Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(1): 105–121.
DOI: 10.3934/mbe.2020006
Received: 25 June 2019
Accepted: 18 September 2019
Published: 26 September 2019

Research article

Deep reinforcement learning based valve scheduling for pollution isolation in water distribution network

Chengyu Hu[1], **Junyi Cai**[1], **Deze Zeng**[1,*], **Xuesong Yan**[1], **Wenyin Gong**[1] **and Ling Wang**[2]

[1] Department of Computer Science, China university of geosciences, Wuhan, China
[2] Department of Automation Control, Tsinghua University, Beijing, China

* **Correspondence:** Email: dazzae@gmail.com

Abstract: Public water supply facilities are vulnerable to intentional intrusion. In particular, Water Distribution Network (WDN) has become one of the most important public facilities that are prone to be attacked because of its wide coverage and constant open operation. In recent years, water contamination incidents happen frequently, causing serious losses and impacts to the society. Various measures have been taken to tackle this issue. Pollution or contamination isolation by localizing the contamination via sensors and scheduling certain valves have been regarded as one of the most promising solutions. The main challenge is how to schedule water valves to effectively isolate contamination and reduce the residual concentration of contaminants in WDN. In this paper, we are motivated to propose a reinforcement learning based method for valve real time scheduling by treating the sensing data from the sensors as state, and the valve scheduling as action, thus we can learn scheduling policy from uncertain contamination events without precise characterization of contamination source. Simulation results show that our proposed algorithm can effectively isolate the contamination and reduce the risk exclosure to the customers.

Keywords: reinforcement learning; scheduling problem; water distribution network; water contamination incident

1. Introduction

Since the terrorist attacks in the United States at September 11, 2001, great efforts have been made worldwide to improve the safety of public health and people's awareness of security threats. Many countries are increasingly concerned about water security, especially the threat of malicious terrorist attacks on water supplies. Water Distribution Network (WDN) has become one of the most important public facilities that are prone to accidents or deliberate pollution invasion due to its wide coverage and continuous open state [1]. Several large scale incidents of sudden drinking water pollution in recent years have warned us that these contamination incidents may lead to social health problems and have adverse political effects [2]. Therefore, drinking water early warning system is necessary to reduce the impact of sudden pollution incident.

In a typical drinking water warning system, a large number of water quality monitoring sensors are deployed to detect contaminant [3, 4, 5]. Based on the sensing data, it is essential to identify the contaminant source, i.e., contaminant source identification [6, 7, 8]. Last, and also the most important, it is significant to take some actions to isolate the contaminant according to the emergency response policy [9].

Upon the water pollution, one intuitive way to ensure the safety of people is to cut the supply of water in the whole WDN. However, this will lead to serious social and economic losses, or may even cause social panic. An alternative way is to well schedule the valves and hydrants in the WDN to ensure the contaminants are isolated, without incurring too much negative impact. By scheduling the valves, the contaminant water can be controlled within certain range; Furthermore, by scheduling the hydrants, it is able to discharge the contaminant in the WDN so as to recover the normal water supply as soon as possible. In this case, the problem is on how to schedule the valves and hydrants based on the water quality monitoring sensing data.

For example, a simple typical water distribution network is shown in Figure 1. When contamination event occurs, there will be serious contamination diffusion if the valve is not timely and reasonably scheduled, as shown in the Figure 1 (a). However, when a reasonable scheduling is performed, the contamination situation can be controlled as shown in Figure 1 (b).

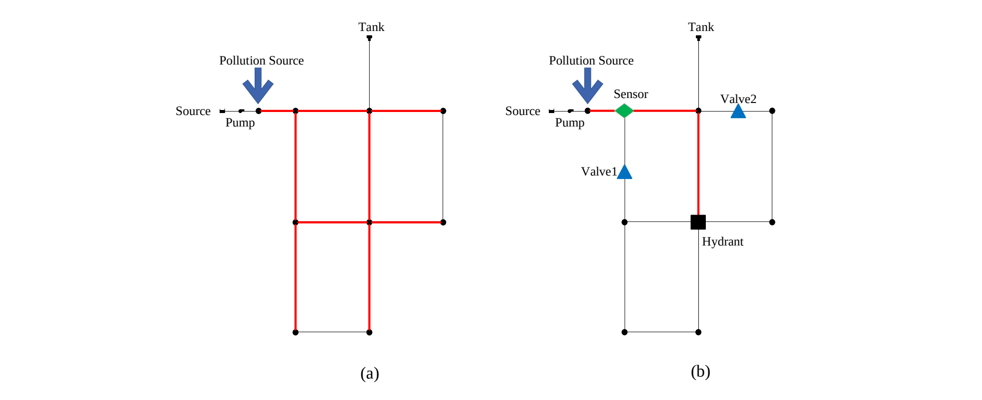


Figure 1. (a) Contaminant spreads without any response action; (b) Contaminant spreads with response actions.

The scheduling of valves and hydrants for contaminant isolation actually has been widely studied. Various optimization algorithms have been applied [10, 11], such as genetic algorithm [12] and ant colony algorithm [13]. These heuristic algorithms have the advantages of easy implementation, but they suffer from complex parameter adjustment, hindering their practical adoption. Most importantly, these heuristic algorithms require precise contamination source information for subsequent scheduling computation, which means that the timeliness of the scheduling is ignored due to the calculation of precise location of contamination source. Fortunately, we notice that the success of AlphaGo [14] has raised many interests in both academic and industry. The core of AlphaGo is deep reinforcement learning, which has been widely applied in vast domains, e.g., intelligent transportation control, computer game, robotics control, etc [15, 16]. By studying these applications, we find that deep reinforcement learning is quite appropriate to be applied for the real time scheduling of valves and hydrants for contaminant isolation in WDN. Therefore, we are motivated to investigate this issue in this paper.

The main contributions of this paper are as follows:

We accurately model the valve and hydrant scheduling problem to fit the reinforcement learning framework. In particular, we treat the sensing data in the WDN as the state, and the valve and hydrant scheduling as the action. Compared with traditional optimization algorithm, our method can model various uncertain contamination scenarios without accurately characterizing the contamination sources, which obey the timeliness principle of emergency scheduling task. By training the scheduling agent offline and deploying them online, real time scheduling can be achieved. To our best knowledge, this is the first work that applies reinforcement learning in valve and hydrant scheduling.

