Proactive and Real-Time Optimal Control of Water Quality in Water Distribution Networks

by

Xiushuang Li

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2021 by the
Graduate Supervisory Committee:

Pitu Mirchandani, Chair
Treavor Boyer
Feng Ju
Giulia Pedrielli

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

Drinking water quality violations are widespread in the United States and elsewhere in the world. More than half of Americans are not confident in the safety of their tap water, especially after the 2014 Flint, Michigan water crisis. Other than accidental contamination events, stagnation is a major cause of water quality degradation. Thus, there is a pressing need to build a real-time control system that can make control decisions quickly and proactively so that the quality of water can be maintained at all times. However, towards this end, modeling the dynamics of water distribution systems are very challenging due to the complex fluid dynamics and chemical reactions in the system. This challenge needs to be addressed before moving on to modeling the optimal control problem. The research in this dissertation leverages statistical machine learning approaches in approximating the complex water system dynamics and then develops different optimization models for proactive and real-time water quality control. This research focuses on two effective ways to maintain water quality, flushing of taps and injection of chlorine or other disinfectants; both of these actions decrease the equivalent "water age", a useful proxy for water quality related to bacteria growth. This research first develops linear predictive models for water quality and subsequently linear programming optimization models for proactive water age control via flushing. The second part of the research considers both flushing and disinfectant injections in the control problem and develops mixed integer quadratically constrained optimization models for controlling water age. Different control strategies for disinfectant injections are also evaluated: binary on-off injections and continuous injections. In the third part of the research, water demand is assumed to be uncertain and stochastic. The developed approach to control the system relates to learning the optimal real-time flushing decisions by combing reinforced temporal-difference learning approaches with linear value function approximation for

solving approximately the underlying Markov decision processes. Computational results on widely used simulation models demonstrates the developed control systems were indeed effective for water quality control with known demands as well as when demands are uncertain and stochastic.

ACKNOWLEDGMENTS

I would like to first thank Dr. Pitu Mirchandani, my committee chair, for his guidance and support throughout this journey. I am grateful that he offered me a research assistant position in the summer of 2015, which started our collaboration. Dr. Mirchandani's guidance and encouragement ever since helped me push myself beyond the limits. I truly appreciate all the research advice he shared with me and all the time he spent in guiding me. I would also like to thank my committee members: Dr. Treavor Boyer, Dr. Feng Ju and Dr. Giulia Pedrielli for their constant and valuable feedback on my research, which has helped improve this dissertation a lot. I also want to express my sincere gratitude to Dr. Linda Chattin and Dr. Daniel McCarville for their support during my teaching assistant jobs. I also want to thank Dr. Jorge Sefair for his help and encouragement during my Ph.D. studies.

I want to shout out a big thank you to all my friends! Thank you Hyunsoo Yoon for being my big brother and always supporting me. Thank you Taeyeong Choi for being a great friend and always listening to me. Thank you Ghazal Shams for your constant support and encouragement, I am grateful to have a friend like you. Thank you Yazhu Song, Guanqi Fang, Mona Khoddam, Dening Peng, Seho Kee, Nathan Gaw, Kun Wang, Sangdi Lin, Viswanath Potluri, Fei Gao, Chao Wang, Xufeng Yao, Feifan Wang, Maziar Kasaei Roodsari, Nooshin Shomal Zadeh, Yeawon Yoo, Arun Bala Subramaniyan, Carly Metcalfe, Logan Mathesen, and Brittany Fischer for making this long journey memorable with all the happy times.

Lastly, I want to thank my parents and my brother for their love and support, as well as always encouraging me to achieve my goals.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

"Only about half of Americans are very confident in the safety of their tap water, and a majority think lead contamination of the tap water in Flint, Michigan, indicates a more widespread problem, rather than an isolated problem"(Swanson, 2016). Indeed, Allaire *et al.* (2018) pointed out that drinking water quality violations are widespread in the United States, and 21 million people used water from water distribution systems that violated health-based quality standards in 2015. Benedict *et al.* (2017) reported that 42 drinking water associated outbreaks occurred and caused over 1000 cases of illness and 13 deaths from 2013 to 2014. Rhoads *et al.* (2016) found out increased water age and degraded water quality occur very often even in green buildings.

Among many other factors causing water quality to violate the standards, stagnation plays a key contributing role. During long periods of stagnation, lead, copper or other metals can leach from pipes, microbial growth will also increase (Proctor *et al.*, 2020; Ley *et al.*, 2020; Ling *et al.*, 2018; Wang *et al.*, 2014). Lautenschlager *et al.* (2010) found out that microbial growth can increase dramatically after overnight stagnation in household buildings, while Bédard *et al.* (2018) investigated the impact of stagnation on water quality in large buildings. They both observed that bacteria level could increase even after a short period of stagnation (1 hour). The microbial growth can be very concerning because one of the bacteria, Legionella, can cause Legionnaires' disease. During Covid-19 shutdown, many people had to work from

home, thus resulting numerous low-occupancy or even empty buildings. Researchers' attentions have since been drawn to water quality in those buildings. And Cassell *et al.* (2021) showed that the reported cases of Legionnaires' disease in US have increased more than five times from 2000 to 2018, according to data from the Center for Disease Control and Prevention (CDC). CDC also created a toolkit in 2017 for developing a water management program to reduce Legionella Growth and spread in buildings (CDC, 2017). However, it involves complicated decision makings throughout the monitoring and control process, for example, what quality measurements to be collected, where and how often they are to be collected, what kind of control actions to be implemented, where and when to trigger the control actions, what is the magnitude of the control, etc.

CDC also noted in their 2017 toolkit that flushing and adding disinfectants (e.g., chlorine) can be effective in improving water quality. Lautenschlager *et al.* (2010) and Bédard *et al.* (2018) both showed that flushing of taps is an effective way to improve the water quality after stagnation. Proctor *et al.* (2020) also developed guidelines that suggest flushing and adding disinfectants for building water management. Hozalski *et al.* (2020) showed that flushing can rapidly restore the chlorine residual (which prohibits the bacteria growth) by analyzing water samples collected before, during and after flushing showers in 5 unoccupied university buildings. Although flushing of taps has been shown effective in improving water quality after stagnation, it also ends up with a significant amount of water being wasted, while 2 billion people globally still lack access to safe drinking water, according to a paper recently published in *Nature* by Everard (2019). The U.S. Environmental Protection Agency (EPA), U.S. Green Building Council (USGBC), and others have recognized the benefits of water conservation. For example, the EPA created "WaterSense at Work", which is a detailed guide to water efficiency in commercial and institutional (CI) buildings (US

Environmental Protection Agency, 2012). One of the most visible signs of water efficiency is the certification by Leadership in Energy and Environmental Design (LEED) or "green" CI buildings, which are designed to use significantly less energy and water than conventional CI buildings (e.g., water use reduced by 60%). The dilemma is how to achieve a good balance between maintaining water quality and wasting less water.

This dissertation aims to address key decision making and water quality control problems by leveraging statistical machine learning in the optimal control of water quality for different water distribution systems. Specifically it will focus on how to derive the optimal control policy for water quality under two different scenarios, under known deterministic water demand and under uncertain stochastic water demand. It will start with different statistical machine learning approaches in approximating the system dynamics; because the water distribution system involves complicated fluid dynamics and chemical reactions, it is very difficult to derive an accurate mathematical model to describe the system dynamics. Then the research focuses on the optimal control of water age (a water quality indicator) via flushing using the approximate model for system dynamics. The problem is then extended by including both flushing and chlorine injection as controls. In the scenario of uncertain stochastic water demand, the dissertation discusses how to develop the optimal control policy in an online fashion using the reinforcement learning framework.

The dissertation chapters are arranged as follows. Chapter 2 focuses on approximating the system dynamics via different statistical machine learning approaches, and the optimal control of water age through flushing. A building water distribution network and a city water distribution network are used as case studies. Chapter 3 then extends the problem by including both flushing and chlorine as controls. Different chlorine injection strategies (binary and continuous) are considered in the problem

formulation. In Chapter 4, water demand is assumed be to stochastic and future demand is unknown. A problem formulation for deriving the optimal control policy based on reinforcement learning is developed. The problem is then solved with temporal difference learning combined with function approximation in an online fashion. Final Chapter 5 summarizes the contributions of this dissertation and discusses potential further research directions.

Background overviews of each of the relevant topics are given in the subsections.

1.2 The General Optimal Control Problem

Optimal control theory is an extension of the calculus of variations, and Sargent (2000) noted that it has a long history dating back to the 17th century when Galileo worked on two shape problems, the catenary and the brachistochrone. Sargent also gives a more detailed history of the optimal control theory in that paper. Since the seminal works Lev Pontryajin and Richard Bellman in the 1950s, usage of optimal control methods in various applications have dramatically increased.

A typical optimal control problem is to find a control policy for a dynamical system over a period of time such that the objective function (usually the total cost) is optimized. A very abstract framework for an optimal control problem is as follows.

Minimize the continuous-time cost functional (the discrete-time case is similar, just substitute the integral with summation)

$$J = \Phi[\boldsymbol{x}(t_0), t_0, t_f] + \int_{t_0}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t]dt \tag{1.1}$$

subject to

$$\dot{\boldsymbol{x}(t)} = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \tag{1.2}$$

$$\boldsymbol{g}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq 0 \tag{1.3}$$

$$\phi[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] = 0 \tag{1.4}$$

where $\boldsymbol{x}(t)$ is the state variable, $\dot{\boldsymbol{x}}(t)$ is the first order derivative of $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$ is the control variable, t_0 is the initial time, t_f is the terminal time. The Φ term in objective function (1.1) is the endpoint cost, and the $\mathcal{L}$ term is called Lagrangian, which represents the cost per unit time.

Constraint (1.2) is the first-order system dynamics (called the state equation), constraint (1.3) is the algebraic path requirement, and constraint (1.4) models the boundary conditions requirements.

In general, this is a nonlinear optimization problem due to the nonlinearity of function $\boldsymbol{f}$, and the global optimal solution is hard to achieve. But in some special cases, for example, when the state equation (1.2) is linear and functions Φ and $\mathcal{L}$ are quadratic, then the problem becomes the linear quadratic (LQ) optimal control problem, to which there exist analytical solutions (Kalman *et al.*, 1960).

Constraint (1.2) may not be as straightforward as it looks. In some cases, the state equation (1.2) can be derived from physics laws, for example, based on the balance of mass and energy (Cembrano *et al.*, 2000; Sakarya and Mays, 2000) when the system is not too complex. Then the effort is mainly spent on how to solve the resulting nonlinear optimization problem. In some other cases, the state equation (1.2) is almost impossible to be derived from physics laws due to the high complexity of the system dynamics, then the resulting optimal control problem is much harder. In such cases, some kind of approximation of (1.2) is necessary to develop a solution approach. One straightforward solution approach is to come up with a way to approximate the system dynamics. Since basically the system dynamics model (1.2) is to predict the state at next time step given current state and control, statistical machine learning allows researchers to take advantage of its prediction power. Because the approximation of system dynamics is to predict the system state at next time step using current system state information and other available current and

past data, approximating system dynamics perfectly fits into the scope of machine learning. Shin *et al.* (2010) applied support vector regression (SVR) in their work of model predictive flight control, they implemented an adaptive version of SVR to predict the state of a fixed-wing unmanned aerial vehicle, including slideslip angle, roll rate, rudder angle, roll angel, etc. They considered the prediction from SVR model as the real plane state in their model predictive control module. Recently, Jain *et al.* (2018) made use of regression tree to predict the power consumption of an office building given historical data, and later incorporated the built regression trees into their formulation of the minimal peak power consumption. Sondermeijer *et al.* (2019) used step-wise regression to approximate the complex inverter output solely based on local measurements, which is included in their formulation of optimal power flow control problem. These are just a few examples of machine learning applications in approximating system dynamics in the areas of flight control and building energy management. In fact the idea is so powerful that it can be applied in almost all kinds of optimal control problems. To give another example, consider the freeway ramp metering problem, the ramp metering rates must depend on total traffic on freeway, the speed of vehicles, and the density of traffic flow, etc. However, in order to formulate the problem properly, a model of the traffic system dynamics, either accurate or approximate, must be built. Statistical machine learning can be used to achieve that goal. Given historic data, a model can be built to approximate the dynamics of freeway traffic. Then the model can be used as the state equation constraint to determine optimal ramp metering rates.

Another approach to optimal control is to model the problem using a dynamic programming (DP) formulation (Bertsekas *et al.*, 2000; Sundström *et al.*, 2010). Because of computing complexity of solving DP exactly, optimization researchers use approximate dynamic programming (ADP) (Bertsekas, 2011; Powell, 2007). This ap-

proach is often called reinforcement learning in computer science (Sutton and Barto, 2018). ADP is very widely used in robotics to control the robots. ADP is becoming increasingly popular in many other complicated control problems, such as the control of autonomous vehicles (Sallab *et al.*, 2017; Shalev-Shwartz *et al.*, 2016; Kiran *et al.*, 2021), the operation of water resources systems (Castelletti *et al.*, 2013; Madani and Hooshyar, 2014), etc. Basically, there are two types of ADP approaches, model-based approaches and model-free ones. The model-based methods normally address models for underlying Markov decision processes and use dynamic programming to solve the problem albeit using approximate value-iteration or policy iteration approaches. Model-free methods include Monte Carlo, temporal difference learning (Q-learning and SARSA), deep Q-learning, etc. They are based on samples from previous history, and do not require a Markov process model of the system. The value function of a state, which is the total expected reward (or cost) over the future starting from that state, is estimated using an iterative approach on historic samples. And the optimal policy can be derived from the estimated value function. More details of such approaches can be found in Sutton and Barto (2018).

1.3 The Approximation of Water Distribution System Dynamics

As mentioned in section 1.2, the approximation of system dynamics, or the state equation (1.2), is necessary when the system dynamics are complex, as they are in water distribution systems where the fluid dynamics and chemical reactions are very complicated to model. What is needed is system transition model that gives the relationship between current system state and system state after a time step. Since continuous data collection is not a major issue nowadays thanks to the advancement in sensors and cyber connectivity, an approximate state transition model is possible to be built by using statistical machine learning. And even if we cannot collect enough data

from the real system, we can make use of simulation modeling. A calibrated simulation model functions like the real system, and we can run the simulation multiple times to gather data for the machine learning model. Then the constraint (1.2) can be replaced by the simpler approximate system transition model in the determination of the optimal control, which now becomes easier to compute.

The approximate system transition model does not only make the optimal control problem solvable, but it also makes proactive control possible that uses prediction of future states. In water distribution systems, it is essential to maintain the water quality at satisfactory level all the times. Currently, the water quality is controlled in a passive way by frequently taking readings from sensors installed in the system, and alerting when the readings are abnormal (Storey *et al.*, 2011). By the time alerts are generated and noticed, the water quality could already be unacceptable and possibly has affected people's water usage and their health. In order to avoid such incidents, a proactive control strategy is necessary. This is where the prediction can play an important role. With the development of machine learning, one can easily build a prediction model of water quality. As long as the predictions are accurate and early enough, control actions can be taken much earlier to avoid water quality incident. With the prediction model, not only the actions can be taken much earlier, but the control system can be cost effective.

As one of the most popular approaches in machine learning, neural network models are often the first choice of many researchers in their work of water quality prediction. Even before the current popularity of artificial intelligence, neural network models have been used by water resources researchers. Maier and Dandy (1996) used neural networks in their prediction of salinity in a river 14 days in advance. They also wrote a review paper in 2000 to discuss the application of neural network models in forecasting water resources variables (Maier and Dandy, 2000). Since 2000, the application of

neural networks in water quality prediction has only become more widely used (Najah *et al.*, 2013; Palani *et al.*, 2008; Singh *et al.*, 2009). In recent years, researchers also explored the application of *Long Short-Term Memory* (LSTM) neural networks in water quality prediction. Wang *et al.* (2017) used LSTM neural network to predict water quality indicators and showed an improved accuracy over traditional neural networks. Jia *et al.* (2018) constructed a hybrid model of physics and data including LSTM neural network to monitor water temperature and quality in lakes. The high accuracy of neural network models makes it very popular in applications. However, the problem with neural network is that they are usually black-box models, and do not have much interpretability in terms of why they make specific predictions. The training of an accurate neural network model also requires much more data than other machine learning predictive models, and the training time increases rapidly with more data.

Lasso regression was first proposed by Tibshirani (1996), and it has since been used in many predictive modeling (both regression and classification), because of its advantage of selecting the most significant predictor variables. Liu *et al.* (2016) incorporated the lasso penalty term in their prediction model of residual chlorine in urban water distribution system to automatically identify important water quality predictor variables. Brooks *et al.* (2016) compared 14 regression methods in their beach water quality prediction and showed adaptive lasso model can achieve the second best accuracy. Trueman *et al.* (2016) applied lasso regression, linear regression, boosted regression tree ensembles etc. in the prediction of disinfection by-product in drinking water and showed lasso model can achieve the lowest mean absolute error in some of their data sets.

Biology inspired algorithms have long been an important subject of artificial intelligence. One such popular technique is genetic programming based symbolic regres-

sion (Koza, 1992). It is a stochastic evolutionary approach that builds mathematical expressions by iteratively fitting a set of data. Singh and Gupta (2012) applied gene expression programming, which is an extension to genetic programming, in the prediction of disinfection by-products in water. They showed the genetic-programming-based algorithm can achieve similar accuracy as support vector machine and neural network models. Savic (2009) discussed the application of evolutionary polynomial regression in modeling the pipe burst rates in their book. Berardi *et al.* (2008) also made used of evolutionary polynomial regression to build prediction models of pipe bursts in water distribution systems. Genetic programming based algorithms are easy to use, and requires no assumption on the form of the model. Therefore, it becomes a very flexible approach to explore the input-output relationship, although it may take longer to find fitting parameters in the models in the corresponding larger feasible search space.

Chapter 2 covers the application of lasso regression, neural network models and symbolic regression on water quality prediction. Their performance is evaluated on the same data set collected from a simple waste water distribution system. The pros and cons of each approach are also summarized in that Chapter. Three case studies are presented to show the idea of approximating water system dynamics using machine learning.

1.4 Water Quality Control with Deterministic Demand

The optimal control of water distribution system can be divided into two categories, depending on whether the system dynamics can be represented with a set of physics-based equations, as mentioned earlier, or data-based models. For the former category, Cembrano *et al.* (2000) formulated the optimal operation of pumps and valves as a nonlinear optimization problem, and solved it using a gradient descent

approach. In their formulation, the dynamic model of the system is derived from the mass and energy balance in the water network. Sakarya and Mays (2000) also formulated the operation of pumps as a nonlinear optimization problem, but they included constraints for substance concentration limits in their model. They used a simulation tool EPANET (Rossman, 1999) combined with GRG2 (a FORTRAN nonlinear program solver, see Lasdon and Waren (1986)) in their solution approach. Their dynamic model of the system was also based on the conservation of mass and energy. In latter cases where the system dynamic model cannot be derived based on physics laws but rather on empirical data, many researchers have used genetic algorithms to help solve the associated optimal control problem for water quality management (Chen and Ni-Bin, 1998; Cho *et al.*, 2004; Dhar and Datta, 2008; Kuo *et al.*, 2006; Kurek and Ostfeld, 2012; Tu *et al.*, 2005), which are usually integrated with some simulation models. There are also researchers who use machine learning to build an approximation model of the system dynamics, instead of using simulation models. Kuo *et al.* (2006) built a neural network model to approximate water quality, and they used a genetic algorithm to solve the optimal control of nutrient loads from the watershed. Qiao *et al.* (2013) also used a neural network model to approximate the wastewater treatment plant, but then used another neural network model to learn the optimal control. Zhou *et al.* (2015) proposed a method called convex piece-wise linear machine for the optimal control problems. Although the purpose of their classifier is not to approximate the system dynamics, it is actually used to generate a set of linear inequalities to approximate the feasible control region. They showed their method is promising in a case study of HVAC energy consumption.

Flushing and disinfectant injections are two common water quality control actions (Lautenschlager *et al.*, 2010; Bédard *et al.*, 2018; Proctor *et al.*, 2020; Hozalski *et al.*, 2020; Islam *et al.*, 2017). Flushing is effective because it quickly brings in fresher

water directly from water treatment plant via a main pipe, however, it could cause large amounts of water to be wasted. Adding disinfectants, for example, chlorine, is also an effective control action since it could significantly decrease or stop bacteria growth and kills most bacteria, parasites and viruses. But it is difficult to control the amount of disinfectants to be added, too much of them is also harmful. For example, when chlorine residual exceeds 2.0 mg/L, it will cause water to taste bad, according to WHO guidelines for drinking water quality (WHO, 2003). Higher concentration of disinfectants will also lead to more disinfection by-products, some of which can lead to cancer, according to CDC guidelines (CDC, 2016) and Islam *et al.* (2017).

When water demand is deterministic, then it is possible to exactly predict the demand in future time steps from historic data and current quality information by a machine learning approach, hence a proactive control policy can be achieved. Islam *et al.* (2017) used simulation to predict chlorine residual and solved the problem of location chlorine boost injection points using a maximum covering heuristic. Kang and Lansey (2010) assumed demand forecast and modeled the operations of valves and chlorine injections as an unconstrained problem with penalty costs, the problem is then solved using a genetic algorithm. Wu *et al.* (2015a) developed a model predictive control system, in which they used artificial neural networks to predict total chlorine and free ammonia levels, and used the predictions in a genetic algorithm to solve for the optimal ammonia dosing rates at the pump station in order to maintain chlorine and ammonia at desired levels.

It is not straightforward to include multiple control measures in formulating the problem of optimal control of water quality, since all of them affect water quality. Proper assumptions need to be made in terms how they are related. Chapter 2 covers the optimal control of water age via flushing using linear models to predict water age, the idea is tested on two different water distribution networks (one for a multi-story

building, one for a city). Chapter 3 addresses the optimal control of effective water age through both flushing and chlorine injection, where chlorine injection is assumed to decrease effective water age. Now water age prediction model becomes a nonlinear one because both flushing and chlorine reduce effective water age nonlinearly. Chapter 3 also studies different chlorine injection controls - binary vs continuous. The modeling of flushing and chlorine injection is applied on a city water distribution network.

1.5 Water Quality Control with Stochastic Demand

When water demand is stochastic, it is difficult to predict the demand in future time steps. A different perspective has been used in this research for prediction and modeling the water quality control.

The aforementioned approximate dynamic programming (ADP) is a promising approach that can help solve such decision-making problems under uncertainty. Because the demand comes from an unknown distribution at every time step, developing exact optimal control actions in advance is not realistic. A more appropriate policy consists of a strategy that learns from the past realized demand and its interaction with the water distribution system. Here the water distribution and quality control systems can be modeled as a Markov decision process, since the system state (e.g., water quality) at next time step only depends on the current demand, system state, and the performed control actions (e.g., flushing). In the ADP approach, the optimal policy is derived by iteratively updating the value function, which is the total expected reward (or cost) over future time steps (till the end of decision time horizon) starting from current state, until the value function converges. However, since the "water quality state" of a parcel of water is usually of continuous value in time and space, it is impossible to store a specific value for every value of the state. Therefore, the approach to be developed must be able to make some predictions for the states it

has not observed, and the value function should be approximated as a parameterized function of the state.

Castelletti *et al.* (2010) formulated the optimal reservoir operation as a standard ADP problem, and used a tree-based algorithm to estimate the value function for grid-discretized states. The policy maps lake storage level to release decision. Lee and Labadie (2007) studied the operations of two river systems and applied Q-learning in ADP to derive the optimal control policy after discretizing the state into grids. Castelletti *et al.* (2014) adopted a batch-mode reinforcement learning algorithm (fitted Q-iteration) in designing the operation of a water reservoir in Japan with water quality targets. Bhattacharya *et al.* (2003) presented a study of controlling the pump operations of a complex water system in Netherlands using artificial neural network based Q-learning.

As mentioned above, continuous water quality state poses a major computational challenge. Discretization is possible, but when the state vector is high dimensional, the state space after discretization may be too large to be handled, even by today's high performance computers. A parameterized function approximation could also be useful in handling continuous state, but it is nontrivial to develop such a model. There are difficulties like how to encode the state, what base model to choose, etc. that need to be addressed. Besides the continuous state issue, there are also different approaches to update the value function, each with its own pros and cons.

Chapter 4 covers the optimal control of water age via flushing when water demand is stochastic. It presents the problem formulation with water age limit restrictions. The action value functions are estimated using a linear model with approximate kernels. Multiple experiments are run with different hyper-parameters and different learning algorithms using the case of a city water distribution network. Chapter 4 essentially presents a framework and an approach for developing the optimal control

policy for any given water distribution systems that uses a learned policy for real time decision making on control actions (e.g., flushing) based on past and current controls and observed sensor data.

Chapter 2

LINEAR PROGRAMMING BASED OPTIMAL CONTROL OF WATER QUALITY

The schematic relationship of proactive and real-time control of water quality in water distribution systems can be illustrated using Figure 2.1. The state of the system $\boldsymbol{x}(z,t)$ over space and time is generated from the real-world or simulated system. The associated data can be observed and collected through measurement sensors. The collected data can then be used to build data-driven estimation model for the system dynamics (since exact mathematical representation is not available). The built estimation models can predict the system state (e.g., water quality, and $\tilde{\boldsymbol{x}}(z,t)$ represents the predicted value) when used in the optimal control model. The proactive and real-time control module contains the optimization model for deriving the optimal control actions. We will first present linear programming based proactive optimal control models in this chapter, and develop nonlinear programming based optimal control models in Chapter 3. A real-time optimal control model will be presented in Chapter 4.

This chapter is divided into two main sections. Section 2.1 is focused on how to approximate the water system dynamics using different machine learning approaches. The approach of approximating system dynamics has long been adopted in the research of water distribution systems, as discussed in Section 1.3. However, the challenge is that there are so many different approximation approaches that can be used, and there is no single one that can be applied to all scenarios. On top of that challenge, it is also a difficult decision to select what states of the system to be used in the approximation model. We will cover three different machine learning approaches

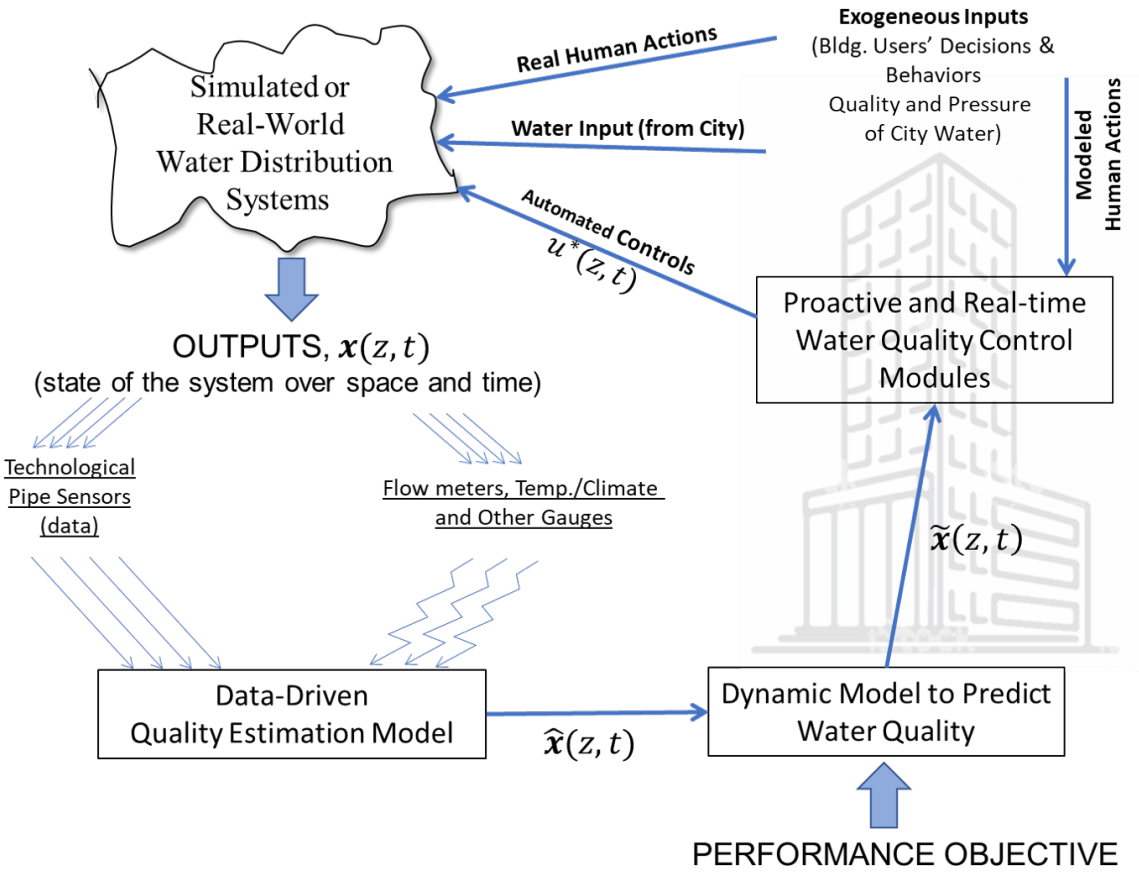


Figure 2.1: The Schematic Diagram of Proactive and Real-time Control of Water Quality in Water Distribution Systems

in Section 2.1, and present three case studies to explain how the system dynamics approximation can be done using such approaches. Section 2.2 addresses the problem of optimal water age control through flushing of taps. Water age is an indicator of water stagnation, and, as aforementioned, stagnation can cause significant water quality degradation. Flushing can quickly restore water quality to the safe level, but it also causes significant water waste. In Section 2.2, we develop a semi-closed loop cost-aware control model for water quality, in which linear prediction models are incorporated in the formulation of min-cost flushing control. The semi-closed loop control model will be evaluated using two different water distribution networks,

where the flushing control actions are applied to the simulation model to evaluate their effectiveness.

2.1 Water Quality Prediction in Water Distribution Systems

When we consider discrete time steps, the state equation (1.2) in the general optimal control problem ((1.1) $\sim$ (1.4)) is equivalent to

$$\boldsymbol{x}(t+1) - \boldsymbol{x}(t) = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t)] \tag{2.1}$$

This indicates that $\boldsymbol{x}(t+1)$ can be expressed as a function of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. This relationship between $\boldsymbol{x}(t+1)$ and $\boldsymbol{x}(t)$ hints that statistical machine learning approaches can be used to approximate equation (2.1) since the exact mathematical expression for $\boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t)]$ cannot be derived.

2.1.1 Prediction Methods

Different machine learning approaches have been applied successfully by researchers in predicting water system states (including water quality), as discussed in Section 1.3. In this section, we briefly explain each of the machine learning approaches we have explored in predicting water quality, and present three cases studies in which they were successfully applied in the prediction of water quality with satisfactory accuracy.

2.1.1.1 Lasso/Ridge Regression

Lasso regression (Tibshirani, 1996) is an approach similar to linear regression, but its advantage is that it can select significant variables by adding the L1 norm of coefficients in the objective function,

$$\min \frac{1}{n} \|X\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \alpha \|\boldsymbol{w}\|_1 \tag{2.2}$$

where X, $\boldsymbol{y}$ are the independent and dependent variables, respectively, n is the number of samples, $\boldsymbol{w}$ is the vector of estimated coefficients, and α is a predefined constant (usually called regularization parameter). If α is set zero, then Lasso regression becomes ordinary least square regression. A list of α values can be provided for iterative fitting, and the best model can be selected via cross-validation. With the L1 norm term, the coefficients of insignificant variables will be pushed to zero thus leaving $\boldsymbol{w}$ a sparse vector. This characteristic of Lasso regression makes it useful when a relatively simple model with only significant variables is preferred.

Ridge regression is similar to Lasso regression, its objective function includes the L2 norm of coefficients:

$$\min \frac{1}{n}\|X\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \alpha\|\boldsymbol{w}\|_2^2 \tag{2.3}$$

The L2 norm term makes Ridge regression useful to handle multicollinearity that causes problem for linear regression.

Both Lasso regression and Ridge regression have an extra term in their objective function, this extra term is actually a penalty on the coefficients. It is this penalty term that gives them extra functionality and enables them to generate better models than linear regression.

2.1.1.2 Neural Networks

Feed-forward neural network models can approximate any continuous functions according to the *universal approximation theorem* (Hornik, 1991). They can also handle multiple outputs very easily using only one network. It can be useful when multiple predictions (e.g., predictions at multiple time steps in the future) are needed. One issue that often arises in machine learning is overfitting. To prevent overfitting in neural network models, the L2 norm of the weights W (parameters of neural networks)

is added in the cost function (objective function),

$$Loss(\hat{\boldsymbol{y}}, \boldsymbol{y}, W) = \frac{1}{2}\|\hat{\boldsymbol{y}} - \boldsymbol{y}\| + \frac{\alpha}{2}\|W\|_2^2 \tag{2.4}$$

where $\boldsymbol{y}$ is the actual independent variable to predict, and $\hat{\boldsymbol{y}}$ is the prediction. This is similar to Lasso regression and Ridge regression, as they all have a penalty of the coefficients/weights in the objective function. With this penalty term, neural network models tend to generalize well. Note that there is also a parameter α (called regularization parameter) in the cost function, the best value of α can be set via grid search (an exhaustive search and cross-validation method). Back-propagation (LeCun *et al.*, 1989) is usually used the calculate the optimal weights.

2.1.1.3 Symbolic Regression

Symbolic regression is a regression approach that can find the best fit model without the need of specifying the model structure. It is a stochastic evolutionary approach that generates initial expressions by randomly combining mathematical building blocks (independent variables, constants, mathematical operators, analytical functions, etc.). From iteration to iteration (the evolution process), genetic programming is used to generate new expressions via various genetic operators such as crossover and mutation.

The mathematical expression is encoded as a tree structure and can be evaluated easily in a recursive manner. An example of the encoding is shown in Figure 2.2. Each internal node represents an operator function, and each leaf node represents an operand.

A set of functions is main input to symbolic regression (other than independent variables), the functions are used to build a mathematical expression. And the model complexity can be controlled by restricting the tree depth and the function set.

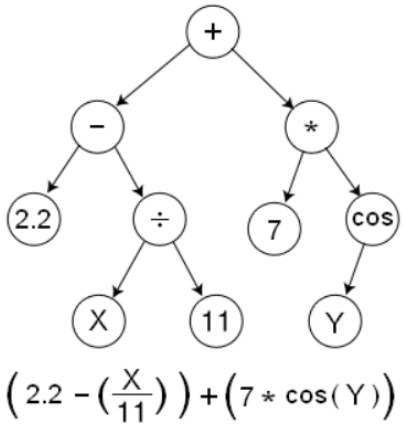


Figure 2.2: An Example of Encoding Mathematical Expression as a Tree (from Wikipedia)

Symbolic regression is useful when not much information about the potential model is known; it is a powerful approach that can build a variety of models. The biggest advantage of symbolic regression is it does not rely on assumptions on the form of the underlying model. However, the advantage comes at the cost of more searching time as it searches in a much larger feasible space for the best fit model(s). And manual tuning is necessary in order to achieve a model with high accuracy and relatively simple form.

