water distribution systems. The general model can be applied to unsteady hydraulic conditions. Later, Shastri and Diwekar (2006) presented a two stage stochastic programming approach for sensor placement in WDNs by incorporating nodal demand uncertainties in the objective function. Also, Rico-Ramirez et al. (2007) proposed a two stage mixed integer program which minimizes the expected population at risk and the cost of sensors. Mukherjee et al. (2017) presented a new approach to solve sensor placement problem in WDNS by incorporating uncertainties in nodal demands and attack locations.

One of the main research works on sensor placements is the Battle of the Water Sensor Networks (BWSN) (Ostfeld et al., 2008) which was a multi-objective network design competition. In this competition, fifteen independent teams have participated to design the sensor network. They considered the following objectives in their formulations: minimize the detection time, the population affected and the amount of contaminated water consumed and maximize the detection likelihood with a limited number of sensors. In this competition the best four solution methods based on the number of non-dominated[1] solutions are obtained (Berry et al., 2006; Dorini et al., 2006; Krause et al., 2006; Wu and Walski, 2006). A major limitation of the BWSN formulation is the way non-detected events were handled. Events which could not be detected were ignored which could result in very promising results on impact reduction with a very small chance of detection.

One of the four solution methods was provided by the research team of Berry et al. (2006), which performed a lot of research on sensor placement in WDNs. They designed a mixed-integer programming (MIP) formulation which was very similar to the p-median facility location problem. In that problem, p facilities should be placed and each customer should be assigned to one facility in order to minimize the distance between the facility and the customer. In the formulation of Berry et al. (2006), each contamination scenario should be detected by one ’witness’ using some number of sensors in order to minimize the impact over all contamination scenarios. The ’witness’ is defined such that it is the first sensor in the network to detect the contamination or it is a dummy location, which means it is a non-detection. Several optimal methods and heuristics are introduced by the research team. This MIP formulation and heuristic solution methods formed the basis of the most used sensor placement toolkit in practice, the TEVA-SPOT Toolkit (Hart et al., 2008).

Further, Laird et al. (2006, 2005) and Perelman and Ostfeld (2013) investigated the problem of contamination source identification based on the sensor deployment in the network. However, their approaches are useful only if the amount of contaminant introduced into the WDN is known. Also, it is also difficult to obtain the unique solution because of the limited number of sensor measurements available. Recently Palleti et al. (2016) used a new and different approach to the sensor placement problem which satisfy observability and identifiability conditions. They were interested in the objectives of detecting the attack and identifying the source of the attack. Observability refers to the ability of the sensor network to detect the contamination where as identifiability refers to the ability of the sensor network to identify the exact location of the intrusion.

2.2. Imperfect sensor placement

The number of papers that take imperfection of sensors into account for the sensor placement problem in WDNs, is quite limited. For instance, a very recent survey paper, see Hu et al. (2018), only mentions one paper, namely Comboul and Ghanem (2013), that considers sensor imperfection, from a list of 18 papers. Likewise, the survey paper Rathi and Gupta (2014), only mentions 3 papers, Berry et al. (2009), Xu et al. (2010), and again Comboul and Ghanem (2013), out of a list of 23 papers that deal with imperfect sensors. In this subsection we will briefly discuss these papers and explain how our work differs from it.

As the data analysis inside a sensor works with some range or threshold, there is a fair chance not to detect every intrusion. For example when the contamination concentration has become too low at that point in the network. Ostfeld and Salomons (2004) and Weickgenannt et al. (2010) incorporated these thresholds in their formulation, but the sensors still work perfectly above some threshold. Other researchers just only considered large enough contamination events to overcome this problem.

In the previous section, the adjusted p-median facility location problem introduced by Berry et al. (2006) was explained. In later work of Berry et al. (2008), they extended their work by allowing the sensors to not detect every contamination. They assign false negative probabilities to each sensor location and change the impact of an intrusion into an expected impact. An intrusion at some point is now with some probability first detected by sensor 1, with some probability by sensor 2, and so on and finally with some probability not detected at all. In contrast to the standard

[1] A solution is non-dominated if none of the objective functions can be improved in value without degrading some of the other objective values.

BWSN formulation, they did add penalty costs to non-detections. In Berry et al. (2008), the formulation is extended for multiple sensor types and thus multiple different probabilities of detecting false negatives across the network. In Berry et al. (2009), experiment with a number of solution methods to deal with imperfect sensors and conclude that it is worth using optimization methods that are aware of the sensor imperfection as more robust solutions are found. Krause et al. (2008) describe their contribution to the BWSN of Krause et al. (2006) along with several extensions. They show ways to handle multi-objective optimization by scaling all objectives and also how to handle sensor failures. Imperfect sensors can be implemented in the framework they described by using a random binary variable associated per location which indicates if a sensors works or not. In that way, the objective function is changed with an extra expectation over all possible failure scenarios as the average impact changes per different failure scenario. In practice, this implementation only worked with a low failure probability and at most one sensor failure per scenario, according to the paper, as the number of different failure scenarios can increase rapidly. Results of this implementation have not been given.

In a research by Xu et al. (2010), a two-stage model is proposed to tackle the problem that sensors in a WDN may provide false positive and false negative signals. They combine a facility location model with Bayesian networks such that the probability that a contamination goes undetected and the false alarm rate are minimized. Comboul and Ghanem (2013) take sensors imperfection into account but imperfection is quanitifed by looking at one sensor i.e, the one that can detect the contamination at the earliest. Also, they assume that detection probability is function of the contaminant concentartion. Preis and Ostfeld (2008) and Shen et al. (2014) coped with the unreliability by using detection redundancy as an objective. This means that most contaminations should be detected by more than one sensor which makes sure that when a sensor fails, other sensors can still detect the contamination.

To summarize, limited works exist in the literature on the imperfect sensor placement. The novel contributions of our work are as follows.

- A method for determining optimal placement of imperfect sensors, based upon a maximum covering model with weighted edges. So far, in literature, the maximum covering model has only been applied for perfect sensors. Note that Xu et al. (2010) considered imperfect sensors, but for the *placement* of sensors, perfection of sensors was assumed.
- Optimization of a weighted combination of four objectives. Existing papers dealing with imperfect sensors only optimize one objective. For instance, in Berry et al. (2009) either “mass consumed” or “population exposed” are optimized.
- Incorporation of identification probability objective. Source identification, one of the objectives of our research, is used to make sure that the problem can be quickly found and fixed. By identifying the source, it is possible to close some valves or pipelines to make sure that the damage to other parts of the network is minimized and that other parts of the water network can get back to operation sooner. Therefore, this objective can also be seen as an important practical objective. This includes a method for giving an upper bound for this probability. Berry et al. (2009) and Comboul and Ghanem (2013) do not consider identification probabilities. Xu et al. (2010) does, but only implicitly through the use of Bayesian networks. We derived an explicit expression for the identification probability.

3. Design for placement of imperfect sensors

In this section, we list the most important assumptions used in this paper in Section 3.1. Then we will mathematically define the sensor placement problem and the objective formulations in Sections 3.2 and 3.3. Afterwards, several solution methods for this problem will be presented in Sections 3.4, 3.5 and 3.6.