We use open source simulator EPANET to evaluate the efficiency of our algorithm. Extensive simulation results show that our reinforcement learning based method can well schedule the valve and hydrant to effectively isolate the contaminant.

The rest of this paper is structured as follows. Section 2 presents some related work on the scheduling of valves and hydrants in WDN, as well as some preliminaries on reinforcement learning and its applications. Then, Section 3 elaborates and formulates the valves and hydrants scheduling problem. Section 4 gives our reinforcement learning based scheduling algorithm and Section 5 shows simulation based performance evaluation results. Finally, Section 6 concludes the paper.

2. Related work

2.1. Sensor deployment and contaminant source identification

The scientific community has devoted a great deal of effort to developing sensor-based contaminant warning systems (CWS) that deploy water quality monitoring sensors in WDN to identify contaminant sources [4, 17]. Most of the previous studies focused on improving the ability of CWS to quickly identify the contaminant source and to increase the reliability of the monitoring system. For example, some new forms of sensors are invented and applied in the WDN [18], especially the mobile sensors, which can flow in the water pipes and move very close to leakage point. As a result, it is reported that the detection accuracy is higher than the traditional static sensors [19]. Once the water quality monitoring sensors detect contamination, we shall identify the contaminant source to derive where, when and how much the contaminant are inject into the WDN. Afterwards, emergency response mechanisms will give a series operations on valves and hydrants to evacuate the contaminated water [20]. Once an effective response policy is adopted, the impact of pollution can be minimized and the water supply system can be recovered to normal running status [21].

2.2. Optimization for valve and hydrant scheduling

By literature survey, we notice that many studies formulate the contaminant source identification and scheduling problems as single objective or multi-objective optimization problems. Regarding the emergency response problem to contamination events in WDNs, the basic optimization goal is to minimize the impact resulting from the contamination. For example, Poulin et al. [22] proposed an emergency response strategy to ensure drinking water safety in which the operational sequence of valves and hydrants was determined with the objective of minimizing the amount of contaminated water consumed. Later on, they further [23] considered the combination of valves and hydrants, and proposed a new operation policy of unidirectional flushing for the contaminated water. In 2012, Gavanelli et al. [24] optimized the scheduling of a set of tasks by genetic algorithm so that the consumed volume of the contaminated water is minimized. In addition to considering the operation of valves and hydrants, there are also some other actions that can be used to minimize the impact on public health. For example, the use of dye injection can act as an alert mechanism, which can discourage the public consumption of potentially contaminated water [25].

Although much of the research in this field considered the scheduling of valves and hydrants as a single objective optimization problem, it inherently involves multiple objectives such as system design costs, operation costs, water quality, and others. Accordingly, some multi-objective optimization algorithms have been proposed in some recent studies [9, 12]. Rasekh et al. [26] proposed a simulation framework via social risk assessment to simulate the dynamics of pollution events by assuming relaxed static, homologous and static responses in traditional engineering methods. They established a multi-objective model and used genetic algorithm approach to trade off the consequences and the probability of occurrence. Afshar et al. [13] propose an ant colony optimization based algorithm, coupled with the WDN simulator EPANET, to minimize the maximum regret and the total regret by selecting the best combination of hydrants and valves. Rasekh et al. [27] propose a contaminant response mechanism where the disposals are optimized using evolutionary algorithms to achieve public health protection with minimum service interruption.

We notice that the optimization algorithms can achieve good performance under deterministic environment. Otherwise, it is hard to achieve better results in dynamic or uncertain environment. For example, it challenges to design optimization algorithm to schedule valves and hydrants when water demand varies which thus leads to the change of flow speed and direction. In some extreme cases, we even cannot identify the location of contaminant source by little sensor data. In these situations, reinforcement learning can be used to schedule the valves and hydrants by reading the information from the sensors.

2.3. Reinforcement learning in scheduling problem

Reinforcement learning has been widely applied to scheduling problems in many other disciplines. For example, Knowles et al [28] use reinforcement learning to improve long term reward for a multistage decision based on feedback given either during or at the end of a sequence of actions. Yau et al [29] present an extensive review on the application of the traditional and enhanced reinforcement learning to various types of scheduling schemes, namely packet, sleep-wake and task schedulers, in wireless networks, as well as the advantages and performance enhancements brought about by reinforcement learning. In order to overcome the challenges of implementing dynamic pricing and energy consumption scheduling, Kim et al. [15] propose a reinforcement learning algorithm that allows each of the service provider and the customers to learn its strategy without a priori information about the micro-grid in electricity grid. Moghadam et al. [16] propose a two-phase reinforcement learning-based algorithm for data-intensive tasks scheduling in cluster-based data grids. These aforementioned studies show that reinforcement learning is an effective alternative for solving scheduling problem. Although with great success in different domains, none of existing studies applies reinforcement learning to solve the contaminant isolation problem via the scheduling of valves and hydrants. We are motivated to address this issue in this paper.

3. System architecture and formulation

3.1. System architecture

To ensure the safety of consumers, it is crucial to monitor water quality and operate the valves or the hydrants. In recent years, the SCADA (Supervisory Control And Data Acquisition) system has been widely deployed for water distribution to facilitate water management. Smart water management mainly includes two functions, one is to monitor portable water, the other is aiming to monitor the water supply distribution, which includes control water flow, speed, rate and tubes conditions.

In a smart water distribution management system, the components includes pipes, valves, reservoirs and clean water pumping stations. The sensors are deployed at any nodes to collect monitoring data of these component. Figure 2 shows a general water distribution architecture , which consists of three layers which are WSN (Wireless Sensor Network) layer, IoT Layer and Cloud layer.