2.1.2 Case Study I: pH Prediction in a Wastewater System

A physical wastewater piping system was built by collaborators for real experiments in this case study (Saetta *et al.*, 2019). The physical wastewater piping system contains three identical urinals, with sensors installed in each of them sampling the pH values and conductivity every 15 seconds. The layout for one urinal in the piping system is shown in Figure 2.3. The artificial urination events are set to occur randomly. Two experiments that lasted over four hours with different urination frequency were implemented. In the first experiment, the time between two consecutive

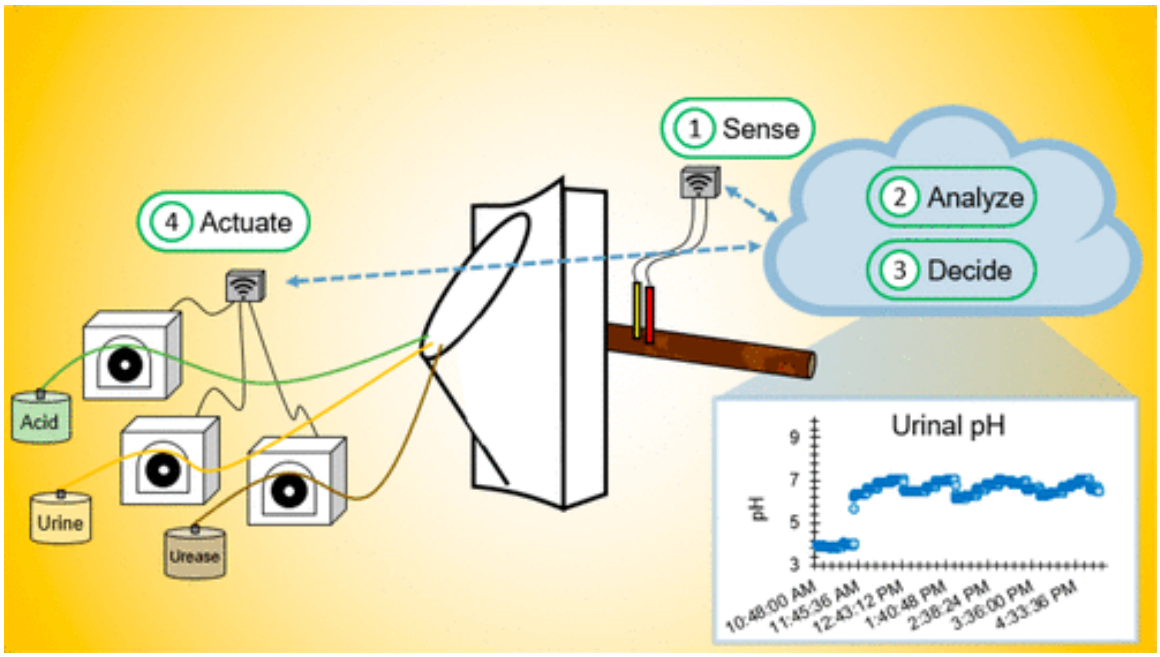


Figure 2.3: The Wastewater Piping System Layout (Only One Urinal Is Displayed, from Saetta *et al.* (2019))

urination events is generated from a continuous uniform distribution from 1 to 10 minutes. In the second experiment, the uniform distribution ranges from 10 to 20 minutes. The frequency of artificial urination events was set up this way to simulate a busy and a not-so-busy restroom, respectively. The trigger time of artificial urination events and the amount of (artificial) urine in each urination event are also recorded. More details about the experiments and data can be found in Saetta *et al.* (2019).

The goal of these experiments is to control the wastewater quality within the pipe system, where the pH is an indicator of the system acidity and bacteria growth relies on a proper range of pH values. The control action is to pump acid into the system based on sensor measurements of pH, or based on the predicted values of pH. Thus, we need models that can predict pH values ahead of time so that the control action (adding acid) can be done proactively.

Sensor data is read every 15 seconds, and we want to predict pH values in the near future (four time-steps ahead, equivalent to 1 minute), using historic pH and conductivity data (collected from sensor readings) and the recorded urination information from both experiments. Twenty historic data points (equivalent to 5-minute history) of both pH and conductivity, plus the time lag from last urination event and the amount of urine in last urination event are used as the input to the model. The mathematical expression of the model was $\boldsymbol{y} = f(\boldsymbol{x})$, where $\boldsymbol{y}$ is the predicted pH value vector of dimension 4, and $\boldsymbol{x}$ is the input variable vector of dimension 42, with x_0 - x_{19} denoting the historic pH, x_{20} - x_{39} denoting the historic conductivity, x_{40} - x_{41} denoting the time lag from last urination event and the amount of urine in last urination event, respectively. The actual form of function f depends on the machine learning approach, it could be either linear or nonlinear in terms of the input variable vector $\boldsymbol{x}$.

Each experiment was run for about four hours in the lab. In total, 6,700 data records were collected from these two experiments. They were randomly split into training and testing sets with 70/30 ratio. The data was also standardized to center around zero with unit variance in the pre-processing step; this was to avoid the prediction to be dominated by conductivity since conductivity has a much higher magnitude than other variables.

The scikit-learn Python package (Pedregosa *et al.*, 2011) was used to build Lasso regression and neural network models for the prediction. The symbolic regression models were built using MATLAB code (Searson, 2009).

Four Lasso regression models were built to predict pH values in the next minute, one prediction for one time step, since each Lasso regression model makes only one prediction. The accuracy of Lasso regression models in terms of R^2 and root mean squared error (RMSE) on training and testing data are shown in Figure 2.4. Overall,

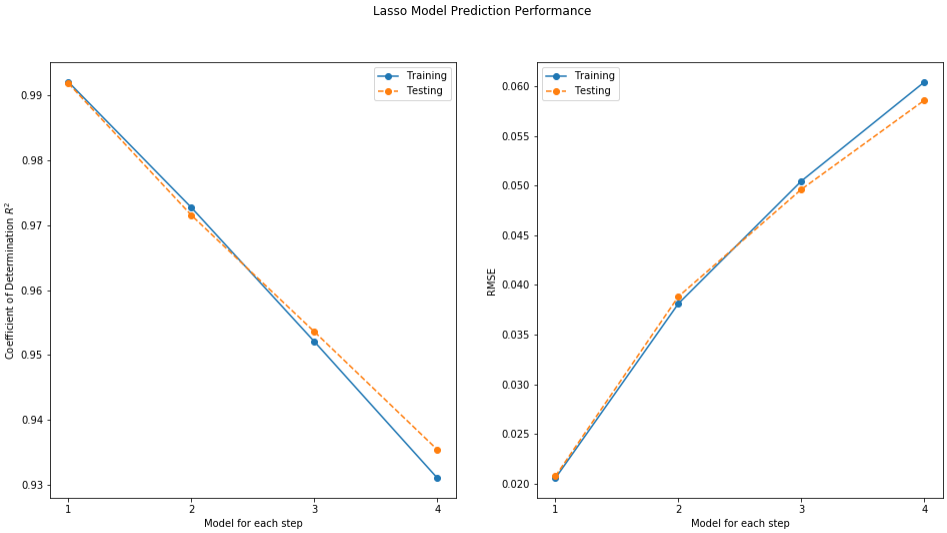


Figure 2.4: Performance of the Lasso Models

the accuracy is decreasing in as the time step increases, and RMSE is increasing. However, even for the fourth-step prediction, the R^2 is close to 0.93, which is considered quite good. The associated RMSE is only 0.06 for pH values ranging from 3.8 to 8.5. The coefficients of Lasso regression models are shown in Table 2.1. Notice that the coefficients of many variables are zero, and this is the advantage of Lasso regression explained earlier. After data standardization, the coefficient can be view as the importance of the corresponding variable. During the fitting process, Lasso regression automatically pushes the coefficients of variables with less prediction power to be zero. In fact, the coefficients for all conductivity variables (bold blue colored indices in the table) are zero, indicating they are not helpful in pH predictions. Another important finding can be noted from the coefficients table is that variable x_{19} (the most recent pH value) has the largest absolute value in all four models, indicating it has the largest impact on future pH predictions. This observation makes sense and tells us to rely on most recent information to make predictions for the future.

For neural networks, since it can predict multiple values using one network struc-

Table 2.1: Coefficients for the Lasso Models

Variable Index	Coefficients				Variable Index	Coefficients			
	Model 1	Model 2	Model 3	Model 4		Model 1	Model 2	Model 3	Model 4
0	0	0	0	-0.00182	20	0	0	0	0
1	-9E-05	0	0	0	21	0	0	0	0
2	0	-0.00099	-0.00192	0	22	0	0	0	0
3	0	0	-0.00042	-0.00099	23	0	0	0	0
4	0	0	0	-0.00076	24	0	0	0	0
5	-0.00077	-0.00148	-0.00112	-0.00078	25	0	0	0	0
6	0	0	0	0	26	0	0	0	0
7	0	0	0	0	27	0	0	0	0
8	0	0	0	0	28	0	0	0	0
9	0	0	0	0	29	0	0	0	0
10	0	-3.7E-05	0	0	30	0	0	0	0
11	0	-0.00108	0	0	31	0	0	0	0
12	-7.7E-05	-0.00148	-0.00498	-0.00551	32	0	0	0	0
13	-0.0036	0	-0.00172	-0.00557	33	0	0	0	0
14	0	0	0	0	34	0	0	0	0
15	0	0	0	0	35	0	0	0	0
16	0	0	0	0	36	0	0	0	0
17	0	0	0	0	37	0	0	0	0
18	-0.04841	0	0	0	38	0	0	0	0
19	0.282166	0.231048	0.23273	0.234536	39	0	0	0	0
*red = pH, blue = conductivity, black = urination					40	0.000482	0.000296	0.000719	0.000919
					41	-0.00059	-0.0011	-0.00185	-0.00265

ture, single model is enough to make predictions for four pH values. The network structure we used contains one input layer, one hidden layer with 10 hidden nodes and one output layer. The rectified linear unit function, $f(x) = \max(0, x)$, was used as the activation function for the hidden layer. There are four output nodes in the output layer since four pH values in future need to be predicted. The identity activation function for the output layer is used because this is a regression problem. A quasi-Newton type optimizer (L-BFGS) is used to numerically optimize the weights with constant learning rate 0.001 and maximum number of iterations 200. The grid search method is used to find the best regularization parameter α in the loss function (2.4) from $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$. The performance is shown in Figure 2.5. The

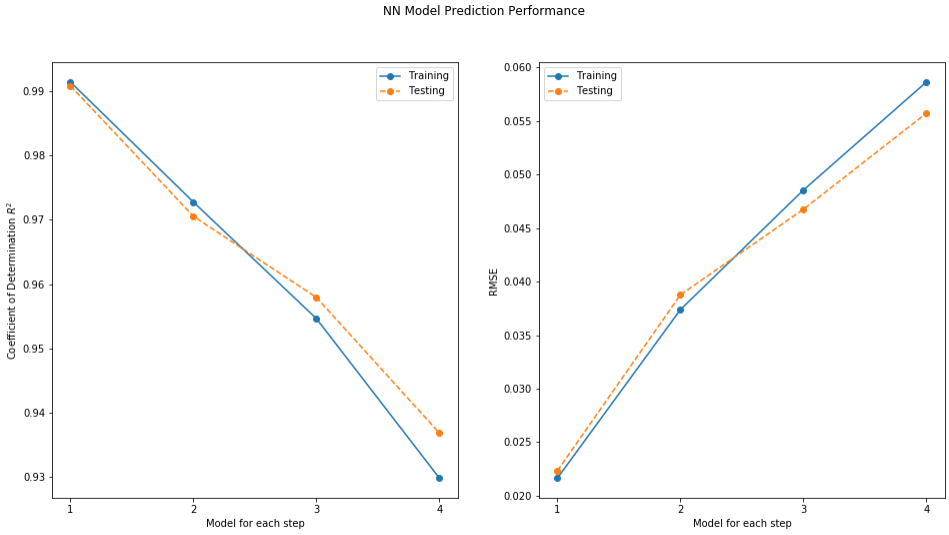


Figure 2.5: Performance of the Neural Network Model

accuracy (both R^2 and RMSE) is very similar to that of Lasso models.

For symbolic regression (SR), only the basic function set $\{+, -, \times, \div\}$ (other than the input variables) was provided for the model development, and the underlying genetic programming procedure was run for 100 iterations with 50 individuals (mathematical expressions) in each iteration. At the end of the algorithm, a set of candidate expressions (prediction models) are drawn in one Pareto plot. We selected models with relative low complexity and high accuracy that are not dominated by others. The final selected models are shown in Table 2.2 and their performances are shown in Figure 2.6. The performance is comparable with that of Lasso regression and neural network models. Notice that most recent pH values (x_{18}, x_{19}) are used in all four models. This confirms our previous observation that future pH values rely on most recent pH values.

In summary, all three approaches can achieve comparable prediction accuracy. Models from all three approaches were evaluated on the same data set, the performance is shown in Figure 2.7. However, the training time differs quite a bit. Symbolic

Table 2.2: Selected Models from Symbolic Regression

Step Number	Selected Model
1	$6.655 - 0.1130x_{18} + 0.3409x_{19}$
2	$6.655 - 0.1377x_{18} + 0.3647x_{19}$
3	$6.655 - 0.1469x_{18} + 0.3710x_{19}$
4	$6.656 - 0.1582x_{18} + 0.3793x_{19}$

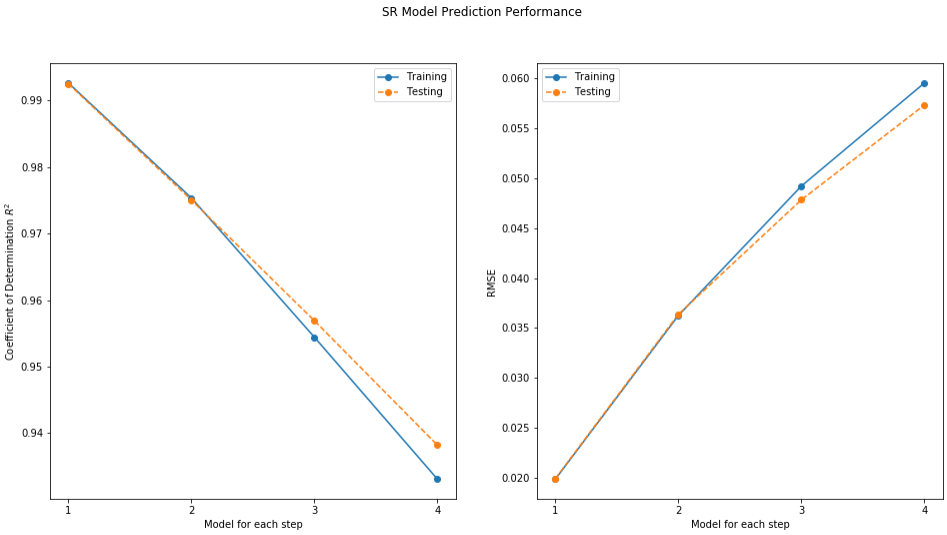


Figure 2.6: Performance of the Symbolic Regression Models

regression took the most time (∼60s), Lasso regression took the least computational time (∼1s), training time for the neural network model was in between (∼20s). This training time difference was observed from a data set that is not too large, the difference will be more with larger data sets. As for model complexity, both Lasso regression and symbolic regression generate linear models, where manual tuning is necessary to obtain linear models from symbolic regression. Both Lasso regression and symbolic regression can be useful in the optimal control of water quality, because the linearity of prediction models makes the solution method for the optimal control

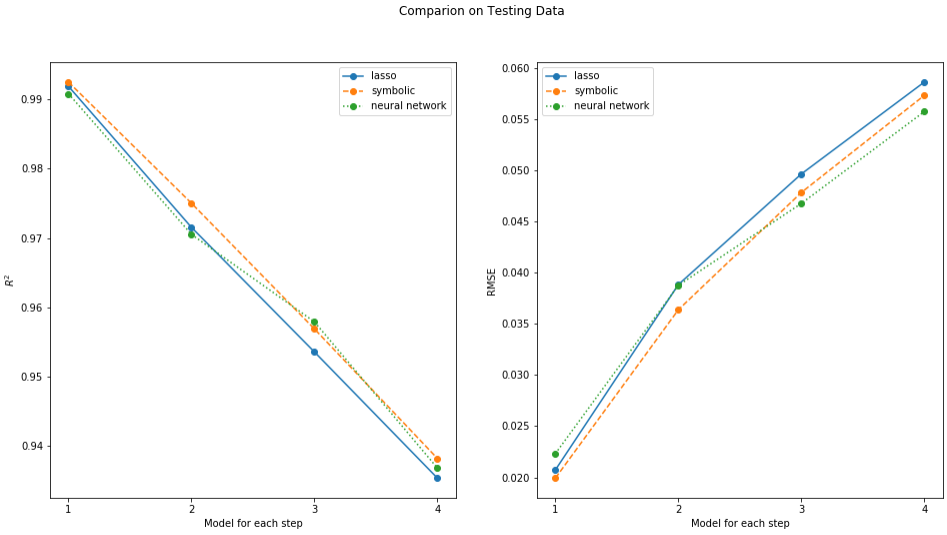


Figure 2.7: The Accuracy of Models from Different Approaches on the Same Test Data

problem simpler. Although the neural network model is more complex, it is still possible to incorporate it in the optimal control method. Actually, such approaches have led a large research area called *neuro-dynamic programming* (Bertsekas and Tsitsiklis, 1996), where neural network models are built to approximate DP's value functions and gradually learn the optimal control policy.

2.1.3 Case Study II: Water Age Prediction in a Building Water System

In this section, a set of linear models are developed to approximate the water age dynamics in a building water distribution system. If the linear models can approximate the water age transition with high accuracy, then state equation (1.2) can be replaced approximately by a set of linear constraints. This approximation does not introduce extra complexity in solving the optimal control problem. Furthermore, if the objective function can be formulated with a linear expression, then the whole control problem becomes a linear program, which will be shown section 2.2.

In this case study, a simulation model for a five-story university building was first built using water distribution systems modeling software EPANET (Rossman, 1999). The water distribution network is shown in Figure 2.8. There are two demand locations on each floor, labeled as Kitchen and Fountain. The demand at different locations can be different, and they may also have different patterns over time. The demand patterns were assumed to be estimated from historic data which are known to the model. In the simulations used in the experiments, for simplicity, demands at the Kitchen and the Fountain on the same floor were assumed to be same, although any arbitrary patterns could be assumed. The demand pattern used for different floors is shown in Figure 2.9.

Our ultimate goal is to model and solve the optimal water age control problem for a water distribution system, and the challenge is how to derive a constraint to describe the system dynamics (equation (1.2), or its discrete time version (2.1)). Inspired by the state space representation of a physical system from control theory

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \tag{2.5}$$

where $\boldsymbol{x}(t)$ represents system state, $\boldsymbol{u}(t)$ is the control input to the system, A, B are called state matrix and input matrix respectively; we assume the water age transition in the building water network (Figure 2.8) can be approximated as follows (using discrete one-hour time step):

$$\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) + \boldsymbol{c} \tag{2.6}$$

where $\boldsymbol{x}(t)$ is the state variable representing water age at hour t, $\boldsymbol{u}(t)$ is the control variable representing the amount of water to be flushed at hour t. Both water age and flushing amount are measured at all 10 demand locations, so $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ are vectors of dimension 10. $A, B, \boldsymbol{c}$ are parameters that we need to estimate, A, B are matrices of dimension 10×10, $\boldsymbol{c}$ is a vector of dimension 10.

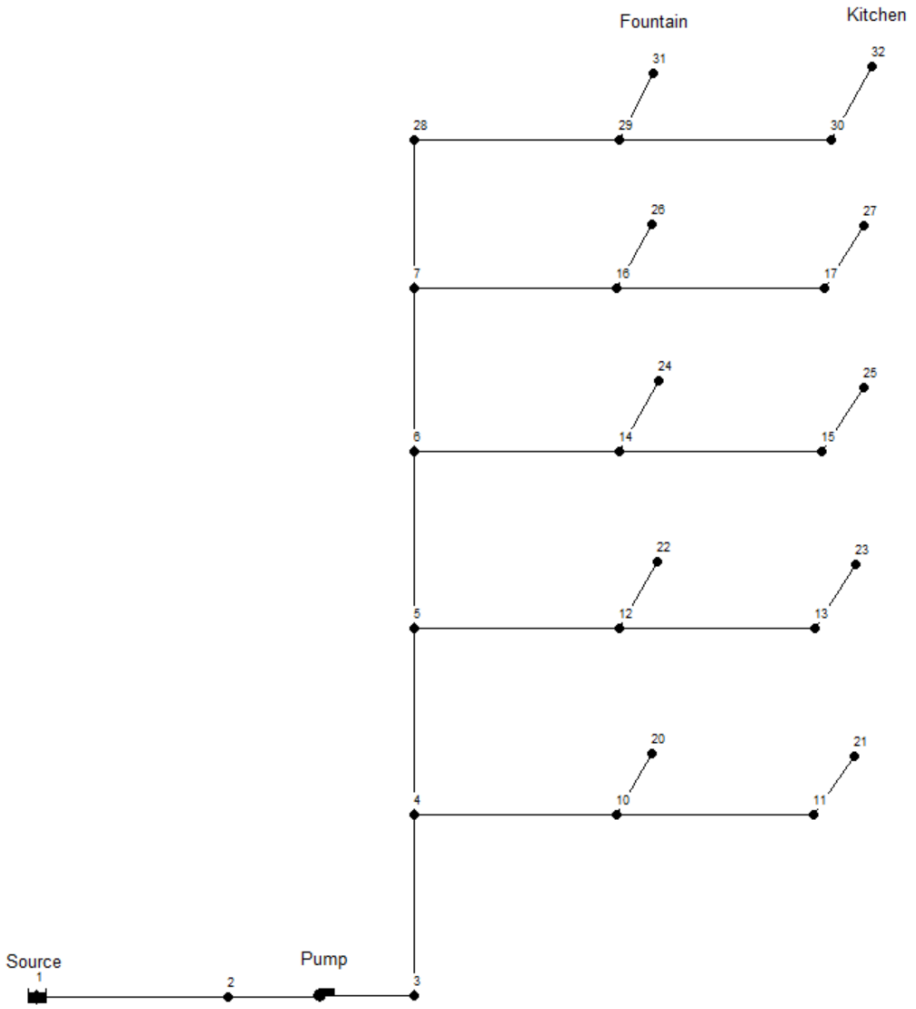


Figure 2.8: EPANET Simulation Model for a 5-story University Building

The simulation was run 1000 times with random flushing (to emulate random water demand) at all locations, to generate some training data. For each simulation, the random amount of water to be flushed is set as β times demand at that location, where β is generated from a discrete uniform distribution of $\{0, 0.5, 1\}$. In total, there are $3^{10} = 59049$ random scenarios, and the 1000 simulations are just a small percentage of them, so we are not enumerating every single scenario. Figure 2.10 shows water age (in hours) at all locations at different times from one of the 1000 simulations. The red vertical dotted line indicates the end of warm-up period for the

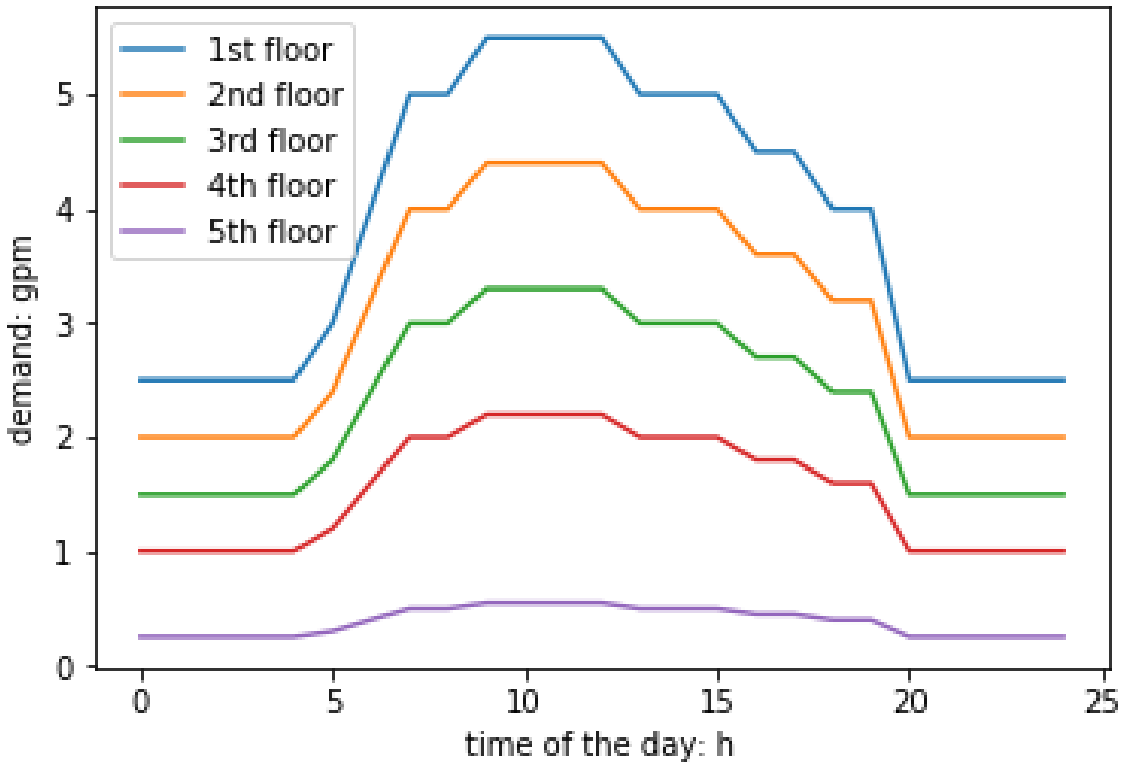


Figure 2.9: The Assumed Demand Patterns on Different Floors

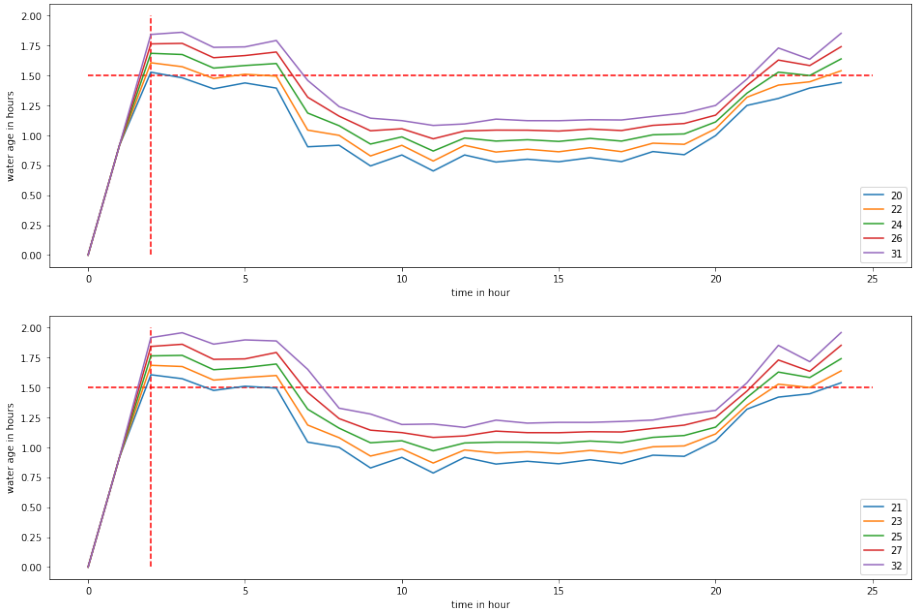


Figure 2.10: Water Age at All Locations from One of 1000 Simulations

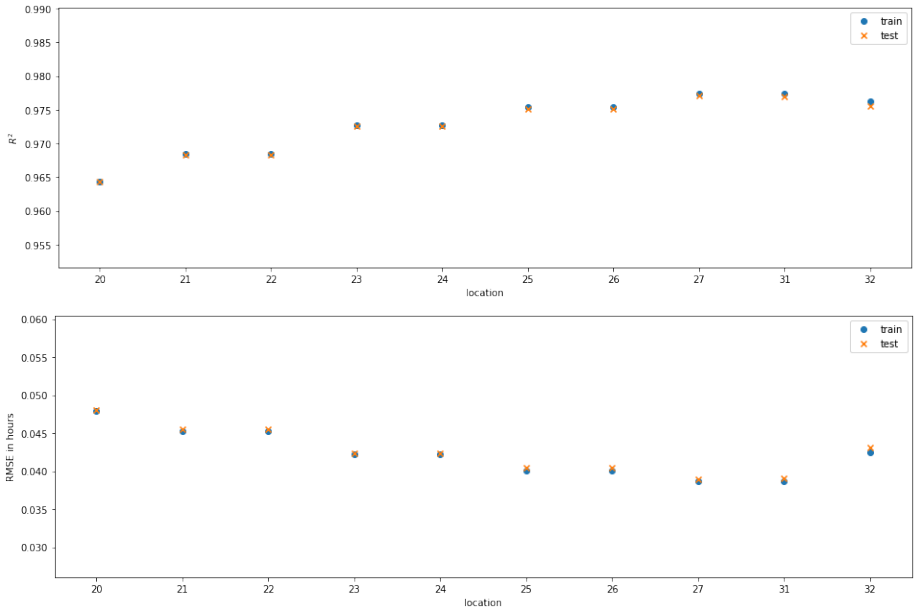


Figure 2.11: The Accuracy of Ridge Regression Models in Training and Testing Data

simulation model, and the red horizontal dotted line shows the constraint on water age limit.

After data was generated from simulation, it was randomly split into training and testing data sets with 70/30 ratio. Then ridge regression was used to estimate parameters $A, B, \boldsymbol{c}$. These are actually coefficients coming from 10 linear models (each location need one model for its water age prediction); that is,

$$x_i(t+1) = \Sigma_{j=0}^{9} a_{ij} x_j(t) + \Sigma_{j=0}^{9} b_{ij} u_j(t) + c_i, \forall i \in \{0, 1, ..., 9\} \tag{2.7}$$

where $x_i(t)$ is water age at location i at time t, $u_j(t)$ is the flushing control at location j at time t, and $a_{i,\cdot}, b_{i,\cdot}, c_i$ are the coefficients of ridge regression model for location i.

The accuracy (R^2 and RMSE) of ridge regression models on training and testing data is shown in Figure 2.11. Models are consistently accurate at all locations; all R^2's are above 96%. Meanwhile, the accuracy in testing data is very close to that in training data, indicating models are not overfitting.

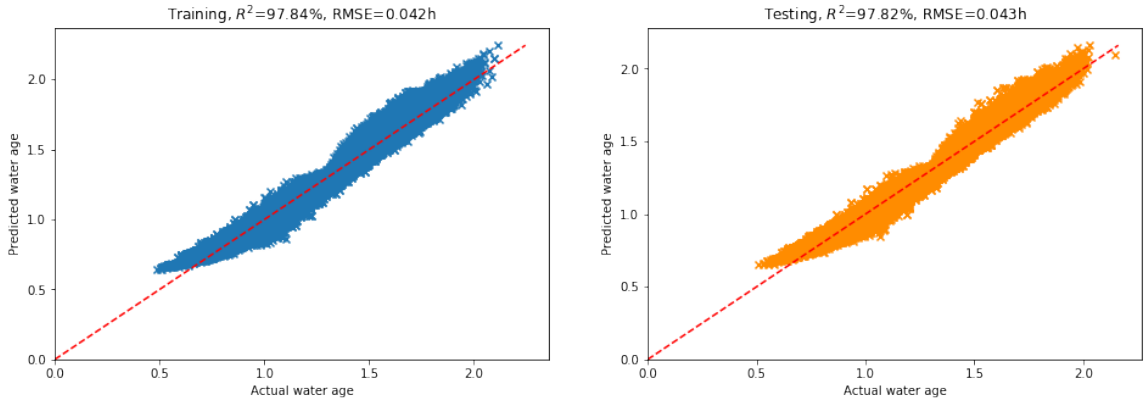


Figure 2.12: The Overall Accuracy (R^2 and RMSE) of Ridge Regression Models on Training and Testing Data

The overall R^2 for training and testing data is 97.84% and 97.82% respectively. The predicted water age and the actual water age are drawn in Figure 2.12. This indicates the linear models derived using ridge regression can approximate the system dynamics very well. The models will be used in the optimal control of water age in this building water distribution system in Section 2.2.

2.1.4 Case Study III: Water Age Prediction in a City Water System

In Section 2.1.3, we presented a case study where the system dynamics can be approximated well with linear models. Although we achieved very high accuracy in the approximation with linear models, it is not guaranteed that linear models always predict well, especially when the system becomes larger and more complex. In Case Study II, only current water age and control action are used as input to the model, but for more complex water distribution systems, this may not be enough to achieve accurate predictions, and more data from past may be needed to improve the prediction accuracy. To extend what we have done in Section 2.1.3, in this case study, we tried to predict water age on the larger water distribution network as shown in

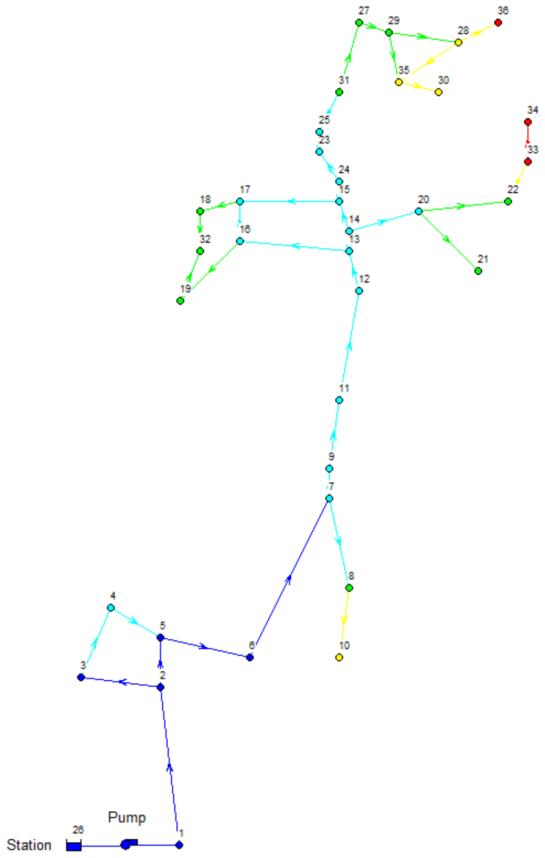


Figure 2.13: EPANET Simulation Model for a City Water Distribution Network

Figure 2.13.