3.1. Main assumptions

- A typical WDN consists of reservoirs, tanks, valves, pumping stations, pipes and fire hydrants etc. As most of the network is burried underground, only a few components of WDN are present above the ground such as reservoirs, tanks, valves and pumps. We assume that the above ground components are more easily accessible for the attackers and considered them as potential target for the contamination (Palleti et al., 2016). These components are termed as vulnerable nodes. Therefore, it is assumed that only a subset of nodes are considered as the potential target for an attack. An attack can happen on each vulnerable node with equal probability. As long as the contamination is not detected, contamination will be added to the network at the intrusion point.
- The contamination will travel through the network with the same speed as the water does.
- It is assumed that each sensor in the network will have the same given probability p of detecting the contamination: $0 \leq p \leq 1$. Besides that, the detection of an attack by a certain sensor is independent of the detection of the attack by other sensors.
- The sensor imperfection is not necessary related to chemical properties but can also reflect the impact of cyber attacks. For instance, in 2015, the U.S. States ICS-CERT (Industrial Control Systems Cyber Emergency Response Team) received and responded to 295 incidents (ICS-CERT, 2016). The Water Sector account for 8.5% of these incidents. Therefore it is assumed that a sensor might be imperfect, namely, that as a result of cyber attacks, the sensor readings might no longer be reliable.
- The detection of an attack by a certain sensor is independent of the detection of the attack by other sensors.
- False positives will not be considered. The effect of a false positive reading by a sensor is negligible with only some economic impact compared to the consequences of a real contamination which is not detected.
- A sensor which is able to detect the contamination will immediately raise an alarm. At this time of detection, it is assumed that several actions will be taken directly such that no more contaminated water will be consumed.

3.2. Sensor placement formulation

We consider a WDN in which water flows from certain sources or tanks to the customers. This network can be represented as a graph $G = (V, E)$, where the nodes V represent sources, demand points, and junctions in the network. Pumping stations, treatment plants, valves and fire hydrants are also illustrated as nodes. E, the set of edges, represents all the pipes between the different nodes. A specific demand pattern introduces flows on the network. The directions of these flows can be modelled by making the graph directed. In this study, EPANET 2 (Rossman, 2000) software is used to simulate the WDN to obtain the corresponding flow directions. A WDN typically consists of hundreds to tens of thousands nodes and pipes. A small subset of the set of vertices is considered to be vulnerable or accessible for an attack. In our research, the same classification for a vulnerable node is used as

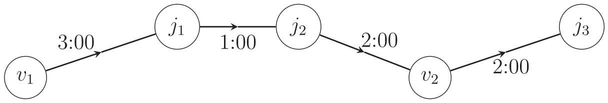

Fig. 1. The Ex-1 network with two vulnerable nodes and three demand nodes.

was done by Palleti et al. (2016), i.e. the vulnerable nodes are water reservoirs and tanks from which water can flow, pumping stations, treatment plants and valves. The set of m vulnerable nodes is denoted by V'.

Sensors can be placed on all nodes j in the set $\{V\backslash V'\}$. When a directed path exists from vulnerable node v to node j, a sensor at node j is able to detect a contamination on node v. The probability that the sensor actually detects the contamination is denoted with p. If p equals 1, the sensor is considered to be perfect. A sensor placement X is a subset of the set of possible sensor locations. The number of sensors to be placed in the network is denoted by B. So, from the set $\{V\backslash V'\}$, a subset X of size B should be picked. This should be done such that the objective formulation used in our research is maximized. The objective function f does not only depend on the sensor placement but also on the probability p that a sensor detects the contamination. Many different objectives can be taken into account. The general sensor placement formulation can be written as follows.

$$\max \quad f(X|p) \tag{1}$$

$$s.t. \quad X \subseteq \{V \setminus V'\} \tag{2}$$

$$|X| = B \tag{3}$$

3.3. Objective formulations

In this paper, we consider four different objectives: the network probability of detection (D), the network probability of identification (F), the average time to detection (T) and the estimated impact of an attack (Z). The first two need to be maximized, the last two minimized. Each objective is scaled between 0 and 1 and weights w are used, depending on which objective is thought to be more important. We will elaborate how values for the objectives can be derived when sensor uncertainty plays a role. For simplicity, all weights are put to $\frac{1}{4}$. In general, the objective formulation of Eq. (1) can be written as follows.

$$\max f(X|p) = \max w_D D + w_F F + w_T(1-T) + w_Z(1-Z) \tag{4}$$

One example will be used in the next sections to illustrate the whole methodology. We consider the simple water distribution network of Fig. 1, which has two vulnerable nodes, v_1 and v_2, and three demand nodes, j_1, j_2 and j_3. This network will be called the Ex-1 Network.

Water in this network flows from node v_1 to node j_3 via nodes j_1, j_2 and v_2. Data concerning the flow time in hours in this network can be seen in Fig. 1. It takes in total 8 h for the water to travel from v_1 to j_3. The demand at each of the three demand nodes is equal to one unit per hour. In this example, sensors can be placed on nodes j_1, j_2 and j_3 and we assume the probability of detection p of each sensor to be 0.8. Now, let us elaborate all objectives using this example.

- *Detection likelihood* (D):

Given a sensor network where each sensor has a probability p of detecting a contamination, it is possible to calculate the detection likelihood of the whole sensor network. For each vulnerable node v, S_v denotes which sensor locations are located downstream of v in the directed graph. The set cardinality of S_v, denoted by n_v, is the number of sensors downstream of v. The probability of detection for each vulnerable node can now be defined as one minus the chance of not detecting the contamination with all n_v sensors. Finally, D is defined as the average of the detection likelihood of all m vulnerable nodes:

$$D = \frac{1}{m}\sum_{v\in V'}\left(1-(1-p)^{n_v}\right) \tag{5}$$

In the case of perfect sensors, i.e. $p = 1$, the sum term within D for a given vulnerable v equals 1 if there is a sensor downstream of v ($n_v > 0$) and 0 if $n_v = 0$. If in the Ex-1 network of Fig. 1 a sensor is placed on nodes j_1 and j_3, there are two sensors which can detect a contamination on node v_1 and only one sensor which can detect an attack on node v_2. If $p = 0.8$, then $D = (0.8 + 0.96)/2 = 0.88$.

- *Identification probability* (F):

Palleti et al. (2016) showed that identifying the source of an attack is fairly straightforward with perfect sensors. It is only necessary to make sure that the set of sensors triggered by an attack on a vulnerable node is unique for each vulnerable node. For example, the sensor network $\{j_1, j_3\}$ in the Ex-1 Network leaves two unique sets for each of the two vulnerable nodes.

With imperfect sensors, this no longer holds. Consider again the placement of sensors on nodes j_1 and j_3. When sensors are known to be imperfect and only j_3 raises an alarm, we are not 100% certain what the source of the contamination is. It could be v_2, but it is also possible that v_1 is the source, if j_1 failed to detect the contamination.

To deal with this uncertainty, we assume node v is contaminated and then consider all possibilities with respect to detection, for each sensor node in S_v. Therefore, from the set S_v, every subset of detecting sensors $c \in \mathcal{P}(S_v)$ is considered, where $\mathcal{P}(S_v)$ is the power set of set S_v excluding the empty set. If we denote the number of sensors in subset c by n_c, then the probability that combination c is precisely the set of sensors alarming when node v is contaminated, is given by

$$P(c|v,p) = p^{n_c}(1-p)^{n_v-n_c} \tag{6}$$

The only combinations c that contribute to the identification probability, are those that do not occur as a combination of detecting sensors for any other vulnerable node. For instance, again considering the Ex-1 Network, assume that $S_{v_1} = \{j_1, j_2, j_3\}$. Then the combinations $\{j_1\}$, $\{j_2\}$ and $\{j_1, j_2\}$ will be able to identify with certainty v_1 as the source of contamination.

For a node v and combination c, define the indicator function $I(v, c)$ as

$$I(v,c) = \begin{cases} 1, & c \notin \mathcal{P}(S_u) \forall u \in V' \setminus v, \\ 0, & otherwise. \end{cases} \tag{7}$$

Using this indicator function, it is possible to define the identification probability F of a sensor network as follows.

$$F = \frac{1}{m}\sum_{v\in V'}\sum_{c\in\mathcal{P}(S_v)} I(v,c)p^{n_c}(1-p)^{n_v-n_c}. \tag{8}$$

- *Detection time* (T):

For the other two objectives, detection time and impact, a method is used which is similar to the one used in Berry et al. (2008). The intuition behind these methods is that a contamination will first pass the first sensor which can detect the contamination with the lowest detection time, and if this sensor does not detect it, the contamination travels further through the network to the next sensor. For the detection time, define the time it takes to detect an attack on node v with a sensor on node j as $q^t_{v,j}$. This time can be computed using the flow speeds in and the volume of the pipes. We can then construct an sorted list L^t_v in which these detection times for each vulnerable node v are sorted such that $L^t_v(i)$ is the ith fastest sensor to detect the contamination.