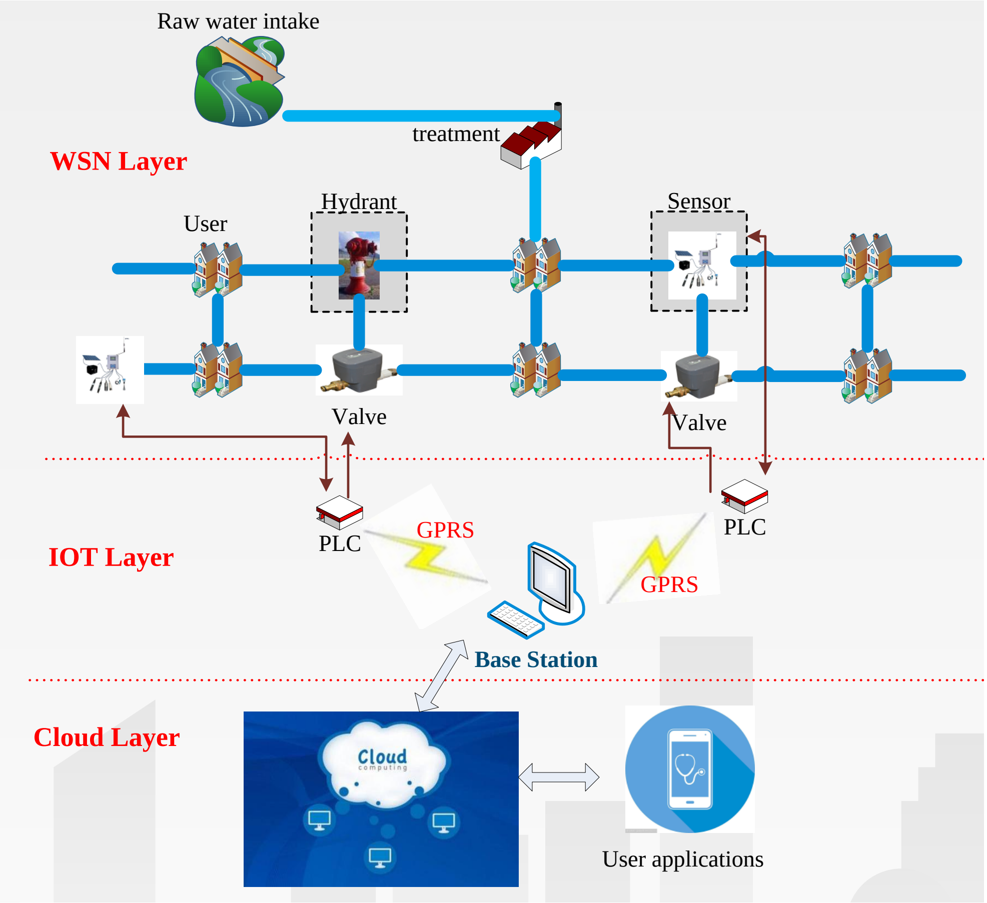


Figure 2. A typical architecture of Water distribution management.

In the WSN layer, sensors are deployed to measure the contaminant concentration and flow data. These continuous level sensors transmit data through a wireless network to a base station. Then, the base station sends the data to the cloud by the Ethernet connection. Programable Logical Controller (PLC) is used to open or close electric valves, thus change the flow direction in WDS.

IoT layer mainly provides WDS with the connectivity which allow sensors and valves to connect to cloud layer. As a bridge between WSN Layer and Cloud Layer, the IoT layer permits the control between valves and PLC, also send the data captured by sensors to cloud layers.

In the cloud layer, user can be aware of water quality and avoid accidental contamination. The authority can make a good decision directly by intelligence computation when sudden water incidents happen.

3.2. System formulation

Water distribution systems, which consist of thousands of pipes, junctions and hydro-valves, may be of loop or branch network topologies, or a combination of both. They are often modelled as a graph $G = (V, E)$, where vertices in V represent junctions, tanks, hydrants or other sources, and edges in E represent pipes and valves. The flow of drinking water depends on demand and pumping capacity, both of which may vary frequently.

There are H hydrants, N valves and M sensors in WDS. Once any contaminant event occurs, sensors may take an alarm when the contaminated water pass by. We need to give an optimal scheduling policy π of hydrants and valves at a period T, thus to maximize disposal of contaminated water as soon as possible, so the performance index can be formulated as Eq (3.1):

$$F = \max \sum_{t=1}^{T} \sum_{h=1}^{H} D_h(\pi(s_t)). \tag{3.1}$$

Here, $D_h(\pi)$ is the disposal of h-th hydrant under policy π and $\pi(s_t)$ is the scheduling policy of hydrants and valves for state s at time t. The disposal D can be simulated by a open source software named EPANET [30]. s_t is a state tensor of sensor readings which can be represented by

$$\{e_1, e_2, \ldots, e_M\} \tag{3.2}$$

Here, e_m is a continuous value acquired from sensor m, because real value reading sensors which can get more information of contamination event is used in our algorithm rather than discrete value reading sensors.

Policy π is a function of state s, and its value is an action a for a certain state s_t that can be defined as a tensor:

$$a = \{v_1, v_2, \ldots, v_N, h_1, h_2, \ldots, h_H\} \tag{3.3}$$

where $v, h \in \{0, 1\}$ is the operation for valve and hydrant respectively in WDN. The action a for state s chosen from scheduling policy π is executed to open or close valves and hydrants.

We can know from Eq (3.3) that there are 2^{N+H} possible actions. If we assume that the step between two actions in scheduling period T is stp, the time complexity of enumeration method to exhaust the optimal solution of policy π is $O(2^{(N+H)*T/stp})$. It's challenging to search the optimal solution in a large WDN with many valves and hydrant.

For the ease of reading, Table 1 summarizes the abbreviations of above technical terms.

Table 1. Notations.

G	A graph is modeled by a WDN
V	V represents junctions, tanks, hydrants or other sources
E	E represents pipes and valves
H	Number of hydrant
N	Number of valve
M	Number of sensor
π	Scheduling policy function
T	Scheduling period
s	State acquired from sensors in WDN
D	Disposal of contaminant
a	Action, a tensor composed of valve operations and hydrant operations

4. Deep reinforcement learning based valve and hydrant scheduling algorithm

4.1. Framework of scheduling algorithm

When a water quality sensor rise an alarm, the control center need to develop an optimal scheduling of valves and hydrants to minimize the impact on the consumers in WDN. Scheduling of hydrant and valve can be carried out according to the monitoring data which collected from sensors. By scheduling valves in an appropriate sequence, we intend to isolate and evacuate contaminated water. Valve can be closed or open, resulting in drinking water-break partly, or limiting movement of contaminated water in WDN. Open hydrant is able to flush contaminants out of WDN. The aim of reducing the concentration of contaminants in WDN can be achieved by scheduling the valves to lead the contaminated water body to the open hydrant. The framework of scheduling algorithm is shown in the Figure 3.