This water distribution network consists of 1 reservoir, 1 pump, 35 junctions and 38 pipe segments in the network. There are 32 (out of 35) junctions with nonzero demand. The simulated duration assumed in this case study was 55 hours.

Similar to previous two case studies, our goal was to build models that can predict future system state using current and past state and control data. Same as Case Study II, water age is used as the indicator of water quality in the distribution system. The controls are amount of water to be flushed at different locations and times.

Again, EPANET (Rossman, 1999) was used to simulate water flow and water age in this city network. The simulation model was run 10,000 times with different

amounts of water in each random flushing to generate data for building the prediction model. The amount of water in each random flushing at each location was set to be β times of its largest demand, where $\beta \in \{0, 0.1, 0.2, 0.3, 0.4\}$, where $\beta = 0$ means no flushing. Each simulation represents a 55-hour operation of the water distribution system. And each location also has its own time-varying demand. After 10,000 simulations, the data was split into training and testing data sets with 70/30 ratio.

After trying different sets of input variables for the model, the models using water age, flushing controls and water demands from most recent two time steps as input achieved satisfactory accuracy:

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{x}(t-1), \boldsymbol{u}(t), \boldsymbol{u}(t-1), \boldsymbol{d}(t), \boldsymbol{d}(t-1)) \tag{2.8}$$

where $\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)$ represent water age, magnitude of flushing control (amount of water flushed) and demand at current time step t, and $t-1$ in parentheses indicates the same set of data from previous time step. State $\boldsymbol{x}(t+1)$ represents water age at next time step. All of them are denoted in bold font, indicating they are all vectors, and each of them is a vector with measurements from all junctions except nodes 2 - 6 (closer to the reservoir, water age is low).

Cross-validation based Lasso models are built for all 27 locations, with one model predicting water age at next time step for each location. The R^2's for all 27 models are plotted in Figure 2.14. Note that R^2 is above 75% for all 27 models. The predicted water age vs actual water age is plotted in Figure 2.15. These Lasso regression models will be used as the approximate water age transition models in the optimal control of water age in the next section.

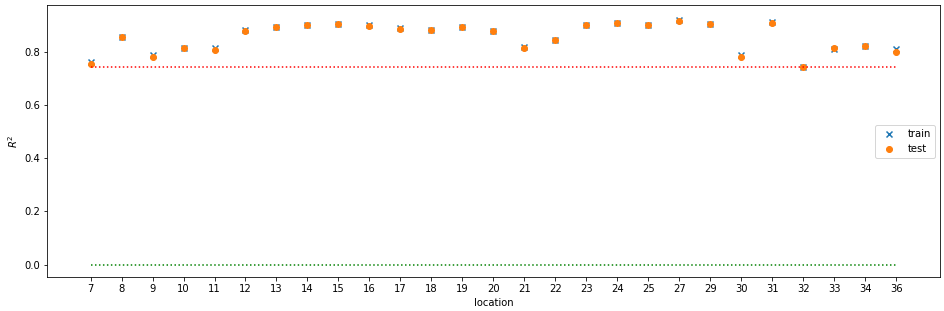


Figure 2.14: The R^2 of Lasso Regression Models for Water Age Prediction at 27 Different Locations

2.2 Optimal Water Age Control with Linear Prediction Models

In water distribution systems, water quality should be maintained at satisfactory level all the times. Water age of a parcel of water indicates how long the water parcel has been stagnant. The longer water stays stagnant, the larger water age, and the more likely bacteria start to grow. Therefore, water age was used as the indicator for water quality. Flushing has been shown effective in improving water quality, as discussed in Chapter 1. But how to quantify the relationship between flushing and water quality improvement is an open research area. In this section, we aim to provide a solution to quantify such relationship and solve for the min-cost flushing schedule using predictions from linear models. We will first present the linear programming based problem formulation, and then show some results by extending two previous case studies.

2.2.1 A Prediction-aided Linear Program Formulation of Optimal Control

Given the approximate water age transition models from previous sections, we are able to model and solve the whole water age optimal control problem as a linear

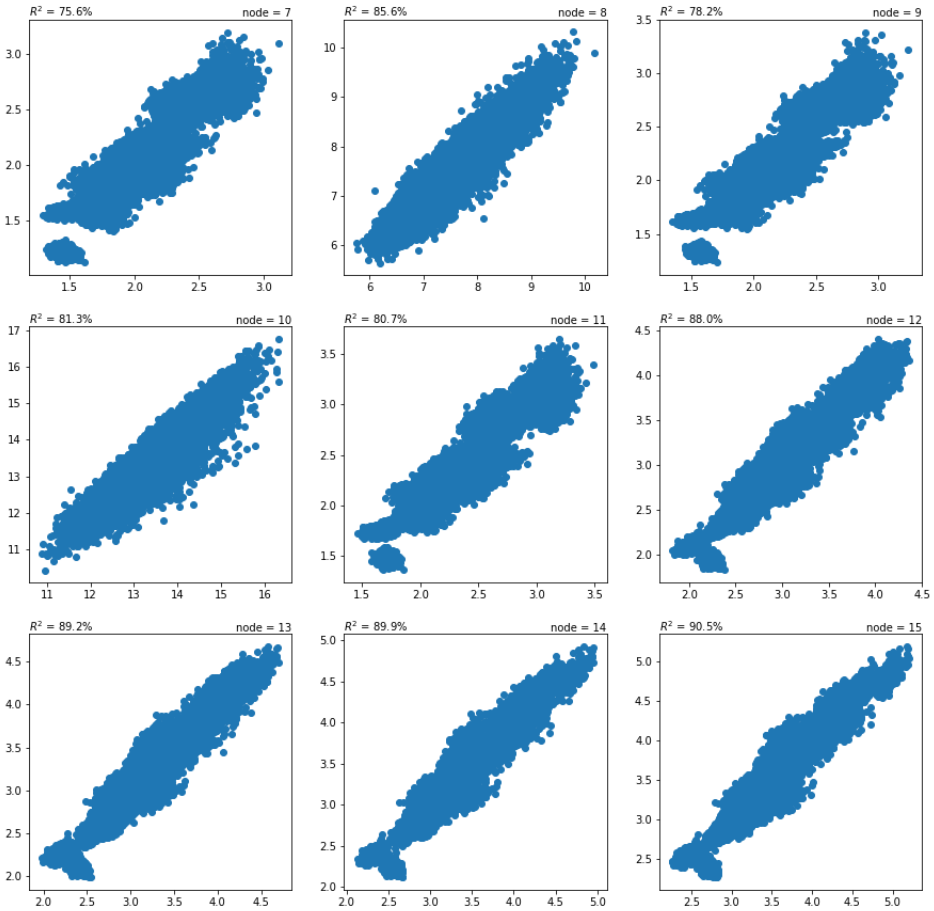


Figure 2.15: The Predicted Age vs Actual Water Age at the First 9 (Out of 27) Locations

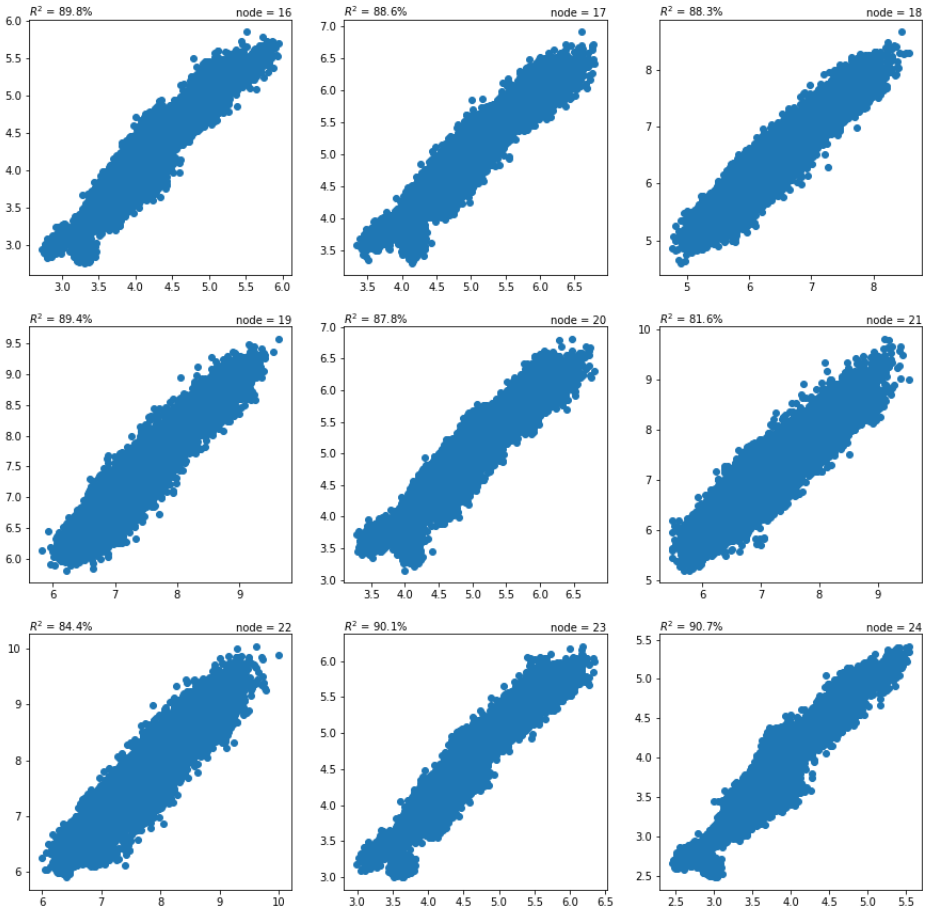


Figure 2.15: The Predicted Age vs Actual Water Age at the Second 9 (Out of 27) Locations

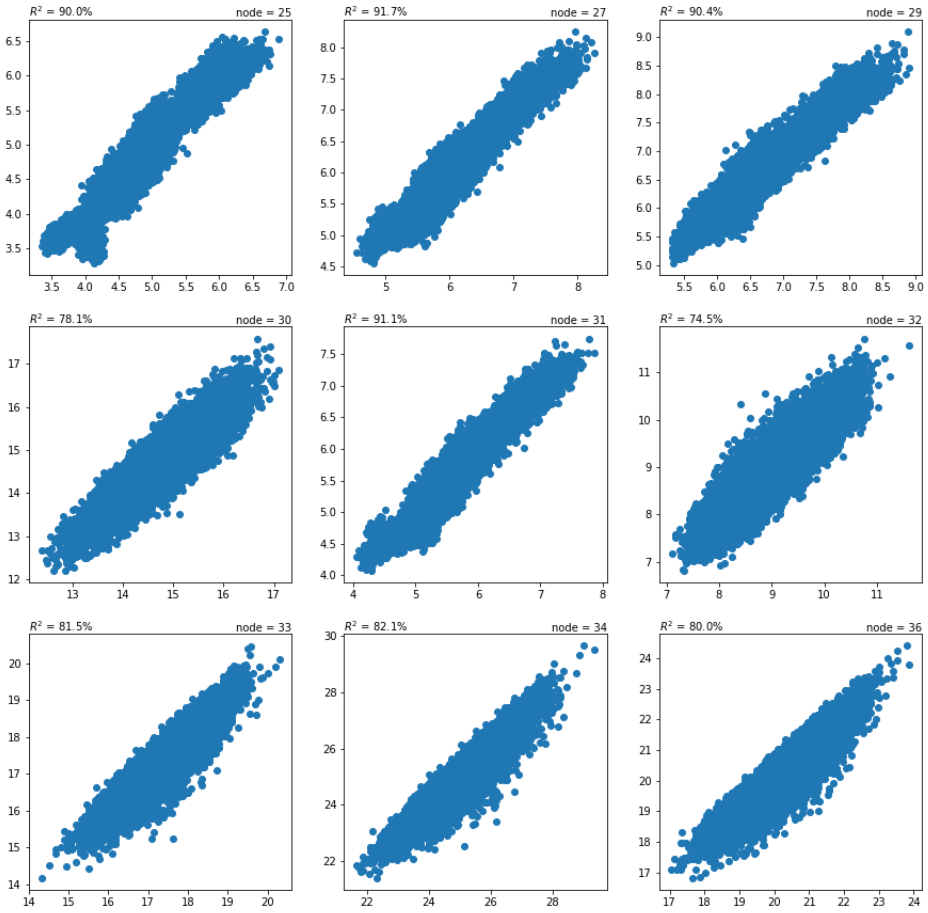


Figure 2.15: The Predicted Age vs Actual Water Age at the Third 9 (Out of 27) Locations

Table 2.3: Definition of Parameters, Indexes and Variables in the Prediction-aided Linear Program Formulation

Parameters/Indexes	Definition
T	the duration of system operation, i.e., final time step
t	index for time step, $t \in \{0, 1, 2, ...T\}$
a_l	the limit of water age
u_l	the limit of flushing control
N_s	the set of locations to measure water age
N_u	the set of locations for flushing controls
d_t	the demand at time t
Variables	**Definition**
x_t	water age at time t for all locations in N_s, $x_t \in \mathbb{R}^+$
u_t	the flushing control (amount of water to be flushed) at time t for all controlled locations in N_a, $u_t \in \mathbb{R}^+$

program (LP). The parameters and variables used in this LP formulation are defined in Table 2.3.

The optimal flushing schedule problem can be formulated as:

$$\min \Sigma_t \|\boldsymbol{u}(t)\|_1 = \Sigma_{i \in N_u} \Sigma_{t=0}^{T} u_i(t) \tag{2.9}$$

subject to

$$x_i(t+1) = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) \qquad \forall i \in N_s, \forall t \neq T \tag{2.10}$$

$$x_i(t) \leq a_l \qquad \forall i \in N_s, \forall t \tag{2.11}$$

$$u_i(t) \leq u_l \qquad \forall i \in N_u, \forall t \tag{2.12}$$

$$\boldsymbol{x}(0) = \boldsymbol{a}_0 \tag{2.13}$$

where $f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d})$ in constraint (2.10) represents a linear expression relating $\boldsymbol{x}$ and $\boldsymbol{u}$, where the time index t is omitted for convenience because the linear expression could include historic values of water and flushing control. Constraint (2.10) is the approximate water age dynamics model resulting from the water age prediction models in previous case studies. Constraints (2.11) and (2.12) represent the required limits of water age and flushing controls, respectively. Constraint (2.13) fixes the initial conditions.

When a linear prediction model can approximate the system state transition accurately enough, the whole control problem becomes a LP, and the optimal flushing schedule problem can be easily solved using off-the-shelf software. However, constraint (2.10) is an approximate relationship of the water age transition. So, as long as the prediction accuracy is not 100% (which is almost always the case for prediction models), the prediction error from these models should not be neglected in the control problem formulation. Therefore, we present an improved LP model with chance constraint to account for prediction errors.

2.2.2 Improved LP Model with Chance Constraint

Use of chance constraints is a typical method for dealing with uncertainties in solving optimization problems. In the formulation of optimal flushing schedule problem, true water age is unknown, but the prediction has a high probability to be close to the true water age, and this is the uncertainty that needs to be addressed.

In the improved LP model, constraint (2.10) is replaced with

$$P(x_i(t) \leq a_l) \geq 1 - \alpha \qquad \forall i \in N_s, \forall t \tag{2.14}$$

Chance constraint (2.14) restricts that the probability of water age $x_i(t)$ is lower than the limit has to be at least $1 - \alpha$, where α is a parameter used to specify the

level of confidence of the water age prediction.

In regression problems, the error term in prediction is usually assumed to be normally distributed with zero mean (Reid *et al.*, 2016). In constraint (2.10), water age prediction $f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d})$ is use to estimate $x_i(t+1)$, and therefore $x_i(t+1)$ also has a normal distribution. Then constraint (2.14) can be transformed as following:

$$P(x_i(t) \leq a_l) \geq 1-\alpha \iff \tag{2.15}$$

$$P(\frac{x_i(t)-\mu_{x_i(t)}}{\sigma_{x_i(t)}} \leq \frac{a_l-\mu_{x_i(t)}}{\sigma_{x_i(t)}}) \geq 1-\alpha \iff \tag{2.16}$$

$$\frac{a_l-\mu_{x_i(t)}}{\sigma_{x_i(t)}} \geq z_{\alpha/2} \iff \tag{2.17}$$

$$\frac{a_l-\hat{x_i}(t)}{\sigma_{x_i(t)}} \geq z_{\alpha/2} \tag{2.18}$$

where $\mu_{x_i(t)}, \sigma_{x_i(t)}$ are the mean and standard deviation of a normally distributed random variable $x_i(t)$, and z is a unit normal random variable with $z_{\alpha/2}$ denoting the $(1-\alpha/2)$ percentile for unit normal distribution. Thus constraint (2.14) simplifies to (2.18) or equivalently

$$\hat{x_i}(t) \leq a_l - z_{\alpha/2}\sigma_{x_i(t)} \tag{2.19}$$

The true value of the standard deviation of the error term can only be estimated from observed data. For Lasso regression models, Reid *et al.* (2016) gave a comprehensive comparison of different error variance estimators, and recommended the cross-validation based Lasso residual sum of square estimator as a good variance estimator. The mathematical expression of their recommended estimator is

$$\hat{\sigma}^2_{L,\lambda} = \frac{1}{n-\hat{s}_{L,\lambda}}\Sigma_{i=1}^{n}(y_i - X_i'\hat{\beta}_\lambda)^2 \tag{2.20}$$

where $\hat{\beta}_\lambda$ is the Lasso coefficient vector estimate, and $\hat{s}_{L,\lambda}$ is the number of nonzero elements of this vector, λ is the "regularization parameter" used in the Lasso models. This estimator is what we used for estimate $\sigma_{x_i(t)}$. The results of the optimal flushing

schedule for water age control, now considering prediction errors in case studies II and III, are extended in Section 2.2.3 and Section 2.2.4.

2.2.3 Case Study II Extension: Optimal Flushing Schedule for Water Age Control of a Building Water System Considering Prediction Errors

The system state (water age) transition model comes from Case Study II in Section 2.1. Replacing constraint (2.10) with the ridge regression models from 2.1.3, we get the following optimization problem

$$\min \Sigma_t \|\boldsymbol{u}(t)\|_1 = \Sigma_{i \in N_u} \Sigma_{t=0}^{T} u_i(t) \tag{2.21}$$

subject to

$$x_i(t+1) = \Sigma_{j=0}^{9} a_{ij} x_j(t) + \Sigma_{j=0}^{9} b_{ij} u_j(t) + c_i, \qquad \forall i \tag{2.22}$$

$$x_i(t) \leq a_l \qquad \forall i, t \tag{2.23}$$

$$u_i(t) \leq u_l \qquad \forall i, t \tag{2.24}$$

$$\boldsymbol{x}(0) = \boldsymbol{a}_0 \tag{2.25}$$

Using water age limit $a_l = 1.5h$, the above linear program is solved to obtain the optimal flushing schedule. Now constraint (2.23) is replaced with a transformed chance constraint (2.19), and the linear program is solved again. After the optimal flushing schedule is obtained, it is applied to the EPANET simulation model to evaluate how effective it is in improving water quality (water age).

With 1.5h water age limit, the percentage of time (over 24 hours simulation time) when water age is above the limit is significantly reduced by flushing, especially when the original demand is low in early morning and late night (comparing Figure 2.16 and Figure 2.17). This metric comparison is summarized in Table 2.4. The optimal flushing schedule from the improved LP model with chance constraint is shown in

Table 2.4: The Comparison of Water Age Constraints Violations (Above 1.5h) Before and After Applying Optimal Flushing Controls

	No Control	Flushing with LP Model	Flushing with Improved LP Model
% Time Water Age Constraint Violated	50.9%	20.0%	13.9%

Figure 2.18. Note that there's no flushing at locations 20 - 24, which are located on the first three floors with higher water demand, it means that the demand at those locations is high enough to keep water fresh.

2.2.4 Case Study III Extension: Optimal Flushing Schedule for Water Age Control of a City Water System Considering Prediction Errors

In the extension to Case Study III, the Lasso regression models from Section 2.1.4 were adopted as the approximate system state (water age) transition model. The model of finding the optimal flushing schedule for water age control in the city water network is formally stated as below. Chance constraint for water age limit is used directly, similarly as was with the building water system in Section 2.2.3.

$$\min \Sigma_t \|\boldsymbol{u}(t)\|_1 = \Sigma_{i \in N_u} \Sigma_{t=0}^{T} u_i(t) \tag{2.26}$$

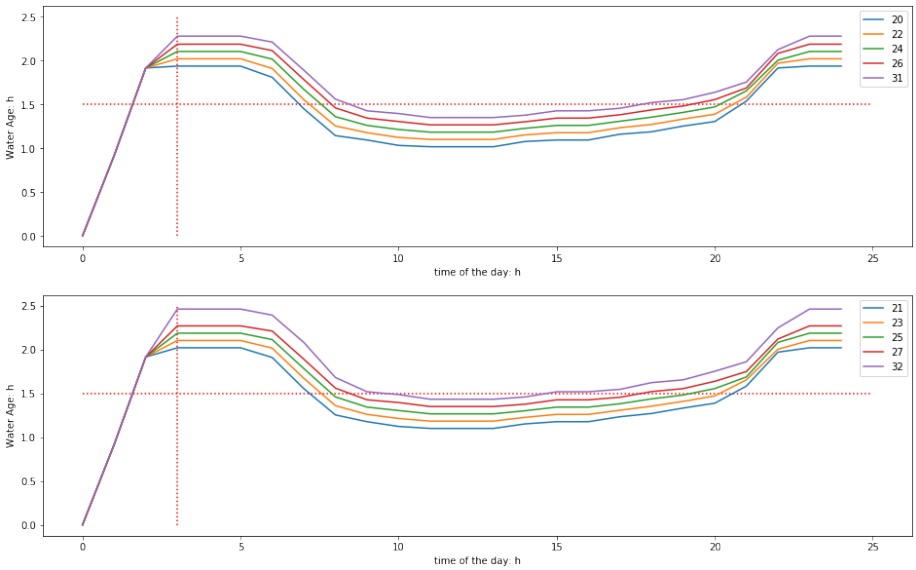


Figure 2.16: Water Age at 10 Different Locations (Five Shown in Each Plot) When No Controls Are Applied

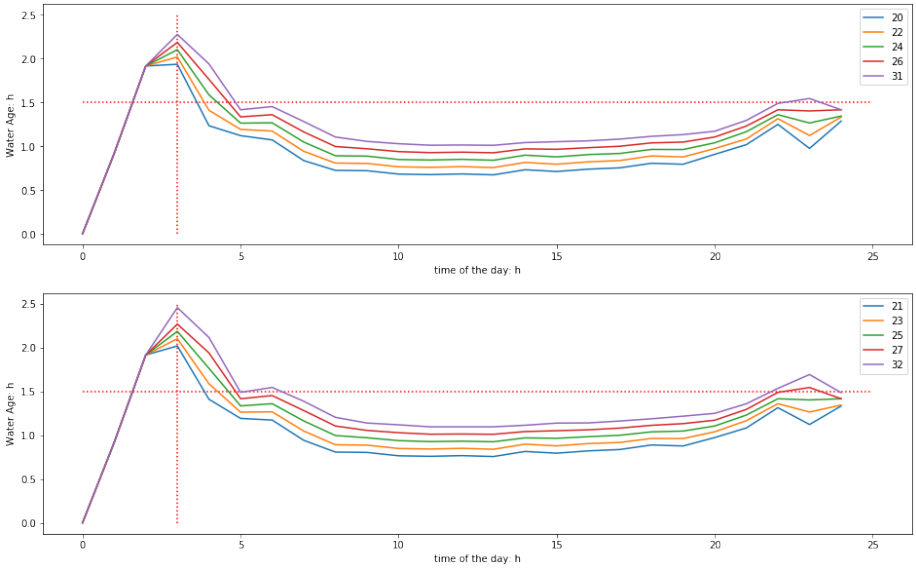


Figure 2.17: Water Age at 10 Different Locations (Five Shown in Each Plot) After Applying Optimal Flushing Control Using Lp Model with Chance Constraints

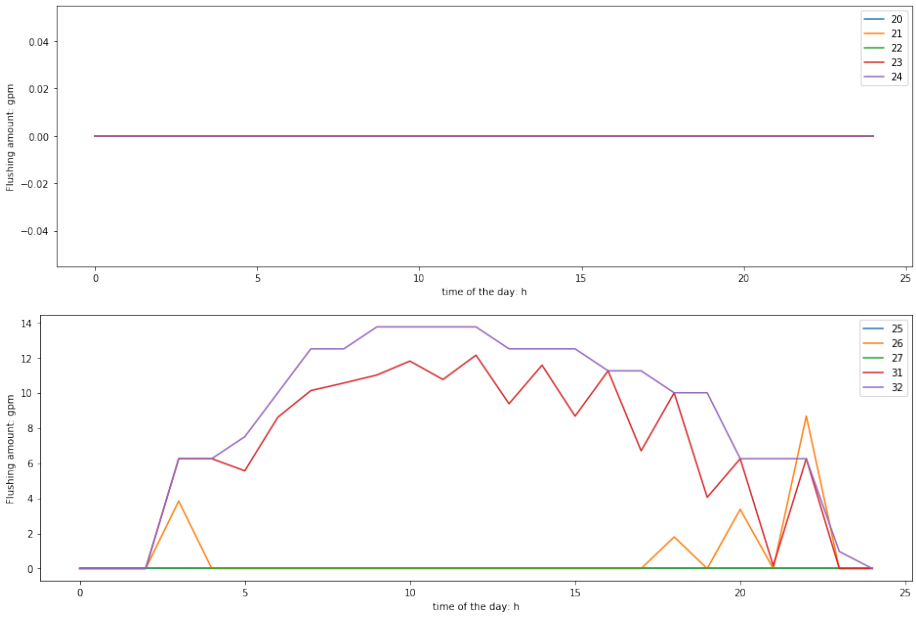


Figure 2.18: The Optimal Flushing Schedules Using LP Model with Chance Constraints

subject to

$$x_i(t+1) = f(\boldsymbol{x}(t), \boldsymbol{x}(t-1), \boldsymbol{u}(t), \boldsymbol{u}(t-1), \boldsymbol{d}(t), \boldsymbol{d}(t-1)) \qquad \forall i \in N_s, \forall t \neq T \tag{2.27}$$

$$P(x_i(t) \leq a_l) \geq 1-\alpha \iff \hat{x}_i(t) \leq a_l - z_{\alpha/2}\sigma_{x_i(t)} \qquad \forall i \in N_s, \forall t \tag{2.28}$$

$$u_i(t) \leq u_l \qquad \forall i \in N_u, \forall t \tag{2.29}$$

$$\boldsymbol{x}(0) = \boldsymbol{a}_0 \tag{2.30}$$

Water age limit a_l is a key parameter in the formulation, it controls the magnitude of flushing. The solution of the above LP gives optimal flushing schedule which was applied to the EPANET model for the city water network. The whole process was

repeated for different values of a_l. Then water age before and after applying the flushing controls were then compared.

The maximum water age over 55h duration before and after applying the optimal flushing controls, and the decrease percentage of maximum water age at different locations are shown in Figure 2.19 for $a_l = 10$. The actual locations that are assigned flushing for $a_l = 10$ are shown in Figure 2.20 (circled in red dotted color). The optimal flushing schedule for $a_l = 10$ is shown in Figure 2.21. (See Appendix A for the same plots for a_l =11 - 15)

The average percentage decrease of maximum water age and the total cost for flushing are summarized in Table 2.5. Note that with lower value of a_l, the average percentage decrease of maximum water age is larger, but total cost is also larger. Comparing locations that has flushing (Figure 2.20 and Appendix B), we can notice that with decreasing value of a_l, the number of locations that triggered flushing are expanding. The set of flushing locations increased from $\{9, 11, 14, 24\}$ for $a_l = 15$ to $\{7, 9, 11, 12, 14, 15, 16, 17, 19, 24, 27, 31\}$ for $a_l = 10$. The results show that we can identify the key locations for flushing in the water age control.

2.3 Conclusions

Water quality can degrade quickly after a short period of stagnation. Water age is a direct indicator of water stagnation. Flushing as one of the main and effective control actions can improve water quality by bringing in fresher water from the water treatment plant. However, quantifying the relationship between flushing controls and water quality improvement to achieve a cost-effective flushing control policy remains as an open research question. The challenge is that water distribution systems involve complicated fluid dynamics and complex chemical reactions that make them very difficult to model. The research work in this chapter addresses this challenge and

Table 2.5: The Average Percentage Decrease of Maximum Water Age

Value of a_l (h)	Average Percentage Decrease	Total Cost of Flushing
15	4.7%	1717.2
14	6.5%	2385.8
13	9.1%	3455.7
12	11.5%	4622.6
11	14.7%	5990.2
10	21.6%	8655.0

developed a linear predictor of water quality and a corresponding LP-based optimal flushing control model with water age limit constraints.

Different machine learning approaches were explored in approximating the dynamics of water distribution systems to predict water quality from observed data. Depending on the actual system and what system states are measured, different approaches could generate very different approximate models. Three case studies on water quality prediction were presented to show how each approach can be applied in specific scenarios. The approximation of water distribution system dynamics makes controlling water quality in networks possible. Integrating the approximate models of water system dynamics within an optimal control framework that uses a LP solution approach, we developed a proactive optimal flushing schedule. This formulation quantifies the relationship between flushing and water quality (water age) improvement. The problem formulation was applied and solved for two different water distribution networks. The optimal flushing solution was evaluated using the EPANET simulation models and the results demonstrated that our flushing control is effective.

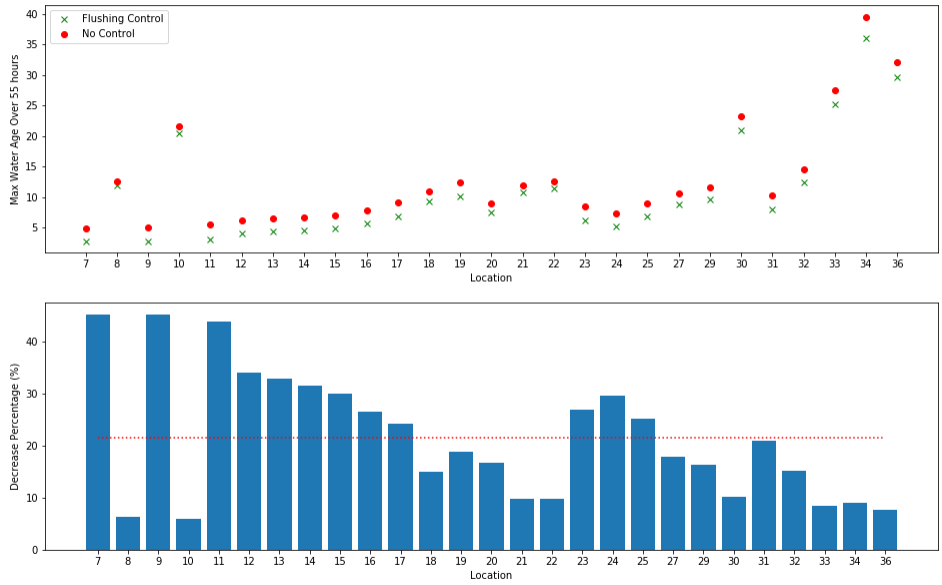


Figure 2.19: The Maximum Water Age over 55h Duration at Different Locations Before and After Applying the Optimal Flushing Controls (Figure above) and the Decrease Percentage of Maximum Water Age at Different Locations (Figure below) for $a_l = 10$

Potential future work could be on fine tuning the approximate models to improve the accuracy, and developing new methods to estimate variance of the predictions. This will improve the associated optimal control policies. Another future direction of research is to consider different water quality indicators that can be measured with high accuracy. Water age is a good indicator of water stagnation, but may not be measured as easily as other water quality indicators. It will be an interesting research direction to combine different water quality indicators into one metric. Multi-objective optimization can be helpful in such problems.