If none of the downstream sensors are able to detect the contamination due to failure, which happens with probability $(1-p)^{n_v}$, a predetermined time $q^t_{v,\infty}$ is defined in which the contamination is detected in another way, for example by observing an outbreak of sickness. In previous research, 48 h is mostly considered for this time Krause et al. (2008). The choose to scale the final detection time to a value between 0 and 1 and not in between 0 and 48. All in all, after derivations given in Appendix, the detection time T of the sensor network is given by:

$$T = \frac{1}{m}\sum_{v\in V'}\frac{1}{q^t_{v,\infty}}\left(\sum_{i=1}^{n_v}\left(p(1-p)^{i-1}q^t_{v,L^t_v(i)}\right) + (1-p)^{n_v}q^t_{v,\infty}\right) \tag{9}$$

- *Contamination Impact* (Z):

The method for the contamination impact is very similar to the method for the detection time, since the estimated volume of contaminated water consumed grows over time. The same ordering as in list L^t_v can thus be used. $q^z_{v,j}$ is defined as the estimated volume of water consumed when a contamination on node v reaches the jth sensor in L^t_v. $q^z_{v,\infty}$ is the estimated volume of water consumed at maximum time $q^t_{v,\infty}$.

As each vulnerable node v can affect a different part of the network with different impacts, the corresponding $q^t_{v,\infty}$'s may vary. For example, after 48 h and a demand of 1 unit per hour in the Ex-1 Network, $q^z_{v_1,\infty} = 129$ and $q^z_{v_2,\infty} = 46$. This should be taken into account by weighing the different impacts of each vulnerable node. By doing this, we place more importance on the parts of the network in which the most water flows. Finally, Z is defined as follows, analogous to the derivation of T in Appendix.

$$Z = \sum_{v\in V'}\frac{1}{\sum_{u\in V'} q^z_{u,\infty}}\left(\sum_{i=1}^{n_v}\left(p(1-p)^{i-1}q^z_{v,L^t_v(i)}\right) + (1-p)^{n_v}q^z_{v,\infty}\right) \tag{10}$$

The formulas for the 4 separate objectives have been derived under the assumption that the detection for every sensor node is the same, namely p. It is easy to extend this to the case that every sensor node has a different detection probability. For instance, if we denote the detection probability of the i^{th} sensor node, located downstream of the vulnerable node v, by $p_{i,v}$, then Eq. (5) becomes

$$D = \frac{1}{m}\sum_{v\in V'}\left(1 - \prod_{i=1}^{n_v}(1-p_{i,v})\right). \tag{11}$$

Similar expressions can be derived for the other three objectives. However, for the remainder of the paper we assume that every sensor has the same detection probability p.

3.4. Greedy algorithm for optimal sensor placement

We are dealing with a multi-objective and non-linear problem. It is a conjecture in literature that the water quality sensor placement problem is in general NP-hard (Krause et al., 2008; Xu et al., 2013). Therefore it is assumed that it is not possible to find the optimal solution for practical problems within reasonable time, which forces the use of heuristics.

In our heuristic approach, the problem is reformulated into a weighted set covering formulation. A static water distribution network with vulnerable nodes can be converted to a bipartite graph as was shown by Palleti et al. (2016). In the bipartite graph, the set of vulnerable nodes is on one side and all other nodes (possible sensor nodes) on the other side. Each vulnerable node v can affect a subset S_v of the set of nodes $\{V\backslash V'\}$ and these arcs are present in the bipartite graph. The bipartite graph representation of the example Ex-1 Network is shown in Fig. 2.

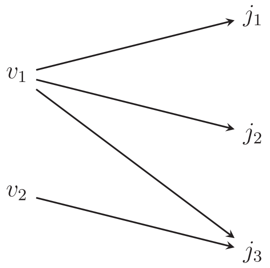

Fig. 2. The Ex-1 network converted to a weighted bipartite graph.

In contrast to the work of Palleti et al. (2016), values are placed on each arc (v, j) for each objective - partial objective values. They describe what the placement of a sensor on node j adds to objectives D, F, $(1-T)$ or $(1-Z)$ for a contamination on vulnerable node v. The contribution of the placement of a certain sensor at node j on the whole network can be calculated by summing all the partial objective values of the arcs entering that node j. However, after one sensor location has been chosen, the objective values on the other arcs should be updated as they are not valid anymore. This method can therefore not be used to place multiple sensors within one step.

3.4.1. Computing partial objective values

It will now be shown how the partial objective values per arc should be calculated and updated for each objective. A partial objective value is denoted by $\Gamma^{\triangle}_{(v,j)}$ where (v, j) is the arc and $\triangle$ the objective for which the value applies. $\Gamma^{\triangle}_{(v,j)}$ is defined as the difference between the part of the objective value for that vulnerable node v with sensors at $S_v \cup j$ minus the one with sensors only at S_v.

For the detection probability D, a specific vulnerable node v and an arc (v, j) in the bipartite graph representation, the derivation of the partial objective value on (v, j) is shown in Eqs. (12)–(14), using the fact that n_v changes to $n_v + 1$.

$$\Gamma^D_{(v,j)} = \frac{1}{m}\left((1-(1-p)^{n_v+1}) - (1-(1-p)^{n_v})\right), \tag{12}$$

$$= \frac{1}{m}((1-p)^{n_v} - (1-p)^{n_v+1}), \tag{13}$$

$$= \frac{1}{m}(p(1-p)^{n_v}). \tag{14}$$

The partial objective values for the time to detection and the impact of the contamination also depend on the place of the possible sensor j in the ordered list L^t_v. For this, we define k as the number of already placed sensors that need to change position in the ordered list when sensor j is added. Therefore, $k = 0$ when the sensor location j is the last sensor to detect and $k = n_v$ when it is the first sensor. After some derivations, see Appendix, it can be shown that the partial objective values for $(1-T)$ and $(1-Z)$ are given by Eqs. (25) and (26):

$$\Gamma^T_{(v,j)}(k) = \frac{1}{mq^t_{v,\infty}} p(1-p)^{n_v-k}\Bigg[(1-p)^k q^t_{v,\infty} + \sum_{h=1}^{k}\left(p(1-p)^{h-1}q^t_{v,L^t_v(n_v-k+h)}\right) - q^t_{v,j}\Bigg], \tag{15}$$

Table 1
The initial partial objective values for the Ex-1 network.

	Γ^D	Γ^F	Γ^T	Γ^Z	Average
(v_1, j_1)	0.4	0.4	0.375	0.590	0.441
(v_1, j_2)	0.4	0.4	0.367	0.585	0.438
(v_1, j_3)	0.4	0.0	0.333	0.549	0.428
(v_2, j_3)	0.4	0.0	0.383	0.210	0.331

$$\Gamma^Z_{(v,j)}(k) = \frac{1}{\sum_{u\in V'} q^z_{u,\infty}} p(1-p)^{n_v-k}\Bigg[(1-p)^k q^z_{v,\infty} + \sum_{h=1}^{k}\left(p(1-p)^{h-1} q^z_{v,L^t_v(n_v-k+h)}\right) - q^z_{v,j}\Bigg]. \tag{16}$$

Using these partial objective values we do not need to calculate every objective value from scratch after only adding one sensor. This improves the required runtime of every algorithm that aims to optimize the objective function by adding one sensor at a time. If the added sensor is only connected to some vulnerable nodes, only the objective values on the arcs leaving those influenced vulnerable nodes need to be changed. For source identification, no simple formula of the partial objective value on an arc could be obtained. Therefore, Eq. (8) will be needed to compute this value.