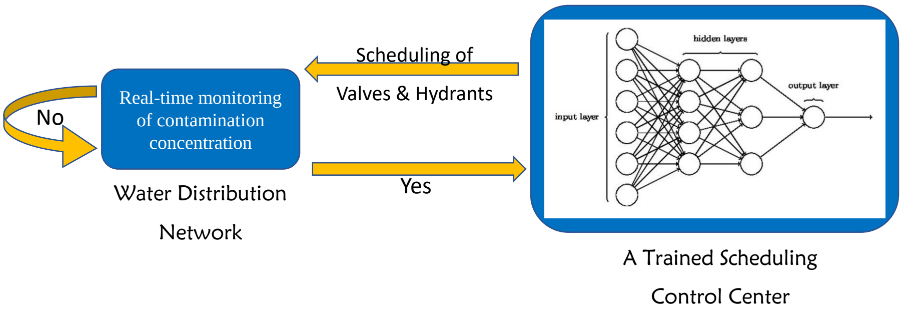

No
Real-time monitoring of contamination concentration
Water Distribution Network
Scheduling of Valves & Hydrants
Yes
hidden layers
input layer
output layer
A Trained Scheduling Control Center

Figure 3. Diagram of scheduling problem.

Control center chooses an action from optimal scheduling policy which is pre-trained by deep learning algorithm proposed in this paper according to the real-time monitoring data collected by water quality sensors in WDN, and applies action to hydrants and valves. Control center will repeat the operation until the reading of water quality sensor shows that it is at the safe level.

4.2. Markov decision processes

Reinforcement learning, distinguishing from supervised learning and unsupervised learning, focuses on the interaction between the reinforcement learning agent and the environment. Agent, as the core of reinforcement learning, obtains the optimal policy by keep learning through trial and error like humans in the different environments to pursue the optimal action, rather than directly be told what action should be done [31]. The principle of reinforcement learning is to learn how to maximize the long-term rewards through a sequence of trial actions. The challenging problem is that any action for a state not only affects the reward of the current state, but also the next state and the states thereafter in the long run. Therefore, it is essential to carefully choose the action for one state with the consideration of future possible states and rewards. Considering that the reading of the sensor is continuous value, we apply deep reinforcement learning [32] to give an optimal scheduling policy.

We first represent the problem of scheduling valves and hydrants as a Markov Decision Processes (MDP), which is defined as a tuple (s, a, p, r). In the valves and hydrants scheduling problem, states s, actions a, transition probabilities p and rewards r are defined as follows:

- s: a tensor which is made up of sensor readings.
- a: a tensor which is made up of the operation of valves and hydrants.
- p: a set of state transition probabilities. In this scheduling problem, the transition probability of the next state after an action is executed is unknown, so it is a model-free problem.
- r: a reward function: $r(s, a)$ is a real-valued immediate reward for taking action a in state s. The goal of reinforcement learning is to enable the agent to continuously interact with the environment to learn an optimal policy, so as to obtain the maximum cumulative reward. We define the mass of contaminant disposal within the time step after each action taken as the reward value, which can be obtained by simulator EPANET.

A MDP unfolds over a series of steps. At each step, the agent observes the current state, s, chooses an action, a, and then receives an immediate reward $r(s, a)$ that depends on the state and action. The agent begins in the initial state s_0, which is assumed to be known. The states transit according to the distribution p, which is hard to directly obtained, especially when the state dimension is large or the state value is continuous. Therefore, we rely on deep reinforcement learning to solve the model-free scheduling problem.

4.3. A customized deep reinforcement learning

We regard the control center in scheduling problem as an agent. The goal of the agent is to interact with the emulator (EPANET) by selecting actions in a way that maximises the future rewards (mass of contaminant disposal). We make the standard assumption that future rewards are discounted by a factor of γ per time-step, and define the future discounted return at time t as $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, where T is the scheduling period. We define the optimal action-value function $Q^*(s, a)$ as the maximum expected return achievable by following any strategy, after seeing some sequence s and then taking some action a, $Q^*(s, a) = \max_\pi \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$, where π is a policy mapping sequences to actions (or distributions over actions). The optimal action-value function obeys an important identity known as the *Bellman equation*. This is based on the following intuition: if the optimal value $Q^*(s', a')$ of the sequence s' at the next time-step is known for all possible actions a', then the optimal strategy is to select the action a' maximising the expected value of $r + \gamma Q^*(s', a')$,

$$Q^*(s,a) = \mathbb{E}_{s'\sim\varepsilon}[r + \gamma \max_{a'} Q^*(s', a')|s, a] \tag{4.1}$$

It is common to use a function approximator to estimate the action-value function, $Q^{(s,a;\theta)} \approx Q^*(s, a)$. Here we refer to a neural network function approximator with weights θ as a Q-network. A Q-network can be trained by minimising a sequence of loss functions $L_i(\theta_i)$ that changes at each iteration i,

$$L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot)}[(y_i - Q(s, a; \theta_i))^2], \tag{4.2}$$

where $y_i = \mathbb{E}_{s'\sim\varepsilon}[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1})|s, a]$ is the target for iteration i and $\rho(s, a)$ is a probability distribution over sequences s and actions a that is generated by an ϵ-greedy strategy.

