SIMULATION-BASED OPTIMIZATION WITH CONSTRAINED SPSA FOR WATER DISTRIBUTION NETWORKS ON MILITARY INSTALLATIONS

by

Aimee Dalsimer

A thesis submitted to The Johns Hopkins University in conformity
with the requirements for the degree of
Master of Science in Engineering

Baltimore, Maryland
December 2022

Abstract

The purpose of this paper is to combine simulation-based optimization and simultaneous perturbation stochastic approximation (SPSA) to create an effective model of a water distribution network and return the optimal diameters for the system. This paper particularly focuses on a distribution network for a military installation. Using a water network simulation that includes random processes to model real world variability, we minimize the monetary cost and amount of the population that receives an inadequate amount of water. We use sequential quadratic programming and projection constraints to add bounds to our model. We conclude by showing that in two case studies, our model using simulation-based optimization performs better than the previously established pipe diameters in the networks.

Primary Reader and Advisor: Dr. James Spall

Acknowledgements

Many thanks to all the people who helped me throughout this research. This project would not exist without LCDR Brendan Bunn, his previous research, and his patience answering all my questions. Responses from Jiahao Shi about SQP and Dr. Burkhardt at EPANET were crucial to this process. Thank you also to Capt Escamilla, USMC and Emma for their support and willingness to help. A special thank you to Andrew W. Slavens, P.E. for his expertise, advice, and feedback. Endless gratitude to Barnes & Nobles for the "office space" and consistent source of caffeine.

Last but certainly not least, thank you to my advisor, Dr. James Spall. Not only did you lay the pathway for this project with your previous work, your kindness and standard of excellence are the reason this research was a success.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Water is at the root of human survival. However, we rarely think about the process behind water traveling from a source to our individual homes. Building a water network system involves critical decisions to create a network that will meet customer demand and be resilient to disasters. The network must also be cost effective and environmentally friendly. As natural resources like water become more limited, minimizing wasted water through leakage in systems becomes even more important. The design of water networks is particularly relevant for the United States military because they are often building or modifying bases in areas with limited funding and limited access to supplies. Among several important design considerations for a new or modified water network are the diameters of the pipes in the network [38]. In this paper, we use a stochastic approximation algorithm with simulation-based optimization to determine the diameter of pipes that will create a resilient and cost-effective water network system.

Simulation-based optimization uses computational tools to model a system and then minimize an objective function through many runs of the simulation. Electronic representations are useful in systems that are complex or costly to build. Simulations offer the advantage of testing multiple configurations or parameters of a system with little effort or cost compared to physical systems. Simulation and optimization are becoming more widely used in the military due to the monetary and personnel costs that can be saved ([5], [6],[40]). In [6], Checco builds a stochastic simulation to model demand for Army deployments to optimize makeup and size of forces and predict future monetary needs in the Army. Again modeling force demand, [40] uses a Monte Carlo simulation to incorporate realistic uncertainty in the system when optimizing skills and effectiveness in the workforce. In this paper, we similarly use a stochastic simulation for optimizing a physical system.

Simultaneous perturbation stochastic approximation (SPSA) is an optimization method that uses a gradient approximation from objective function outputs [31]. It then returns an optimal solution to the problem statement after many iterations. SPSA is particularly useful when only stochastic values are available from a system or there are other aspects of the system that make a direct calculation of the gradient difficult or impossible. Most realistic simulations have inherent randomness to more closely model complex and unpredictable systems.

Simulation optimization with simultaneous perturbation stochastic approximation (SPSA) has been explored in traffic and transportation problems. Ros-Roca et al. [10] explore how well SPSA performs in traffic simulations. They conclude that SPSA has benefits and drawbacks, but is recommended and works well in some transportation problems. Our paper furthers the design used by Ros-Roca by testing different configurations within the SPSA method. Li and Spall [18] apply discrete SPSA (DSPSA) to a public health model. The goal is to reduce the impact of diseases to a population using simulation optimization and DSPSA. With a similar goal in mind of minimizing negative impacts to society, we apply the SPSA algorithm on our model. In order to run simulation-based optimization, first we have to build a simulation of a water distribution network.

The Water Network Tool for Resilience (WNTR) [17] is a Python package for simulating and analyzing water distribution networks. WNTR allows the user to build a water distribution network or use a pre-made model and then simulate water flow or different scenarios in the system. We use WNTR to simulate network relationships like water flow, pump output, and water supplied to a region. In his report, *An Operational Model of Interdependent Water and Power Distribution Infrastructure Systems* [4], LCDR Brendan Bunn uses WNTR to model an interdependent water and electric system. The aim of [4] is to simulate infrastructures that are realistic for military installations or small island territories. Using his model as a template, we add optimization capabilities to the simulation. In particular, we consider the diameters of pipes in the network as the design variables and use WNTR together with a stochastic optimization algorithm to find the optimal pipe diameters for a water distribution network to minimize cost and maximize resilience to leaking pipes. Simulating the operation of a water distribution system prior to building or installing it saves time and money.

WNTR includes a stochastic simulation for pipe leak scenarios. The program calculates a probability of failure for pipes based on their diameter and chooses the duration and number of leaks from a probability distribution, which introduces randomness to the system. Stochastic simulations are more realistic in this scenario since the real world is variable and unpredictable; however, randomness makes optimization more difficult. In this paper, we modify the WNTR simulation to work with our water distribution network and apply SPSA to determine pipe diameters that minimize leakage while meeting as much population demand as possible. Having a valid and verified model is key to stochastic optimization. We discuss the capabilities and qualifications of WNTR in Chapter 3 to show it is a good program to use for simulation optimization.

In Chapter 2 we introduce the algorithms for simulation-based optimization and simultaneous perturbation stochastic approximation, which we combine for our model.

In Chapter 3 we describe our specific network configuration and parameters. We also explain disturbances to the network and detail all the randomness in the system. In Chapter 4, we formalize our objective function measurements and explore different methods of adding constraints to SPSA and we conclude the chapter by deciding on the best combination of hyperparameters and optimization methods for SPSA. In Chapter 5, we run tests to check the performance and efficiency of our model and then discuss generalization of our methods to other water distribution networks. Finally, in Chapter 6, we discuss real world applicability and further uses for this research.