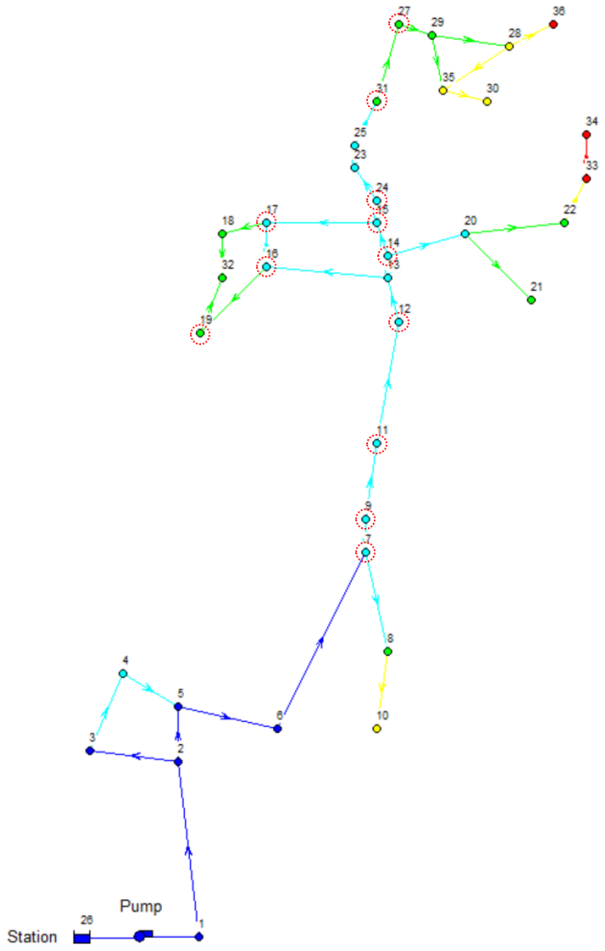


Figure 2.20: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 10$

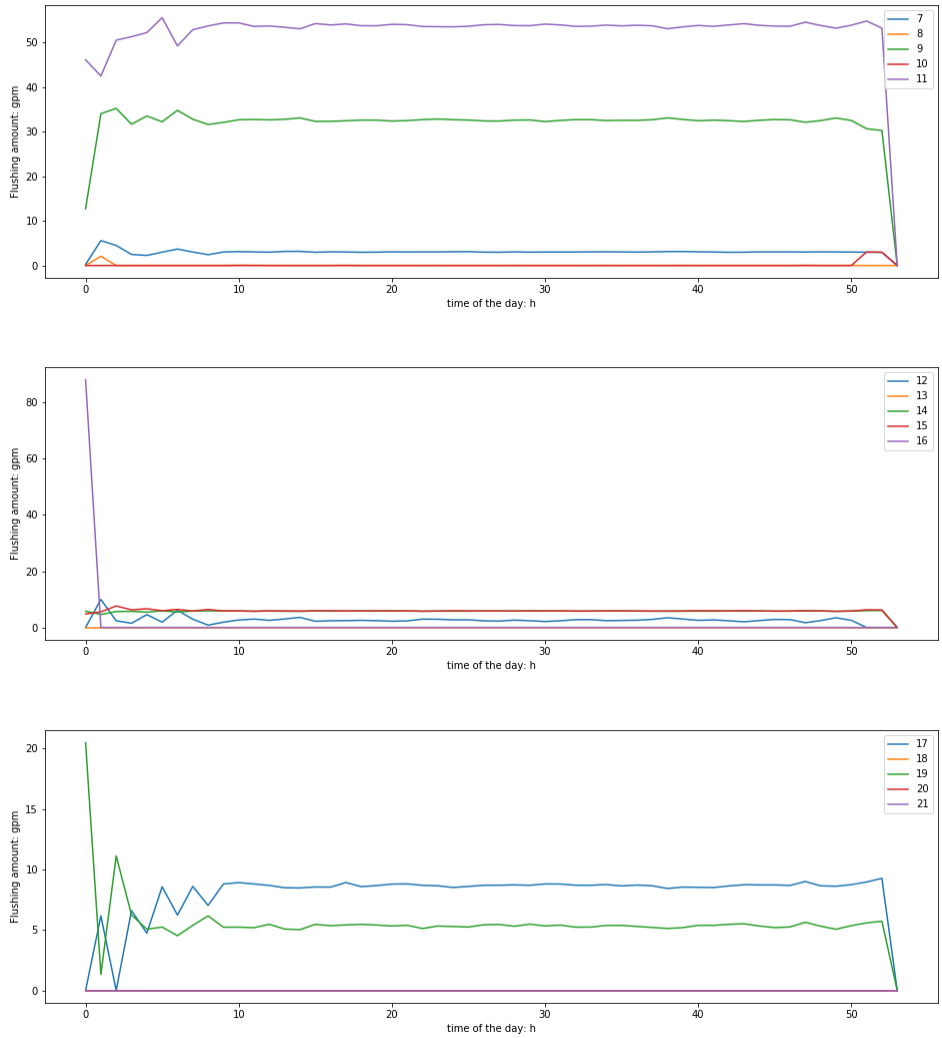


Figure 2.21: The Actual Flushing over Time at Different Locations for $a_l = 10$

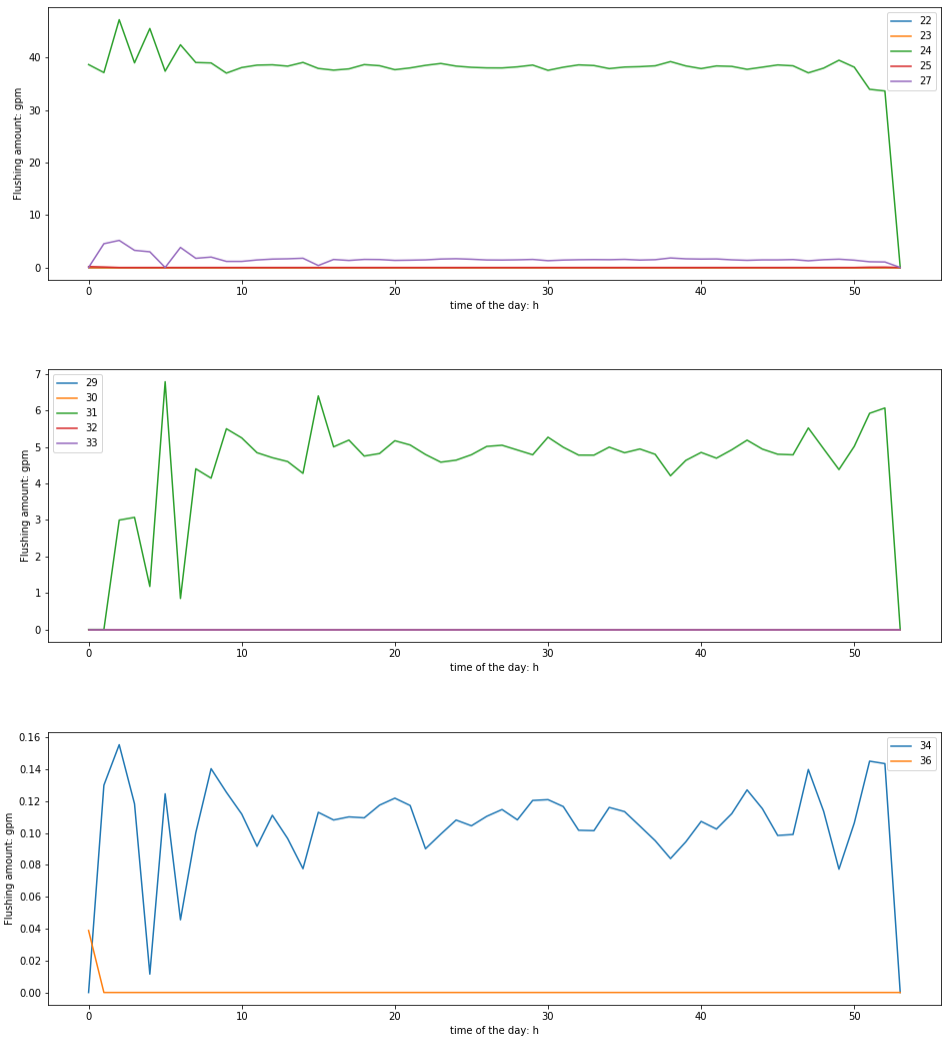


Figure 2.21: The Actual Flushing over Time at Different Locations for $a_l = 10$ (continued)

Chapter 3

NONLINEAR PROGRAMMING BASED OPTIMAL CONTROL OF WATER QUALITY

In Chapter 1, we discussed that flushing and adding disinfectants are two common and effective water quality control actions. In Chapter 2, we showed how to model the optimal control of water quality via flushing. In this chapter, we will consider both control actions (flushing and chlorine injection) in the optimal control of water quality.

Mala-Jetmarova *et al.* (2017) provided a comprehensive review on optimization models and solution approaches for the operation of water distribution systems. After reviewing over 100 papers in total, they categorized water quality related optimization models into three types: the first one included models having objective functions with different costs subject to water quality constraints, the second one included models that minimize the cost of disinfectant mass dose, and the third one had models that minimize disinfectant concentration deviations from desired values at the demand nodes. These models were mainly linear programs or mixed integer nonlinear programs. Also, some researchers had used metaheuristic algorithms combined with EPANET simulations models to solve the control problem.

Flushing is an effective way to improve water quality as it removes bad-quality water out of the system and quickly brings fresh water with satisfactory quality from the source (water treatment plants). Adding disinfectants is also effective as disinfectants strongly prohibit the bacteria growth and kill parasites, bacteria and viruses in water. Chlorine and chloramine are major disinfectants used in public water systems, according to CDC. However, given the complex fluid dynamics and chemical

reactions, the relationship between water quality and different control actions are very likely to be nonlinear. The challenge lies in how to model such relationship in the formulation of the optimal control problem. When more than one control action is being considered, the synergistic effect of different control actions on water quality should be accommodated. This chapter addresses the optimal control problem of water age through both flushing and chlorine injection. A mixed integer quadratically constrained program (MIQCP) formulation is developed in this chapter, and different instances of the formulated problem with different cost ratio parameters are solved. Various chlorine injection controls are also considered in different formulations.

The remaining of this chapter is organized as follows: Section 3.1 explains in more details how flushing and chlorine injection can improve water quality as well as their synergistic effect in water age given certain assumptions. Section 3.2 presents two problem formulations with different chlorine control strategies. In Section 3.3, we explain the solution methods used for solving multiple problem instances. Section 3.4 presents and discusses the numerical examples for problems instances with different cost ratio parameters. Lastly, Section 3.5 concludes this chapter with a summary of contributions and future research directions.

3.1 Water Quality Control via Flushing and Chlorine Injection

We have shown flushing control can be effective in improving water quality and developed proactive optimal flushing schedule for water age control in Chapter 2. In this section, we explain how to model the synergistic impact of flushing and chlorine injection on water age in the problem of water age control.

Flushing is suggested by many researchers for water quality improvement (Lautenschlager *et al.*, 2010; Bédard *et al.*, 2018; Proctor *et al.*, 2020; Hozalski *et al.*, 2020; Islam *et al.*, 2017). The relationship between flushing and water age can be approxi-

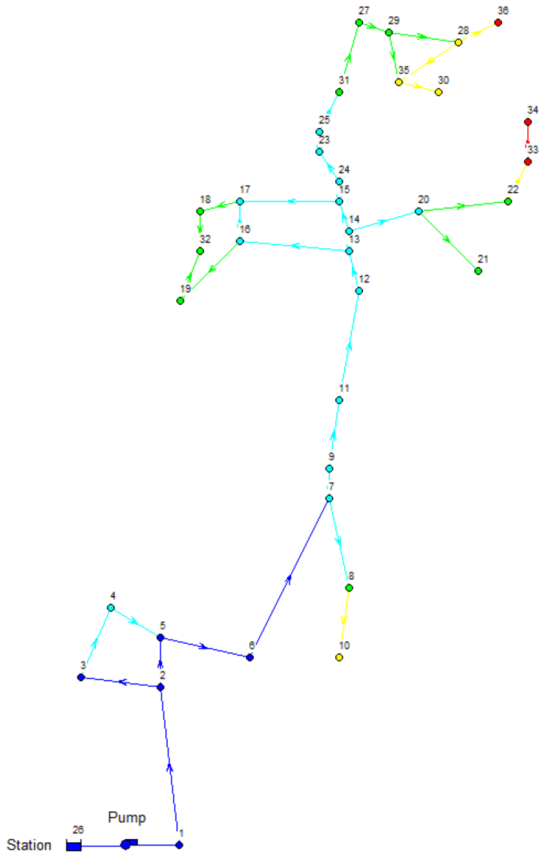


Figure 3.1: A City Water Distribution Network for Flushing and Chlorine Injection Controls

mated using methods mentioned in Section 2.1. In this chapter, we continue to use the city water distribution network (Figure 3.1) for the case study to modeling, analysis and evaluation of our developed approaches. Now we model the relationship between flushing and water age with the Lasso regression approach studied in Section 2.1.4. We end up with a set of linear models that use current and past water demands, flushing controls and water ages as inputs and outputs "water age" quality at next time step.

Mala-Jetmarova *et al.* (2017) summarized that 74% of their reviewed optimization models uses one or two types of decision variables (for control actions). It is a

challenge to include multiple different control actions in the problem formulation due to their synergistic effects on water quality, and the resulting constraints are usually nonlinear. In our formulation of both flushing and chlorine injection controls, we made a reasonable assumption on the relationship of water age and chlorine injection: when certain amount of chlorine is injected to the system, the water age at that location will become zero after a delay of one time step. With this assumption, the relationship between water age, flushing control and chlorine injection control can be modeled approximately as:

$$x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1}) \times (1 - y_{i,t}) \quad 1 \leq t \leq T-1, i \in S \qquad (3.1)$$

where $x_{i,t+1}$ represents water age at location i at time step $t+1$, $f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1})$ represents the fitted Lasso regression models with current time's (i.e., using variables indexed with t) and previous time's (i.e., using variables indexed with $t-1$) water age $x_{\cdot,\cdot}$, flushing control $u_{\cdot,\cdot}$ and water demand $d_{\cdot,\cdot}$, $y_{i,t}$ represents the chlorine injection at location i at time step t.

Variable $y_{i,t}$ in equation (3.1) can be binary or continuous depending on the actual chlorine injection control. If chlorine injection is an on/off control, then $y_{i,t}$ will be binary. Then according to our assumption, Equation (3.1) becomes $x_{i,t+1} = 0$, which makes water age at time step $t+1$ to be zero, when $y_{i,t} = 1$. On the other hand, when $y_{i,t} = 0$, equation (3.1) becomes $x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1})$, meaning water age at location i at time step $t+1$ will use the prediction value from the Lasso model. If chlorine injection is continuous between zero and a maximum amount, then variable $y_{i,t}$ will be continuous between 0 and 1, i.e., $y_{i,t} \in [0, 1]$. In such case, water age at location i at time step $t+1$ will be a fraction $(1 - y_{i,t})$ of the prediction value from the Lasso model represented by function $f(\cdot)$ in equation (3.1).

Considering chlorine injection is not very common at consumer demand locations,

we also included a fixed cost for the installation of required devices (e.g., chlorine booster equipment). This will generate extra constraints, i.e., chlorine injection is only allowed after the required device has been installed at a specific location.

3.2 Model Formulation for Two Different Control Strategies

Two different model formulations based on two different control strategies are developed in this section. Both control strategies apply both flushing and chlorine injection as the control actions. The difference is how chlorine is injected. The first control strategy contains continuous flushing and binary chlorine injection, while the second one consists of continuous flushing combined and continuous chlorine injection.

3.2.1 Continuous Flushing and Binary Chlorine Injection

The control strategy in this section is continuous flushing and binary chlorine injection. We assume there is a fixed cost associated with installing the chlorine booster equipment at the demand location, and there is also an operational cost of chlorine injection. We use two ratios γ and β to model the equipment installation cost and injection cost relative to flushing. We also assume there is some delay for chlorine to be effective after injection; we set such delay to be one time step (one hour) in our example formulation although it is simple to use another time. The set of chlorine injection locations may not necessarily be the same as flushing locations. As the overall goal is to achieve the optimal control of water age through both flushing and chlorine injection, we can identify some locations with very high water age (meaning stagnating water) and consider only those locations for potential chlorine injection.

Same as in Chapter 2, we use $u_{i,t}$ to denote the amount of water to be flushed at location i at time t, and $x_{i,t}$ to denote the water age at location i at time t. The full definition of all parameters, indexes and decision variables are summarized in

Table 3.1.

The objective function is the total cost, which includes cost of flushing, operational cost of chlorine injection and fixed cost of chlorine booster installation. The full formulation for binary chlorine injection is given below.

$$\min \Sigma_{i=1}^{n}\Sigma_{t=1}^{T}u_{i,t} + \beta\Sigma_{i\in S}\Sigma_{t=1}^{T}y_{i,t} + \gamma\Sigma_{i\in S}z_i \tag{3.2}$$

subject to

$$x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1}) \times (1 - y_{i,t}) \quad 1 \le t \le T-1, i \in S \tag{3.3}$$

$$x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1}) \quad 1 \le t \le T-1, i \notin S \tag{3.4}$$

$$y_{i,t} \le z_i \quad t \ge 1, i \in S \tag{3.5}$$

$$P(x_{i,t} \le a_l) \ge 1 - \alpha \quad t \ge 2, \forall i \tag{3.6}$$

$$u_{i,t} \le u_l \quad t \ge 1, \forall i \tag{3.7}$$

$$x_{i,t} = a_0 \quad t = 0, 1, \forall i \tag{3.8}$$

$$u_{i,t} \in \mathbb{R}^+, x_{i,t} \in \mathbb{R}^+ \quad \forall i, \forall t \tag{3.9}$$

$$y_{i,t} \in \{0,1\}, z_i \in \{0,1\} \quad i \in S, \forall t \tag{3.10}$$

where $x_{\cdot,t}$ in constraints (3.3) and (3.4) represents all $x_{1,t}, x_{2,t}, ..., x_{n,t}$, and the same logic applies to notations $u_{\cdot,t}$ and $d_{\cdot,t}$. Function $f(\cdot)$ is the set of Lasso regression models for water age prediction developed in Section 2.1.4.

Constraints (3.3) model water age transition at locations where chlorine booster could be potentially installed, where S denotes the set of potential locations for chlorine booster installations; S could be the whole set of demand location or just a subset of them. If chlorine is injected at location i at time step t (i.e., $y_{i,t} = 1$), then $x_{i,t+1} = 0$ meaning water age at the same location at time step $t+1$ becomes zero due to chlorine injection. Otherwise, water age is restricted to follow the prediction from

Table 3.1: Definition of Parameters, Indexes and Variables for the Optimal Water Age Control Formulation with Continuous Flushing and Binary Chlorine Injection

Parameters/Indexes	Definition
n	total number of demand locations
T	duration of simulation
i	index for location, $i \in \{1, 2, ..., n\}$
t	index for time, $t \in \{0, 1, 2, ...T\}$
S	set of chlorine injection locations
α	the parameter used in the chance constraint to control the confidence level
β	cost of chlorine injection relative to flushing
γ	cost of chlorine booster installation relative to flushing
$d_{i,t}$	water demand at location i at time step t
a_l	universal water age limit for al times and locations
a_0	initial water age for time steps 0 and 1 (using one parameter for simplicity)
u_l	flushing limit (can vary for different times and locations)
Variables	**Definition**
$u_{i,t}$	amount of water to be flushed at location i at time t, $u_{i,t} \in \mathbb{R}^+$
$x_{i,t}$	water age at location i at time t, $x_{i,t} \in \mathbb{R}^+$
$y_{i,t}$	indicator variable, whether to inject chlorine at location i at time step t, $i \in S, t \geq 1, y_{i,t} \in \{0, 1\}$
z_i	indicator variable, whether to install a chlorine booster at location i, $i \in S, z_i \in \{0, 1\}$

the Lasso model, same as what is indicated in constraint (3.4) for locations where there are no chlorine boosters. Constraint (3.5) indicates chlorine injection is only allowed when the booster equipment is installed at that location. Constraint (3.6) is the same chance constraint for water age limit as in Section 2.2.2. In solving the problem instances, we will use the equivalent constrain (2.19). Constraint (3.7) limits the maximum flushing. Constraint (3.8) is the initial condition for water age at time steps 0 and 1. Constraints (3.9) and (3.10) define the domain for possible values of decision/control variables.

Note that functions $f(\cdot)$ represent the Lasso models from Section 2.1.4 and each one is a linear function of $x_{i,t}, u_{i,t}$ and $d_{i,t}$, and therefore constraint (3.3) is quadratic. Thus, objective function (3.2) along with constraints (3.3) - (3.10) forms a *mixed integer quadratically constrained program* (MIQCP).

3.2.2 Continuous Flushing and Continuous Chlorine Injection

In this section, a different chlorine injection strategy is addressed. Unlike in Section 3.2.1, the chlorine injection in this section can be continuous between zero to a maximum amount. Thus variable $y_{i,t}$ becomes a continuous value between 0 and 1. Objective function and all constraints except the possible values for $y_{i,t}$ stay the same. For completeness and better reading experience, the full problem formulation with this change is presented again.

Firstly Table 3.2 summarizes the parameters, indexes and variables used in the model formulation.

Table 3.2: Definition of Parameters, Indexes and Variables for the Optimal Water Age Control Formulation with Continuous Flushing and Continuous Chlorine Injection

Parameters/Indexes	Definition
n	total number of demand locations
T	duration of simulation
i	index for location, $i \in \{1, 2, ..., n\}$
t	index for time, $t \in \{0, 1, 2, ...T\}$
S	set of chlorine injection locations
α	the parameter used in the chance constraint to control the confidence level
β	cost of chlorine injection relative to flushing
γ	cost of chlorine booster installation relative to flushing
$d_{i,t}$	water demand at location i at time step t
a_l	universal water age limit for al times and locations
a_0	initial water age for time steps 0 and 1 (using one parameter for simplicity)
u_l	flushing limit (can vary for different times and locations)
Variables	**Definition**
$u_{i,t}$	amount of water to be flushed at location i at time t, $u_{i,t} \in \mathbb{R}^+$
$x_{i,t}$	water age at location i at time t, $x_{i,t} \in \mathbb{R}^+$
$y_{i,t}$	chlorine injection variable, as a fraction of the maximum amount at location i at time step t, $i \in S, t \geq 1, y_{i,t} \in [0, 1]$
z_i	indicator variable, whether to install a chlorine booster at location i, $i \in S, z_i \in \{0, 1\}$

Now the full problem formulation for continuous flushing and continuous chlorine injection is:

$$\min \Sigma_{i=1}^{n}\Sigma_{t=1}^{T}u_{i,t} + \beta\Sigma_{i\in S}\Sigma_{t=1}^{T}y_{i,t} + \gamma\Sigma_{i\in S}z_i \tag{3.11}$$

subject to

$$x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1}) \times (1 - y_{i,t}) \quad 1 \le t \le T-1, i \in S \tag{3.12}$$

$$x_{i,t+1} = f(x_{\cdot,t}, x_{\cdot,t-1}, u_{\cdot,t}, u_{\cdot,t-1}, d_{\cdot,t}, d_{\cdot,t-1}) \quad 1 \le t \le T-1, i \notin S \tag{3.13}$$

$$y_{i,t} \le z_i \quad t \ge 1, i \in S \tag{3.14}$$

$$P(x_{i,t} \le a_l) \ge 1 - \alpha \quad t \ge 2, \forall i \tag{3.15}$$

$$u_{i,t} \le u_l \quad t \ge 1, \forall i \tag{3.16}$$

$$x_{i,t} = a_0 \quad t = 0, 1, \forall i \tag{3.17}$$

$$u_{i,t} \in \mathbb{R}^+, x_{i,t} \in \mathbb{R}^+ \quad \forall i, \forall t \tag{3.18}$$

$$y_{i,t} \in [0,1], z_i \in \{0,1\} \quad i \in S, \forall t \tag{3.19}$$

The meaning of all constraints are same as those explained in Section 3.2.1.

3.3 Solution Methods

Mixed integer quadratically constrained programs are usually handled by cutting planes and other integer programming methods. When solving problem instances with different cost ratios parameters in next section, we used the Gurobi solver. The solution process by Gurobi in solving our problem instances is a branch-and-bound algorithm combined with various cutting planes methods including mixed integer rounding (MIR) cuts, flow cover cuts, relaxation linearization technique (RLT) cuts, lift-and-relax cuts, etc.

The branch-and-bound algorithm iteratively relaxes the restriction on integer variables, and solved the relaxed problem with all continuous variables. The solution of

the relaxed problem serves as lower or upper bound to the original problem. The process repeats the branch-and-bound logic until the solution gap reaches zero (the solution is optimal), or early stops when reaching the time limit. Within the branch-and-bound process, the branch can be pruned when the solution is inferior than the best known solution, or when the branch becomes infeasible.

The quadratic constraint can be temporally replaced with appropriate cuts, i.e., a set of linear equations representing hyper-planes in high dimensional space (thus the name cutting plane methods).

The MIR cuts are based on the simple principle that if $a \leq b$ and a is an integer, then $a \leq \lfloor b \rfloor$, where $\lfloor b \rfloor$ represents the largest integer less than or equal to b (Osiadacz, 1990). The MIR cuts generates new valid inequalities by rounding the coefficients according to this principle.

The RLT cuts generate new valid inequalities by using relaxation variables to linearize intermediate nonlinear constraints. Many of the cutting plane methods implemented by Gurobi combine simple principals to be more efficient.

3.4 Numerical Experiments

Problem instances of the formulations in Section 3.2 with different cost ratios γ and β were solved using the Gurobi solver on the ASU research computing cluster. The evaluation metrics of all addressed problem instances are shown in Table 3.3.

From Table 3.3, we can notice that in Scenario 1 when chlorine control (installation and injection) costs (1000 and 200) are very large, the total cost is exactly same as that of flushing only control (Table 2.5, $a_l = 10$), and the solution is also the same, meaning chlorine controls (installation and injection) are not triggered at all due to their large cost. The chlorine cost for Scenario 2 is probably still too large to trigger the chlorine controls, given that the total cost is still same as that of flushing only

Table 3.3: Numerical Experiments with Different Cost Ratio Parameters

Scenario	Installation Cost γ	Injection Cost β	Chlorine Injection	Run Time	Total Cost	Optimality Gap
1	1000	200	Binary	412s	8655.0	0%
2	1000	100	Binary	10h	8655.0	11.1%
3	1000	20	Binary	10h	5180.8	16.6%
4	**1000**	**10**	**Binary**	**10h**	**4120.8**	**6.12%**
5	100	2	Binary	10h	905.6	19.6%
6	100	2	Binary	20h	905.6	16.0%
7	100	2	Binary	30h	905.6	15.4%
8	**1000**	**10**	**Continuous**	**20h**	**3411.0**	**2.67%**

control after 10 hours (CPU time) solving. As the cost of chlorine control is reduced in Scenarios 3 to 5, near optimal solution where chlorine injections are triggered starts to appear. In the solution of Scenario 4, the controls are a mix of flushing and chlorine injection. More details are provided below. Scenarios 5 to 7 represent the case when the chlorine control cost is relatively low compared with flushing cost; in this case, most chlorine injections are triggered. From Scenario 5 to Scenario 7 we can notice that the total cost stays the same even when the time limit for the solution process is increased from 10 hours to 30 hours, and we expect the bound will keep increasing and thus reducing the optimality gap if it is given more run time. We will show later that the control in this solution is mainly chlorine injections. Scenario 8 is a problem instance for continuous flushing and continuous chlorine injection. The comparison of Scenario 4 and Scenario 8 is discussed later.

Figure 3.2 shows the chlorine booster installation decisions, and Figure 3.3 shows

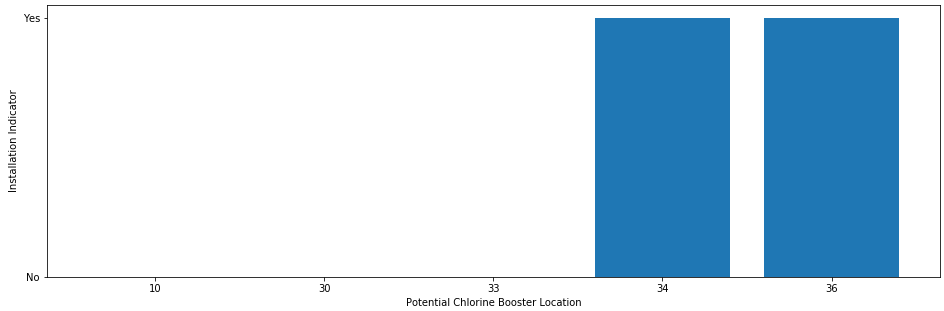


Figure 3.2: Scenario 4: Decisions on Chlorine Booster Installations

the chlorine injection decisions for Scenario 4. Chlorine booster is installed only at location 34 and 36 (out of 5 potential locations), and chlorine is injected at both locations for all times except the first hour. Figure 3.4 shows the flushing schedule at all locations. Note that flushing has been significantly reduced due to chlorine injections (comparing Figure 3.4 with Figure 2.21), flushing is mainly triggered at location 7, location 12 and location 15.

The chlorine controls (chlorine booster installations and chlorine injections) in Scenario 5 are shown in Figure 3.5 and Figure 3.6. The flushing schedules at different locations for Scenario 5 are shown in Figure 3.7, respectively. And we can notice that only a small amount of flushing is triggered at location 14 (Figure 3.7), one of the key locations identified in Section 2.2.4. And chlorine booster is installed at 4 out of 5 potential locations (Figure 3.5). Chlorine injection is triggered at almost all times once the chlorine booster is installed (Figure 3.6).

In Scenario 8, the chlorine injection is continuous between zero and a maximum amount. Compared with Scenario 4, all other parameters are the same. We can notice that continuous chlorine injection has a lower cost, and a smaller solution gap. This result makes sense because with continuous chlorine injection, the problem has a larger feasible region, therefore a better solution can possibly be found. The

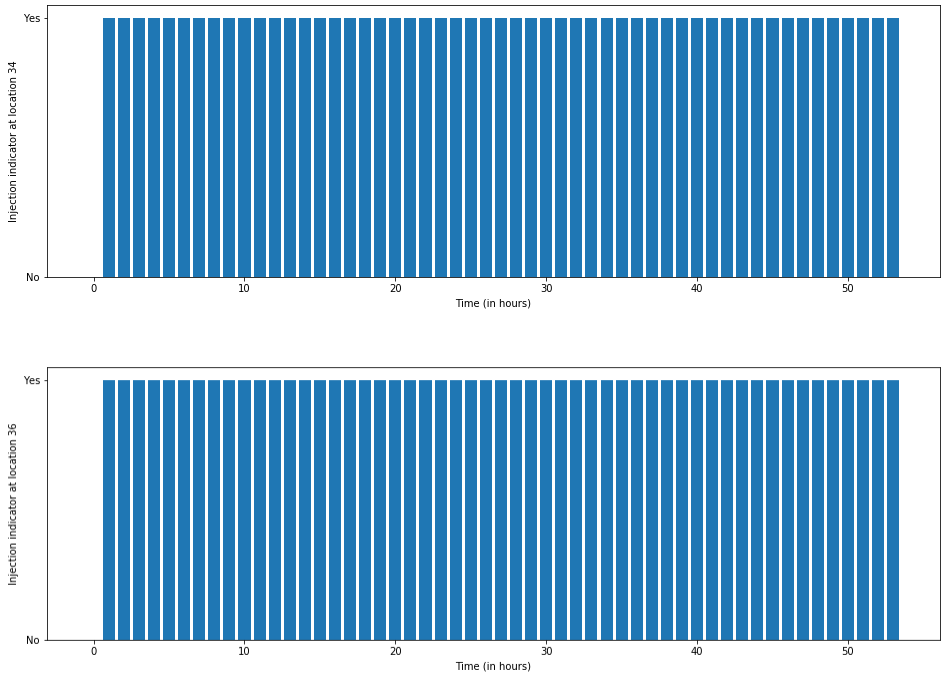


Figure 3.3: Scenario 4: Decisions on Chlorine Injections over Time

chlorine booster locations are same as Scenario 4, only location 34 and location 36 are selected. Another observation is that chlorine injection amount is saved by about half (comparing Figure 3.9 with Figure 3.3), and the amount of flushed water is slightly saved (comparing Figure 3.10 with Figure 3.4).

3.5 Conclusions

Flushing and adding disinfectants are two common and effective water quality control actions. However, their synergistic impacts on water quality makes the optimal control of water quality from these coupled control actions challenging. We have developed two MIQCP-based optimal control models that includes both flushing and chlorine injections. The first model assumes continuous flushing and binary chlorine injections; the second model assumes continuous flushing and continuous chlorine in-

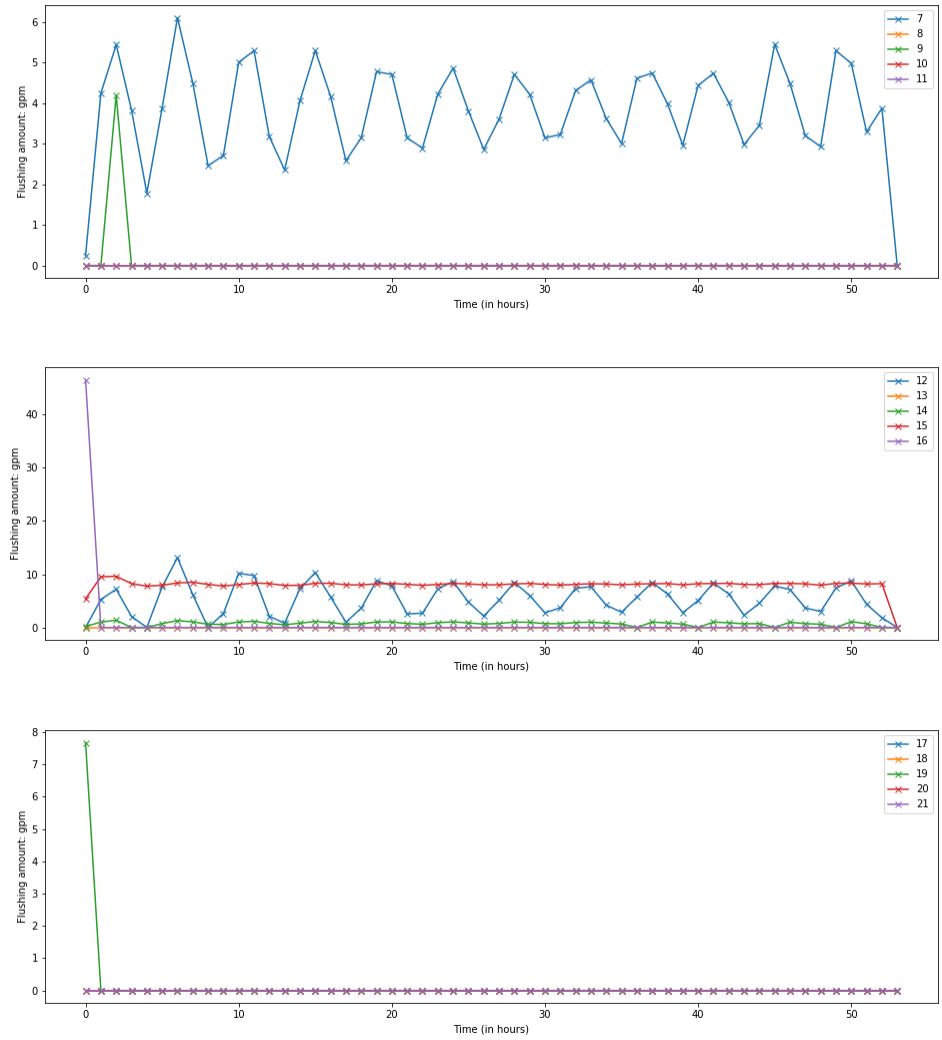


Figure 3.4: Scenario 4: Flushing Controls over Time at Different Locations

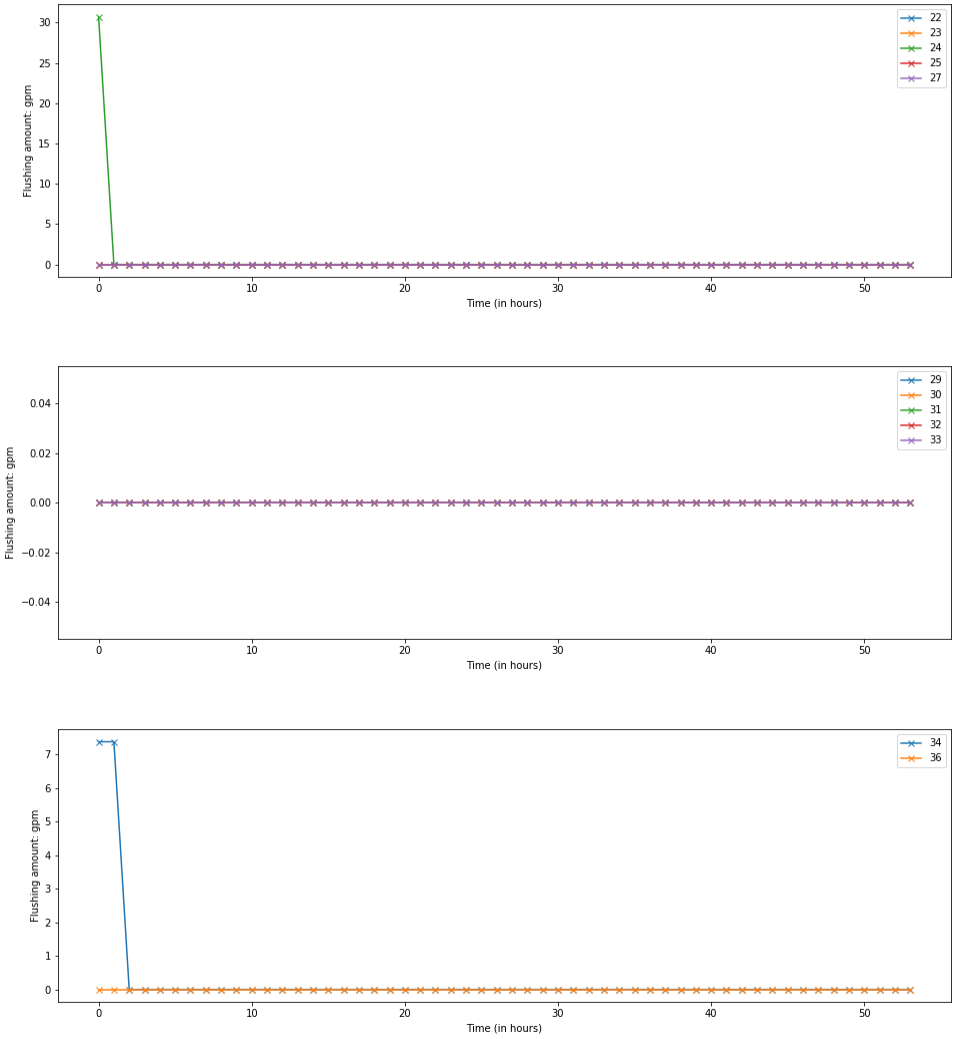


Figure 3.4: Scenario 4: Flushing Controls over Time at Different Locations (Continued)

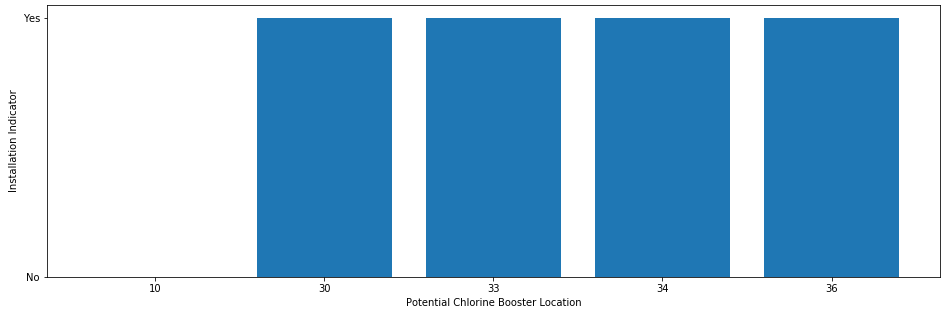


Figure 3.5: Scenario 5: Decisions on Chlorine Booster Installations

jections. Problem instances with different cost ratio parameters are solved using the Gurobi solver for mathematical programming. We obtained meaningful insights from the computational results. The inclusion of chlorine injection control can reduce the total control cost when compared with flushing-only control, and continuous chlorine injections can further reduce the total cost, saving about half of the chlorine control cost, when compared with binary chlorine injection controls.