We utilize a technique known as experience replay where we store the experiences of agent at each time-step, $e_t = (s_t, a_t, r_t, s_{t+1})$ in a data-set $D = \{e_1, \dots, e_N\}$, pooled over many episodes into a replay memory. During the inner loop of the algorithm, we apply Q-learning updates, or mini-batch updates, to samples of experience, $e \sim D$, drawn at random from the pool of stored samples. After performing experience replay, the agent selects and executes an action according to an ϵ-greedy policy. It should be noticed that the experience with contamination source not detected is useless, which means we should not store the experience when the all the readings of sensors below safe threshold. In this case, we execute nothing as default if none contamination was detected by sensors within scheduling period T. The customized deep Q-learning algorithm (CDQA), which is used to train a intelligent agent (control center), is presented in Algorithm 1.

Algorithm 1 The customized deep Q-learning algorithm.

```
Initialize replay memory $D$ to capacity $N$;
Initialize action-value function $Q$ network with random weights;
for $episode \in [1, L]$ do
    Sample a random junction from WDN as a contamination source and generate a contamination event $e$;
    Observe the initial state $s_1$ according to the event $e$;
    for $t \in [1, T]$ do
        With probability $\epsilon$ select a random action $a_t$, otherwise select $a_t = \max_a Q^*(s_t, a; \theta)$;
        Execute action $a_t$ in simulator and observe reward $r_t$ and next state $s_{t+1}$;
        if contaminant concentration in $s_t$ is not less than safe threshold $\phi$ then
            Store transition $(s_t, a_t, r_t, s_{t+1})$ in $D$;
            Sample random mini-batch of transition $(s_j, a_j, r_j, s_{j+1})$ from $D$;
            Set $y_j = r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta)$ for non-terminal $s_{j+1}$, or $y_j = r_j$ for terminal $s_{j+1}$;
            Perform a gradient descent step on $(y_j - Q(s_j, a_j; \theta))^2$;
        end if
    end for
end for
```

In practice, our algorithm only stores the last N experience tuples in the replay memory, and samples uniformly at random from D when performing updates. Our goal is to train an effective and general model which is independent of adjustment of the parameters for minimizing the impact of contaminant event in WDN. In Algorithm 1, the reason why we sample random contamination events from WDN is to train a general agent which is able to work well in most real contamination events. In other words, our agents may perform well without having to locate the source of the contamination, which takes a lot of online time of computation. There are two loops in CDQA, so the time complexity of our algorithm is $O(L * T)$, where T is scheduling period and L is iterations. As the iterations L increases, the policy π given by agents trained by CDQA can approach the optimal solution. The performance of CDQA may depend on a very large L, but our CDQA requires less time complexity than the enumeration method mentioned in Section 3. Considering the cost of online training is extremely expensive because of severity the of real contamination events, we train agent offline and test it in water quality simulator (EPANET).

5. Result and discussion

In order to demonstrate the ability of the agent trained by CDQA, two experiments with different number of contaminant scenario are performed. A real-world WDN [6, 33] as depicted in Figure 4 are used to simulate in our experiments. The WDN includes 97 nodes, 3 of which are hydrants and 4 of which are sensors, 119 pipes, 3 of which are valves. Assuming that the maximum scheduling period T is 24 hours; Scheduling step is 30 minutes. The water demand of each hydrant is 400 gallons per minute. For each contamination event, contaminant is continuously injected into the node of WDN at the first hour. Noted valves (blue triangle) and hydrants (black square) are located in the pipelines and nodes respectively, and sensors (red triangle) are deployed at the nodes.

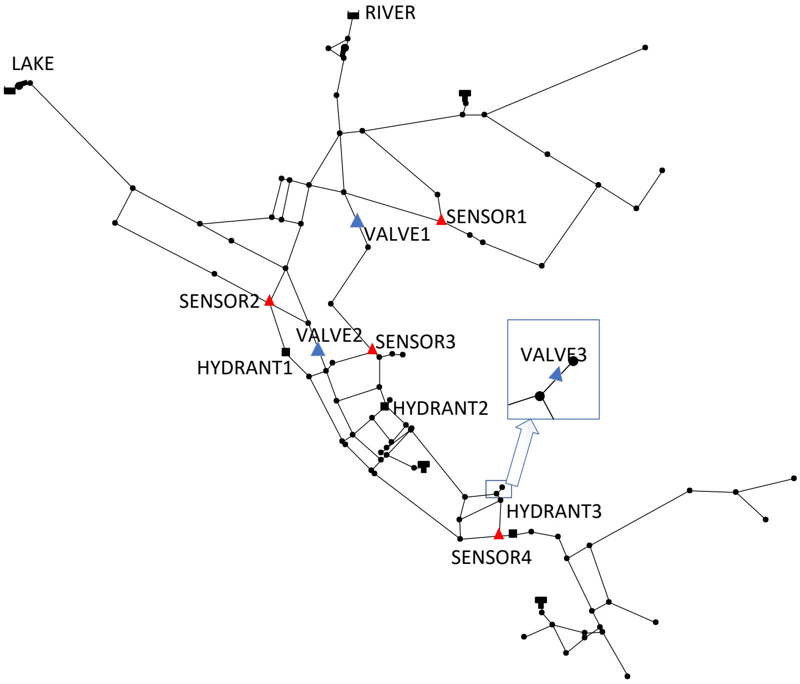


Figure 4. Four sensors (red triangle), three valves (blue triangle), and three hydrants (black square) are deployed in the WDN of 97 nodes.

In the following experiments, initial discount factor γ is set to 0.5 and we use the Adam algorithm with mini-batches of size 32 to train the Q network. The behavior policy during training was ϵ-greedy with ϵ annealed linearly from 1 to 0.1 over the half of training process and 0.1 over the other training process. We set episode iterations L to 5000 and capacity N of replay memory to 1000. Safe concentration ϕ is set to 0.2 mg/L. The structure of Q network shown in Table 2 is a classical fully-connected network where there are 3 hidden layers, 1 input layer and 1 output layer. The input tensor is concatenation of action tensor a shown in Eq 3.3 and sensor reading tensor shown in Eq 3.2. The output is prediction of Q value.Gavanelli2012.

Table 2. Structure of Q network.

Layer	Units	Activation Function
fully connected	15	ReLU
fully connected	6	ReLU
fully connected	6	ReLU
fully connected	1	Linear

In order to show the performance of agents in each episode, we apply EPANET to simulate two cases, one is that all the valves and hydrants are open, the other is that all the valves are closed and all the hydrants are open, then we can compute the mass of contamination disposal VOHO and VCHO, respectively.