Chapter 2

Methods

Selection of pipe diameters in water distribution networks affect the flow of water through a system as well as the cost of installation and repairs. Optimal pipe diameters are typically calculated from complex and involved equations. In *Selecting the Optimum Pipe Size*, Randall Whitesides, P.E. [38] details how to determine pipe sizes for a network using charts and equations. These equations are based on what the pipe is being used for and involve multiple parameters and variables. Calculations quickly become complicated in a network with many pipes and do not account for stochastic effects. Engineers spend a significant amount of time and energy deciding what diameter pipes to use, with Whitesides showing an entire block diagram to illustrate the multi-step process. In this paper, we introduce simulation optimization to explore a more streamlined and efficient method for pipe diameter selection. Simulation-based optimization allows users to build a computer program rather than a physical network, saving time and money. Avoiding calculations by hand, SPSA optimizes the pipe diameters while taking into consideration the randomness in the system. That is, there is an element of randomness that influences the operation and outputs of the system ([32], Chap. 1).

There are multiple optimization techniques available when using simulation-based optimization. We choose stochastic approximation (SA) because it does not depend on gradient calculations for optimization. SA is particularly useful for noisy simulations because it only depends on the simulation output and requires no knowledge of internal equations. Finite difference stochastic approximation (FDSA) and SPSA do not require direct gradient calculations and rely only on simulation output. We choose SPSA because it perturbs all parameters in the system at once, while FDSA only varies one parameter at a time. Spall [32] shows in Chapters 6 and 7 that SPSA is more efficient than FDSA and performs better in numerical tests. Simulation optimization with SPSA is a logical choice for this model because we are relying on stochastic outputs from a simulation of a water distribution network. There are multiple parameters to consider and complicated equations built in to the simulation.

2.1 Simulation-Based Optimization

Simulation-based optimization is a technique developed to find optimal solutions when a system is difficult or impossible to represent with closed-form functions [13][25]. Simulation allows us to avoid the significant assumptions or simplifications of a model that are often required for an analytical or gradient-based solution. Complex or uncertain scenarios are common in both the civilian and military sectors. Reference [5] describes how NATO can use Monte Carlo simulations of operational demand to optimize choices of mission modules and ultimately assist military decision makers. Simulation optimization integrates computer simulations into optimization methods that only need numeric inputs from objective functions. Figure 2-1 shows how a loop is created between the simulation runs and optimization method picked by the user. Starting at the initial input to the simulation, the computer program provides output values that are used to calculate the objective function value. The optimization program then makes appropriate adjustments to the variables and either continues the loop by running the simulation again or ends the loop once some stopping criterion is met. Stopping criteria usually consist of a certain number of runs, a maximum time, or a set level of convergence in values.

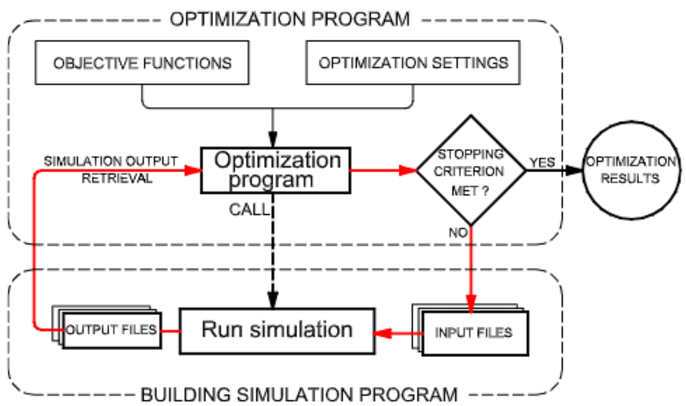


Figure 2-1: Diagram of general steps in simulation-based optimization. This figure was taken from [22].

The objective function for simulation based optimization is of the form

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} E[f(\boldsymbol{\theta}, \boldsymbol{V})]$$

where $\boldsymbol{\theta}$ is a set of decision variables, $\boldsymbol{\Theta}$ is the sample space, and $\boldsymbol{V}$ represents the different aspects of randomness in the system. The loss function can be summarized as $L(\boldsymbol{\theta}) = E[f(\boldsymbol{\theta}, \boldsymbol{V})]$, and the noise in the loss function measurements ε is derived from $\varepsilon = f(\boldsymbol{\theta}, \boldsymbol{V}) - L(\boldsymbol{\theta})$ [32]. Then $f(\boldsymbol{\theta}, \boldsymbol{V})$ represents the output from one iteration of the simulation [34]. The goal of simulation-based optimization is to use values of $f(\boldsymbol{\theta}, \boldsymbol{V})$ to minimize the loss function, $L(\boldsymbol{\theta})$.

In this paper we study the resilience of a system to leakage, which in the physical world would require building multiple physical water distribution networks then adding leaks. Multiple configurations can be tested in a few minutes with simulation optimization. Amaran et al. claim that simulation optimization is useful in complex or stochastic scenarios [1]; our system is both. Amaran also discusses multiple optimization techniques available with simulation-based optimization.

2.2 Simultaneous Perturbation Stochastic Approx.

We focus on simultaneous perturbation stochastic approximation (SPSA) ([31],[32]) as our optimization method. Stochastic optimization in general is useful for models that may be nonlinear, high dimensional, or intrinsically noisy. Spall shows that stochastic approximation methods perform better than random search in numerical tests. Similar to the well-known stochastic techniques simulated annealing and genetic algorithms, SPSA only uses numerical objective function outputs to calculate gradients. However, it differs in rigorously accommodating the noisy loss measurements discussed above by using gradient approximations. We use continuous SPSA rather than the discrete method since our variables are measured in inches. Fractional values exist in inches and more exact predictions can help engineers make decisions based on the resources or time available. Spall [30] details that SPSA may converge faster than non-gradient based techniques to a locally optimal solution. Since multiple configurations of pipe diameters could be good designs, a local solution is sufficient for our model. Amaran [1] states that SPSA requires fewer samples for the same accuracy as other stochastic approximation methods. Due to the complexity and randomness of our specific model and the multiple benefits described, we decide to use SPSA for optimization.