3.4.2. The greedy algorithm

A greedy algorithm will be used to place sensors in the network based on the weighted set covering formulation. In the greedy algorithm, sensors are placed one at a time such that it is possible to update the partial objective values after placing a sensor. At each iteration of the algorithm, we choose to pick the best possible sensor location given the objective values at that time and place a new sensor at that location.

Given a static demand pattern for a WDN, hydraulic network simulations can be performed to construct the directed graph and obtain values needed to compute time and impact. Using the directed graph, a bipartite graph can be constructed and the initial partial objective values for each arc can be calculated, using $n_v = k = 0$. The best sensor candidate j can now be chosen. This is the sensor location j with the largest value when summing all partial objective values $\Gamma^\triangle_{(v,j)}$ of the incoming arcs on that node j, multiplied with the corresponding weight $w_\triangle$ of each objective $\triangle$. After one sensor is placed, the objective value and all necessary partial objective values are updated. The partial objective values entering node j are put to zero. Next, it is again investigated which sensor location is the best to pick and this will continue until B sensors have been placed. The final sensor placement X consists of all the chosen sensor locations.

In the Ex-1 Network, it can be shown which sensor location should be chosen first using the given objective values formulas. All initial partial objective values and the average of the four objectives for each arc (i.e. we assume every objective has the same weight $w = 1/4$) can be found in Table 1.

Even though location j_3 does not contribute to objective F, as it is able to detect both possible contamination attacks, it is the best possible sensor location in the first step. This is true as 0.428 + 0.331 is larger than 0.441 or 0.438. After a sensor is placed on location j_3, the partial objective values for arcs (v_1, j_1) and (v_1, j_2) need to be recalculated.

3.5. Local search

Greedy algorithms are in general very fast and may result in good solutions. However, the optimal solution is not always found. When a sensor location is chosen as the Bth sensor in the previous step, all updated sensor placements with $|X| > B$ contain that sensor location.

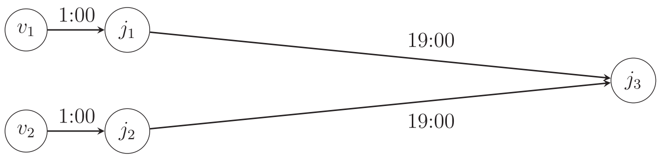

Fig. 3. A Network with two vulnerable nodes and three demand nodes. The flow time of each arc is shown at each arc.

A small example network in which this is a problem is shown in Fig. 3.

The demand is again one unit per hour, p is again 0.8 and the flow times are shown in the figure. In the first step of the greedy algorithm, sensor location j_3 will be chosen as it can detect both possible contaminations. The greedy solution of two sensors must contain location j_3 whereas the optimal sensor placement of two sensors consists of j_1 and j_2.

A simple way to try and improve the greedy solutions is to use a local search algorithm in order to improve the solution found by the greedy algorithm. This local search method is defined as follows. Consider a sensor placement X. In the local search step, each neighboring solution is investigated to check whether it has a better objective value than the current sensor placement. If so, the best-found neighboring solution will be the new placement X and we perform another local search step until a local optimum is found. Neighboring solutions are defined by replacing one sensor node in X with one of the set $\{\{V \backslash V'\} \backslash X\}$. Note that Berry et al. (2009) also suggested a local search method. However, his local search start from the optimal placement found under the assumption of perfect sensors while our starting position comes from running the Greedy algorithm first, already taking sensor imperfection into account.

3.6. Theoretical upper bound for the objective

To quantify the performance of the heuristics, it is useful to compare the found objective values, with a theoretical upper bound. It is easy to see that the following bounds hold: $D \leq 1$, $T \geq 0$ and $Z \geq 0$. In fact, if we assume that p is sufficiently close to 1, or if the number of sensors is sufficiently large (i.e. all $n_v's$ are sufficiently large), then the values of D, T and Z, will be very close to their bounds, i.e. 1, 0 and 0, respectively. The situation for the identification probability F is different. Recall that V' denotes the set of vulnerable nodes, with $|V'| = m$. Next, define the subset W of V', consisting of vulnerable nodes, that are not downstream of any other vulnerable nodes in V'. Then, only the vulnerable nodes in W can be identified as the source of contamination, with certainty. Therefore, F, will have an upper bound $\frac{|W|}{m}$. Denoting the objective by O, and assuming all weights in Eq. (4) are equal, we obtain the following upper bound for the objective.

$$O \leq \frac{3m + |W|}{4m}. \tag{17}$$

We will use Equation (17) to assess the performance of our heuristics for a specific use case, in the next section.

4. Case study

We will use an EPANET benchmark instance of a WDN, corresponding to a reduced representation of the WDN of Bangalore, to demonstrate the methodology introduced in the previous section. In Section 4.1, the Bangalore Network is introduced. This is a medium-sized network with a relatively high number of vulnerable

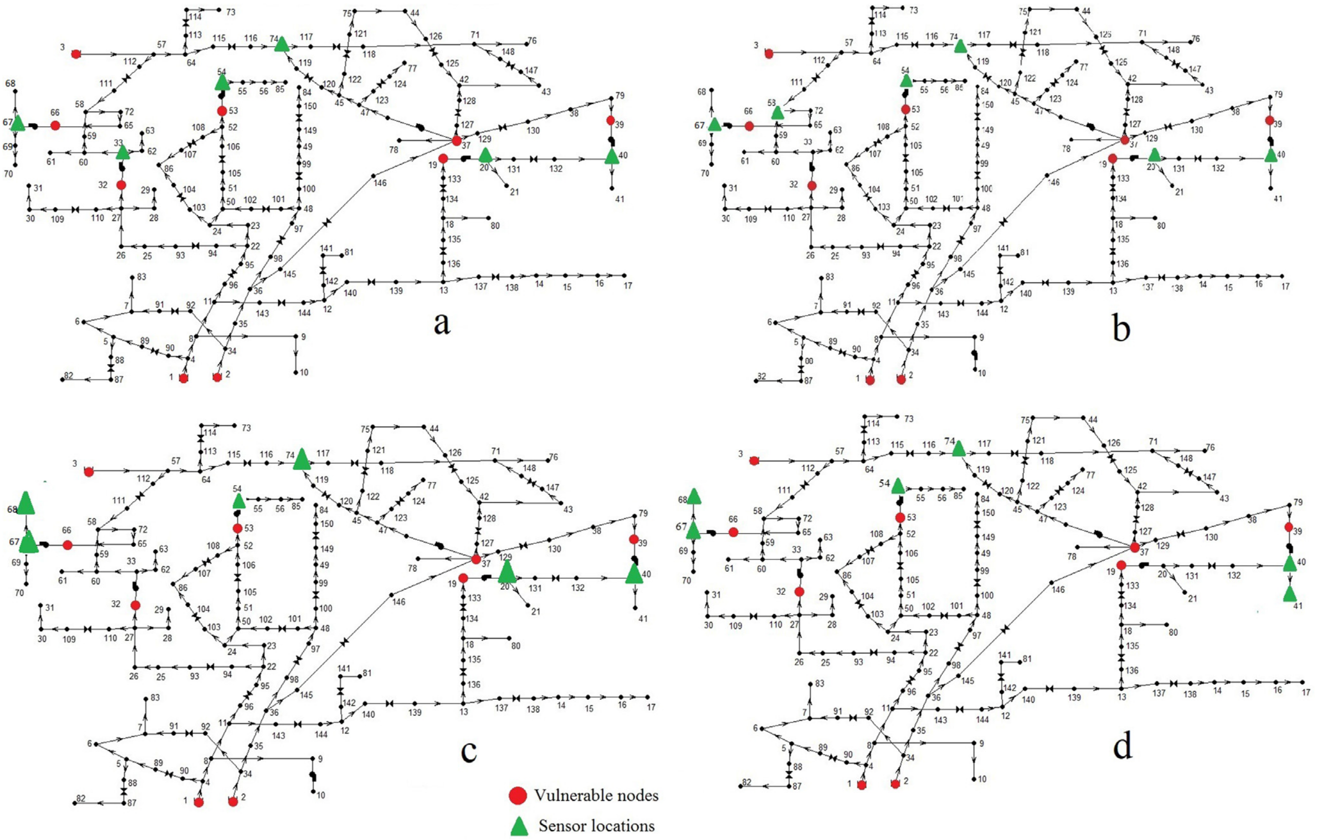

Fig. 4. The Bangalore Network with V' in red and the sensor placement in green. (a), (b), (c), (d) represent sensor locations for X_P, X_{G1}, X_{G2} and X_{G3} respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

nodes. More information on the Bangalore Network can be found in Datta (1992).