5.1. *Single contaminant event happens in WDS*

In this experiment, we test three single contaminant events which occur at the node SOURCE1, SOURCE2 and SOURCE3, respectively. The locations of contamination sources are marked with blue arrow in Figure 5. It is should be noticed that these test contaminant events are all chosen without deliberation. Each contamination events is evaluated in every episode of the algorithm.

Figure 6 shows how the mass of contaminants disposal evolves during training on the contamination events SOURCE1, SOURCE2 and SOURCE3. From the Figure 6, we can see that plots are not stable, but the algorithm tends to converge at the later stage of training. The most important thing is that the performance of our CDQA are much better than VOHO and VCHO performed on these three plots.

5.2. *Multiple contaminant events happen in WDS*

We set the three contamination events used in the last experiment (SOURCE1, SOURCE2, and SOURCE3) to happen simultaneously as a test. Two new contamination events SOURCE4 and SOURCE5 are added as another test. Both tests are evaluated in every episode of the algorithm. The locations of the five contamination sources are marked with arrow in Figure 7.

Figure 8 shows how the mass of contaminants disposal evolves during training on the contamination events where SOURCE1, SOURCE2 and SOURCE3 occur simultaneously and SOURCE4 and SOURCE5 occur simultaneously. As shown in the figure, the CDQA still has advantages over VOHO and VCHO because most of the points on the CDQA are above VOHO and VCHO.

In order to further explore the final performance of the agent trained by CDQA, we recorded the state within a scheduling period at the last episode of training process of CDQA, which is shown in Table 3. In this table, the mass of contaminant disposal is 23648 gram and the contamination event is that three

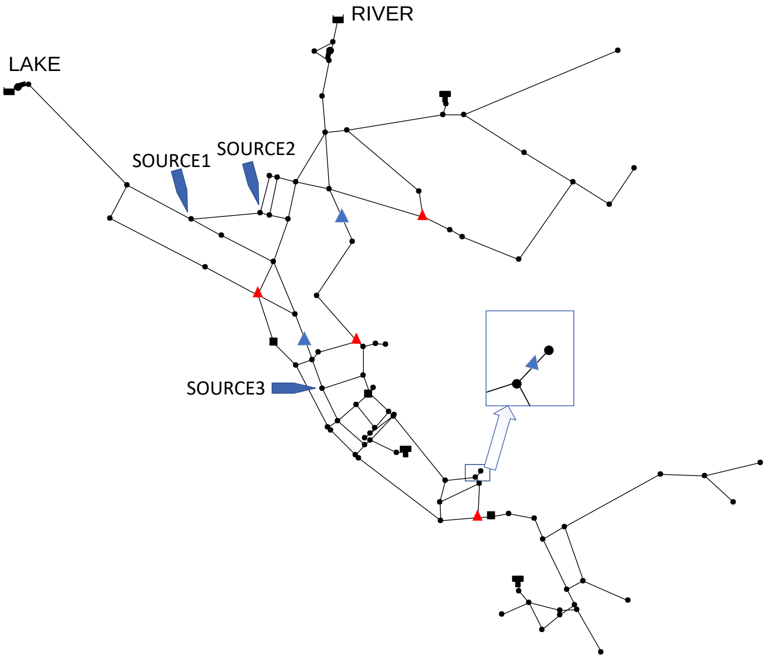
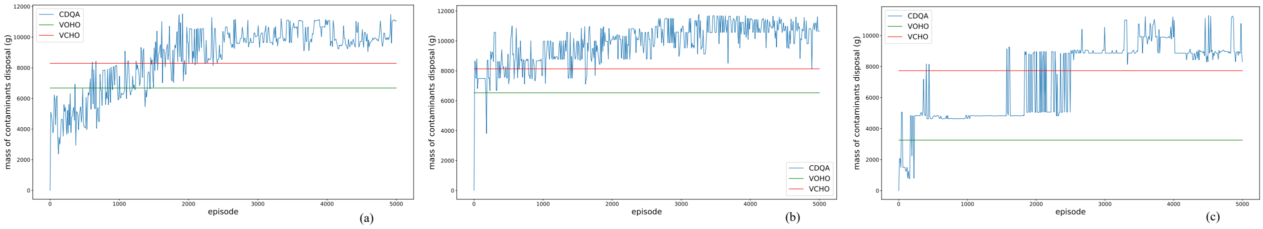


Figure 5. Locations of contaminant event at the node SOURCE1, SOURCE2 and SOURCE3.



(a)

(b)

(c)

Figure 6. Plot (a), (b) and (c) show how the mass of contaminants disposal evolves during training on the contamination events SOURCE1, SOURCE2 and SOURCE3, respectively.

contamination sources SOURCE1, SOURCE2 and SOURCE3 occur simultaneously. The locations of sensors, valves and hydrants were shown in Figure 4. The reading of Sensor2 is 264.842 at time 1 and readings of all the sensors are below safe concentration after time 33, which determine that the scheduling period with scheduling step 30 minutes of this contamination event is 16 hours. Scheduled by our agent, the contaminant concentration drops from 264.842 (reading of Sensor2) to 0.201 (reading of Sensor4). Table 3 has shown that hydrants are open in most situations that the readings of sensors are greater than the safe concentration threshold, which is consistent with common sense, because we always expect to discharge as much contaminants as possible, which indirectly shows that our method is feasible and effective.

In each episode of CDQA of training process, we use contamination scenes of a single source, but the experiment show that we can also get a desired result in multiple source scenes, which indicates that agent trained by CDQA algorithm have certain generalization for other contamination scenes without sampled in algorithm. In other words, we obtain a general agent which can solve the scheduling

Table 3. State within a scheduling period at the last episode of training process of CDQA. 0 or 1 means open or closed for a valve respectively, while 0 or 1 means closed or open for a hydrant respectively.