The basic algorithm for simulation-based optimization with SPSA used in this paper follows these steps [31]:

1. Define parameters $\gamma, \alpha, A, a, c,$ and M, set $k = 0$, and initialize $\hat{\boldsymbol{\theta}}_0$.

2. Calculate $a_k = \frac{a}{(k+1+A)^\alpha}$ and $c_k = \frac{c}{(k+1)^\gamma}$. Generate $\boldsymbol{\Delta}_k$ from a predetermined distribution (Bernoulli(1/2) with support $-1, 1$ in our model).

3. Define $\hat{\boldsymbol{\theta}}_k^+ = \hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k$ and $\hat{\boldsymbol{\theta}}_k^- = \hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k$.

4. Run the simulation and return noisy objective function values $f(\hat{\boldsymbol{\theta}}_k^+)$ and $f(\hat{\boldsymbol{\theta}}_k^-)$.

5. Calculate the gradient approximation $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k) = \frac{f(\hat{\boldsymbol{\theta}}_k^+)-f(\hat{\boldsymbol{\theta}}_k^-)}{2c_k\boldsymbol{\Delta_k}}$, where $\boldsymbol{\Delta_k}^{-1} = [\boldsymbol{\Delta_{k1}}, \boldsymbol{\Delta_{k2}}, ...]$ are the random perturbations on each variable.

6. Generate the next value $\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k\boldsymbol{g_k}$ then set $k = k+1$.

7. Repeat steps 2–5 until $k = M$, the maximum number of iterations.

8. Return the terminal value (hopefully $\approx \boldsymbol{\theta}^*$).

Chapter 3

Application

3.1 Network

In simulation optimization, a verified and validated simulation model is integral to a realistic optimization process. We have to ensure our network is following the correct equations so it behaves as water actually would in a system. Additionally, we should inspect the program and check that it has been successfully used elsewhere or tested against real world models to show it is simulating the system correctly. We use the Water Network Tool for Resilience (WNTR) ([16], [17]) code and examples to build our specific water distribution network for simulation runs. The foundation for WNTR is EPANET, which is a water distribution modeling application developed by the Environmental Protection Agency (EPA). WNTR is copyrighted to the National Technology and Engineering Solutions of Sandia, LLC. EPANET and WNTR have been used in a significant number of models for water systems and disaster scenario studies [19]. The WNTR manual includes the equations for all mass balance and flow calculations and uses the Hazen-Williams formula for head loss and pressure, which is used in real world design of water networks [39]. The methods in the user's manual are consistent with general mass balance and flow equations in physics and engineering

[21]. Dr. Burkhardt, and environmental engineer at the EPA, has confirmed that the program has been used for decades and the equations have been well tested. Based on a 2018 paper [15] there were plans to conduct a series of case studies working with physical systems, but according to Dr. Burkhardt and the lack of results online these have not yet taken place.

3.1.1 System Components

We base our network off of the water distribution system built in [4] to ensure the model is useful for military installations. Our paper focuses on a water distribution network delivering drinking water to customers. The network consists of one source, a reservoir, which uses gravity and three pumps to distribute water to the rest of the system through pipes and nodes.

We build the water network system with WNTR and rely on the parameters given in [4]. We use the diameters presented in [4] for an initial functionality testing of our system and as a baseline for our statistical testing in Chapter 5. Reference [4] builds a realistic water distribution network with specific diameters, so we assume these are practical. It is useful to look at the network in [4] since it was supposed to model military applications, which is also a goal of this paper. In water distribution networks, demand is the amount of water customers need at each output source. Demand varies depending on the time of day and variation in population needs and climate. We also include hourly demand multipliers based on typical household demands and the model in [4]. Elevation impacts pressure and head in a water distribution network due to gravity helping water flow faster at higher elevations. We include elevation in Table 3.1 for repeatability of the network or to allow for alternate calculations. Figure 3-1 illustrates the structure of the system and Tables 3.1, 3.2, and 3.3 list specifics of our water distribution network.

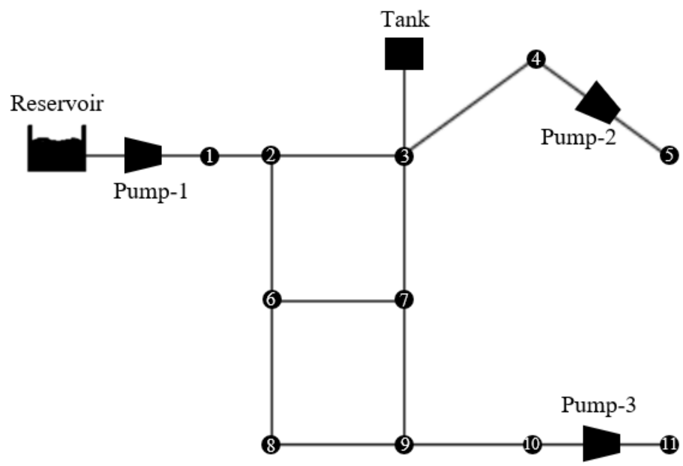


Figure 3-1: A basic diagram of our water distribution network. This network is based on the realistic water network in [4].

Label	**Elevation (ft)**	**Base Demand~$\boldsymbol{N}(\mu,(\frac{1}{3}\mu)^2)$**
Node 1	710	0,0
Node 2	710	(9.46,9.95)
Node 3	700	(9.46,9.95)
Node 4	695	(6.31,4.42)
Node 5	1000	(6.31,4.42)
Node 6	700	(9.46,9.95)
Node 7	695	(12.62,17.69)
Node 8	700	(6.31,4.42)
Node 9	710	(6.31,4.42)
Node 10	710	(6.31,4.42)
Node 11	1000	(3.15,1.11)
Reservoir	800	-

Table 3.1: Elevation and base demand parameters in the water distribution network. Base demand is normally distributed with indicated mean and variance in liters per second (L/s).

	Elevation	**Initial Level**	**Min Level**	**Max Level**	**Diameter**
Tank	850 ft	120 ft	100 ft	150 ft	50.5 ft

Table 3.2: Tank specifications for network. Elevation is to the bottom of the tank.

	Start Node	End Node	Flow (GPM), Head (ft)
Pump 1	Reservoir	Node 1	1500, 250
Pump 2	Node 4	Node 5	158, 131
Pump 3	Node 10	Node 11	158, 131

Table 3.3: Single-point pump values for network.