4.1. Bangalore network

The Bangalore Network, as seen in Fig. 4, is a reduced representation of the water distribution network of Bangalore, a city in India. The network has three sources, which are represented in the network by the reservoir nodes 1, 2 and 3. A clarification of all symbols used in the representation of the WDN can be found in Palleti et al. (2016). This network contains 150 nodes in total, 116 normal pipes, 32 valves and six pumps. Pumps and valves are represented in the networks with an edge between two nodes with the specific symbol on the edge. For this reason, when pumps or valves are considered as a vulnerable node, the node before the pump or valve is labeled as vulnerable. For example, consider the pump between node 66 and 67 on the far left of the network. As water flows from node 66 to node 67, node 66 is considered as a vulnerable node and nodes 67, 68, 69 and 70 will be affected by an attack on this node. Even though the network contains several different demand conditions, the consequences for the flow directions are minimal.

This network was also used by Palleti et al. (2016) in their research focusing on detection and identification with perfect sensors. The main advantage of this network is the large number of reservoirs from which water flows and possible vulnerable nodes, such that the sets of nodes affected by an attack can be very different. Palleti et al. (2016) used nine vulnerable nodes in their research: the three reservoirs and the 6 pumps. The large number of 32 valves were omitted for simplicity of presenting the results as their target was 100% detection and identification, which would otherwise require around 30 sensors. In our research, both 9 and 41 vulnerable nodes will be used. The case with 9 vulnerable nodes will be referred to as the standard Bangalore Network. The case with 41 vulnerable nodes will mainly be used to see how well our methods and objectives can handle larger numbers of vulnerable nodes.

4.2. Perfect-Sensor placement vs imperfect-sensor placement

In Section 4.2.1, a perfect-sensor network for the Bangalore Network will be presented. After that, in Section 4.2.2, we will show how the best-found sensor placement may change, as the probability of detection p is lowered. In these sections, equal weights for each objective will be considered.

4.2.1. Perfect-sensor placement

In our research, the same static loading condition and flow patterns are used as in Palleti et al. (2016). The nodes before the pumps and the three reservoirs were considered to be vulnerable nodes. The resulting nine vulnerable nodes were nodes 1, 2, 3, 19, 32, 37, 39, 53 and 66. As a consequence, there are 141 nodes left where sensors could be placed.

They found out that six sensors are necessary in the Bangalore Network to detect and identify all possible attacks on the nine vulnerable nodes. The sensor network they present is the sensor set {20, 33, 40, 54, 67, 71}, which indeed gives value 1 to objectives D and F. However, using the presented greedy algorithm with $p = 1$ and adding objectives T and Z, a slightly different sensor placement is found. The sensor on location 71 is moved to location 74 as this location is reached somewhat earlier in the network. The small differences for the objectives can be seen in Table 2.

The sensor placement {20, 33, 40, 54, 67, 74} will be used in the next sections as the perfect-sensor placement for comparison and

Table 2
Comparison of the solution found by Palleti et al. (2016) and the solution found by our greedy algorithm.

Sensor placement	*D*	*F*	*T*	*Z*	Average
Palleti et al. (2016): {20, 33, 40, 54, 67, 71}	1.000	1.000	0.970	0.981	0.988
Greedy algorithm: {20, 33, 40, 54, 67, 74}	1.000	1.000	0.980	0.995	0.994

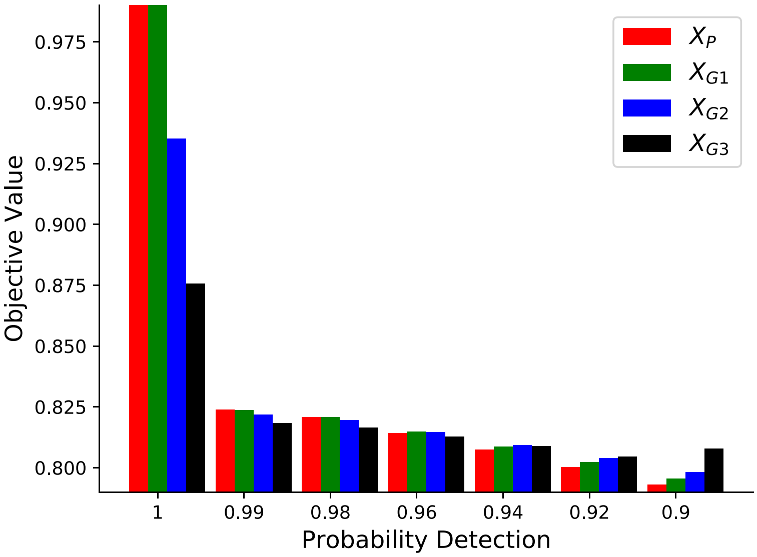

Fig. 5. Comparison of objective values with different detection probabilities.

Table 3
Different sensor placements and the range of *p* for which that sensor placement is optimal according to the greedy algorithm.

Sensor placement	Range of *p* for which the placement is optimal
X_P: {20, 33, 40, 54, 67, 74}	p = 1.000 to 0.981
X_{G1}: {20, 40, 54, 58, 67, 74}	p = 0.980 to 0.955
X_{G2}: {20, 40, 54, 67, 68, 74}	p = 0.954 to 0.932
X_{G3}: {40, 41, 54, 67, 68, 74}	p = 0.931 to 0.900

to see how this placement changes when *p* decreases. This placement will be referred to as X_P. In Fig. 4, the Bangalore Network is represented in which you could see the nine vulnerable nodes in red and the six sensor locations from X_P in green. The flow directions are also represented in this network by arrows, such that the paths from the vulnerable nodes to the sensors are visible.

4.2.2. Imperfect-sensor placement

We will now gradually decrease the detection probability *p*, using a step size of 0.001, starting at $p = 1$, until we reach $p = 0.9$. For each value of *p*, the greedy algorithm is executed to find the optimal sensor placement. This leads to sensor placements which are optimal for a certain range of *p*, see Table 3.

The perfect-sensor placement is only the best-found sensor placement for detection probabilities larger than 0.98. At that point, changing one sensor location results in a better overall objective value. Table 4 shows how the optimal objective values change, as a function of decreasing values of *p*. The optimal objective values for a given *p* are shown bold-faced. Fig. 5 shows the comparison of objective values of different sensor placement and detection probabilities.

When *p* decreases, it can be seen that the objective values for sensor placement X_{G3} decrease slower than the ones for X_P. The reason behind this can clearly be seen when considering the sensor network X_{G3} (see Fig. 4). At some places in the network, two sensors are placed next to each other to compensate the imperfection of the other sensor. This occurs with location pairs 40 & 41 and 67 & 68 in X_{G3}. We have shown that for the Bangalore Network the transition value for the detection probability is $p = 0.98$, assuming nine nodes are vulnerable and six sensors are placed in the network. In addition we have also run experiments to study to what extent this transition probability depends on the number of placed sensors, the number of vulnerable nodes and the network topology. The results are reported in de Winter (2018). For example, we found that for the Bangalore Network with 41 vulnerable nodes, even for $p = 0.999$ with nine sensors deployed in the network, the sensor placement differs from the perfect-sensor placement at three places.

From the results reported in this subsection, we can clearly see the added value of our proposed algorithm, as the assumption that the sensors are perfect in general do not lead to optimal sensor placements.