In this chapter, a set of potential chlorine control locations were identified by observing locations with high water age due to local stagnation. A future research direction could be to develop a systematic method for identifying candidate locations for chlorine injection controls. In our formulations, the injected chlorine at one location only affects "water age" at other locations later in time by the water age prediction model. Another future research direction could be on setting up a coverage range of instantaneous impact for the injected chlorine, and explicitly incorporating a chlorine decay model in formulating the optimal control problem.

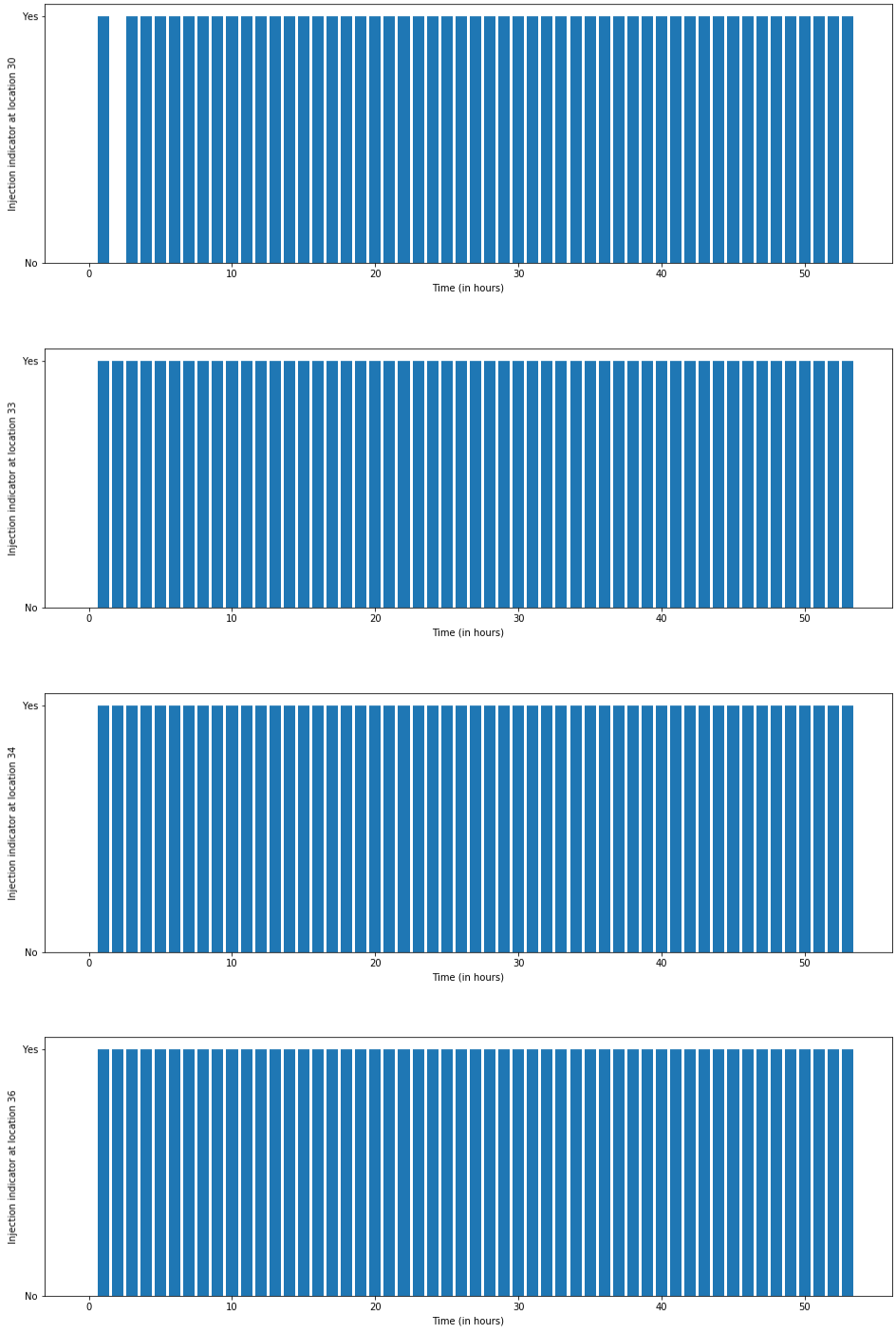


Figure 3.6: Scenario 5: Decisions on Chlorine Injections over Time

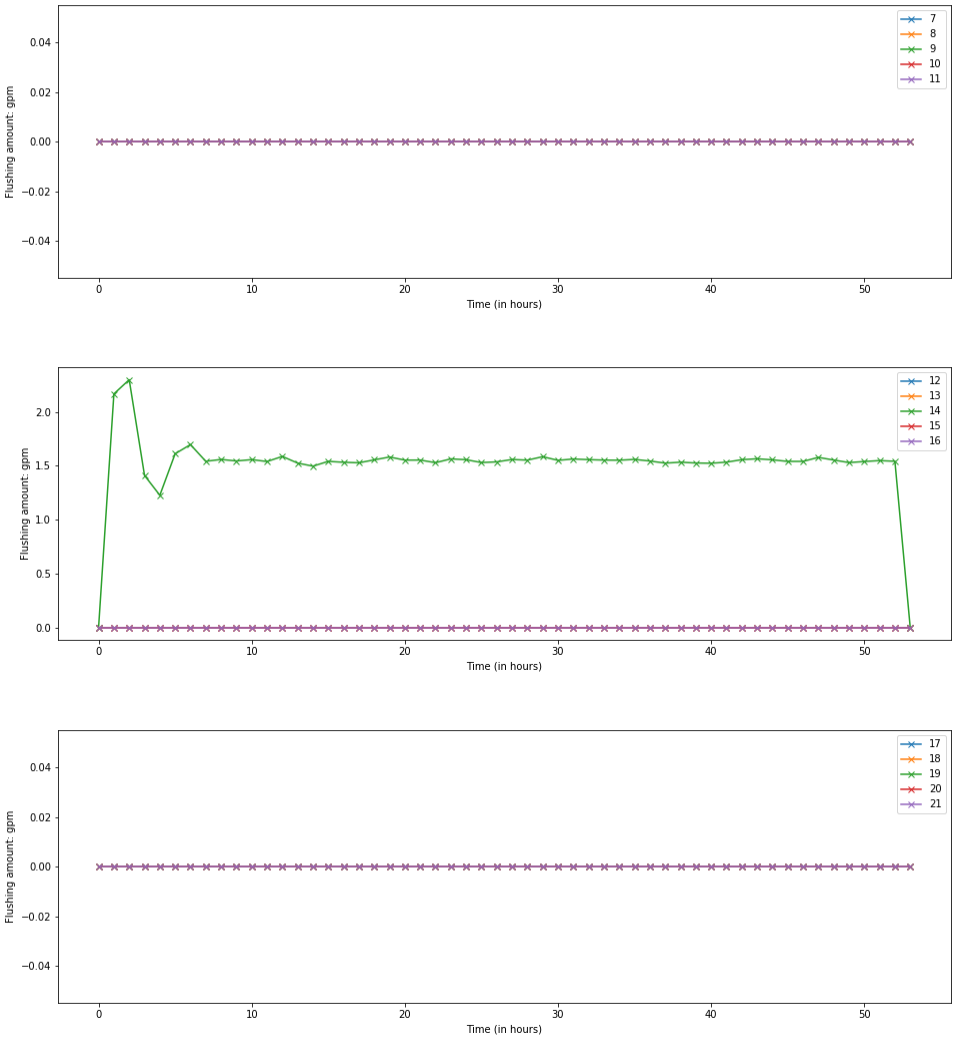


Figure 3.7: Scenario 5: Flushing Controls over Time at Different Locations

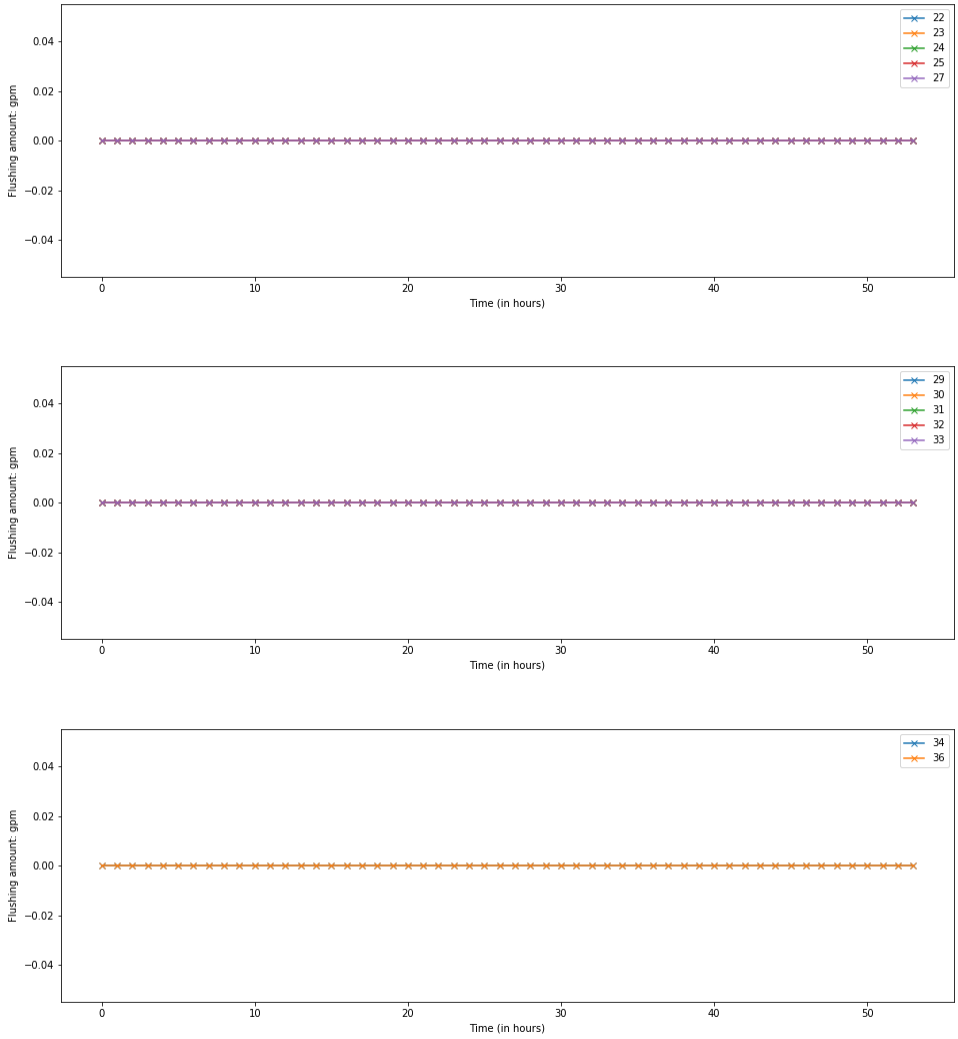


Figure 3.7: Scenario 5: Flushing Controls over Time at Different Locations (Continued)

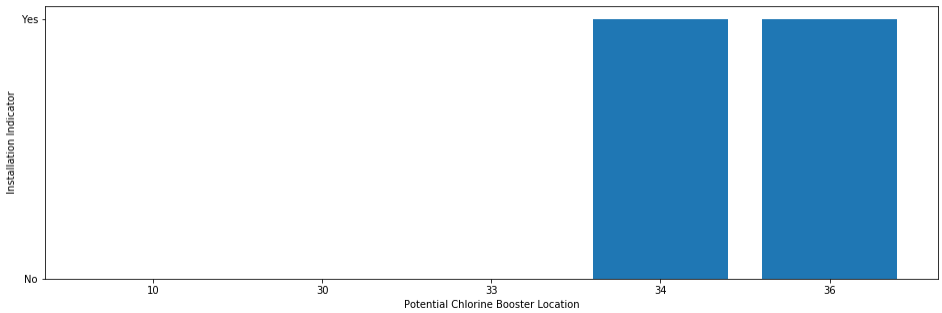


Figure 3.8: Scenario 8: Decisions on Chlorine Booster Installations

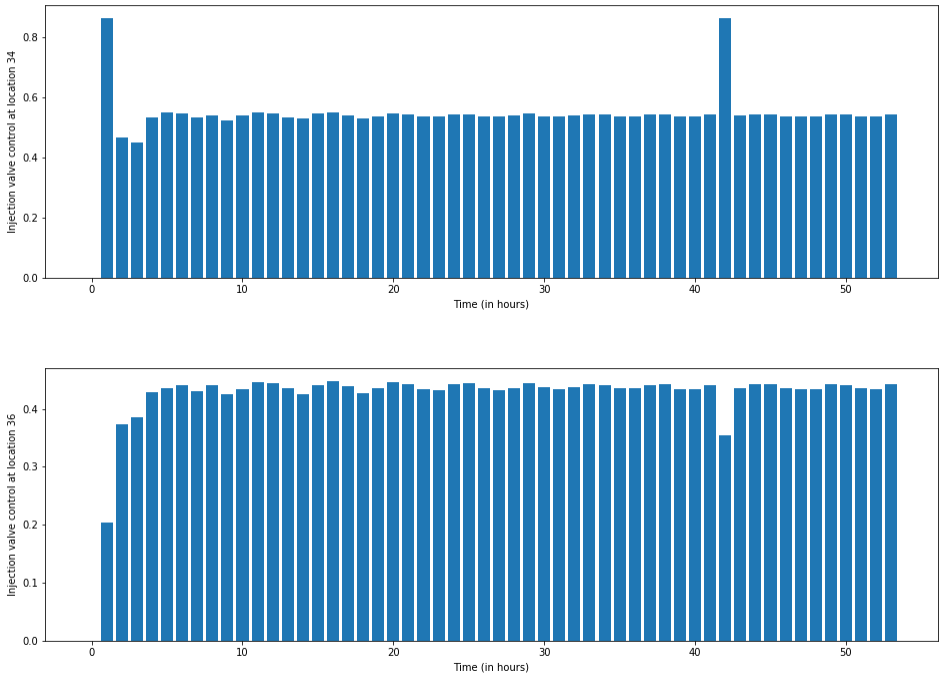


Figure 3.9: Scenario 8: Decisions on Chlorine Injections over Time

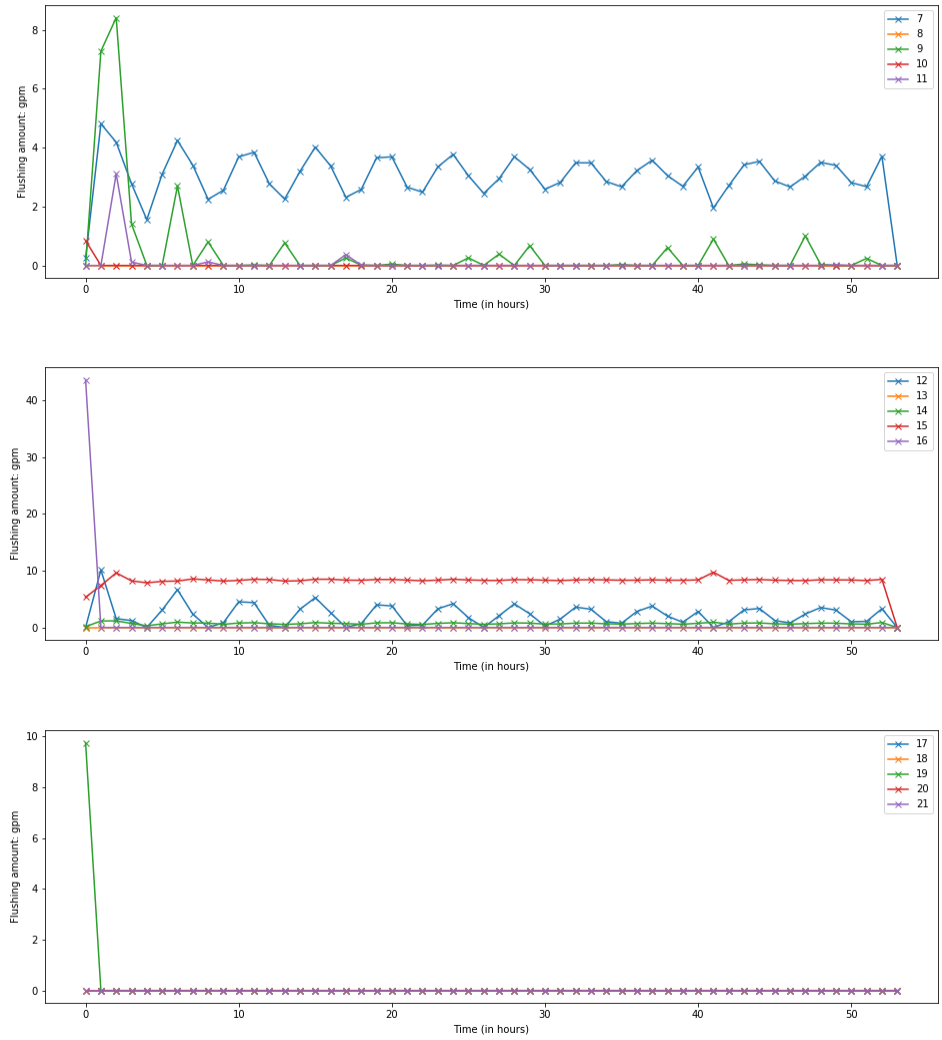


Figure 3.10: Scenario 8: Flushing Controls over Time at Different Locations

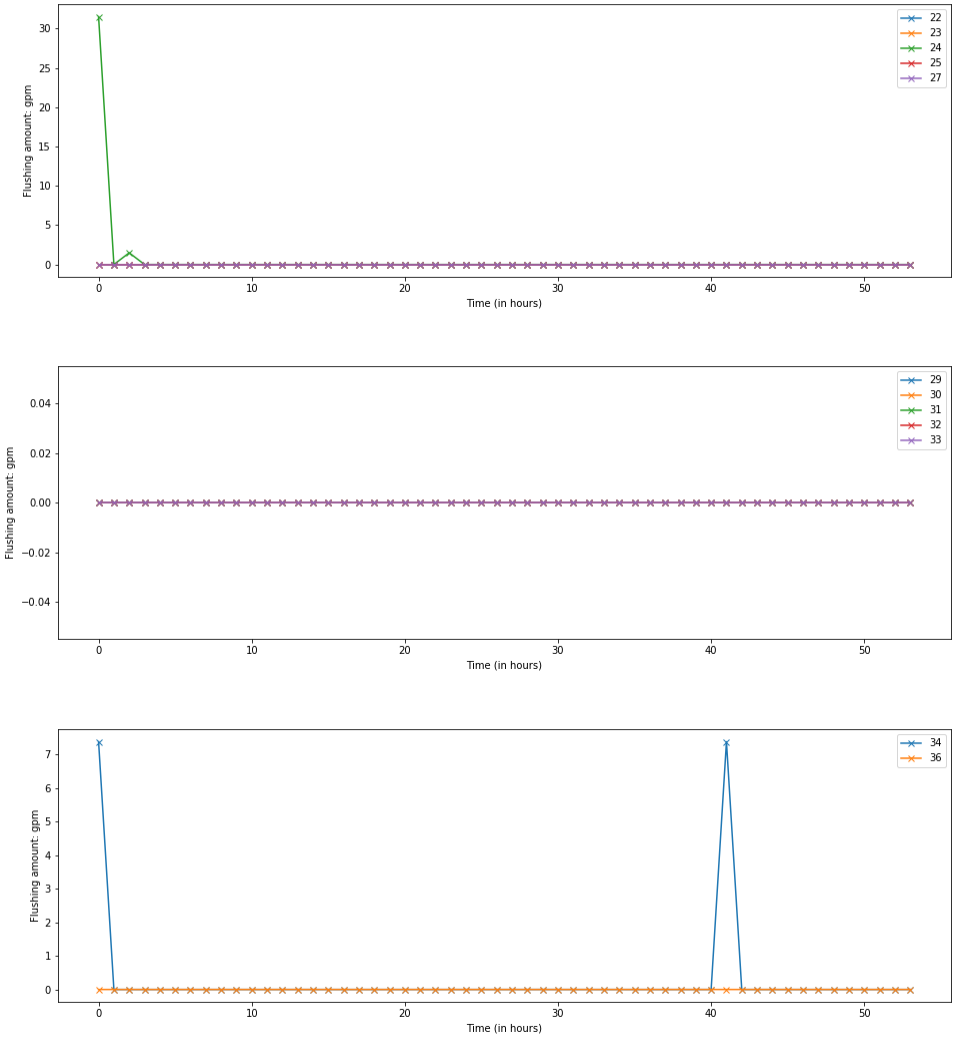


Figure 3.10: Scenario 8: Flushing Controls over Time at Different Locations (Continued)

Chapter 4

OPTIMAL CONTROL OF WATER QUALITY WITH STOCHASTIC DEMAND

In the previous chapters, future water demand is assumed to be known and deterministic. This assumption is reasonable when there is a long history data for water demand that is quite steady with little or no fluctuations. However, realistically, in many scenarios, fluctuations in water demand over space and time may need it to be modeled as being stochastic.

Water demand in a water distribution system varies throughout the day, it also varies from day to day, as people do not perform water-consuming activities at the same time of every day, nor do they perform the same activities from day to day. For example, water usage in university buildings is generally higher during weekdays than during weekends. Also, water usage in office buildings in the daytime is significantly higher than in the nighttime. In fact, Alcocer-Yamanaka *et al.* (2012) have pointed out that residential water demand is sporadic and characterized by sudden demand pulses. For water distribution systems with such varying demand, it is necessary to model the demand as stochastic and develop a robust control policy to maintain water quality throughout the system and at all times. Although methods in previous chapters still apply with demand prediction models, they highly depend on the quality of the prediction, and would be less effective for such water distribution systems with highly fluctuating water demand. A more robust and systematic control method is needed for better water quality control under stochastic for uncertain demand.

This chapter focuses on the optimal control of water quality (water age) with uncertain and stochastic demand. The research work in this chapter takes a different perspective than previous chapters in terms of the prediction and modeling the

optimal control. The optimal control problem is modeled using the *reinforcement learning* based framework, and an optimal policy that is generated through learning from data can recommend real-time control decisions.

The structure of this chapter is organized as follows. Section 4.1 gives more details about the research problem. In Section 4.2, related work in the literature is summarized. The complete problem formulation for optimal water age control via flushing is developed in Section 4.3. In Section 4.4, main solution methodologies are discussed. A complete algorithm combining SARSA temporal-difference learning (SARSA is further explained in Section 4.4.1) with linear function approximation for deriving the optimal control policy is presented in that section. Then, Section 4.5 presents the results of different experiments with different hyper-parameters and different update rules in learning. Lastly, Section 4.6 summarizes the research contributions of this Chapter and provides potential future research directions.

4.1 Dynamic Optimal Control Problem Description

In Chapter 2 and Chapter 3, we discussed the formulation and optimal control of water quality in water distribution systems with assumed deterministic and known demand. We will relax this assumption on water demand in this chapter and focus on the modeling of optimal control of water quality with uncertain stochastic water demand.

As in previous chapters, water age is used as water quality indicator, and flushing is the primary control action. The difference is that water demand is no longer deterministic, it is stochastic and future demand is uncertain. The goal is to model the optimal control of water age with such demand, and then derives the optimal control policy that can help real-time decision making for the flushing control.

Water distribution systems with stochastic water demand can be modeled as finite

Markov Decision Processes (MDP) with states representing the water quality (e.g., water age) at different locations. Standard MDP problems require a probability function,

$$p(s', r|s, a) = Pr(S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a) \qquad \forall s', s \in \mathcal{S}, a \in \mathcal{A}(s) \quad (4.1)$$

to describe the dynamics of the MDP. Equation (4.1) defines the probability of system reaching state s' generating reward r when taking action a in state s. However, the states (water age, chlorine concentration, etc.) of water distribution systems are usually of continuous values, it is almost impossible to derive a probability function like (4.1). The challenge is how to model the MDP dynamics if such probability functions cannot be derived. Model-free methods like *Q-learning* in *reinforcement learning* do not require complete knowledge of the probability function representing the MDP dynamics, and can be potentially helpful in modeling and solving the optimal control problem.

Another challenge we are faced with in controlling water distribution systems is that the system states are usually described with continuous values. The problem can be converted to one with discrete states after discretization (Yoo and Kim, 2016; Castelletti *et al.*, 2010; Lee and Labadie, 2007). However, the resolution of discretization is hard to determine. Also, when the state vector is high dimensional, discretization will generate an enormous amount of states which makes the optimal control problem intractable. In order to avoid the dilemma of using discretization, parameterized methods that can deal with continuous state directly are considered in Section 4.4.3.

4.2 Related Work

The optimal control of water distribution systems has been studied by many researchers. Many of them have focused on the controlling of reservoir/pump operations of complex water distribution systems (Castelletti *et al.*, 2010; Lee and Labadie, 2007; Castelletti *et al.*, 2014; Bhattacharya *et al.*, 2003). Wang and Hong (2020) reported that 12% of the 77 studies on building controls they have reviewed are on domestic hot water control (Ruelens *et al.*, 2014; De Somer *et al.*, 2017). Many of the reinforcement learning approaches use deep neural networks in approximating the value function (Wu *et al.*, 2015b; Mocanu *et al.*, 2018; Zhang *et al.*, 2018). Deep neural networks can achieve high accuracy in the value function approximation in these optimal control problems, but the complicated network structure makes it different to interpret their predictions.

When water distribution systems are modeled as MDPs, if the probability function of the underlying MDP dynamics can be derived, then *dynamic programming* (DP) can be used to solve the optimal control problem. In cases where the MDP dynamics cannot be derived, especially in systems with uncertain demand, the optimization problem is usually solved using approximation methods. Here we will use the method called *approximate dynamic programming* (ADP). In ADP approaches, any supervised learning methods can be used to approximate the value function (Sutton and Barto, 2018), for example, regression tree based approach (Castelletti *et al.*, 2010), support vector regression (Shin *et al.*, 2010), and linear models with Fourier basis (Konidaris *et al.*, 2011).

ADP approaches, or *reinforcement learning* are becoming increasingly utilized in the stochastic optimal control, and various implementations have been established by different researchers in solving such optimization problems. This is also true in the

control of water distribution systems. However, to our knowledge, there is not yet much literature on modeling the optimal control of water age in water distribution systems, especially with flushing controls. Our main contribution in this chapter is the development of a complete optimal flushing control model using ADP. Computational results from our successful implementation of our solution algorithm show that the learned policy for flushing is very effective.

4.3 Water Quality Control Model Based on Reinforcement Learning

4.3.1 Key Concepts

Reinforcement learning methods are suitable for the optimal control of systems that can be modeled as Markov decision processes. They iteratively learn what to do in different situations and derive the optimal control policy (see Sutton and Barto (2018) for a more comprehensive introduction). Some of the key concepts are briefly explained below.

Markov decision processes (MDP) are discrete time stochastic control processes. A MDP is usually represented by a four-tuple (S, A, P, R), where

- S is the state space (a set of possible states),
- A is the action space (a set of possible actions),
- $P : S \times A \times S \to [0, 1]$ is the state transition probability map, and $P(s, a, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability of ending in state s' at $t+1$ when taking action a_t at time t,
- $R : S \times A \times S \to \mathbb{R}$ is the reward map, and $R(s, a, s')$ represents the reward generated after taking action a at time t and reaching state s'.

If a MDP has a finite number of forward time steps, it is called a finite horizon MDP. The water quality control problem we consider in this chapter is based on a finite horizon MDP.

A policy π is a mapping function of the state-action pair (s, a); it defines the probability of taking action a given state s, that is,

$$\pi(a|s) = P(a_t = a|s_t = s) \tag{4.2}$$

In our water age control problem, the policy gives us the probability of taking each action (flushing control) given a state (water age) of the water distribution system.

The reward R_t comes from the goal of the control task, and reinforcement learning methods usually try to maximize the expected total reward G_t,

$$G_t = \Sigma_{i=1}^{T} R_{t+i} \tag{4.3}$$

There could be a discounting factor for rewards in future time steps, but we did not consider discounting in our problem since our finite horizon was not long. Our goal is to control the water distribution system with minimum cost over the given finite horizon; in our case, the reward is the negative cost (maximizing the expected total reward is equivalent to minimizing the expected total cost).

Value function $v_\pi(s)$ associated with a policy π for a state s is defined as the expected total reward over all future time steps when starting from state s and following policy π afterwards, that is,

$$v_\pi(s) = E_\pi(G_t|S_t = s) \tag{4.4}$$

Similarly, the action-value function $q_\pi(s, a)$ is the value function for state-action pair (s, a), and is defined as the expected total reward over all future time steps when taking action a in state s and following policy π afterwards,

$$q_\pi(s, a) = E_\pi(G_t|S_t = s, A_t = a) \tag{4.5}$$

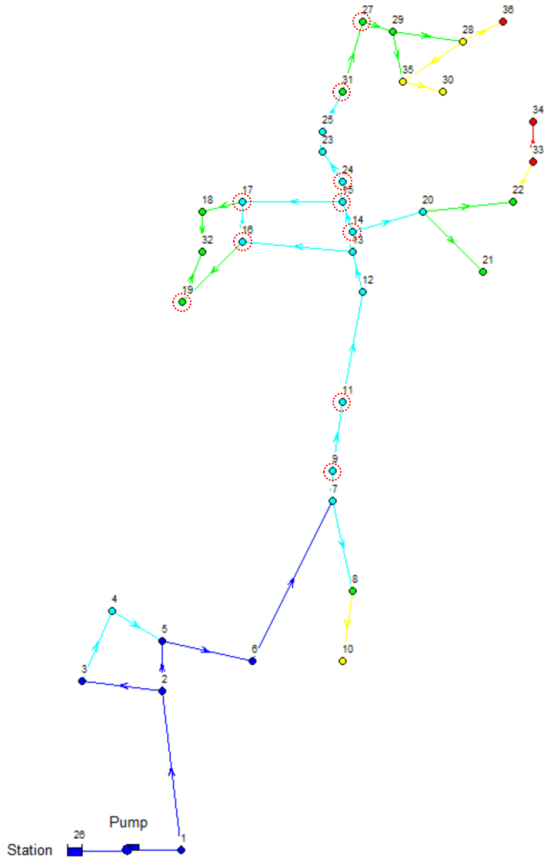


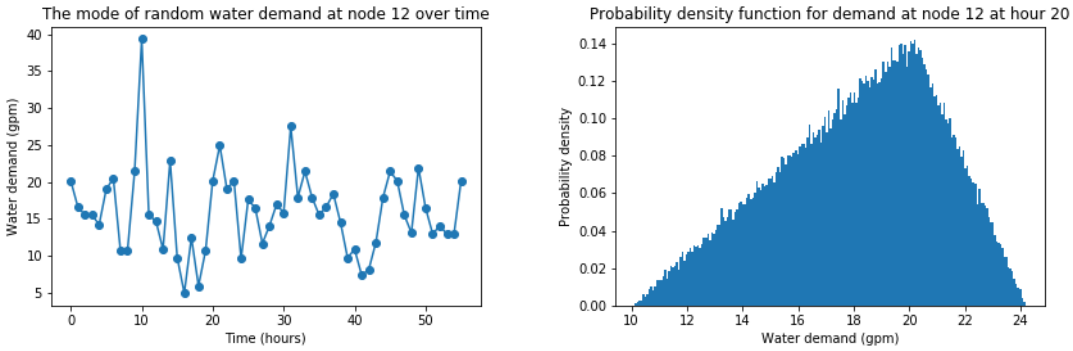
Figure 4.1: The Water Distribution Network for a City

Value functions are key to solve the stochastic optimal control problem, but they cannot be solved exactly using DP methods due to computation complexity. Instead, they are often approximated with different approaches.