Another important thing to note about the system is that Pump 1 is open when the tank level is below 110 feet and closed when the tank level is above 140 feet. We use a pressure dependent demand (PDD) model, which means that the amount of water leaving a node, or used to supply a customer, is determined by pressure at the node. If there is not enough pressure, the population demand at that node may not be fully met. When no leakage is introduced to the system and the baseline diameters are used, the water network meets all population demand for a 24-hour period. Figure 3-2 displays the operation of the water network system without any disturbances. Using the diameters from [4], we run the simulation with the mean base demands and no leakage to show that all desired demand can be met by the configuration and settings of our network.

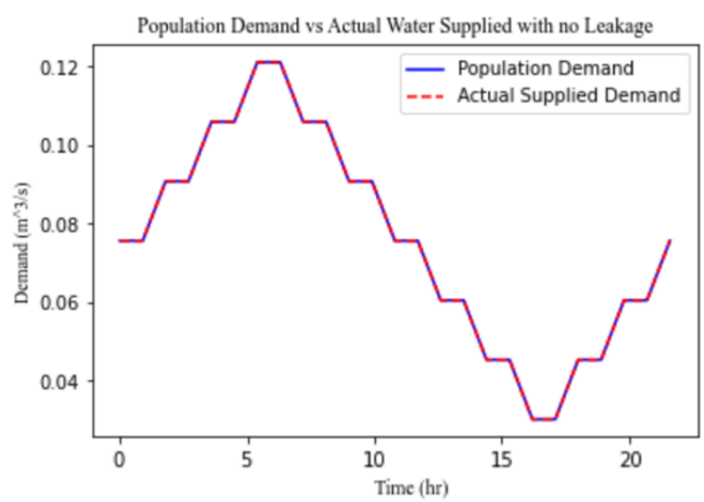


Figure 3-2: The baseline water distribution network meets all required demand when there are no leaks in the system.

3.1.2 Pipe Leakage

Resilience is an important property of a water distribution network because the consequences of people living without water are significant [28]. Evaluating the resilience of a water distribution system can be done with various metrics. WNTR includes the capability of introducing earthquakes, power outages, and fires to a system while providing output on hydraulic metrics like pressure and demand. Most disaster scenarios and other factors that impact water networks, like aging or climate change, can be grouped in two categories: additional unexpected demand or pipe leaks [9]. In this paper we focus on pipe leakage because it can be caused by disaster situations or normal disturbances to a system like freezing or thawing pipes, or pressure changes.

We add leaks to the system based on probability distributions, which are discussed in Section 3.1.3. The leak area and diameter are dependent on the diameter of the pipe by the following equations. We base these equations on a demo stochastic simulation for pipe leaks in WNTR [17].

$$\text{Leak Diameter} = \text{pipe diameter} \times .3$$

$$\text{Leak Area} = \pi\left(\frac{\text{leak diameter}}{2}\right)^2$$

The leaks are added to the system at simulated breaks in the pipes, which impacts the pressure and demand met throughout the water network system. Figure 3-3 illustrates a case where pipe leakage is introduced to the system. The mean base demands are used from each demand distribution and the baseline diameters are used. This shows that for the duration of the leaks, actual supplied demand does not meet requested demand in the system.

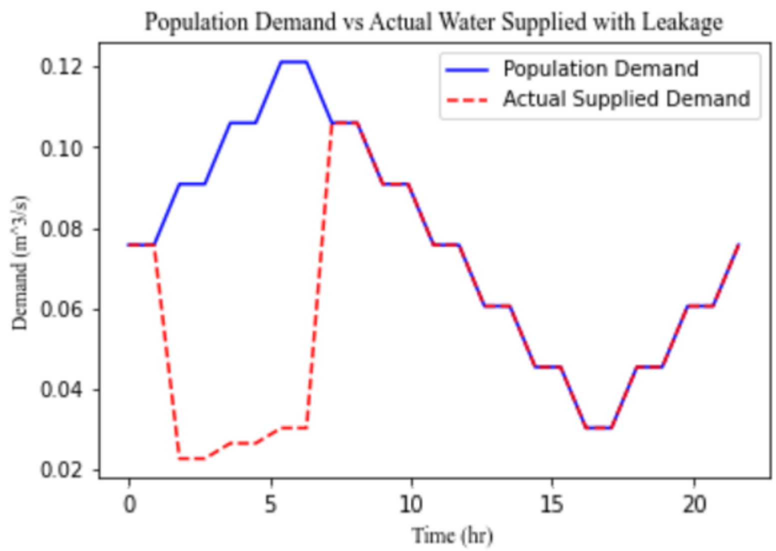


Figure 3-3: Pipe leakage at time 2 introduces large difference between population demand and actual supplied demand in a water distribution network. Leak duration is 3 hours.

3.1.3 Noisy Measurements

Noise or randomness can be introduced to a simulation to better model the unpredictability of real life. Water networks are physical structures that are subject to many external elements that cannot be fully controlled or predicted. A realistic simulation for a water distribution network should acknowledge this, which our system does by introducing randomness in the population demand requested, the number and location of pipe leaks, and the start time and duration of leaks. WNTR has a stochastic pipe leak demo simulation, which we use as an outline for our simulation model [17].

Table 3.4 is a summary of the randomness in our simulation, which makes it more realistic. Using similar notation as [32], the randomness in our system is accounted for in the $\boldsymbol{V}$ term of 2.1. Chapter 14 in [32] discusses the inclusion of direct random processes in simulation optimization and calls $\boldsymbol{V}$ the "amalgamation of the (pseudo) random effects". Details on the different elements of noise in our model come after the

Variable/Parameter	Distribution
Demand	Normal (μ, $(\frac{1}{3}\mu^2)$)
# Pipes to Fail	Random Integer (0,6)
Location of Failures	Random Non-uniform (failure probabilities)
Time of Failure	Random Non-Discrete Uniform (1.00,6.00)
Duration of Failure	Random Exponential (4$\times$ # pipes to fail)

Table 3.4: Summary of distributions of different elements of randomness in the simulation.

table.