4.3. Impact of uncertainty in the demand

In the previous subsection we have evaluated our model under static load conditions. In this subsection we will quantify the impact of uncertainty of demand on the imperfect sensor placement.

The starting point of the analysis is the same as before, i.e. the Bangalore network with static loading conditions, as in Palleti et al. (2016). We model uncertainty in demand by varying the original demand W_i at demand node *i*, by a factor *P*. We assume the demand is uniformly distributed on the interval $[(1-P)W_i, (1+P)W_i]$. For the uncertainty in demand, we have considered the following values for *P*: {5%, 10%, 15%, 20%, 25%}. The cases with uncertainty in demand are bench marked against the case with static loading. We denote the sensor placement obtained for this benchmark case by X_{orig}. So for instance, for the case $p = 0.9$, according to Table 3, $X_{orig} = \{40, 41, 54, 67, 68, 74\}$.

For a given value of *P*, we change the demand of every node, by multiplying it with a random number between $1-P$ and $1+P$.

Table 4
Objective values for the four sensor placement for different values of p.

Sensor placement	$p = 1$	0.99	0.98	0.96	0.94	0.92	0.90
X_P	**0.9937**	**0.8239**	0.8207	0.8141	0.8073	0.8003	0.7930
X_{G1}	0.9932	0.8236	**0.8207**	**0.8148**	0.8086	0.8022	0.7956
X_{G2}	0.9353	0.8218	0.8195	0.8145	**0.8093**	0.8039	0.7981
X_{G3}	0.8755	0.8182	0.8165	0.8127	0.8088	**0.8046**	**0.8001**

Table 5
Fraction of experiments where uncertainty in demand gives same result as static loading.

p	$P = 5\%$	10%	15%	20%	25%
0.9	1	0.940	0.878	0.754	0.682
0.95	1	0.915	0.726	0.679	0.574

Table 6
Ratio of objective values for X_{orig} and newly found optimal placement.

p	$P = 5\%$	10%	15%	20%	25%
0.9	1	0.99855	0.99724	0.99561	0.99455
0.95	1	0.99903	0.99753	0.99454	0.99236

Table 7
Maximun number of different nodes in X_{orig} and newly found optimal placement.

p	$P = 5\%$	10%	15%	20%	25%
0.9	0	1	1	1	2
0.95	0	1	1	2	2

For the obtained WDN we again run EPANET, and subsequently our algorithm, to determine the best sensor placement. As a measure of performance we look at the number of nodes in the new sensor placement, that where not in X_{orig}. We also look at the ratio between the overall objective of X_{orig} and that for the new found placement.

This experiment has been repeated 500 times. For the imperfection of the sensors, two values were considered, namely $p = 0.9$ and $p = 0.95$. The results of the experiment are report in Tables 5, 6 and 7.

We conclude from Table 5 that inclusion of demand uncertainty has an impact on sensor placement. Still, even for uncertainty up to 25%, for more than 50% of the time, the sensor placement found through the static demand assumption is the optimal one.

Table 6 shows that even though X_{orig} no longer is the optimal sensor placement, the difference in objective values with the optimal placements are relatively small.

Finally, Table 7 shows that for the considered scenarios, the maximum number of new nodes in the optimal sensor placements, is at most two.

4.4. Performance of the greedy algorithm

In this section, the performance of our greedy algorithm will be compared with its variant which in addition implements a local search, introduced in Section 3.5. The comparison will be done using the Bangalore Network with nine vulnerable nodes. A total of 100 different scenarios are considered, varying the number of placed sensors B (between 5 and 14) and the detection probability p (between 0.9 and 0.99, in steps of 0.01). Again, it is assumed that the weights for each objective is equal. Table 8 reports the average objective value, the average run time and the number of times the solution of the algorithm was equal to the best-found solution by all heuristics. All algorithms were implemented in Python 2.7 and were run on a computer with an Intel Core i5-5300U processor with 2.30 GHz CPU and 8 GB of RAM.

The results for the cases corresponding to the placement of 5–9 sensors and that of 10–14 sensors, are split in the table.

By definition, the Greedy + LS heuristic performs at least as good as the greedy algorithm. We see that in 59 out of the 100 cases the basic greedy algorithm finds solutions which are equally good as solutions found by Greedy + LS. The more sensors need to be placed, the less likely it is that Greedy is able to find the best-known solution. However, the differences are very small. For the 50 cases with 10–14 sensors, the difference between the average objective value of Greedy and Greedy + LS is 0.00034 while the maximum deviation in one case from the best-known objective value is 0.00144. This largest deviation occurs with the case $p = 0.96$ and $B = 12$ for which the best-found objective value was 0.83038. This means that the score of the solution of Greedy is 99.83% if the best-found solution is considered to be 100%.

While this is only a fairly small difference, the greedy algorithm clearly outperforms the other heuristic with respect to run times. The run time of Greedy is on average 1 s over all 100 cases in this small network with a maximum run time of 4 s. The run times of Greedy + LS increases much faster. The average run time for this heuristic for the 50 largest cases is over a minute.

Finally we also look at the performance of the heuristics, by comparing the outcomes with a brute force method that finds the optimal solution and a theoretical upper bound. For the former, we looked at a reduced Bangalore Network, where only 36 out of the original possible sensor locations are considered. Then, for the scenario of 6 placed sensors and detection probability $p = 0.95$, we find that both Greedy and Greedy + LS give the same objective value 0.81195 as the brute force method. The run time for this case was 0.061 s for Greedy, while the brute force method took 704 s. A second way to quantify the performance of the heuristics is to compare them with the theoretical upper bound in Eq. (17). It is

Table 8
Comparison of the two heuristics on the Bangalore Network with nine vulnerable nodes.

Bangalore	5–9 sensors, 10 different p's			10–14 sensors, 10 different p's		
$m = 9$	Avg. obj.	#best-found[a]	Avg. time	Avg. obj.	#best-found	Avg. time
Greedy	0.81509	46	0.251 s	0.82853	13	1.905 s
Greedy + LS	0.81511	50	3.226 s	0.82887	50	64.970 s

[a] Number of solutions found with the heuristic, which are not outperformed by the other heuristic.

Table 9
Comparison of greedy with theoretical upper bound; 15 sensors are placed and $p = 0.95$.

Instance	Greedy: 15 sensors, $p = 0.95$	Upper-bound	Optimality gap
Bangalore, $m = 9$	0.83110	0.83333	0.27%
Bangalore, $m = 41$	0.74586	0.76829	2.982%

easy to verify that for the Bangalore network with $m = 9$, three vulnerable nodes are not downstream of other vulnerable nodes. Therefore $|W| = 3$. As a result, according to Eq. (17), the objective value is upper bounded by 0.833. Also for the Bangalore network with 41 vulnerable nodes we find $|W| = 3$, leading to the upper bound 0.76829. In Table 9, these upper bounds are compared to the solution value found with the greedy algorithm for 15 sensors and $p = 0.95$. Using these results, the optimality gap, defined as the relative distance between the found objective value and the upper bound, can be calculated.

We conclude from Table 9 that the sensor placements found through Greedy are close to the optimal placement.

5. Conclusions and future work

Worldwide, there is an upcoming threat of water pollution and terrorist attacks on water distribution networks. Accidental or deliberate incidents will affect the quality of the drinking water and can cause many fatalities and a huge economic impact. To prevent these consequences, WDNs should use a sensor network to monitor the quality of the drinking water. The main problem we consider is how to place sensors in a WDN in an optimal way.

In previous sensor placement studies, it was mostly assumed that sensors are perfect. However, sensors can fail to detect a contamination due to errors, failures, maintenance difficulties, degradation, drifting or hacking. In this paper it was found that large differences in the optimal sensor locations may occur when considering slightly imperfect sensors compared to perfect sensors. We have shown that the imperfectness of contamination sensors influences the optimal placement of these sensors within the network. In particular, we conclude that it is essential to take the imperfectness of the sensors into account when designing a sensor placement.