4.3.2 Complete Formulation of the Optimal Control Problem

We will continue to use the city water distribution network (Figure 4.1) case for the model evaluation in this section.

This city water distribution network contains one reservoir node (labeled as Station in the network). The reservoir node supplies water to the system, connected to other nodes with a pump to provide enough pressure. The model assumes a finite



(a) The Mode of Random Water Demand at Node 12 Over Time

(b) The Probability Density Function of Water Demand at Node 12 at Hour 20

Figure 4.2: Modeling the Random Water Demand at Node 12

horizon of 55 hours. There are 32 nodes in the system representing different locations for consuming water, plus nodes 1, 28 and 35 which are just connection nodes with no water consumption (i.e., the demand is zero). All the consumption locations have a stochastic demand, and the demand distribution could be different for every hour. To model the demand at each location at each hour, the demand distribution in our case study is assumed to have a triangular distribution, where the minimum is 50% of its mode value and the maximum is 120% of its mode value (the users of this method could have any distribution for their problem). Figure 4.2 uses node 12 as an example to show what the random demand looks like. The mode of the stochastic demand at node 12 for all 55 hours are shown in Figure 4.2a, and Figure 4.2b shows the (simulated) probability density function of demand at node 12 at hour 20. Same as previous chapters, water quality is measured in terms of water age in hours. The water age at all locations with consumption demand are used as system states. The potential flushing control locations are identified using results from previous chapters, and are circled in red color in Figure 4.1. The associated EPANET simulation model is used to simulate the real water distribution system.

The parameters and variables used in the model formulation are summarized in Table 4.1.

Then the goal is to find a solution to the below optimization problem:

$$\max_{\pi}\{q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]\} \tag{4.6}$$

subject to

$$S_t \leq S_l \quad \forall 1 \leq t \leq T \tag{4.7}$$

$$S_{t+1} = f(S_t, A_t) \quad \forall 0 \leq t < T \tag{4.8}$$

where $G_t = R_{t+1} + R_{t+2} + ... + R_T$, R_t is the reward (negative cost) at time step t. The system starts from S_0 and follows the sequence $A_0, R_1, S_1, A_1, R_2, S_2, A_2, S_3, ...$ under a policy π, and $f(\cdot)$ represents the unknown system dynamics, the expectation $E_\pi(\cdot)$ is performed over the state distribution generated by the stochastic demand and control action (from using policy π).

The water age limit constraint (4.7) is eventually relaxed and added to the objective function with a Lagrangian penalty multiplier β, i.e.,

$$R_t = -C_t^f - C_t^p = -\Sigma_{i\in N_a} A_{i,t} - \beta\Sigma_{i\in N_s} \max(S_l - S_{i,t}, 0) \tag{4.9}$$

where C_t^f denotes the flushing cost, C_t^p denotes the penalty cost, β is the penalty cost ratio.

4.4 Solution Methods

Reinforcement learning problems are typically solved in an iterative way by interacting with the system in an online fashion. There are two main steps, value function evaluation and policy improvement. The value function evaluation step tries to approximate the value function of a policy, then the policy improvement step greedily

Table 4.1: Definition of Parameters and Variables in the Formulation of Optimal Flushing Control with Stochastic Demand

Parameters	Definition
T	the duration of system operation, i.e., final time step
t	index for time step, $t \in \{0, 1, 2, ...T\}$
S_l	the limit of system state (water age)
N_s	the set of locations to measure system states
N_a	the set of locations for flushing controls
β	penalty cost ratio for water age limit violation
Variables	**Definition**
S_t	the system state vector at time t for all locations in N_s, $S_t \in \mathbb{R}^+$
A_t	the control action vector at time t for all controlled locations in N_a, i.e., whether to trigger flushing for each controlled location at time t, $A_t \in \mathbb{B}$
R_t	the reward (negative cost) at time t
G_t	total reward starting from time t till the end of operation T
π	a control policy, it maps the state-action pair to a probability, i.e., $\pi(a\|s) \in [0, 1]$
$q_\pi(s, a)$	the action-value function associated with a policy π, defined as the total expected reward G_t starting from state s and taking action a at time t and following policy π afterwards, i.e., $q_\pi(s, a) = E_\pi(G_t\|S_t = s, A_t = a)$, and t is any time step

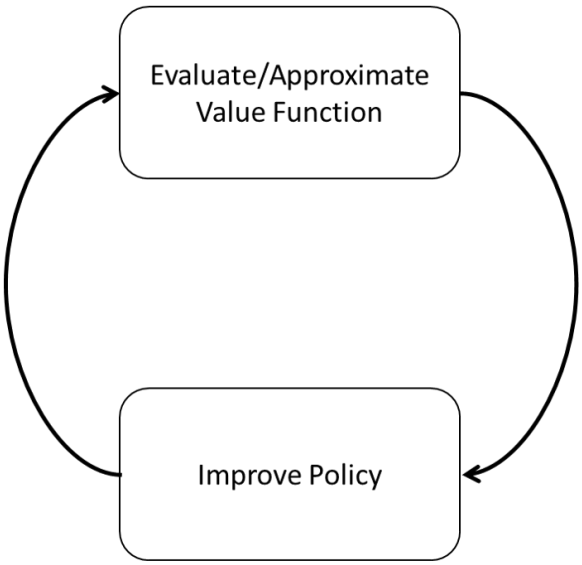


Figure 4.3: The General Process for Reinforcement Learning Approaches

selects the best action associated with each state. These two steps are performed iteratively, the process stops when the convergence is achieved, as illustrated by Figure 4.3.

The Bellman optimality equation is the foundation for reinforcement learning methods, and the iterative methods for solving the optimal control policy is based on Bellman's optimality equation. The Bellman optimality equation for the optimal action-value function $q_*(s, a)$ is

$$q_*(s, a) = E(R_{t+1} + \max_{a'} q_*(S_{t+1}, a')) \qquad \forall s, a \in \mathcal{A}(s) \tag{4.10}$$

where s denotes current system state at time t, $\mathcal{A}(s)$ is the set of possible actions that can be taken given current system state s at time t, $a \in \mathcal{A}(s)$ is the actual action that may be taken at time t, S_{t+1} is the system state (that has a stochastic property due to uncertain input to the system) at time $t + 1$ after taking action a, R_{t+1} is the reward generated at time $t + 1$, the expectation E is performed over all possible values of state S_{t+1}.

Note that in equation (4.10), the action-value function is specified for each state-action pair (s, a). However, water age and other water quality metrics in the water distribution system have continuous values. Enumerating all its values and storing the expected total reward for each possible value of the state-action is not realistic. A more efficient way to estimate the action-value function is needed, so that the value function for unseen states can be predicted. We will develop a weighted iterative online learning approach combined with value function approximation in Section 4.4.3.

We first explain two fundamental weighted iterative methods based on equation (4.10) in the following sections, then explain the value function approximation we have used in the case study.

4.4.1 SARSA: On-policy Temporal-Difference Learning Control

Temporal-difference (TD) learning combines the ideas of Monte Carlo and dynamic programming (DP). It uses previous experiences like Monte Carlo methods, and updates estimates using other learned estimates like DP (Sutton and Barto, 2018). Each experience is a sequence of $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, ...$, generated when control actions are applied at different time steps. SARSA algorithm updates $Q(s, a)$ at every time step t using this tuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ (and hence the name $SARSA$). The action-value function $Q(s, a)$ is updated in SARSA as follows:

$$Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + Q(S_{t+1}, A_{t+1})] \tag{4.11}$$

where $R_{t+1}+Q(S_{t+1}, A_{t+1})$ is the new estimated value for $Q(S_t, A_t)$. That is, $Q(S_t, A_t)$ is updated by a weighted average of its old estimate (with weight $1 - \alpha$) and the new estimate (with weigh α). The process generally converges faster when the weight α (often called step size) decreases over time.

SARSA is called "on-policy" TD learning because the policy it is optimizing is

also used to make explorations and learning. When the method does not concurrently update the policy and learn using the policy, then it is called an "off-policy" method. Q-learning is an off-policy TD learning control which we will discuss next.

4.4.2 Q-learning: Off-policy Temporal-Difference Learning Control

Q-learning (Watkins, 1989) algorithm is similar to SARSA, but unlike SARSA using all elements of the tuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ in the update, it does not need the second action A_{t+1}. Instead, it selects the best value of state S_{t+1}, and uses $R_{t+1} + \max_a Q(S_{t+1}, a)$ as the new estimate. The complete update rule is shown below.

$$Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \max_a Q(S_{t+1}, a)] \tag{4.12}$$

Note the similarity and difference between (4.11) and (4.12). The difference lies in the new estimate, one is using $Q(S_{t+1}, A_{t+1})$, the other is using $\max_a Q(S_{t+1}, a)$.

4.4.3 Value Function Approximation

In TD learning methods, when the state space is discrete, SARSA and Q-learning can be applied directly. But for continuous state, saving the expected total reward for every state-action pair after discretization in large tables is not computationally efficient, especially with high dimensional states. Parameterized methods are more promising in approximating the value function, as they do not have to save the numerical value of the value function for each state-action pair (s, a). Instead the method builds a parameterized model to fit the value function for all (s, a). Another advantage with parameterized methods is they can make predictions for the needed value function in the update (4.11) or (4.12) when the state has not been experienced.

We eventually used weighted linear models with approximate radial basis function

(RBF) kernels to estimate the action-value function $q_\pi(s, a)$.

A kernel is a transformation function that calculates the similarity of two states in the transformed space using original attributes. The RBF kernel is defined as

$$K(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}} \tag{4.13}$$

We did not use the RBF kernel directly in the value function approximation, instead, a kernel approximation method called RBFSampler from the Python scikit-learn package (Pedregosa *et al.*, 2011) was used to build features. RBFSampler approximates the feature map of an RBF kernel by Monte Carlo approximation of its Fourier transform. The RBFSampler in the scikit-learn package implements a variant of Random Kitchen Sinks by Rahimi and Recht (2008).

The linear models with approximate radial basis function (RBF) kernels for action-value function approximation can be abstractly expresses as:

$$\hat{q}(s, a, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\Phi}(s, a) = \Sigma_{i=1}^{d} w_i \Phi_i(s, a) \tag{4.14}$$

where $\boldsymbol{\Phi}(s, a)$ represents the approximate RBF kernels coming from the RBFSampler method, d is total approximate RBF kernels used, $\boldsymbol{w}$ is the coefficients of the linear model (linear in terms of $\boldsymbol{\Phi}(s, a)$).

Because an approximation for the action-value function is used, and it is characterized by its coefficients $\boldsymbol{w}$, updating $\hat{q}(s, a, \boldsymbol{w})$ is equivalent to updating $\boldsymbol{w}$. The update of $\boldsymbol{w}$ in the SARSA TD learning control using stochastic gradient descent approach is as follows,

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} + \boldsymbol{w}_t^T \boldsymbol{\Phi}(S_{t+1}, A_{t+1}) - \boldsymbol{w}_t^T \boldsymbol{\Phi}(S_t, A_t)]\boldsymbol{\Phi}(S_t, A_t) \tag{4.15}$$

The framework of learning based control process is presented in the schematic diagram of Figure 4.4. The whole process is repeated in a loop, by continuously

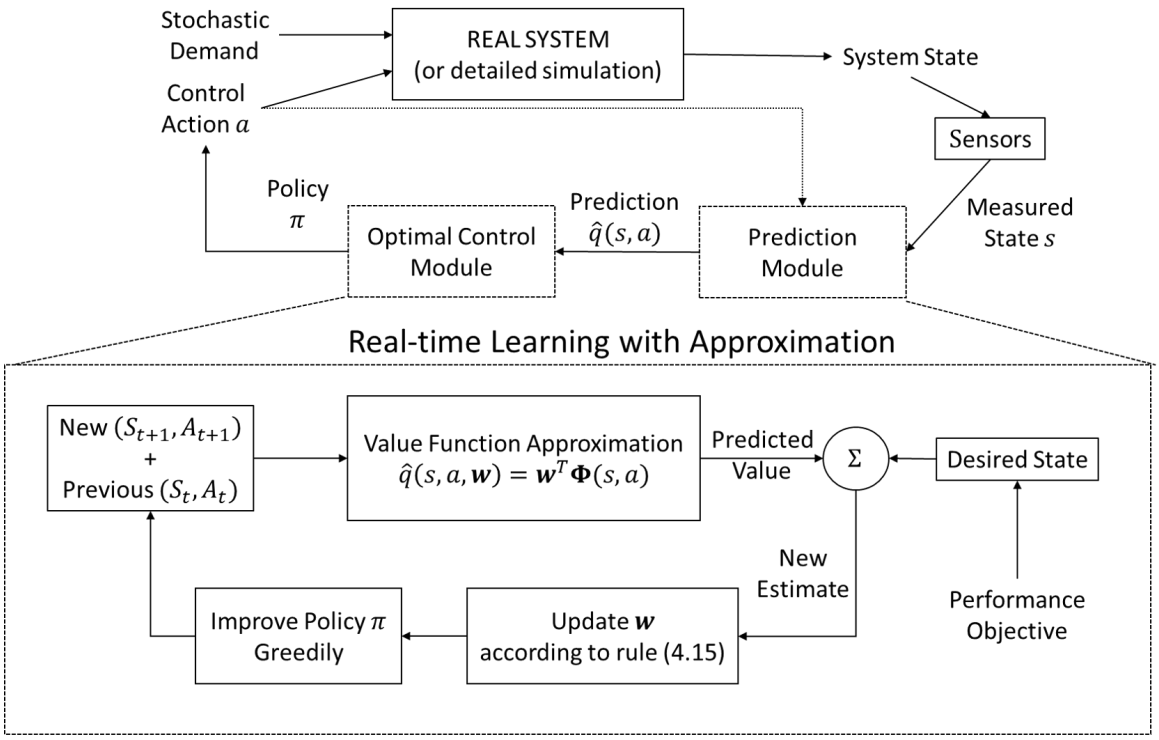


Figure 4.4: The Diagram for Real-time Learning-based Optimal Control Process with Value Function Approximation

observing new states from the system, approximating their value functions using a linear model, updating existing value function estimates using new data, taking action, and observing states... until reaching the end of the operation horizon.

The complete SARSA TD learning control algorithm with linear value function approximation using approximate RBF kernels is given in Figure 4.5.

4.5 Experiments and Results

The complete method was successfully implemented with Python code, and was run for different experiments with different parameters. The settings for all experiments are summarized in Table 4.2.

The total reward (negative cost) over episodes/simulations for Experiments 2 to

Set number of episodes (simulations) N

Set number of time steps in each episode T (horizon)

Set step size α

Set small ϵ (for ϵ-greedy policy)

Set water age limit S_l

Set penalty cost ratio β

Set the approximate RBF kernels $\boldsymbol{\Phi}$

Initialize weights $\boldsymbol{w}_0$ (for the linear model to approximate value function $\hat{q}$) arbitrarily

For $i = 1, 2, 3, \dots, N$

    Initialize state S_0

    Choose action A_0 from ϵ-greedy policy derived from $\hat{q}(s, a, \boldsymbol{w}_0) = \boldsymbol{w}_0^T \boldsymbol{\Phi}(\mathrm{s}, \mathrm{a})$

    For $t = 0, 1, 2, \dots, T$

        Take action A_t, observe S_{t+1}

        Calculate flushing cost C_{t+1}^f and penalty cost C_{t+1}^p

        Calculate equivalent reward $R_{t+1} \leftarrow -C_{t+1}^f - C_{t+1}^p$

        Choose action A_{t+1} using ϵ-greedy policy derived from $\hat{q}(s, a, \boldsymbol{w}) = \boldsymbol{w}_t^T \boldsymbol{\Phi}$

        Update $\boldsymbol{w}$:

        $$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[R_{t+1} + \boldsymbol{w}_t^T \boldsymbol{\Phi}(\mathrm{S_{t+1}}, \mathrm{A_{t+1}}) - \boldsymbol{w}_t^T \boldsymbol{\Phi}(\mathrm{S_t}, \mathrm{A_t})]\boldsymbol{\Phi}(\mathrm{S_t}, A_t)$$

        $S_t \leftarrow S_{t+1}$

        $A_t \leftarrow A_{t+1}$

Figure 4.5: The SARSA TD Learning Control Algorithm with Linearly Weighted Value Function Approximation Using Approximate RBF Kernels

Table 4.2: Parameter Settings for Different Reinforcement Learning Experiments

Experiment #	Update Rule	# of Approx RBF Kernels	Penalty Cost β	Total Episodes
1	Q-learning	5	100	1000
2	Q-learning	5	100	2000
3	SARSA	5	100	2000
4	SARSA	5	20	2000
5	SARSA	5	10	2000
6	SARSA	5	2	2000
7	SARSA	5	10	1000
8	SARSA	4	10	1000

8 are shown in Figures 4.6 to 4.12.

The trend of total reward over episodes in Experiment 2 (Figure 4.6) is not really converging. This experiment applied value function approximation, bootstraping (updating value function estimates using existing estimates) and off-policy training. Sutton and Barto (2018) have pointed out that these three elements (which they referred to as *the deadly triad*) could cause divergence when used together. This might explain why the result of Experiment 2 did not converge after 2000 episodes.

Comparing Experiment 2 and Experiment 3 (Figure 4.6 vs Figure 4.7), we can notice the result is improved a lot by replacing Q-learning update rule (4.12) with SARSA update rule (4.11). This comparison once again verifies the statement about *the deadly triad* by Sutton and Barto (2018).

In Experiments 3 to 6, different penalty cost β was used to see how it affects the learning result. By comparing Figure 4.7 to Figure 4.10, we can find that the

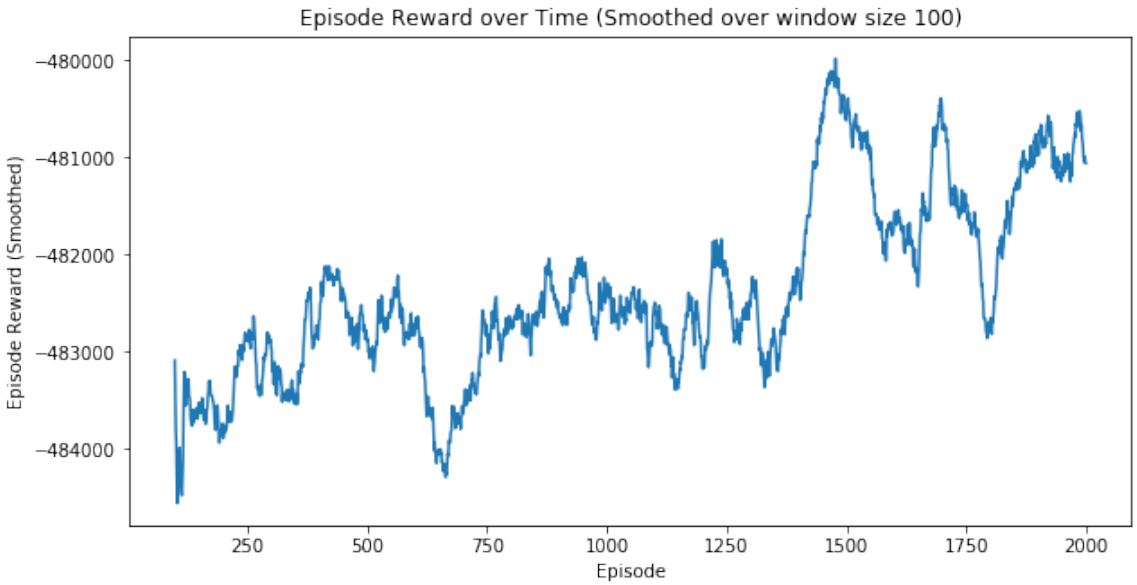


Figure 4.6: Experiment 2: Total Reward (Negative Cost) over Episodes (Simulations)

convergence is reached earlier in the latter cases. In Figure 4.7 with $\beta = 100$, the convergence trend is still not very obvious. In Figure 4.8 with $\beta = 20$, it is starting to fluctuate in a much smaller range after about 1000 episodes. In Figure 4.9 with $\beta = 10$, the convergence is reached after about 600 episodes. In Figure 4.10 with $\beta = 2$, the total reward quickly converges after about 500 episodes.

Experiment 7 and Experiment 8 are implemented to see the impact of number of RBFSamplers used in the linear model. The results (Figure 4.11 vs Figure 4.12) are very similar, with Figure 4.12 showing slightly better convergence. We can conclude that in our problem setting, 4 or 5 RBFSamplers do not make a difference. But the features used in the linear models to approximate the value function are surely important, and they affect the prediction accuracy directly.

After these experiments were done, the policy derived through learning using the algorithm in Figure 4.5 from Experiment 5 is compared with two other policies, a rule-based policy (to trigger flushing at all controlled locations whenever the observed

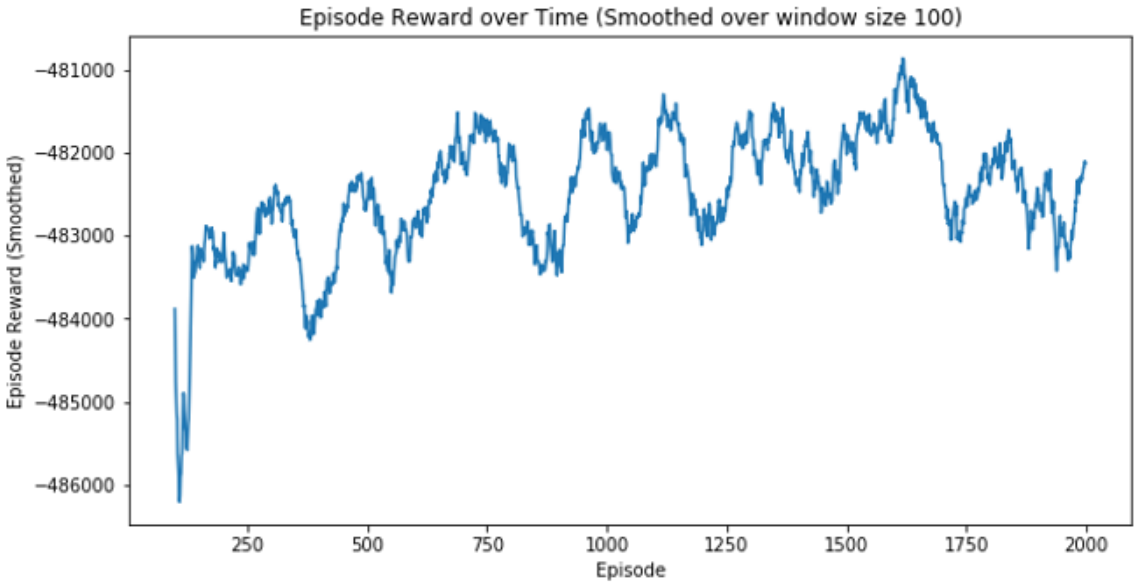


Figure 4.7: Experiment 3: Total Reward (Negative Cost) over Episodes (Simulations)

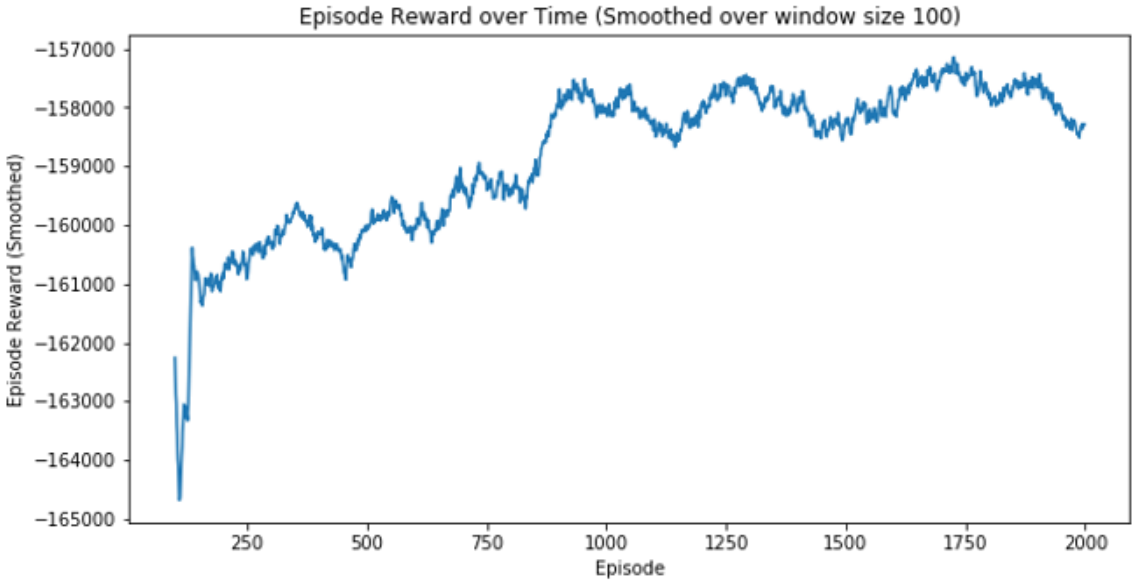


Figure 4.8: Experiment 4: Total Reward (Negative Cost) over Episodes (Simulations)

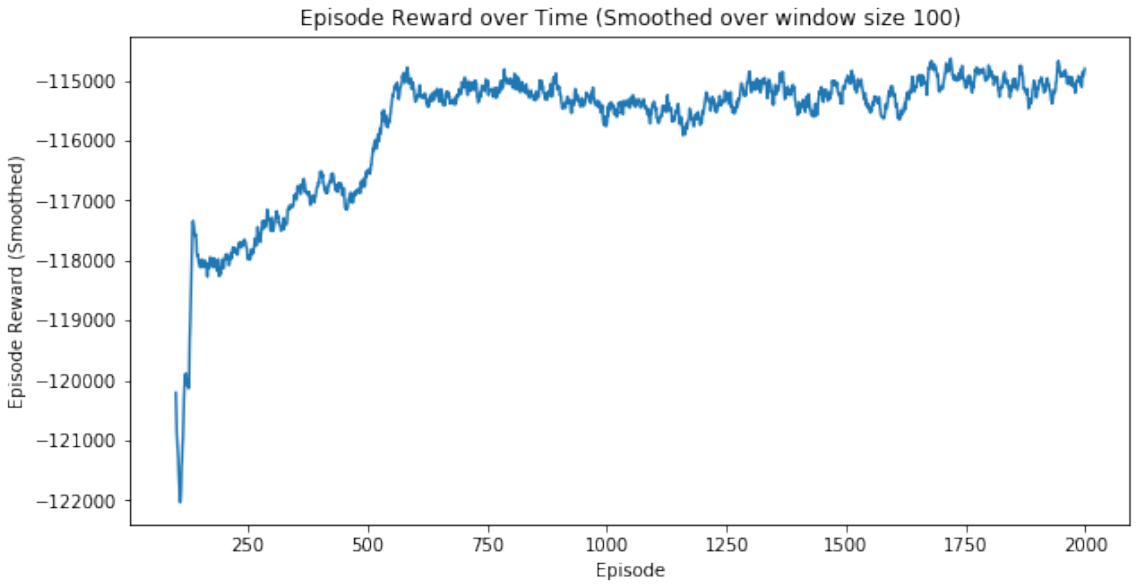


Figure 4.9: Experiment 5: Total Reward (Negative Cost) over Episodes (Simulations)

Figure 4.10: Experiment 6: Total Reward (Negative Cost) over Episodes (Simulations)

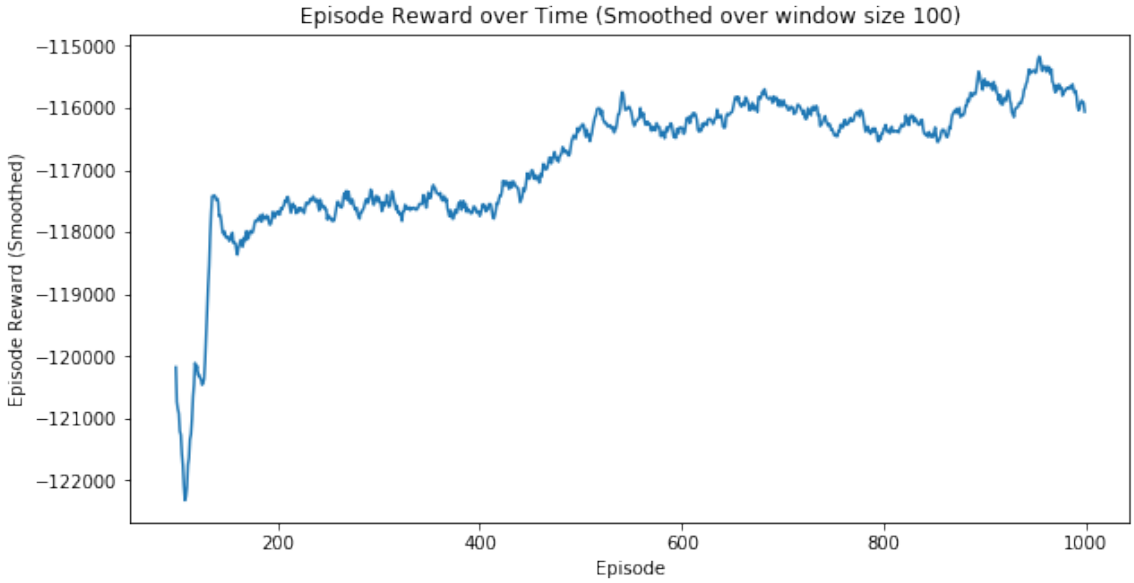


Figure 4.11: Experiment 7: Total Reward (Negative Cost) over Episodes (Simulations)

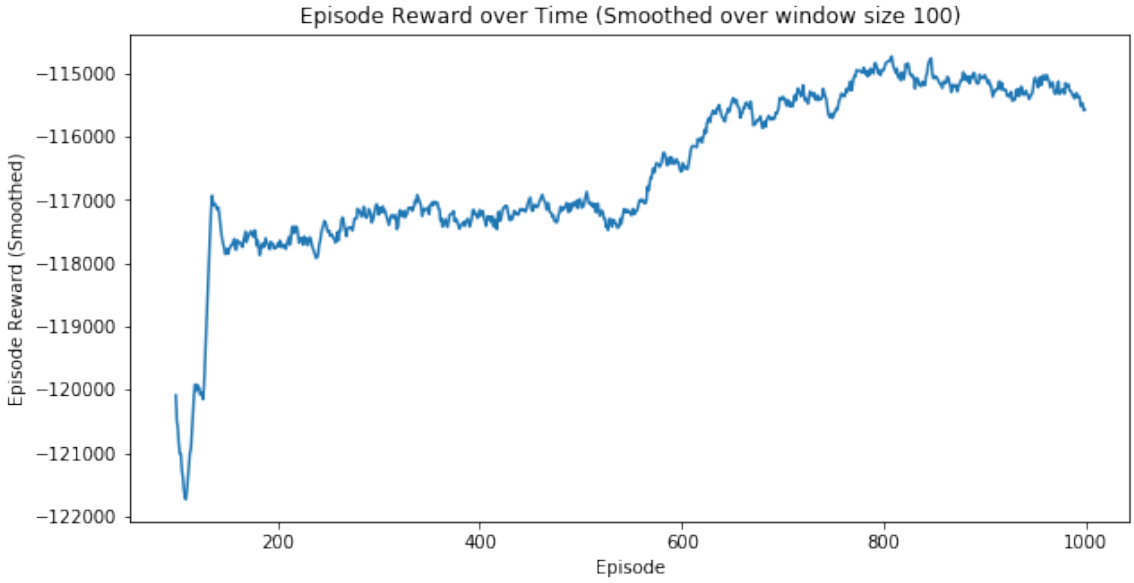


Figure 4.12: Experiment 8: Total Reward (Negative Cost) over Episodes (Simulations)

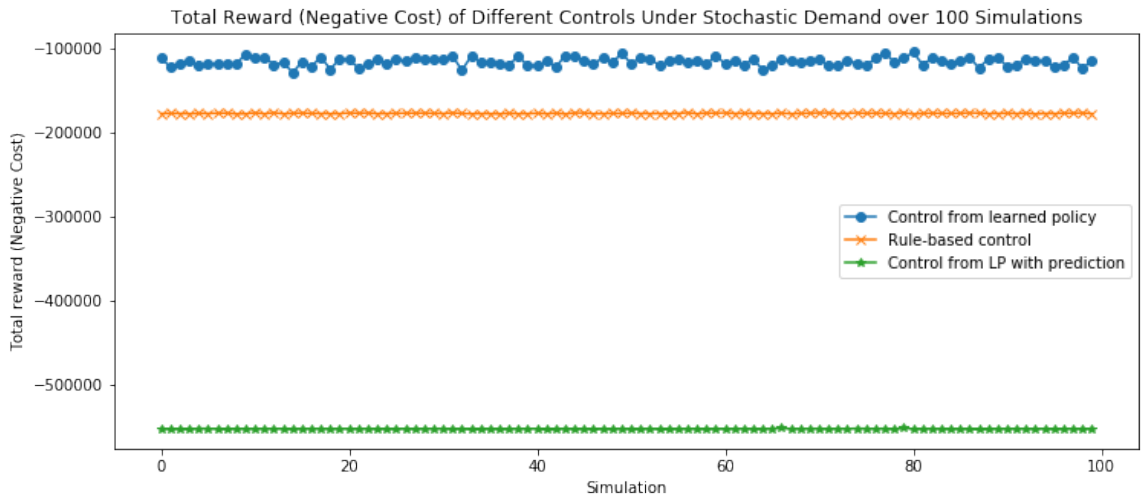


Figure 4.13: The Comparison of Total Reward (Negative Cost) for Three Flushing Control Policies

water age is above the limit), and the optimal flushing schedule policy from LP model in Section 2.2.4. All three policies are used for flushing control in 100 simulation runs with stochastic demand (triangular distribution, Figure 4.2). The total reward in all 100 runs from these three policies are plotted in Figure 4.13. The average cost of each control policy is given in Table 4.3.