Randomizing the extent and duration of leakage allows us to create a system that is resistant to both small scale distresses and larger disasters that could result in entire segment breaks or more significant leaks. We want to look specifically at systems with leakage, so on each iteration of our simulation, we randomly assign an integer number of leaks between zero and six. We choose this based on the statistic that most water distribution networks experience between 5% and 55% leakage, from [20] and [3]. Next, we calculate the failure probability for each pipe, which is inversely proportional to the diameters of the pipes since larger pipes have thicker walls and are less likely to leak [9]. Governed by the failure probabilities, we randomly choose the predefined number of pipes to leak from the list of all pipes without replacement. The failure probabilities provide a realistic way to determine the pipes that would most likely leak in a physical distribution network. After these steps, we have the specific number and location of the leaky pipes in our system.

Next, we use a continuous random uniform distribution from one to six hours to select a start time of the leaks, assuming the system is initially leak-free. In reality, age of the system and pressure through the pipes will impact when a leak develops in a system. Since we are studying resilience to leaks and only modeling the system for a day, we assume the leakage happens in an observable amount of time in the 24-hour period. We also assume leaks all start at the same time and that the duration of all leaks in the system is equal. Next we define the duration of the leaks, or the amount of time before they are all repaired. It can take a long time to locate and reach pipes in water distribution networks so the repair time varies significantly. Online sources range from a few hours to multiple days, but we base our decision on the claim that all leaks can be fixed in 24 hours [29],[2]. It is reasonable to assume it is more likely for pipes to be fixed as time goes on, and that repairs will depend on the number of pipes that are leaking. We generate the duration of leakage in the system using an exponential distribution with mean of $4N$ where N is the number of pipes leaking. Four hours was consistently in the range of repair times reported online and the maximum number of pipes that will leak in our system is six, so all pipes would typically be fixed in 24 hours using this distribution.

The final element of randomness in our system is the population demand requested at each node. The desired population demand is the amount of water required at each output node to serve the needs of the population being considered. The actual population demand is supplied based on the pressure and flow throughout the network during the simulation. We set the base demand at each node with a random normal distribution and set hourly demand multipliers to model common variance throughout a 24 hour period. Demand in a water network system can be modeled well with a normal distribution according to [26]. We set the mean at the base demands in [4], which are estimations from experience working with water systems on island locations and personnel from WNTR. Since the demand is in L/s in [26], we convert our values to these units and then generate a random normal distribution with the means previously established and the variance set as $(1/3 \times mean)^2$. Our choice of variance comes from [26] and ensures that there is a small chance that the base demand will be negative. A negative base demand is not plausible in real life so we want to minimize the opportunity for this to occur. A new base demand value is generated with each iteration of the simulation, which should capture the changes in population demand with seasonal shifts or inconsistent customer levels.

Chapter 4

Implementation

4.1 Problem Formulation

WNTR has several built-in metrics to calculate how effective a water distribution network is, and we combine two of these for our objective function. We use the desired demand and actual demand to calculate the demand not met due to leakages in the system. The desired population demand at each node is calculated from the base demand and hourly demand multipliers. The actual demand is calculated while the simulation is run and is related to the pressure at each node in the system. The demand not met is the difference of these two values, and represents part of the population that will not be supplied with enough water for the 24-hour period.

While minimizing the number of people affected by leaks, we still want to maintain low cost in building a system. Pipes with larger diameters are more costly to build, transport, install, and maintain. Using this logic, we want to minimize the sum of the diameters used in the system to minimize cost while maintaining the capability of meeting customer demand. Additionally, we add constraints for the diameter of the pipes to ensure they are realistic for a water network. Formulation (4.1) displays the objective function with constraints.

$$\min_{\theta} \quad L(\boldsymbol{\theta}) = \min_{\theta} \quad E[w_1 I(\boldsymbol{\theta}, \boldsymbol{V}) + w_2 D(\boldsymbol{\theta}, \boldsymbol{V})] \tag{4.1}$$

$$s.t. \quad \frac{3}{4} \leq \hat{\boldsymbol{\theta}}_{ki} \leq 30 \qquad \forall i, k \tag{4.2}$$

where $I(\boldsymbol{\theta}, \boldsymbol{V})$ is the difference between desired and actual demand and $D(\boldsymbol{\theta}, \boldsymbol{V})$ is the sum of the pipe diameters. The constraints apply to all i components of each k^{th} iteration of $\hat{\boldsymbol{\theta}}_k$ and bound our pipe diameters between 3/4 inch and 30 inches. Demand not met is calculated in m^3/s and the pipe diameters are converted to meters since these units are most compatible with WNTR. The weights w_1 and w_2 are assigned to each part of the objective function are defined in the next section.

4.1.1 Objective Function Weighting

Prior to running optimization on the water distribution network, we must decide on the weights, w_1 and w_2, to use in the objective function. The percentage of our function allotted towards the resiliency versus cost consideration is up to the user and can be easily adjusted. As climate change and urbanization increase the likelihood and impact of extreme weather across the world, sustainability and resiliency are becoming more valued in the planning of water networks. Ciprian Sanchez et al. [28] discuss the importance of resilient water supply systems in the future. We combine this information with the weights used by Suribabu [36] to write our new objective function.

$$E[f(\boldsymbol{\theta}, \boldsymbol{V})] = E[.8I(\boldsymbol{\theta}, \boldsymbol{V}) + .2D(\boldsymbol{\theta}, \boldsymbol{V})]$$

We choose these from the standard options in Suribabu's paper since we care significantly more about minimizing the people impacted by inadequate demand in the system than the project cost. We also consulted a civil engineer, Andrew Slavens, P.E., who works with water distribution networks in Georgia and confirmed 80:20 was a reasonable distribution of weights.