Time	Sensor1	Sensor2	Sensor3	Sensor4	Valve1	Valve2	Valve3	Hydrant1	Hydrant2	Hydrant3
1	0	264.842	0	0	0	0	0	0	0	0
2	0	64.093	0	0	0	0	1	0	1	0
3	0	21.283	0	0	1	0	0	1	1	0
4	0	197.593	73.643	0	1	0	0	1	1	0
5	0	57.586	2.113	0	0	0	1	1	1	1
6	0	99.784	129.751	3.506	1	0	0	1	1	0
7	0	13.664	1.488	2.045	0	0	1	1	0	0
8	0	0	190.991	92.851	1	0	0	1	1	1
9	0	0	83.131	131.996	1	0	0	1	1	1
10	0	0	2.717	83.362	0	0	1	1	0	0
11	0	0	0	90.745	0	1	0	1	0	1
12	0	27.993	0	90.888	0	1	0	1	0	0
13	0	47.11	0	69.01	0	0	1	1	0	1
14	0	7.576	0	60.77	0	0	0	0	1	1
15	0	56.247	0	53.248	1	0	0	0	0	1
16	0.251	4.562	36.005	13.280	1	0	0	1	1	1
17	0	22.855	36.492	7.804	1	0	0	1	1	1
18	0	19.992	37.552	1.577	1	0	0	1	1	1
19	0	0	38.871	0	1	0	0	1	1	1
20	0	9.321	18.865	21.84	1	0	0	1	1	1
21	0	15.722	25.558	43.041	1	0	0	1	1	1
22	0	12.69	1.896	22.235	0	0	0	1	0	0
23	0	5.269	13.398	27.858	1	1	1	1	0	1
24	0	0	10.096	5.447	1	1	0	1	1	1
25	0	0	15.722	4.598	1	1	0	1	1	1
26	0	0	5.653	3.12	1	1	0	1	1	1
27	0	0	0	0.234	1	1	0	1	1	1
28	0	0	0	0.234	1	1	0	1	1	1
29	0	0	0	0.219	1	1	0	1	1	1
30	0	0	0	0.216	1	1	0	1	1	1
31	0	0	0	0.212	1	1	0	1	1	1
32	0	0	0	0.208	1	1	0	1	1	1
33	0	0	0	0.201	1	1	0	1	1	1

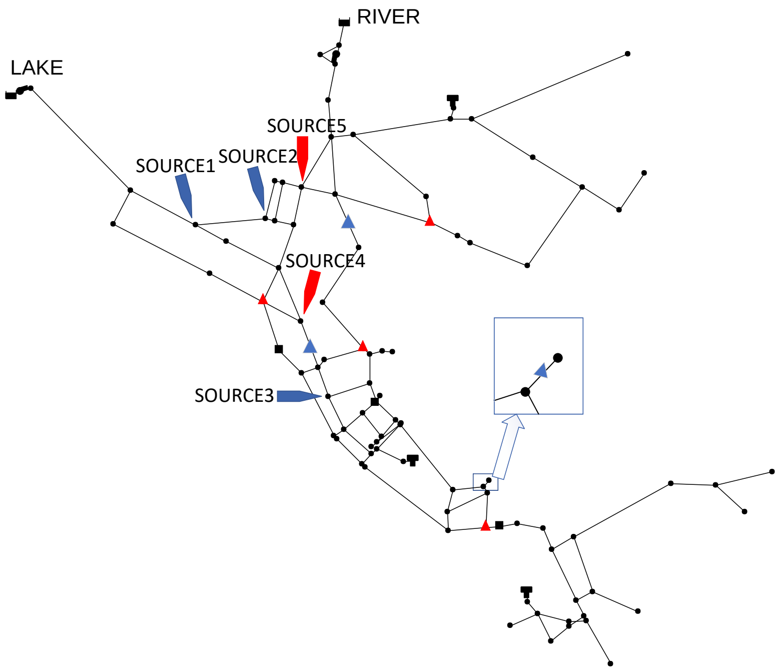


Figure 7. The locations of contamination sources SOURCE1, SOURCE2, and SOURCE3 in the first test are marked with blue arrow and SOURCE4 and SOURCE5 in the second test are marked with red arrow.

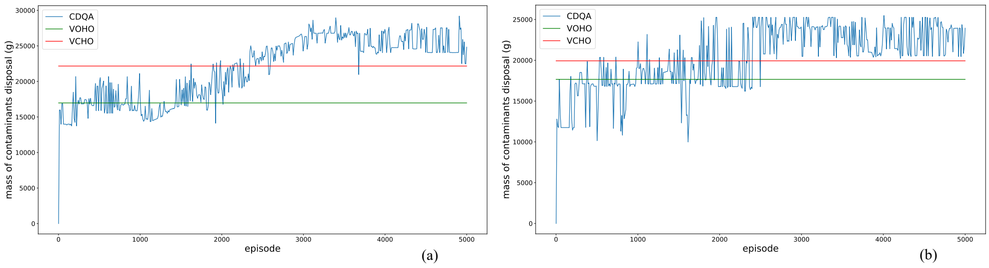


Figure 8. The plot (a) shows the variation of the mass of contaminants disposal during training process when contamination events SOURCE1, SOURCE2 and SOURCE3 occur simultaneously. The plot (b) shows the variation of the mass of contaminants disposal when SOURCE4 and SOURCE5 occur simultaneously.

problem of uncertain contaminant event. Moreover, the trained agent receive the real time state and give a action instantly for scheduling of valves and hydrants, which saves lots of expensive computing time after contaminant event occurs.

6. Conclusion

In this paper, we investigate the problem of valve and hydrant scheduling for contaminant water evacuation and water supply recovery in WDNs. We first give a comprehensive survey on existing solutions to this problem and notice that all previous studies need to precisely locate the contamination source before scheduling and then cost some time to search scheduling strategy. We therefore are motivated to propose a customized deep Q-learning based algorithm, with well design of the state, action and reward, to address this issue. This is the first time that deep reinforcement learning has been used to solve such problems as a real time scheduling problem. To evaluate the performance of our algorithm, we adopt EPANET to simulate various contaminant injection incidents in a typical WDN. The experiment results show that our algorithm can not only achieve good experimental results in single contamination source events, but also perform well in multiple contamination source events. Our work proves the feasibility and efficiency of applying deep reinforcement learning for valve and hydrant scheduling for contaminant water evacuation in WDNs.