We can observe that the learned flushing control policy performs consistently better than the other two policies. Note that the optimal flushing schedule from the LP model performs very badly in these 100 simulation runs, because (1) the flushing schedule is solved for a problem with deterministic water demand; and (2) the somewhat poor accuracy of water age prediction models used in the LP formulation (although the R^2 for all used models is above 75%, it may not be good enough for controlling the water distribution system when demand is stochastic).

The actual binary flushing decisions from the learned policy (from Experiment 5) and from the rule-based policy for one simulation are shown in Figure 4.14a and Figure 4.14b, respectively. The horizontal dimension shows different locations, the

Table 4.3: The Average Cost of Three Flushing Control Policies over 100 Simulations

	Learned Policy	Rule-based Policy	Policy from LP
Average Cost	116,033	177,634	553,418

corresponding node IDs are (9, 11, 14, 15, 16, 17, 19, 24, 27, 31). The vertical dimension shows different hours, from 0 to 55. We can observe that the flushing decisions from the learned policy varies for different hours, while the rule-based policy triggers flushing controls after hour 9 and remains flushing at all locations till hour 55.

4.6 Conclusions

The water demand in water distribution systems could indeed be stochastic due to people's uncertain water-consuming activities. Thus, water quality control in such systems involves sequential decision making under stochastic uncertain demand patterns. Reinforcement learning (RL) based approaches generally can deal with the stochastic uncertainty fairly well, because they iteratively learn the demand patterns and system dynamics from past system behaviors with various demands. RL methods are essentially computational methods approximate dynamic programs (ADP). In ADP approaches, the value function can be estimated through interactive learning. However, for systems with continuous high dimensional states, it is difficult to apply the dynamic programming logic without further approximation.

The research in this chapter is focused on the problem of water quality (water age) control through flushing, in presence of stochastic and uncertain demand. We developed a real-time control system that is based on SARSA temporal-difference learning combined with linear value function approximation for controlling water

```
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [1, 0, 1, 0, 0, 1, 1, 0, 1, 0],
 [0, 1, 1, 0, 0, 0, 1, 0, 1, 0],
 [0, 1, 1, 1, 0, 0, 0, 1, 1, 0],
 [0, 0, 0, 1, 0, 0, 0, 0, 1, 0],
 [0, 0, 0, 1, 0, 1, 0, 0, 1, 0],
 [0, 1, 1, 1, 0, 1, 0, 0, 1, 0],
 [1, 0, 1, 1, 1, 1, 0, 0, 0, 1],
 [1, 1, 0, 1, 1, 0, 0, 1, 0, 0],
 [1, 1, 0, 1, 0, 0, 0, 1, 0, 0],
 [0, 0, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 0, 0, 1, 0, 0, 1, 0, 0, 0],
 [1, 1, 1, 1, 0, 1, 1, 1, 0, 0],
 [0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
 [0, 0, 1, 1, 0, 0, 1, 0, 1, 0],
 [0, 1, 1, 0, 1, 1, 0, 1, 1, 0],
 [0, 1, 1, 0, 0, 0, 1, 0, 1, 0],
 [0, 0, 0, 0, 0, 1, 1, 0, 0, 0],
 [1, 1, 1, 0, 1, 1, 0, 0, 1, 0],
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
 [0, 0, 1, 1, 0, 0, 1, 0, 1, 1],
 [0, 1, 1, 0, 0, 1, 1, 0, 1, 0],
 [0, 0, 0, 1, 1, 0, 0, 1, 0, 1],
 [0, 0, 0, 0, 1, 0, 0, 1, 1, 1],
 [0, 0, 1, 1, 0, 1, 0, 0, 1, 0],
 [0, 1, 0, 1, 0, 0, 0, 1, 1, 0],
 [0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
 [1, 0, 1, 1, 0, 0, 1, 1, 0, 1],
 [0, 0, 1, 0, 1, 1, 0, 1, 1, 1],
 [0, 1, 1, 0, 0, 1, 1, 1, 1, 1],
 [1, 0, 0, 1, 1, 1, 1, 0, 1, 1],
 [0, 0, 1, 1, 1, 1, 1, 0, 0, 1],
 [0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
 [0, 0, 1, 1, 0, 1, 1, 0, 0, 0],
 [0, 1, 1, 1, 0, 1, 0, 0, 0, 0],
 [0, 0, 0, 1, 0, 1, 1, 1, 1, 0],
 [1, 0, 0, 0, 0, 0, 1, 1, 1, 0],
 [0, 1, 1, 0, 0, 1, 1, 0, 1, 0],
 [0, 1, 1, 1, 0, 1, 0, 0, 1, 0],
 [1, 1, 0, 0, 1, 1, 1, 1, 0, 0],
 [1, 0, 1, 0, 1, 1, 0, 1, 1, 0],
 [1, 0, 0, 1, 0, 1, 1, 0, 1, 1],
 [0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
 [0, 1, 0, 0, 0, 0, 1, 1, 1, 1],
 [0, 1, 0, 1, 0, 0, 0, 0, 0, 1],
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
 [0, 0, 1, 1, 0, 1, 0, 1, 1, 1],
 [1, 1, 1, 0, 1, 1, 0, 1, 0, 1],
 [1, 0, 1, 1, 0, 0, 0, 0, 0, 1],
 [0, 0, 1, 1, 0, 1, 0, 1, 1, 1],
 [1, 0, 0, 0, 1, 0, 1, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0, 0, 1, 1]]
```

(a) The Binary Flushing Control Actions over Time at Different Locations from Learned Policy

```
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
```

(b) The Binary Flushing Control Actions over Time at Different Locations from Rule-based Policy

Figure 4.14: The Actual Flushing Decisions for One Simulation

quality via flushing controls. The policy for flushing control actions derived from the approximated action-value functions is shown to perform better than a rule-based policy and the control policy derived from a linear programming optimization model.

The water quality control for water distribution systems through flushing of taps is a control problem with continuous states and continuous control actions. In this chapter, the high dimensional states are handled by linear approximate models for value functions, and we assumed binary flushing control action for each location. A future research direction is to relax this binary flushing control assumption, and replace it with multiple discrete values for the control action. Even further, to model the flushing as a continuous value between zero and a maximum amount. Neural network models, especially deep neural networks (having multiple hidden layers) can be helpful in this research direction.

Flushing through taps is effective in improving water quality, but it is not the only method. We also modeled the water quality control problem by including both flushing and disinfectant (chlorine) injection in Chapter 3, but only flushing is used as the control action in this chapter. Another future research direction is to include both flushing and disinfectant injections as the control actions in optimal water quality control problem with stochastic demand.

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions and Dissertation Contributions

This dissertation addresses the optimal control problem of water quality in water distribution networks. Different proactive and real-time optimal control models with water quality constraints were developed for water distribution networks in various scenarios, (1) when water demands are deterministic, when predictions of water age are linear, and objective for flushing control can be approximated as linear, (2) when demands are deterministic but now controls include both flushing and chlorine injections that lead to non-linear control methods, and (3) when water demands are stochastic and uncertain where the control method resulted in learning both the demand and the dynamics of the water distribution system in an online fashion. The last control method used reinforcement learning with temporal difference learning and linear value function approximation.

Modeling the dynamics of water distribution systems is challenging due to the complex fluid dynamics and chemical reactions. In Chapter 2, different statistical machine learning approaches were explored to build approximate models for water system dynamics. Three case studies were performed to evaluate the efficacy of different approaches. Then these approximate models for water system dynamics were integrated in a linear programming (LP) based optimization model for the flushing control. Flushing was selected as the control action because it is effective in improving water quality as it quickly brings fresher water from the water treatment plant. The LP based optimization model for flushing control was solved using LP solver, and

the derived flushing control was applied to EPANET simulation models for a 5-story building and a city water distribution network. The simulation results demonstrated that the flushing control from the solution of the LP model was very effective.

In Chapter 3, two control actions were considered in the control of water age: flushing and disinfectant injections. Adding chlorine as a disinfectant is recommended by CDC to control water quality. With regard to the coupled control actions in this part of the research, their synergistic impacts on water quality makes the optimal control of water quality challenging. Two mixed integer quadratically constrained optimization models were developed for controlling water age through flushing and chlorine injections. The first model assumes continuous flushing and binary chlorine injections; the second model assumes continuous flushing and continuous chlorine injections. Problem instances with different cost ratio parameters were solved using the Gurobi solver for mathematical programming. The computational results from problem instances of these models gave us meaningful insights about the relationship of flushing control and chlorine injection control. The inclusion of chlorine injection control can significantly reduce the total control cost when compared with flushing-only control. Continuous chlorine injections can further reduce the total cost, saving about half of the chlorine control cost, when compared with binary chlorine injection controls.

In Chapter 2 and Chapter 3, water demands were assumed to be deterministic and known. However, the water demand in water distribution systems could indeed be stochastic due to people's fluctuating water-consuming activities. Chapter 4 addresses the problem of optimal water quality control for water systems with stochastic and uncertain demand. Based on approximate dynamic programming and reinforcement learning, this research developed a real-time control system that is based on SARSA temporal-difference learning combined with linear value function approxima-

tion for controlling water quality (water age) via flushing controls. The policy for flushing control actions derived from the approximate value functions is shown to perform better than a rule-based policy and the control policy derived from LP-based optimization.

5.2 Directions for Future Research

In Chapter 2, different machine learning models were used to approximate the water system dynamics. But the accuracy of some of the developed models can be further improved. Potential future work is to fine tune these models to improve their accuracy, so that their improved accuracy can help improve the proactive control actions obtained from any associated optimization model utilized. Another future direction of research is to consider different water quality indicators that can be measured with high accuracy. Water age is a good proxy indicator of water stagnation, but may not be measured as easily as other water quality indicators. It will be an interesting research direction to combine different water quality indicators into one metric. Multi-objective optimization models and approaches can be helpful in corresponding problems.

In Chapter 3, a set of potential chlorine control locations were identified by observing locations with high water age due to local stagnation. A future research direction could be to develop a systematic method for identifying candidate locations for chlorine injection controls. One implicit assumption in developing our formulations was that the injected chlorine at one location only affects water quality (water age) at that location while later in time at other locations due to water dynamics and water age prediction models. This assumption can be modified by an appropriate one that considers chlorine decay and the effective coverage of injected chlorine. Therefore, another future research direction could be on setting up a coverage range of instanta-

neous impact for the injected chlorine, and explicitly incorporating a chlorine decay model in formulating the optimal control problem.

The water quality control for water distribution systems through flushing of taps is a control problem with continuous states and continuous control actions. In Chapter 4, the high dimensional states were handled by linear approximate models for value functions, and binary flushing control action was assumed at each location. A future research direction is to relax this binary flushing control assumption, and replace it with multiple discrete values for the control action. Even further, to model the flushing as a continuous value between zero and a maximum amount. Neural network models, especially deep neural networks (having multiple hidden layers) can be helpful in this research direction (see Bertsekas and Tsitsiklis (1996)).

Flushing through taps is effective in improving water quality, but it is not the only method. In Chapter 3, the water quality control model is formulated by including both flushing and disinfectant (chlorine) injections, but only flushing is used as the control action in the scenario with stochastic and uncertain demand. Another future research direction is to include both flushing and disinfectant injections as the control actions in the optimal water quality control problem with stochastic demand.

REFERENCES

Alcocer-Yamanaka, V. H., V. G. Tzatchkov and F. I. Arreguin-Cortes, "Modeling of drinking water distribution networks using stochastic demand", Water resources management **26**, 7, 1779–1792 (2012).

Allaire, M., H. Wu and U. Lall, "National trends in drinking water quality violations", Proceedings of the National Academy of Sciences **115**, 9, 2078–2083 (2018).

Bédard, E., C. Laferrière, E. Déziel and M. Prévost, "Impact of stagnation and sampling volume on water microbial quality monitoring in large buildings", PLoS One **13**, 6, e0199429 (2018).

Benedict, K. M., H. Reses, M. Vigar, D. M. Roth, V. A. Roberts, M. Mattioli, L. A. Cooley, E. D. Hilborn, T. J. Wade, K. E. Fullerton *et al.*, "Surveillance for waterborne disease outbreaks associated with drinking water—united states, 2013–2014", MMWR. Morbidity and mortality weekly report **66**, 44, 1216 (2017).

Berardi, L., O. Giustolisi, Z. Kapelan and D. Savic, "Development of pipe deterioration models for water distribution systems using epr", Journal of Hydroinformatics **10**, 2, 113–126 (2008).

Bertsekas, D. P., "Dynamic programming and optimal control 3rd edition, volume ii", Belmont, MA: Athena Scientific (2011).

Bertsekas, D. P. and J. N. Tsitsiklis, *Neuro-dynamic programming*, vol. 5 (Athena Scientific Belmont, MA, 1996).

Bertsekas, D. P. *et al.*, *Dynamic programming and optimal control: Vol. 1* (Athena scientific Belmont, 2000).

Bhattacharya, B., A. Lobbrecht and D. Solomatine, "Neural networks and reinforcement learning in control of water systems", Journal of Water Resources Planning and Management **129**, 6, 458–465 (2003).

Brooks, W., S. Corsi, M. Fienen and R. Carvin, "Predicting recreational water quality advisories: A comparison of statistical methods", Environmental modelling & software **76**, 81–94 (2016).

Cassell, K., J. L. Davis and R. Berkelman, "Legionnaires' disease in the time of covid-19", Pneumonia **13**, 1, 1–3 (2021).

Castelletti, A., S. Galelli, M. Restelli and R. Soncini-Sessa, "Tree-based reinforcement learning for optimal water reservoir operation", Water Resources Research **46**, 9 (2010).

Castelletti, A., F. Pianosi and M. Restelli, "A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run", Water Resources Research **49**, 6, 3476–3486 (2013).

Castelletti, A., H. Yajima, M. Giuliani, R. Soncini-Sessa and E. Weber, "Planning the optimal operation of a multioutlet water reservoir with water quality and quantity targets", Journal of Water Resources Planning and Management **140**, 4, 496–510 (2014).

CDC, "Disinfection by-products", Centers for Disease Control and Prevention URL `https://www.cdc.gov/safewater/chlorination-byproducts.html` (2016).

CDC, "Developing a water management program to reduce legionella growth & spread in buildings", Centers for Disease Control and Prevention URL `https://www.cdc.gov/legionella/downloads/toolkit.pdf` (2017).

Cembrano, G., G. Wells, J. Quevedo, R. Pérez and R. Argelaguet, "Optimal control of a water distribution network in a supervisory control system", Control engineering practice **8**, 10, 1177–1188 (2000).

Chen, H. and C. Ni-Bin, "Water pollution control in the river basin by fuzzy genetic algorithm-based multiobjective programming modeling", Water Science and Technology **37**, 8, 55 (1998).

Cho, J. H., K. S. Sung and S. R. Ha, "A river water quality management model for optimising regional wastewater treatment using a genetic algorithm", Journal of Environmental Management **73**, 3, 229–242 (2004).

De Somer, O., A. Soares, K. Vanthournout, F. Spiessens, T. Kuijpers and K. Vossen, "Using reinforcement learning for demand response of domestic hot water buffers: A real-life demonstration", in "2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)", pp. 1–7 (IEEE, 2017).

Dhar, A. and B. Datta, "Optimal operation of reservoirs for downstream water quality control using linked simulation optimization", Hydrological Processes: An International Journal **22**, 6, 842–853 (2008).

Everard, M., "Meeting global drinking water needs", Nature Sustainability **2**, 5, 360–361 (2019).

Hornik, K., "Approximation capabilities of multilayer feedforward networks", Neural networks **4**, 2, 251–257 (1991).

Hozalski, R. M., T. M. LaPara, X. Zhao, T. Kim, M. B. Waak, T. Burch and M. McCarty, "Flushing of stagnant premise water systems after the covid-19 shutdown can reduce infection risk by legionella and mycobacterium spp.", Environmental Science & Technology **54**, 24, 15914–15924 (2020).

Islam, N., R. Sadiq and M. J. Rodriguez, "Optimizing locations for chlorine booster stations in small water distribution networks", Journal of Water Resources Planning and Management **143**, 7, 04017021 (2017).

Jain, A., F. Smarra, M. Behl and R. Mangharam, "Data-driven model predictive control with regression trees—an application to building energy management", ACM Transactions on Cyber-Physical Systems **2**, 1, 4 (2018).

Jia, X., A. Karpatne, J. Willard, M. Steinbach, J. Read, P. C. Hanson, H. A. Dugan and V. Kumar, "Physics guided recurrent neural networks for modeling dynamical systems: Application to monitoring water temperature and quality in lakes", arXiv preprint arXiv:1810.02880 (2018).

Kalman, R. E. *et al.*, "Contributions to the theory of optimal control", Bol. soc. mat. mexicana **5**, 2, 102–119 (1960).

Kang, D. and K. Lansey, "Real-time optimal valve operation and booster disinfection for water quality in water distribution systems", Journal of Water Resources Planning and Management **136**, 4, 463–473 (2010).

Kiran, B. R., I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey", IEEE Transactions on Intelligent Transportation Systems (2021).

Konidaris, G., S. Osentoski and P. Thomas, "Value function approximation in reinforcement learning using the fourier basis", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 25 (2011).

Koza, J. R., *Genetic programming: on the programming of computers by means of natural selection*, vol. 1 (MIT press, 1992).

Kuo, J.-T., Y.-Y. Wang and W.-S. Lung, "A hybrid neural–genetic algorithm for reservoir water quality management", Water research **40**, 7, 1367–1376 (2006).

Kurek, W. and A. Ostfeld, "Multiobjective water distribution systems control of pumping cost, water quality, and storage-reliability constraints", Journal of Water Resources Planning and Management **140**, 2, 184–193 (2012).

Lasdon, L. S. and A. D. Waren, "Grg2 user's guide", School of Business Administration, University of Texas at Austin (1986).

Lautenschlager, K., N. Boon, Y. Wang, T. Egli and F. Hammes, "Overnight stagnation of drinking water in household taps induces microbial growth and changes in community composition", Water research **44**, 17, 4868–4877 (2010).

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural computation **1**, 4, 541–551 (1989).

Lee, J.-H. and J. W. Labadie, "Stochastic optimization of multireservoir systems via reinforcement learning", Water resources research **43**, 11 (2007).

Ley, C. J., C. R. Proctor, G. Singh, K. Ra, Y. Noh, T. Odimayomi, M. Salehi, R. Julien, J. Mitchell, A. P. Nejadhashemi *et al.*, "Drinking water microbiology in a water-efficient building: stagnation, seasonality, and physicochemical effects on opportunistic pathogen and total bacteria proliferation", Environmental Science: Water Research & Technology **6**, 10, 2902–2913 (2020).

Ling, F., R. Whitaker, M. W. LeChevallier and W.-T. Liu, "Drinking water microbiome assembly induced by water stagnation", The ISME journal **12**, 6, 1520–1531 (2018).

Liu, Y., Y. Zheng, Y. Liang, S. Liu and D. S. Rosenblum, "Urban water quality prediction based on multi-task multi-view learning", (2016).

Madani, K. and M. Hooshyar, "A game theory–reinforcement learning (gt–rl) method to develop optimal operation policies for multi-operator reservoir systems", Journal of hydrology **519**, 732–742 (2014).

Maier, H. R. and G. C. Dandy, "The use of artificial neural networks for the prediction of water quality parameters", Water resources research **32**, 4, 1013–1022 (1996).

Maier, H. R. and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications", Environmental modelling & software **15**, 1, 101–124 (2000).

Mala-Jetmarova, H., N. Sultanova and D. Savic, "Lost in optimisation of water distribution systems? a literature review of system operation", Environmental modelling & software **93**, 209–254 (2017).

Mocanu, E., D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning", IEEE transactions on smart grid **10**, 4, 3698–3708 (2018).

Najah, A., A. El-Shafie, O. A. Karim and A. H. El-Shafie, "Application of artificial neural networks for water quality prediction", Neural Computing and Applications **22**, 1, 187–201 (2013).

Osiadacz, A. J., "Integer and combinatorial optimization, george l. nemhauser and laurence a. wolsey, wiley-interscience series in discrete mathematics and optimization, new york, 1988, isbn 0-471-82819-x, 763pp,£ 71.90", (1990).

Palani, S., S.-Y. Liong and P. Tkalich, "An ann application for water quality forecasting", Marine Pollution Bulletin **56**, 9, 1586–1597 (2008).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python", Journal of machine learning research **12**, Oct, 2825–2830 (2011).

Powell, W. B., *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703 (John Wiley & Sons, 2007).

Proctor, C. R., W. J. Rhoads, T. Keane, M. Salehi, K. Hamilton, K. J. Pieper, D. M. Cwiertny, M. Prévost and A. J. Whelton, "Considerations for large building water quality after extended stagnation", AWWA water science **2**, 4, e1186 (2020).

Qiao, J.-F., Y.-C. Bo, W. Chai and H.-G. Han, "Adaptive optimal control for a wastewater treatment plant based on a data-driven method", Water science and technology **67**, 10, 2314–2320 (2013).

Rahimi, A. and B. Recht, "Weighted sums of random kitchen sinks: replacing minimization with randomization in learning.", in "Nips", pp. 1313–1320 (Citeseer, 2008).

Reid, S., R. Tibshirani and J. Friedman, "A study of error variance estimation in lasso regression", Statistica Sinica pp. 35–67 (2016).

Rhoads, W. J., A. Pruden and M. A. Edwards, "Survey of green building water systems reveals elevated water age and water quality concerns", Environmental Science: Water Research & Technology **2**, 1, 164–173 (2016).

Rossman, L. A., "The epanet programmer's toolkit for analysis of water distribution systems", in "WRPMD'99: Preparing for the 21st Century", pp. 1–10 (1999).

Ruelens, F., B. J. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets and R. Belmans, "Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning", in "2014 Power Systems Computation Conference", pp. 1–7 (IEEE, 2014).

Saetta, D., A. Padda, X. Li, C. Leyva, P. B. Mirchandani, D. Boscovic and T. H. Boyer, "Real-time monitoring and control of urea hydrolysis in cyber-enabled non-water urinal system", Environmental science & technology **53**, 6, 3187–3197 (2019).

Sakarya, A. B. A. and L. W. Mays, "Optimal operation of water distribution pumps considering water quality", Journal of Water Resources Planning and Management **126**, 4, 210–220 (2000).

Sallab, A. E., M. Abdou, E. Perot and S. Yogamani, "Deep reinforcement learning framework for autonomous driving", Electronic Imaging **2017**, 19, 70–76 (2017).

Sargent, R., "Optimal control", Journal of Computational and Applied Mathematics **124**, 1-2, 361–371 (2000).

Savic, D. A., "The use of data-driven methodologies for prediction of water and wastewater asset failures", in "Risk Management of Water Supply and Sanitation Systems", pp. 181–190 (Springer, 2009).

Searson, D., "Gptips: Genetic programming and symbolic regression for matlab", (2009).

Shalev-Shwartz, S., S. Shammah and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving", arXiv preprint arXiv:1610.03295 (2016).

Shin, J., H. J. Kim, S. Park and Y. Kim, "Model predictive flight control using adaptive support vector regression", Neurocomputing **73**, 4-6, 1031–1037 (2010).

Singh, K. P., A. Basant, A. Malik and G. Jain, "Artificial neural network modeling of the river water quality—a case study", Ecological Modelling **220**, 6, 888–895 (2009).

Singh, K. P. and S. Gupta, "Artificial intelligence based modeling for predicting the disinfection by-products in water", Chemometrics and Intelligent Laboratory Systems **114**, 122–131 (2012).

Sondermeijer, O., R. Dobbe, D. Arnold, C. Tomlin and T. Keviczky, "Regression-based inverter control for decentralized optimal power flow and voltage regulation", arXiv preprint arXiv:1902.08594 (2019).

Storey, M. V., B. Van der Gaag and B. P. Burns, "Advances in on-line drinking water quality monitoring and early warning systems", Water research **45**, 2, 741–747 (2011).

Sundström, O., D. Ambühl and L. Guzzella, "On implementation of dynamic programming for optimal control problems with final state constraints", Oil & Gas Science and Technology–Revue de l'Institut Français du Pétrole **65**, 1, 91–102 (2010).

Sutton, R. S. and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

Swanson, E., "Poll: About half of americans are 'very confident' in tap water", The Christian Science Monitor URL `https://www.csmonitor.com/Environment/2016/0305/Poll-About-half-of-Americans-are-very-confident-in-tap-water` (2016).

Tibshirani, R., "Regression shrinkage and selection via the lasso", Journal of the Royal Statistical Society: Series B (Methodological) **58**, 1, 267–288 (1996).

Trueman, B. F., S. A. MacIsaac, A. K. Stoddart and G. A. Gagnon, "Prediction of disinfection by-product formation in drinking water via fluorescence spectroscopy", Environmental Science: Water Research & Technology **2**, 2, 383–389 (2016).

Tu, M.-Y., F. T.-C. Tsai and W. W.-G. Yeh, "Optimization of water distribution and water quality by hybrid genetic algorithm", Journal of water resources planning and management **131**, 6, 431–440 (2005).

US Environmental Protection Agency, "Watersense at work: best management practices for commercial and institutional facilities", (2012).

Wang, H., S. Masters, M. A. Edwards, J. O. Falkinham III and A. Pruden, "Effect of disinfectant, water age, and pipe materials on bacterial and eukaryotic community structure in drinking water biofilm", Environmental science & technology **48**, 3, 1426–1435 (2014).

Wang, Y., J. Zhou, K. Chen, Y. Wang and L. Liu, "Water quality prediction method based on lstm neural network", in "2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)", pp. 1–5 (IEEE, 2017).

Wang, Z. and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges", Applied Energy **269**, 115036 (2020).

Watkins, C. J. C. H., "Learning from delayed rewards", (1989).

WHO, "Chlorine in drinking-water", World Health Organization URL `https://www.who.int/water_sanitation_health/dwq/chlorine.pdf` (2003).

Wu, W., G. Dandy and H. Maier, "Optimal control of total chlorine and free ammonia levels in a water transmission pipeline using artificial neural networks and genetic algorithms", Journal of Water Resources Planning and Management **141**, 7, 04014085 (2015a).

Wu, Z. Y., M. El-Maghraby and S. Pathak, "Applications of deep learning for smart water networks", Procedia Engineering **119**, 479–485 (2015b).

Yoo, B. and J. Kim, "Path optimization for marine vehicles in ocean currents using reinforcement learning", Journal of Marine Science and Technology **21**, 2, 334–343 (2016).

Zhang, Z., A. Chong, Y. Pan, C. Zhang, S. Lu and K. P. Lam, "A deep reinforcement learning approach to using whole building energy model for hvac optimal control", in "2018 Building Performance Analysis Conference and SimBuild", vol. 3, pp. 22–23 (2018).

Zhou, Y., B. Jin and C. J. Spanos, "Learning convex piecewise linear machine for data-driven optimal control", in "2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)", pp. 966–972 (IEEE, 2015).

APPENDIX A

THE OPTIMAL FLUSHING SCHEDULES FOR DIFFERENT WATER AGE LIMIT FOR CASE III EXTENSION

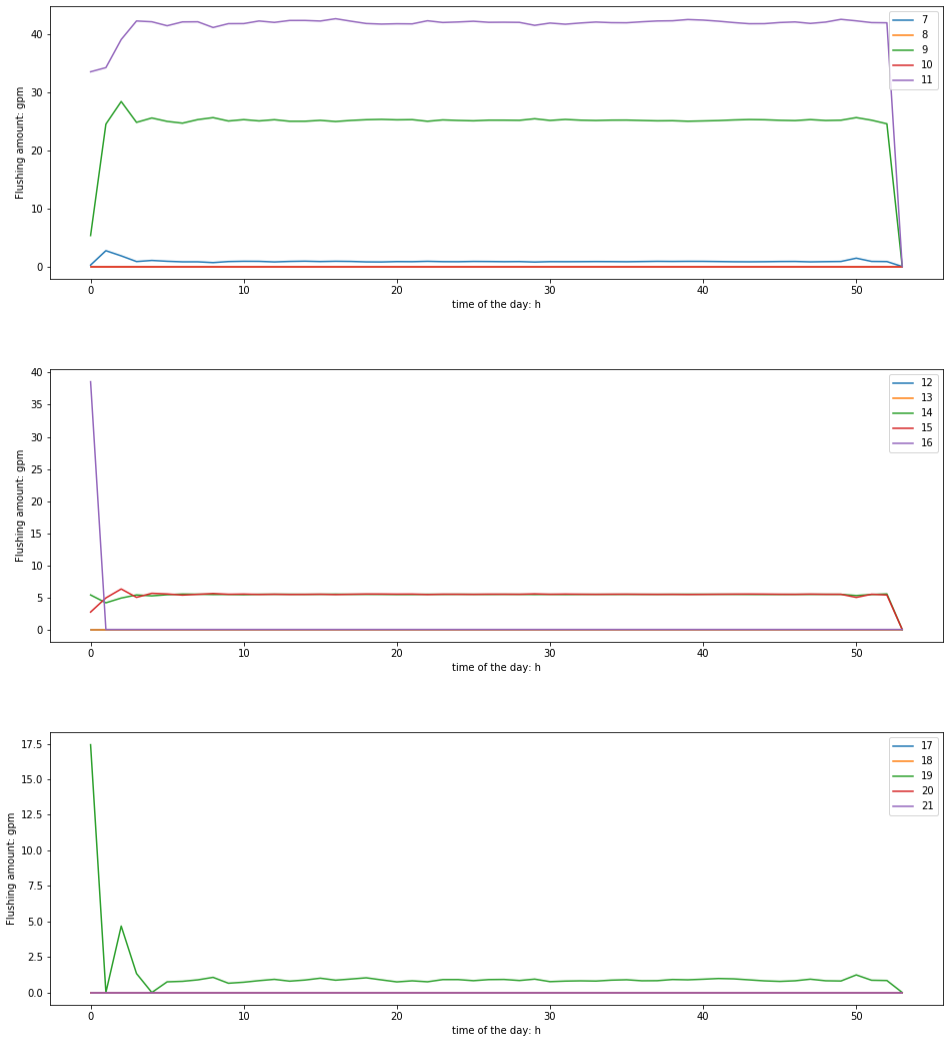


Figure A.1: The Actual Flushing over Time at Different Locations for $a_l = 11$

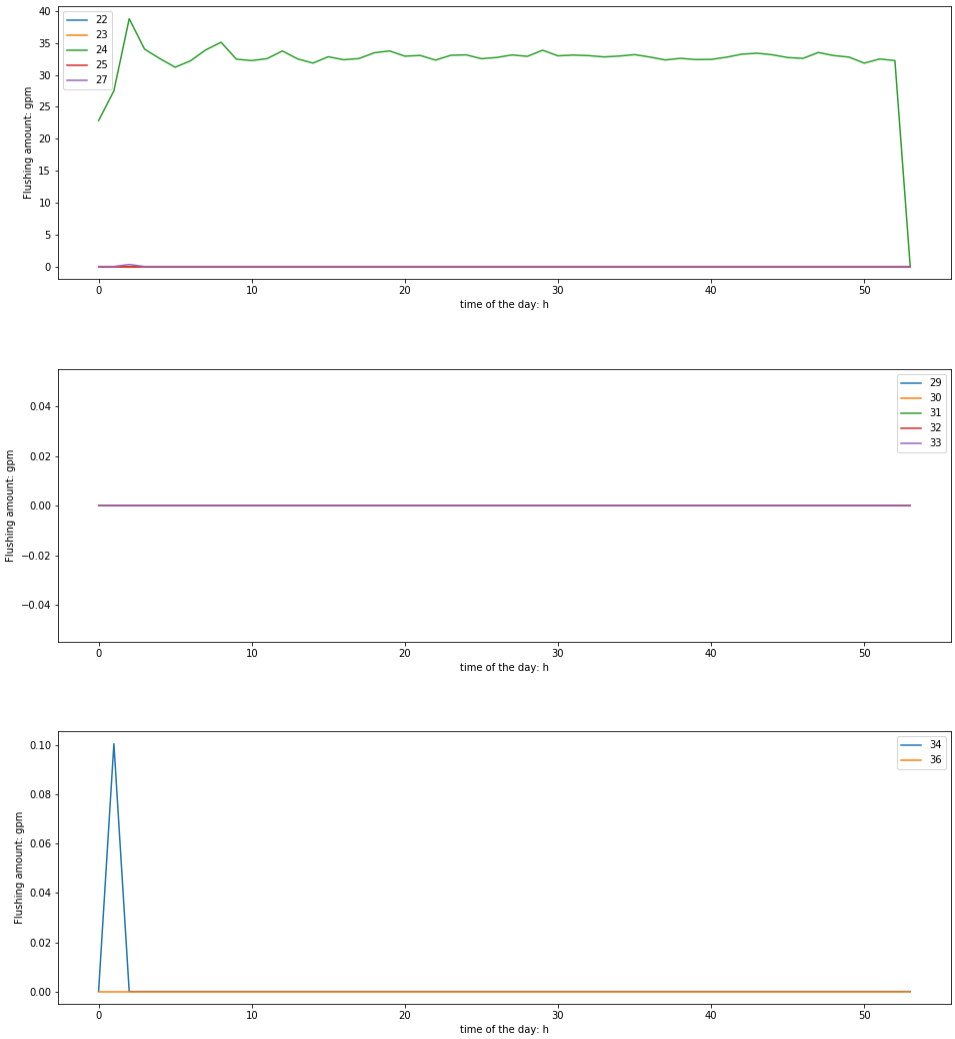


Figure A.1: The Actual Flushing over Time at Different Locations for $a_l = 11$ (continued)