4.2 SPSA

Following the general algorithm in Chapter 2, we implement SPSA for optimization with our objective function measurements from the simulation runs. Recall $\boldsymbol{\theta}$ is a vector of the eleven pipe diameters, one for each pipe in the network. We use $\hat{\boldsymbol{\theta}}_0 = [15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15]^T$ for our initial value since it is approximately the midpoint of the viable pipe diameters. We use the common choice of a Bernoulli ± 1 distribution for each component of $\boldsymbol{\Delta}_k$ since it is symmetric, has a mean of zero, and has a finite inverse moment ([32], Sect 7.5). We calculate gradient approximations with the noisy objective function values from our simulation, then iterate steps 2–5 of the algorithm for M runs. The algorithm is terminated when it reaches M iterations, the maximum allowable number, which is defined prior to initialization. Spall [32] states that $\alpha = .602$ and $\gamma = .101$ are "practically effective and theoretically valid." Again following the guidelines from Spall, we set A equal to 10% of the maximum iterations allowed. Next, we use semiautomatic tuning to find the parameters a and c.

4.2.1 Hyperparameter Tuning

Automatic gain selection guidelines are discussed in ([32], Sect 7.5). First, we set c approximately equal to the standard deviation of the noisy measurements from the simulation. We calculate the sample variance at the initial diameters $\hat{\boldsymbol{\theta}}_0$ using Equation 4.3 with $n = 50$ samples. We set c equal to $\sqrt{s^2} \approx .475$.

$$\text{Sample Variance} = s^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1} = .2258 \tag{4.3}$$

Next, we modify the automatic gain calculation code from Spall [33] for our system to estimate the hyperparameter a for different amounts of iterations. We use a step size of .25 to stabilize the system while still allowing the design variables to span the whole feasible range in 100 iterations. For any calculations on our first network, SPSA is run with the hyperparameters in Table 4.1.

A	γ	α	c	a (100)	a (500)	a (1000)	a (2000)
$.10M$	.101	.602	.475	45	110	172	362

Table 4.1: Hyperparameters used for numerical results of network. Hyperparameter a is displayed for different amounts of iterations.

4.2.2 Common Random Numbers

Common random numbers (CRNs) is a variance reduction method that can be used to decrease run time and improve performance of a system. CRNs increase the correlation between random variables in an algorithm so they have positive covariance, which decreases overall variance when we take a difference. We expect the difference between noisy measurements to have less variance if there is monotonicity and synchronization in the system. A simple way to run SPSA with CRNs is by setting the same seed for the random numbers generated in the loss measurement calculations for $\hat{\boldsymbol{\theta}}_k^+$ and $\hat{\boldsymbol{\theta}}_k^-$ at a given k [14].

Monotonicity means that the two simulations move the same way when small perturbations are applied to the random variables. To complete a small empirical test for this concept, we look at the loss function output for $\boldsymbol{\theta} = [12, 12, ..., 12]^T$ and $\boldsymbol{\theta} = [18, 18, ..., 18]^T$, modeling a shift up and down from our initial $\hat{\boldsymbol{\theta}}_0$. Using these test inputs, we set the random variables to specific values. For variables with a normal or exponential distribution we set them equal to the mean, and we choose the first three pipes to have leaks starting at six hours. We average 50 noisy measurements for this case for comparison, and display these values as the ‘standard’ in Table 4.2. We add and subtract a set amount from each random variable and get average loss function values to represent "small perturbations" of the random variables. We illustrate subtracting a small amount from the standard values with '-' and a larger amount with '- -' in Table 4.2, and do the same thing but in the positive direction with '+' and '++'.

Table 4.2 shows that with the same small changes in the randomness, the averaged loss function values for $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ change in the same direction. For two sets of diameters near $\hat{\boldsymbol{\theta}}_0$, the average loss function gets smaller when all randomness in the system decreases and increases when more noise is added to the system. Similar behavior between the two sample diameters is what we expect for a monotonic system, so this gives us an indication CRNs may be successful with our implementation.

Diameters	12 inches	18 inches
- -	.737	1.07
-	.848	1.20
Standard	.671	1.99
+	1.45	2.16
++	1.70	2.33

Table 4.2: Average loss function values after small (- and +) and larger (- - and ++) changes in the random variables in the positive and negative direction.

Synchronization holds if the random variables are acted on and used in the same way in the two calculations. Because we are running the same simulation, it is guaranteed that there will be the same number of random variables in both simulations. In each run of WNTR, the simulation will follow the same equations and programmed rules. These consistencies suggest the randomness may apply the same way to both runs of the simulation.

The practical way to see if CRNs work is to test a small sample. We test our model with and without CRNs and return the time it takes to run SPSA and the final $\boldsymbol{\theta}^*$ after 100 iterations. Then we average 50 noisy loss measurements and compare the two outputs. Including CRNs decreased the run time by over 10% and the objective function value by more than 20%. We conclude that CRNs improve performance and decrease runtime for our system, so we include them in our model.

4.2.3 Constraint Implementation

In this paper $\hat{\boldsymbol{\theta}}_k$ is a vector of physical properties, pipe diameters, which means we have to limit the feasible domain. Measured in inches, we constrain our pipes to the range of 3/4 inch to 30 inches. Small supply and service pipes can be as small as 3/4 inch while larger distribution pipes in a simple network like ours are likely to be in the 20–30 inch range [23].

Constraints are added to optimization methods in a variety of ways. We explore three popular methods: penalty functions, projection functions, and sequential quadratic programming.

4.2.3.1 Penalty Functions

We reproduce the use of penalty functions applied by Wang and Spall in [12]. A penalty function extends the objective function with a part that penalizes the system when diameters outside of our bounds are used. Following the model in [35] our modified objective function is:

$$E[f(\hat{\boldsymbol{\theta}}, \boldsymbol{V})] = E[.8I(\hat{\boldsymbol{\theta}}, \boldsymbol{V}) + .2D(\hat{\boldsymbol{\theta}}, \boldsymbol{V}) + P|\sum_i \min(0, \hat{\boldsymbol{\theta}}_{ki} - 3/4) + \sum_i \max(0, \hat{\boldsymbol{\theta}}_{ki} - 30)|] \tag{4.4}$$

where P determines how severe the penalty is when the $\hat{\boldsymbol{\theta}}_{ki}$ values are out of bounds. Good values of P depend on the system and could be obtained through theory or trial and error. After increasing P from 1 to 10, the system continued to give values outside of our bounds. Since the sample mean of our function outputs without the penalty function was near 1.5, having to use such a high penalty value suggests this may not be a good approach for our system. We ultimately decided not to use a penalty function since it also increased our runtime by 200%.