In order to design a sensor placement, the problem has been converted to a changing weighted set covering formulation with a bipartite graph. A greedy algorithm and a variant deploying local search have been introduced to solve this problem. They can be used to obtain a close to optimal sensor placement, taking into account the failure probability of the sensors as well as four different objectives: minimizing the time to detection and the impact of a contamination and maximizing the probability of detection and the probability of identifying the source of the attack. The algorithms have been tested using the Bangalore network.

The greedy method is very fast, and close to optimal in small networks. Local search improvements only result in minor improvements to the sensor placement, while increasing the computation time drastically.

The developed method can be easily extended to take dynamic demand patterns into account, i.e. demand patterns that vary over time. For each demand pattern a corresponding flow pattern over the WDN can be derived. The contribution of this flow pattern to the objective values for a given sensor placement can be computed. Proper weighing of these contributions for different demand patterns allows to compute the objective values for a dynamic demand pattern over a day, which can be used as input to derive good sensor placements. In addition, while the formulas described in this paper assume equal failure probability p over all sensors, they can easily be adapted to take different failure probabilities into account.

In order to apply the introduced methods in practice, it is important to be able to estimate the failure probability p of a sensor, since p has a large influence on the best sensor placement. KPIs given by the producer of the sensor may help to estimate p; however, independent measurements will also be needed to fine tune such estimations - especially to see how the failure probability depends on the age of the sensor.

Acknowledgement

This work was supported in part by the National Research Foundation (NRF), Prime Ministerâs Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2015NCR-NCR003-001) and administered by the National Cybersecurity R&D Directorate.

Appendix

Derivation of Objective Values for T and Z. In Section 3.3 it was shown that the objective value T can be formulated as can be seen in Eq. (9), which is shown again below:

$$T = \frac{1}{m}\sum_{v\in V'}\frac{1}{q^t_{v,\infty}}\left(\sum_{i=1}^{n_v}\left(p(1-p)^{i-1}q^t_{v,L^t_v(i)}\right)+(1-p)^{n_v}q^t_{v,\infty}\right) \tag{(9) revisited}$$

This will now be clarified using the previously defined definitions of p, $q^t_{v,j}$ and L^t_v. First, consider only one vulnerable node v. Given a chance p that a sensor detects a contamination, the first sensor $L^t_v(1)$ contributes $pq^t_{v,L^t_v(1)}$ to the average detection time of vulnerable node v, the second sensor $p(1-p)q^t_{v,L^t_v(2)}$, etcetera. With probability $(1-p)^{n_v}$, no sensor will give an alarm which results in a detection time defined as $q^t_{v,\infty}$. In total, summing all these terms this results in the part between brackets in Eq. (9). To make sure the objective value is between 0 and 1, we have to divide this value by the maximum detection time $q^t_{v,\infty}$. When no sensors are placed after vulnerable node v, T for that v should give 1 instead of $q^t_{v,\infty}$. The normalized detection time of the whole sensor network is then the average over all m vulnerable nodes in V'. The derivation of Z is similar.

Derivation of Partial Objective Values for T and Z. The derivation of the partial objective values for T and Z also need some further clarification. In Section 3.4.1, it was only stated that the objective value for an arc also depends on the position of the new sensor j in the ordered list L^t_v. The order in L^t_v determines how much each sensor contributes to the objective function as the first sensor is the first to raise an alarm with probability p. When the detection time of the added sensor is less than $L^t_v(1)$, this sensor will be the first sensor, along with its contribution p, and all other sensors contribute less to the objective than before. When the added sensor is placed at the end of the network, only the contribution of the non-detection time of $q^t_{v,\infty}$ will be less than before and the sensor itself will also contribute a smaller fraction to the total objective.

Define the normalized detection of one vulnerable node as T_v:

$$T_v = \frac{1}{q^t_{v,\infty}}\left(\sum_{i=1}^{n_v} p(1-p)^{i-1}q^t_{v,L^t_v(i)}+(1-p)^{n_v}q^t_{v,\infty}\right) \tag{18}$$

First, we will look at the easiest example, a sensor on node j placed further away than the already placed n_v sensors such that $q^t_{v,L^t_v(n_v)} < q^t_{v,j} < q^t_{v,\infty}$. Only the last term of T_v (which is $(1-p)^{n_v}q^t_{v,\infty}$) will be split and changes. The sensor with location j will be the sensor to detect the contamination when all n_v sensors fail and this added sensor works. This happens with probability $p(1-p)^{n_v}$. With probability $(1-p)(1-p)^{n_v}$, there is a non-detection. The new formula for T_v, denoted as T^*_v, can be seen in Eq. (19).

$$T^*_v = \frac{1}{q^t_{v,\infty}}\left(\sum_{i=1}^{n_v} p(1-p)^{i-1}q^t_{v,L^t_v(i)} + p(1-p)^{n_v}q^t_{v,j}\right.$$
$$\left. +(1-p)(1-p)^{n_v}q^t_{v,\infty}\right) \qquad (19)$$

It should also be noted that the contribution of vulnerable node v to the total objective T is T_v divided by m and that our objective is to maximize $(1-T)$. The difference between $\frac{T^*_v}{m}$ and $\frac{T_v}{m}$ is equal to the decrease of objective T via vulnerable node v if sensor j is added to the end of the ordered list. So, $\frac{T_v}{m} - \frac{T^*_v}{m}$ is equal to the objective value of arc (v, j) as we consider $(1-T)$ as an objective. This objective value is denoted by $\Gamma^T_{(v,j)}(k)$ in which k is used to show the position of the possible new sensor j in the ordered list. The variable k represents the number of already placed sensors which need to change a position in the ordered list when sensor j is added. In this example, k is equal to zero as the added sensor is added to the end. $\Gamma^T_{(v,j)}(0)$ is calculated in Eqs. (20), (21) and (22).

$$\Gamma^T_{(v,j)}(k=0) = \frac{T_v}{m} - \frac{T^*_v}{m} = \frac{1}{mq^t_{v,\infty}}\big((1-p)^{n_v}q^t_{v,\infty} - (p(1-p)^{n_v}q^t_{v,j}$$
$$+(1-p)(1-p)^{n_v}q^t_{v,\infty})\big) \qquad (20)$$

$$= \frac{1}{mq^t_{v,\infty}}\left(p(1-p)^{n_v}q^t_{v,\infty} - p(1-p)^{n_v}q^t_{v,j}\right) \qquad (21)$$

$$= \frac{1}{mq^t_{v,\infty}}p(1-p)^{n_v}(q^t_{v,\infty} - q^t_{v,j}) \qquad (22)$$

In the same way, $\Gamma^T_{(v,j)}(1)$ and $\Gamma^T_{(v,j)}(2)$ can be calculated.

$$\Gamma^T_{(v,j)}(1) = \frac{1}{mq^t_{v,\infty}}p(1-p)^{n_v-1}((1-p)q^t_{v,\infty} + pq_{v,L^t_v(n_v)} - q^t_{v,j}) \qquad (23)$$

$$\Gamma^T_{(v,j)}(2) = \frac{1}{mq^t_{v,\infty}}p(1-p)^{n_v-2}\Big((1-p)^2q^t_{v,\infty} + p(1-p)q^t_{v,L^t_v(n_v)}$$
$$+pq^t_{v,L^t_v(n_v-1)} - q^t_{v,j}\Big) \qquad (24)$$

A pattern becomes visible in these formulas. For larger values of k, more original sensor contributions change and more terms need to be added. When $k = n_v$, all sensor contributions change. In general, we can define the partial objective value for arc (v, j) and position variable k as follows.

$$\Gamma^T_{(v,j)}(k) = \frac{1}{mq^t_{v,\infty}}p(1-p)^{n_v-k}\Bigg[(1-p)^k q^t_{v,\infty}$$
$$+\sum_{h=1}^{k}\left(p(1-p)^{h-1}q^t_{v,L^t_v(n_v-k+h)}\right) - q^t_{v,j}\Bigg] \qquad (25)$$

In the same way, the partial objective value of objective $(1-Z)$ can be derived. The main difference is that we should account for the weighing of the impact of an attack on each vulnerable node. The formula for $\Gamma^Z_{(v,j)}(k)$ can be seen in Eq. (26).

$$\Gamma^Z_{(v,j)}(k) = \frac{1}{\sum_{u\in V'} q^z_{u,\infty}}p(1-p)^{n_v-k}\Bigg[(1-p)^k q^z_{v,\infty}$$
$$+\sum_{h=1}^{k}\left(p(1-p)^{h-1}q^z_{v,L^t_v(n_v-k+h)}\right) - q^z_{v,j}\Bigg] \qquad (26)$$

References

Berry, J., Carr, R.D., Hart, W.E., Leung, V.J., Phillips, C.A., Watson, J.-P., 2008. On the placement of imperfect sensors in municipal water networks. In: Water Distribution Systems Analysis Symposium 2006, pp. 1–13.