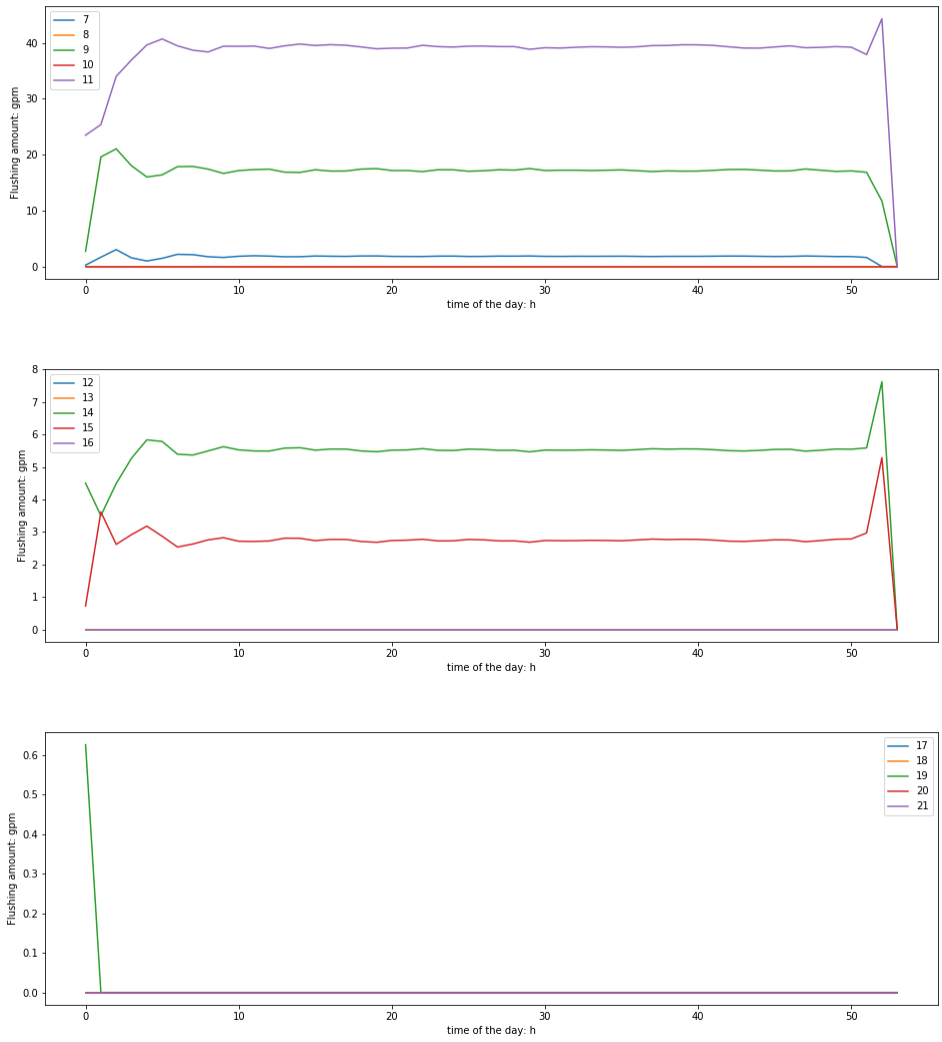


Figure A.2: The Actual Flushing over Time at Different Locations for $a_l = 12$

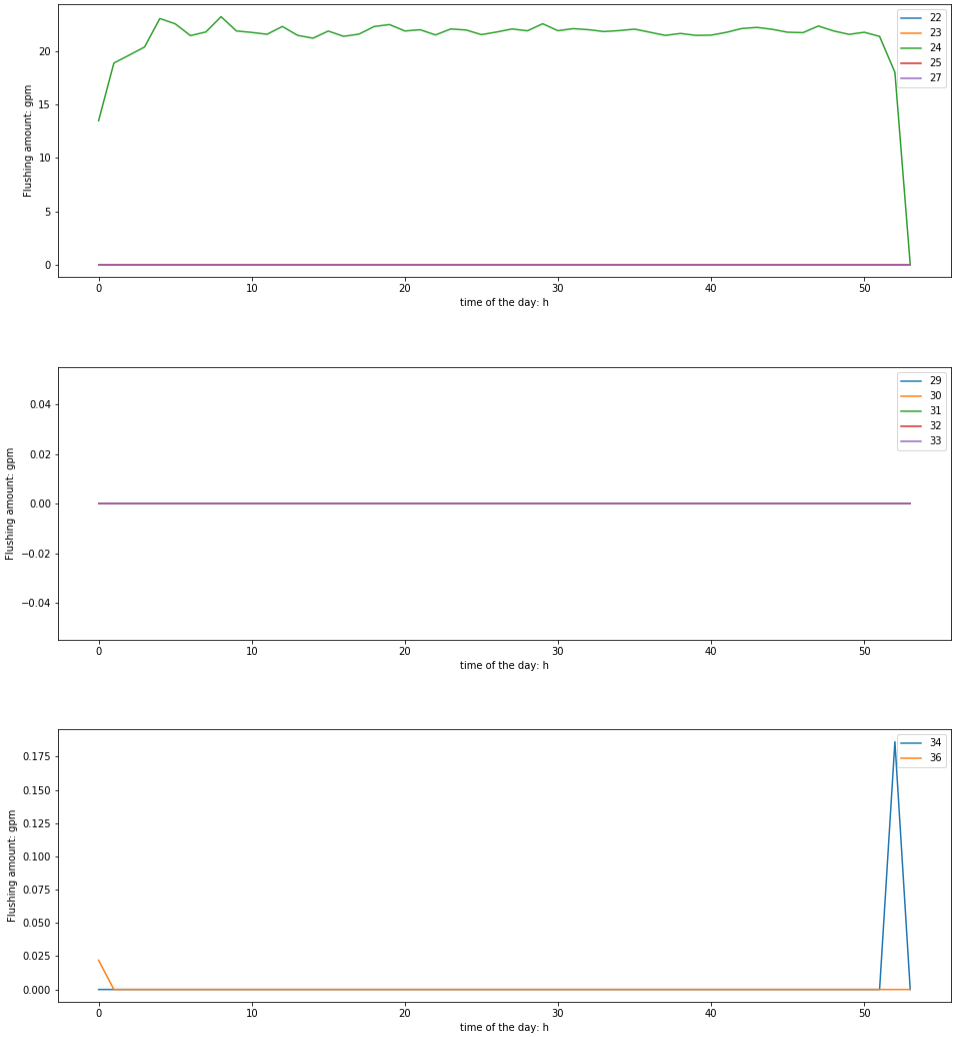


Figure A.2: The Actual Flushing over Time at Different Locations for $a_l = 12$ (continued)

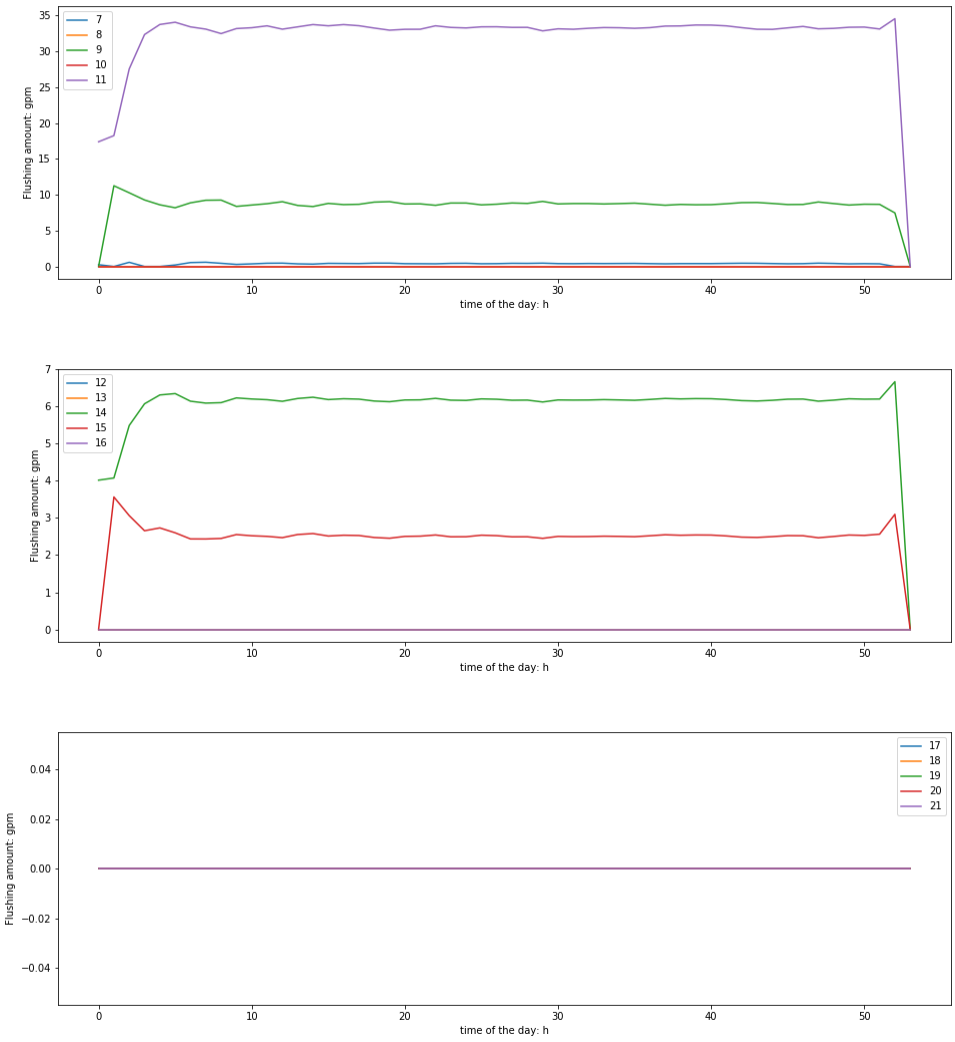


Figure A.3: The Actual Flushing over Time at Different Locations for $a_l = 13$

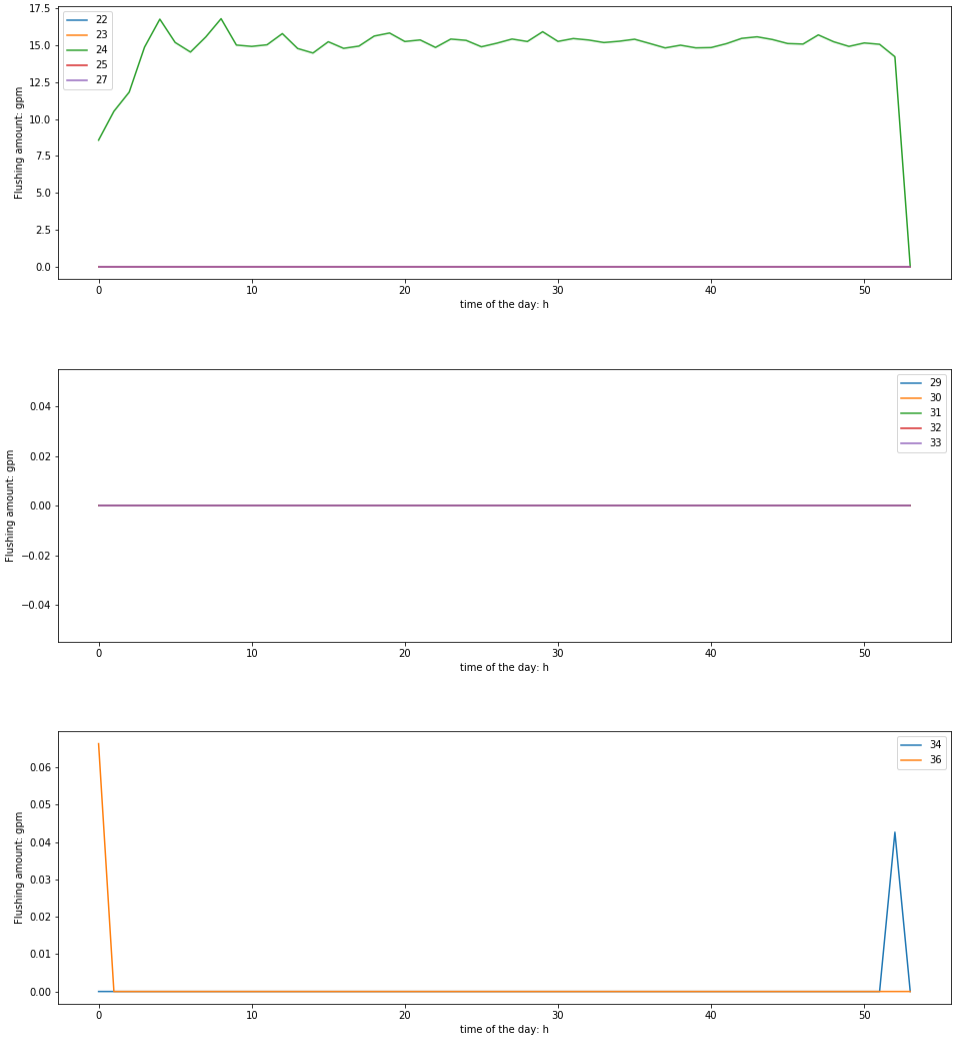


Figure A.3: The Actual Flushing over Time at Different Locations for $a_l = 13$ (continued)

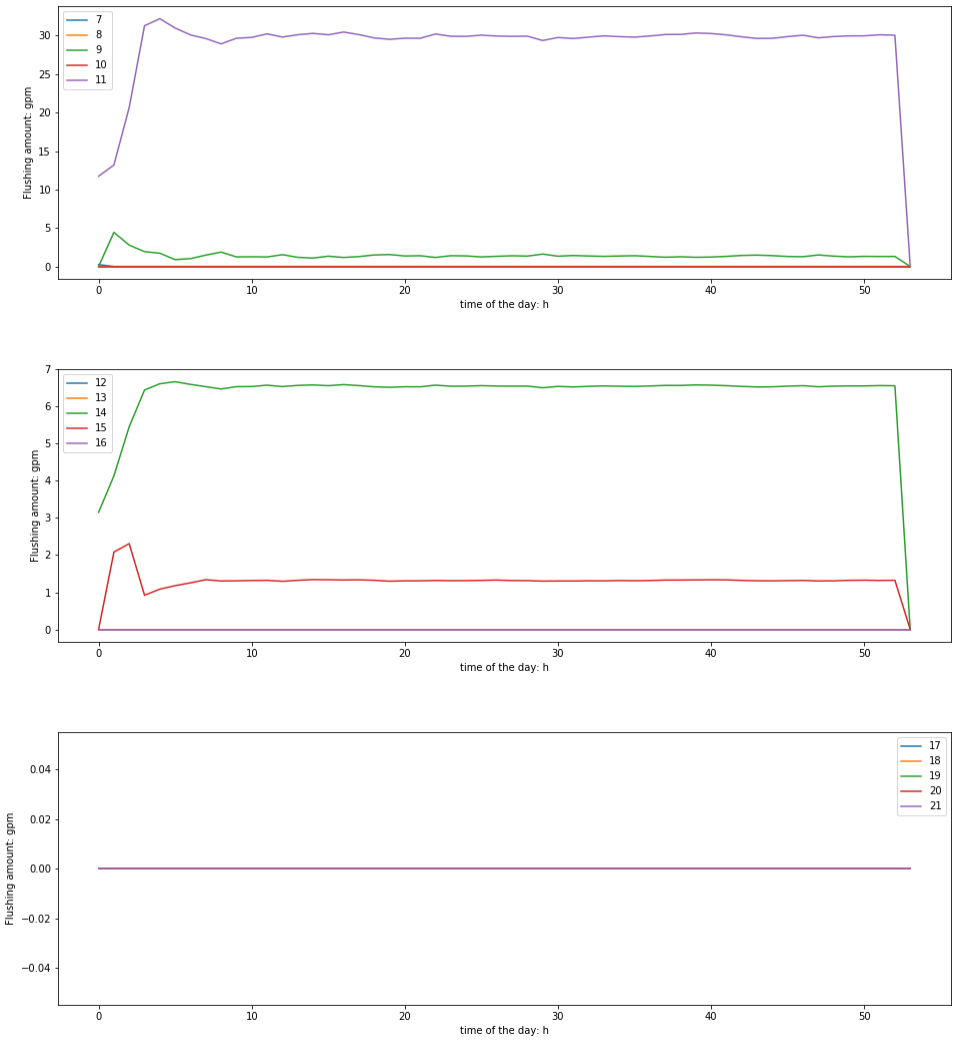


Figure A.4: The Actual Flushing over Time at Different Locations for $a_l = 14$

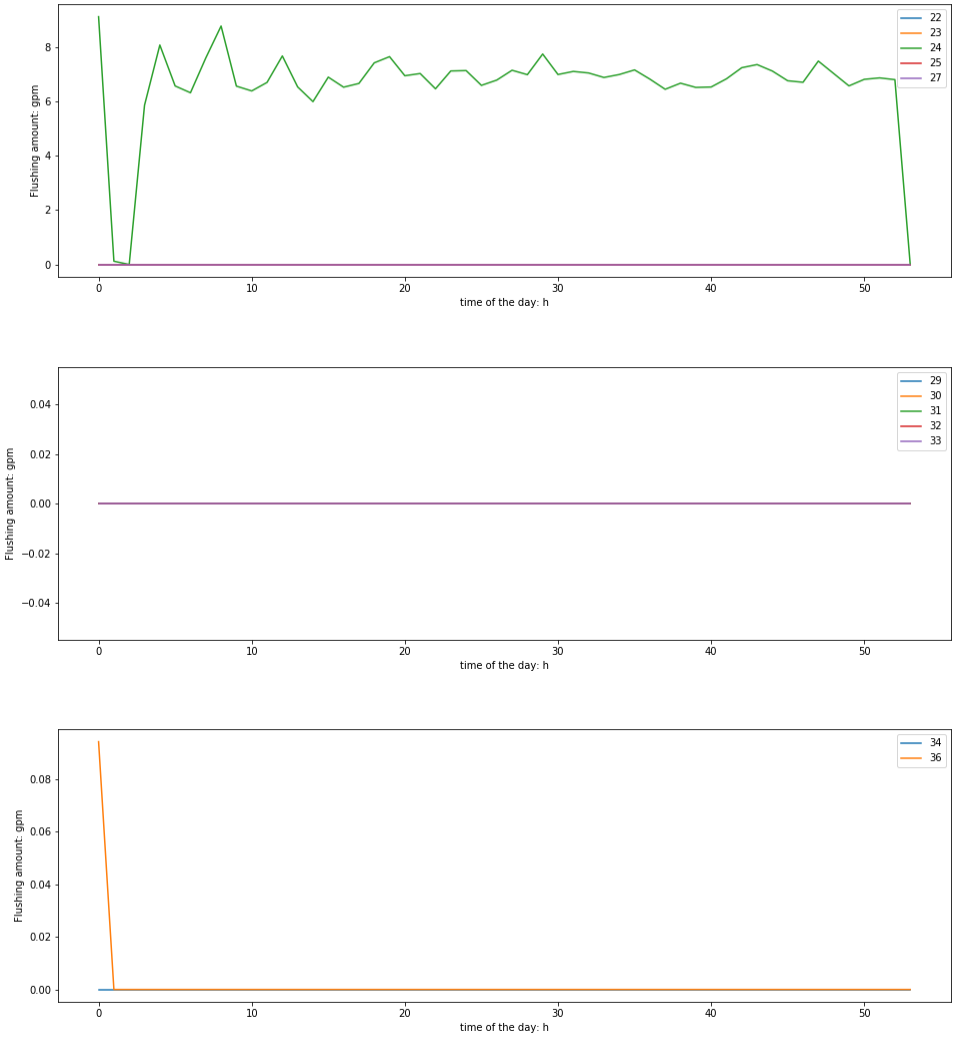


Figure A.4: The Actual Flushing over Time at Different Locations for $a_l = 14$ (continued)

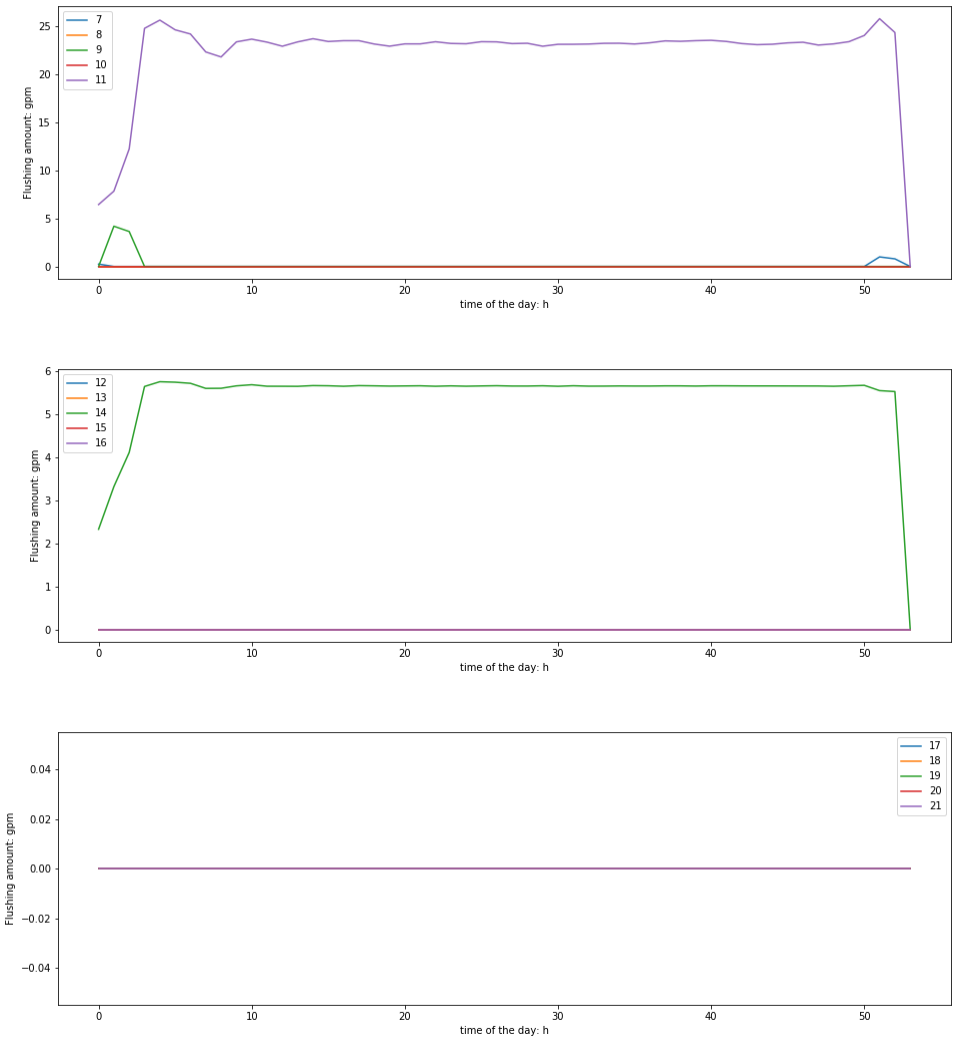


Figure A.5: The Actual Flushing over Time at Different Locations for $a_l = 15$

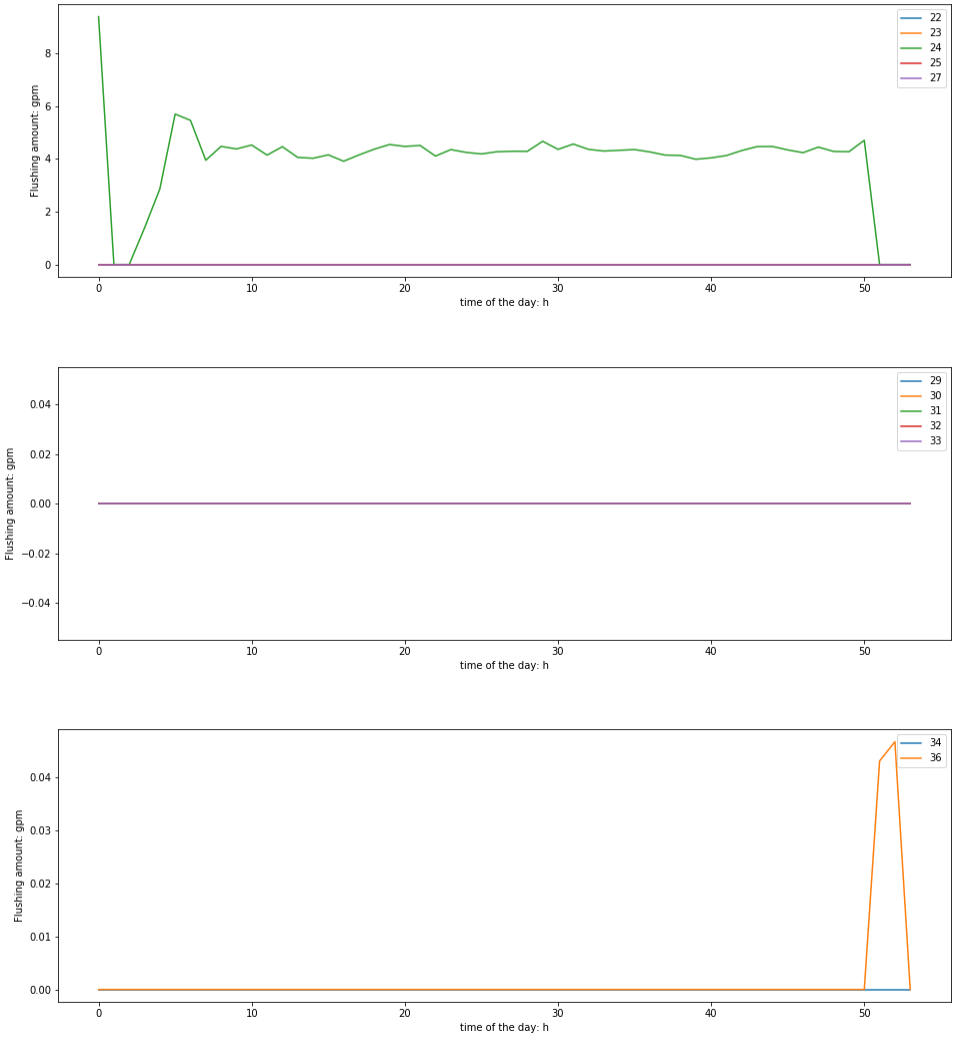


Figure A.5: The Actual Flushing over Time at Different Locations for $a_l = 15$ (continued)

APPENDIX B

THE ACTUAL LOCATIONS WITH FLUSHING FOR DIFFERENT WATER AGE LIMIT FOR CASE III EXTENSION

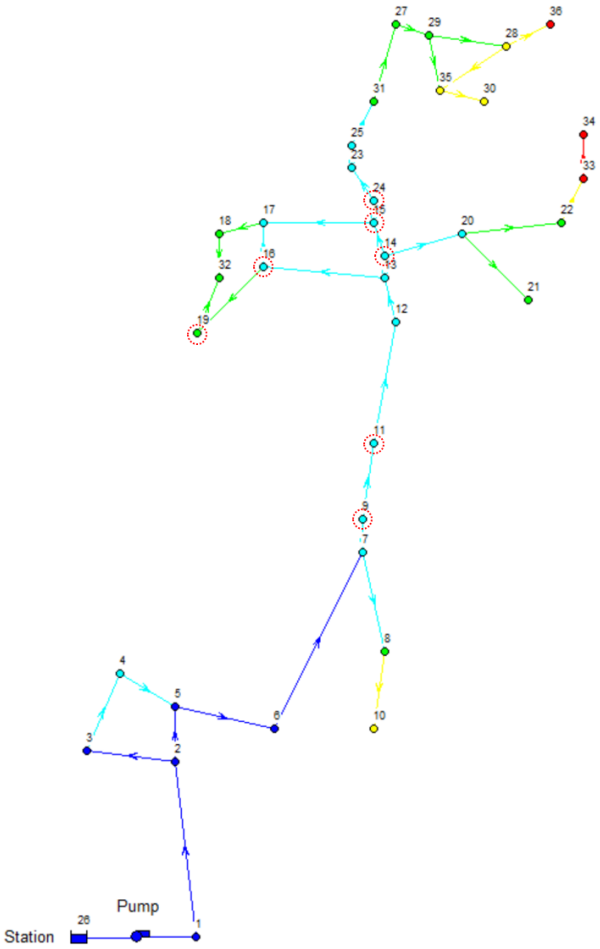


Figure B.1: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 11$

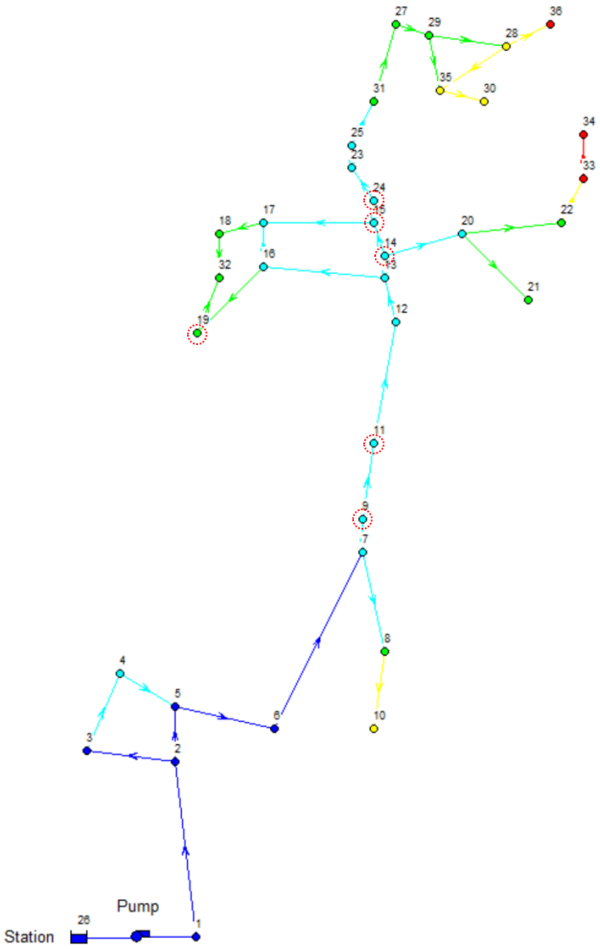


Figure B.2: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 12$

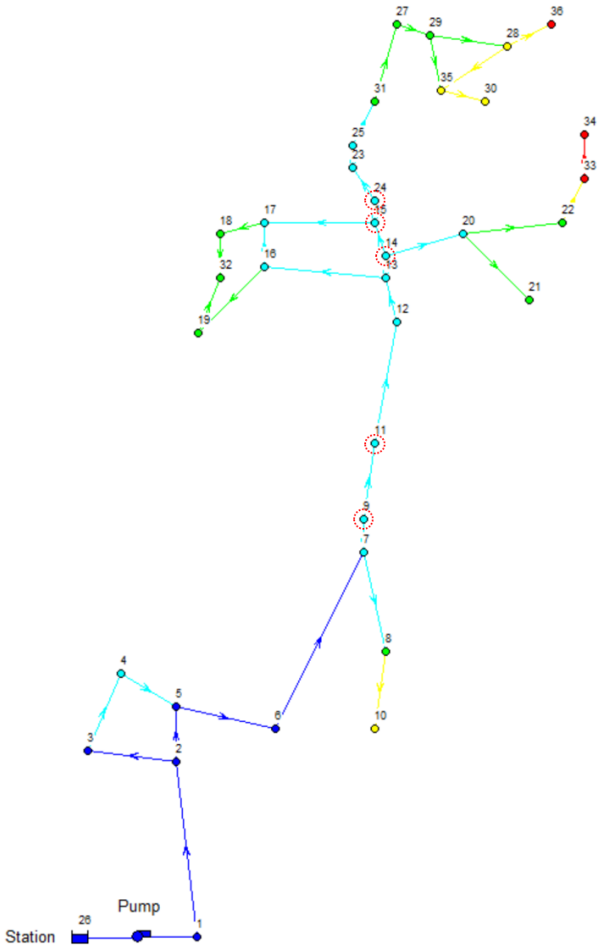


Figure B.3: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 13$

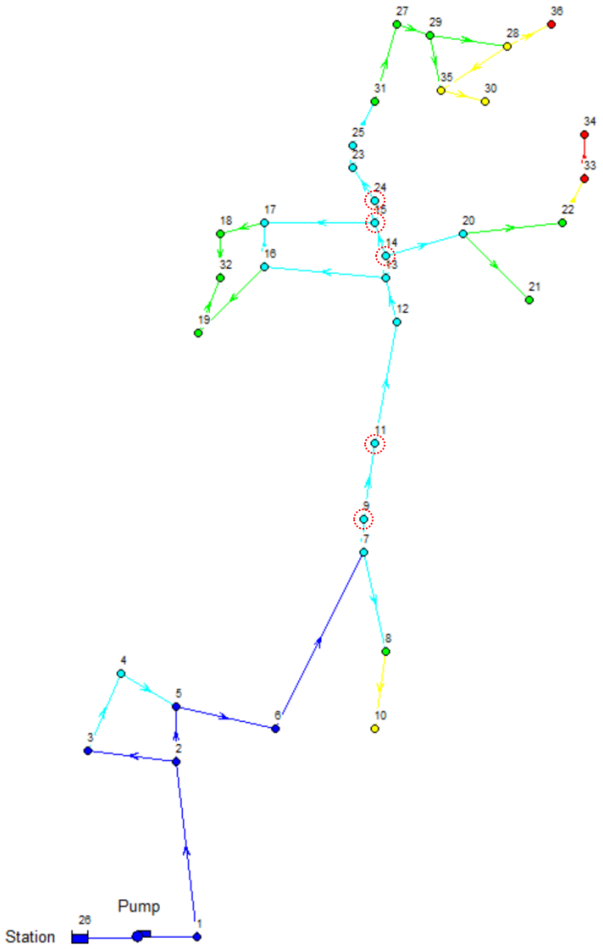


Figure B.4: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 14$

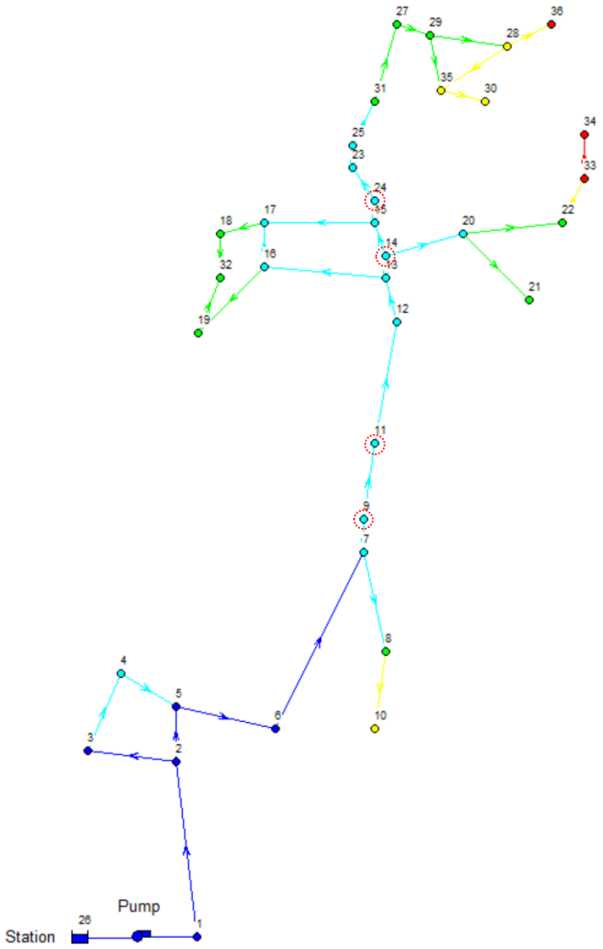


Figure B.5: The Actual Locations (Circled in Red Color) with Flushing for $a_l = 15$