4.2.3.2 Projection Constraints

Another method for adding constraints to our $\hat{\boldsymbol{\theta}}$ values is to check the value of the components of $\hat{\boldsymbol{\theta}}_k$ after each SPSA iteration and send any values that are outside of the bounds back to, or inside, the boundary point.

To use projection constraints, we add equation 4.5 to our algorithm in the final calculation of $\hat{\boldsymbol{\theta}}_{k+1}$ on each iteration [37]. This sends diameters less than 3/4 or greater than 30 back to the constraint bounds.

$$\hat{\boldsymbol{\theta}}_{k+1} = \min_i\{\max_i\{\hat{\boldsymbol{\theta}}_{k_i}, 3/4\}, 30\} \tag{4.5}$$

4.2.3.3 Sequential Quadratic Programming

Shi and Spall [27] discuss sequential quadratic programming (SQP) for constrained optimization problems. SQP is used to iteratively solve optimization problems within the constraints for our model by minimizing the distance between our $\hat{\boldsymbol{\theta}}_k$ and values within the bounds. Since we require positive diameters in order to calculate the loss functions, we still include projection functions to send any $\boldsymbol{\theta}^+_{k_i}$ or $\boldsymbol{\theta}^-_{k_i}$ values that are less than or equal to zero back to .1 inches. Then we perform the normal SPSA calculations, until we return a temporary $\hat{\boldsymbol{\theta}}_{k+1}$, which we label $\tilde{\boldsymbol{\theta}}_{k+1}$. If all components of $\tilde{\boldsymbol{\theta}}_{k+1}$ are within our bounds of 3/4 inches and 30 inches, we do not perform SQP on that iteration and return $\hat{\boldsymbol{\theta}}_k = \tilde{\boldsymbol{\theta}}_k$. If any component is outside of the bounds, we use the Python package 'scipy.optimize.minimize' with method 'SLSQP' to perform SQP on this iteration and return a value within the bounds. $\boldsymbol{\theta}^{SQP}_{k+1}$ represents the method minimizing the design variables using sequential least squares programming

(SLSQP). We use the Euclidean norm as the function, $\hat{\boldsymbol{\theta}}_k$ as our initial value, and allow a maximum of 100 iterations for SQP. Algorithm 4.6 shows the operations for SQP within the SPSA calculation of $\hat{\boldsymbol{\theta}}_{k+1}$.

$$\tilde{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \boldsymbol{g}_k$$

$$\hat{\boldsymbol{\theta}}_{k+1} = \begin{cases} \tilde{\boldsymbol{\theta}}_{k+1} & \text{if } \tilde{\boldsymbol{\theta}}_{k+1,i} \in [\frac{3}{4}, 30] \\ \boldsymbol{\theta}_{k+1}^{SQP} & \text{if } \tilde{\boldsymbol{\theta}}_{k+1,i} \notin [\frac{3}{4}, 30] \end{cases} \tag{4.6}$$

4.2.4 Gradient Averaging

In some instances, it may be advantageous to perform multiple gradient calculations within the SPSA algorithm. This can be helpful if there is a large amount of noise in the loss function measurements. It can also minimize the randomness in the system from $\boldsymbol{\Delta}_k$. This can easily be implemented by generating n values of $\boldsymbol{\Delta}_k$ then taking an average of n conditionally independent values of $\hat{g}_k(\hat{\boldsymbol{\theta}}_k)$. We use the average of the gradient approximations for our calculation of $\hat{\boldsymbol{\theta}}_{k+1}$.

Gradient averaging creates a trade-off between the amount of total SPSA iterations and the amount of gradients averaged. More averaging may give a more reliable value, but requires additional runs of the simulation for each $\hat{\boldsymbol{\theta}}_k^+$ and $\hat{\boldsymbol{\theta}}_k^-$ calculation averaged. In our model, we test $n = 5$ gradient averages so it has an impact but still allows us to run a large number of overall SPSA iterations. We integrate gradient averaging and the different methods for constrained SPSA to test which combination performs best.

4.2.5 Performance of Constraint Methods and Averaging

In order to decide which combination of optimization methods to use in our model, we compare different configurations. We consider using SQP versus projection constraints, and look at a version with and without gradient averaging. We consider the cost and effectiveness of these options for our model.

Table 4.3 shows the results of our model with different optimization methods. In order to compare the cost and benefit of each configuration we use the same number of total simulation runs. This counts each time the loss function needs to be recalculated in our model, so each basic SPSA iteration uses 2: one for $\hat{\boldsymbol{\theta}}^+$ and one for $\hat{\boldsymbol{\theta}}^-$. When we use gradient averaging over 5 values, we have to do $2 \times 5 = 10$ simulation runs for each SPSA iteration. To compare outputs, we calculate the loss function values for 50 simulation runs and return the average. We use the hyperparameters from Table 4.1 with $a(500)$ and the iteration counts listed. All runs were done in one sitting on the same computer to maintain fairness on runtimes.

	Projection		**SQP**	
	No Avg	Gradient Avg	No Avg	Gradient Avg
# SPSA Iterations	500	100	500	100
# Simulation Runs	1000	1000	1000	1000
Run Time (mins)	8:47	21:44	9:45	20:17
Avg Loss Value	1.134	1.206	1.048	1.231

Table 4.3: Comparison of different optimization methods.

Based on our numerical tests, we decide to use SQP to add constraints to the model and do not include averaging of gradient approximations. The runtime without averaging was significantly lower than with for the same number of simulation runs. The simulation is dynamic, so the larger runtime could be due to having to store more values in the averaging loop or poorly tuned parameters. The average loss function measurements were not largely different, so we make our decision due to the runtime and slightly better performance of SQP and no averaging. We implement SQP without gradient averaging for any future use of the model.

Chapter 5

Numerical Results

Using the parameters and methods described previously, we test the performance of our model. First we check graphically if the process is doing what we expect. In one run of SPSA, we expect a lot of noise so it may be difficult to infer much from the graph. By averaging multiple loss function outputs over runs of 2000 SPSA iterations, we expect to visually see a decrease in the loss function outputs. If there was no randomness in the system, both plots should consistently decrease over the number of iterations. Figure 5-1 displays a single run of SPSA and Figure 5-2 displays 10 SPSA runs with hopes of averaging out some of the noise in single runs.