Berry, J., Carr, R.D., Hart, W.E., Leung, V.J., Phillips, C.A., Watson, J.-P., 2009. Designing contamination warning systems for municipal water networks using imperfect sensors. J. Water Resour. Plann. Manage. 135 (4), 253–263.

Berry, J., Hart, W.E., Phillips, C.A., Uber, J.G., Watson, J.-P., 2006. Sensor placement in municipal water networks with temporal integer programming models. J. Water Resour. Plann. Manage. 132 (4), 218–224.

Comboul, M., Ghanem, R., 2013. Value of information in the design of resilient water distribution sensor networks. J. Water Resour. Plann. Manage. 139 (4), 449–455.

Datta, R., 1992. General and Sensitivity Analysis of Water Distribution Networks. Ph. D. dissertation, Indian Institute of Science, Bangalore, India Ph.D. thesis.

Dorini, G., Jonkergouw, P., Kapelan, Z., Di Pierro, F., Khu, S., Savic, D., 2006. An efficient algorithm for sensor placement in water distribution systems. In: Water Distribution Systems Analysis Symposium 2006, pp. 1–13.

Gleick, P.H., Heberger, M., 2014. Water conflict chronology. In: The World Water. Springer, pp. 173–219.

Hart, W.E., Berry, J.W., Boman, E.G., Murray, R., Phillips, C.A., Riesen, L.A., Watson, J.-P., 2008. The teva-spot toolkit for drinking water contaminant warning system design. In: World Environmental and Water Resources Congress 2008, pp. 1–12.

Hu, C., Li, M., Zeng, D., Guo, S., 2018. A survey on sensor placement for contamination detection in water distribution systems. Wireless Netw. 24 (2), 647–661.

ICS-CERT, 2016. NCCIC/ICS-CERT Year in Review: FY 2015. U.S. Department of Homeland Security Industrial Control Systems-Cyber Emergency Response Team, Washington, D.C (2016).

Kessler, A., Ostfeld, A., Sinai, G., 1998. Detecting accidental contaminations in municipal water networks. J. Water Resour. Plann. Manage. 124 (4), 192–198.

Krause, A., Leskovec, J., Guestrin, C., Van Briesen, J., Faloutsos, C., 2008. Efficient sensor placement optimization for securing large water distribution networks. J. Water Resour. Plann. Manage. 134 (6), 516–526.

Krause, A., Leskovec, J., Isovitsch, S., Xu, J., Guestrin, C., Van Briesen, J., Small, M., Fischbeck, P., 2006. Optimizing sensor placements in water distribution systems using submodular function maximization. In: Water Distribution Systems Analysis Symposium 2006, pp. 1–17.

Laird, C., Biegler, L., van Bloemen Waanders, B., 2006. Mixed-integer approach for obtaining unique solutions in source inversion of water networks. J. Water Resour. Plann. Manage. 132 (4), 242–251.

Laird, C., Biegler, L., van Bloemen Waanders, B., Bartlett, R., 2005. Contamination source determination for water networks. J. Water Resour. Plann. Manage. 131 (2), 125–134.

Lee, B.H., Deininger, R.A., 1992. Optimal locations of monitoring stations in water distribution system. J. Environ. Eng. 118 (1), 4–16.

Lee, B.H., Deininger, R.A., Clark, R.M., 1991. Locating monitoring stations in water distribution systems. J. Am. Water Works Assoc. 60–66.

Mukherjee, R., Diwekar, U.M., Vaseashta, A., 2017. Optimal sensor placement with mitigation strategy for water network systems under uncertainty. Comput. Chem. Eng. 103, 91–102.

Murray, R., Hart, W.E., Phillips, C.A., Berry, J., Boman, E.G., Carr, R.D., Riesen, L.A., Watson, J.-P., Haxton, T., Herrmann, J.G., et al., 2009. US environmental protection agency uses operations research to reduce contamination risks in drinking water. Interfaces 39 (1), 57–68.

Ostfeld, A., Salomons, E., 2004. Optimal layout of early warning detection stations for water distribution systems security. J. Water Resour. Plann. Manage. 130 (5), 377–385.

Ostfeld, A., Uber, J.G., Salomons, E., Berry, J.W., Hart, W.E., Phillips, C.A., Watson, J.-P., Dorini, G., Jonkergouw, P., Kapelan, Z., et al., 2008. The battle of the water sensor networks (BWSN): a design challenge for engineers and algorithms. J. Water Resour. Plann. Manage. 134 (6), 556–568.

Palleti, V.R., Narasimhan, S., Rengaswamy, R., Teja, R., Bhallamudi, S.M., 2016. Sensor network design for contaminant detection and identification in water distribution networks. Comput. Chem. Eng. 87, 246–256.

Perelman, L., Ostfeld, A., 2013. Bayesian networks for source intrusion detection. J. Water Resour. Plann. Manage. 139 (4), 426–432.

Preis, A., Ostfeld, A., 2008. Multiobjective contaminant sensor network design for water distribution systems. J. Water Resour. Plann. Manage. 134 (4), 366–377.

Propato, M., 2006. Contamination warning in water networks: general mixed-integer linear models for sensor location design. J. Water Resour. Plann. Manage. 132 (4), 225–233.

Rathi, S., Gupta, R., 2014. Sensor placement methods for contamination detection in water distribution networks: a review. Procedia Eng. 89, 181–188. 16th Water Distribution System Analysis Conference, WDSA2014.

Rico-Ramirez, V., Frausto-Hernandez, S., Diwekar, U.M., Hernandez-Castro, S., 2007. Water networks security: a two-stage mixed-integer stochastic program for sensor placement under uncertainty. Comput. Chem. Eng. 31 (5), 565–573.

Rossman, L.A., 2000. EPANET 2 User's Manual. U. S. Environmental protective Agency, Cincinnati.

Shastri, Y., Diwekar, U., 2006. Sensor placement in water networks: a stochastic programming approach. J. Water Resour. Plann. Manage. 132 (3), 192–203.

Shen, H., McBean, E., Wang, Y., 2014. Sensor placement under nodal demand uncertainty for water distribution systems. In: Securing Water and Wastewater Systems. Springer, pp. 123–133.

Weickgenannt, M., Kapelan, Z., Blokker, M., Savic, D.A., 2010. Risk-based sensor placement for contaminant detection in water distribution systems. J. Water Resour. Plann. Manage. 136 (6), 629–636.

de Winter, C., 2018. Robust Placement of Water Quality Sensors in Water Distribution Networks MSc Graduation Thesis.

Wu, Z.Y., Walski, T., 2006. Multiobjective optimization of sensor placement in water distribution systems. In: Water Distribution System Analysis Symposium 2006, pp. 1–14.

Xu, J., Small, M., Fischbeck, P., Van Briesen, J., 2010. Integrating location models with Bayesian analysis to inform decision making. J. Water Resour. Plann. Manage. 136 (2), 209–216.

Xu, X., Lu, Y., Huang, S., Xiao, Y., Wang, W., 2013. Incremental sensor placement optimization on water network. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 467–482.