Acknowledgment

This research was partially supported by NSF of China (Grant No. 61673354) and National Key Research and Development Project(Grant No. 2016YFE0104400) and the State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology (DMETKF2018020, DMETKF2019018). This paper has been subjected to Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. S. E. Hrudey, Safe drinking water: lessons from recent outbreaks in affluent nations, 2005.
2. D. J. Kroll, *Securing our water supply: protecting a vulnerable resource*. PennWell Books, 2006.
3. S. Rathi and R. Gupta, A simple sensor placement approach for regular monitoring and contamination detection in water distribution networks, *KSCE J. Civ. Eng.*, **20** (2016), 597–608.
4. C. Hu, G. Ren, C. Liu, et al., A spark-based genetic algorithm for sensor placement in large scale drinking water distribution systems, *Cluster Comput.*, **20** (2017), 1089–1099.
5. X. Yan, K. Yang, C. Hu, et al., Pollution source positioning in a water supply network based on expensive optimization, *Desalin Water Treat*, **110** (2018), 308–318.
6. H. Wang and K. W. Harrison, Improving efficiency of the bayesian approach to water distribution contaminant source characterization with support vector regression, *J. Water Res. Pl.*, **140** (2012), 3–11.
7. X. Yan, J. Zhao, C. Hu, et al., Multimodal optimization problem in contamination source determination of water supply networks, *Swarm Evol. Comput.*, 2017.
8. W. Gong, Y. Wang, Z. Cai, et al., Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution, *IEEE T. Syst. Man Cy.*, (2018), 1–15.
9. A. Afshar and M. A. Marino, Multiobjective consequent management of a contaminated network under pressure-deficient conditions, *J. Am. Water Works. Ass.*, **106** 2014.

10. T. Ren, S. Li, X. Zhang, et al., Maximum and minimum solutions for a nonlocal p-laplacian fractional differential system from eco-economical processes, *Bound Value Probl.*, **2017** (2017), 118.

11. T. Ren, X. H. Lu, Z. Z. Bao, et al., The iterative scheme and the convergence analysis of unique solution for a singular fractional differential equation from the eco-economic complex system's co-evolution process, *Complexity*, 2019.

12. A. Preis and A. Ostfeld, Multiobjective contaminant response modeling for water distribution systems security, *J. Hydroinform.*, **10** (2008), 267–274.

13. A. Afshar and E. Najafi, Consequence management of chemical intrusion in water distribution networks under inexact scenarios, *J. Hydroinform.*, **16** (2014), 178–188.

14. D. Silver, A. Huang, C. J. Maddison, et al., Mastering the game of go with deep neural networks and tree search, *Nature*, **529** (2016), 484.

15. B. G. Kim, Y. Zhang, M. van der Schaar, et al., Dynamic pricing and energy consumption scheduling with reinforcement learning, *IEEE T. Smart Grid*, **7** (2016), 2187–2198.

16. M. H. Moghadam and S. M. Babamir, Makespan reduction for dynamic workloads in cluster-based data grids using reinforcement-learning based scheduling, *J. Comput. Sci-neth.*, 2017.

17. S. Rathi and R. Gupta, A critical review of sensor location methods for contamination detection in water distribution networks, *Water Qual. Res. J. Can.*, **50** (2015), 95–108.

18. T. P. Lambrou, C. C. Anastasiou, C. G. Panayiotou, et al., A low-cost sensor network for real-time monitoring and contamination detection in drinking water distribution systems, *IEEE Sens. J.*, **14** (2014), 2765–2772.

19. D. Zeng, L. Gu, L. Lian, et al., On cost-efficient sensor placement for contaminant detection in water distribution systems, *IEEE T. Ind. Inform.*, (2016), 1–1.

20. L. Perelman and A. Ostfeld, Operation of remote mobile sensors for security of drinking water distribution systems, *Water Res.*, **47** (2013), 4217–4226.

21. M. R. Bazargan-Lari, An evidential reasoning approach to optimal monitoring of drinking water distribution systems for detecting deliberate contamination events, *J. Clean Prod.*, **78** (2014), 1–14.

22. A. Poulin, A. Mailhot, P. Grondin, et al., Optimization of operational response to contamination in water networks, in *WDSA 2006*, (2008), 1–15.

23. A. Poulin, A. Mailhot, N. Periche, et al., "Planning unidirectional flushing operations as a response to drinking water distribution system contamination," *J. Water Res Pl.*, **136** (2010), 647–657.

24. M. Gavanelli, M. Nonato, A. Peano, et al., Genetic algorithms for scheduling devices operation in a water distribution system in response to contamination events, **7245** (2012), 124–135.

25. A. Rasekh and K. Brumbelow, Water as warning medium: Food-grade dye injection for drinking water contamination emergency response, *J. Water Res. Pl.*, **140** (2014), 12–21.

26. A. Rasekh, M. E. Shafiee, E. Zechman, et al., Sociotechnical risk assessment for water distribution system contamination threats, *J. Hydroinform*, **16** (2014), 531–549.

27. A. Rasekh and K. Brumbelow, Drinking water distribution systems contamination management to reduce public health impacts and system service interruptions, *Environ. Model Softw.*, **51** (2014), 12–25.

28. M. Knowles, D. Baglee and S. Wermter, Reinforcement learning for scheduling of maintenance, in *Research and Development in Intelligent Systems XXVII*. Springer, (2011), 409–422.

29. K. L. A. Yau, K. H. Kwong and C. Shen, Reinforcement learning models for scheduling in wireless networks, *Front Comput. Sci-Chi.*, **7** (2013), 754–766.

30. L. A. Rossman *et al.*, Epanet 2: users manual, 2000.

31. R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, in *Neural Information Processing Systems*, 1999.

32. V. Mnih, K. Kavukcuoglu, D. Silver, et al., Playing atari with deep reinforcement learning, *ArXiv*, **abs/1312.5602**, 2013.

33. Y. Xuesong, S. Jie, and H. Chengyu, Research on contaminant sources identification of uncertainty water demand using genetic algorithm, *Cluster Comput.*, **20**, (2017), 1007–1016.