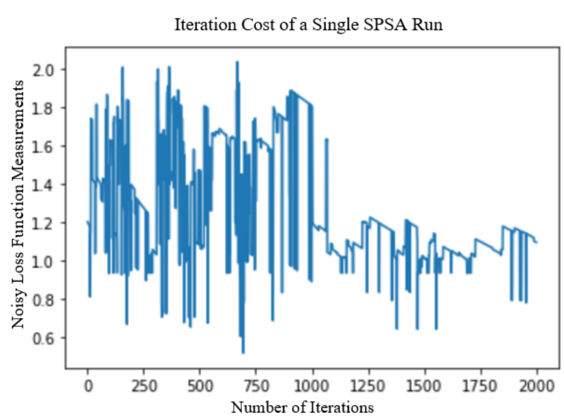


Figure 5-1: A single run of the system to show how SPSA actually runs on the model.

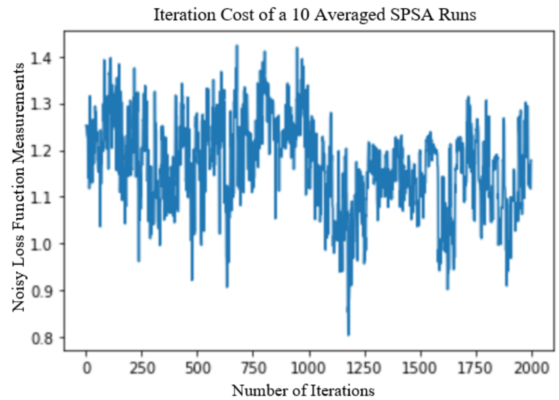


Figure 5-2: 10 SPSA runs averaged in attempt to reduce the visualization of noise and check if the loss function outputs are decreasing.

We do not see the strong downward trend we expect to see in the graphical results, and we see dramatic and sudden shifts in the loss function outputs. Prior to discussing explanations for this, we explore the statistical significance of the results.

To see how significant the results are, we perform statistical t-tests and get p-values. We use a two sample test for matched pairs to calculate our test statistic. We follow the formulas in [32] to calculate the test statistic. In order to evaluate whether one result is better than the other, we use 50 samples of each and calculate the following t-statistic:

$$t = \frac{\bar{X} - \bar{Y}}{(\frac{s^2_{X-Y}}{n})^{\frac{1}{2}}}$$

where $s^2_{X_{base}-X_{test}} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - Y_i - (\bar{X} - \bar{Y}))^2$ and we use $n-1$ degrees of freedom.

We compare the final $\boldsymbol{\theta}^*$ output after different amounts of SPSA iterations to both our initial $\boldsymbol{\theta}_0$ and a baseline model. For a baseline model, we use the diameters chosen in [4] for the realistic water distribution network. We also compare our system to the initial diameters to see if our output has changed a significant amount from our input. Table 5.1 shows the numerical results from these statistical tests. Based on the SPSA theory and algorithm, we assume that the average simulation output for the terminal $\boldsymbol{\theta}^*$ will be less than the average output for simulation runs of $\hat{\boldsymbol{\theta}}_0$. Since we know the value decreases, we report a one-tailed p-value when running a statistical test against the initial $\boldsymbol{\theta}_0$. Since we do not know whether the terminal diameters after SPSA will be better than the baseline model in [4], we rely on a two-tailed p-value for any statistics against the baseline.

# of Iterations	**100**	**500**	**1000**	**2000**
% Better vs Initial	6.29	8.49	10.25	26.95
p-value vs Initial	.080	.077	.050	$.207 \times 10^{-7}$
% Better vs Baseline	-10.0	5.28	7.48	15.52
p-value vs Baseline	–	.294	.203	1.86×10^{-3}

Table 5.1: Statistical tests for different amounts of SPSA iterations. Negative percentages or no value indicates that the baseline diameters are better than those after SPSA.

Small one-tailed and two-tailed p-values indicate that we reject the null hypothesis of the two samples having the same mean [8]. The p-value is significant against the initial pipe diameters in each number of iterations we studied, and implies that our diameters after the SPSA optimization algorithm improve from the input. After 2000 iterations, Table 5.1 shows that the average diameters from simulation optimization are significantly better than those in [4]. We can also see from the percent of improvement in the averaged measurements that our algorithm eventually seems to return a better result than the diameters used in [4]. Looking more practically at the results, the terminal $\hat{\boldsymbol{\theta}}_{2000}$ at one run of 2000 iterations in inches is:

[8.283, 17.095, 13.655, 8.283, 22.467, 0.75, 26.556, 8.283, 0.75, 0.75, 8.283]

We can convert this to meters and subtract it from the average noisy loss function value to analyze the two parts of the objective function separately. We have a difference between desired demand and actual demand of

$$\frac{.7972 - .2(2.925)}{.8} = \text{difference in demand} = .2653$$

We run calculations with this value in m^3/s but we can use other units to make it more comprehensible. Compared to the baseline, after optimization the model saves us over 2000 gallons per minute of water throughout the water distribution network. The average household uses a flow rate of 6–12 gallons per minute of water [11], so this is supplying between 150 and 333 more houses with water than the diameters in the baseline network. This suggests our algorithm produces a significantly more resilient water distribution network.

Focusing on the second part of the objective function, we sum the diameters of $\hat{\boldsymbol{\theta}}_{2000}$ then we compare this value to the sum of the baseline diameters.

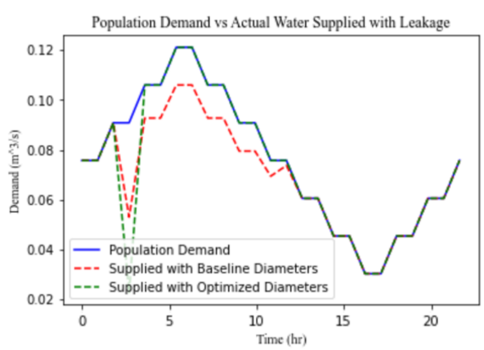
$$\text{Sum of diameters at 2000 iterations} \approx 115 \text{ inches}$$

$$\text{Sum of baseline diameters} = 124 \text{ inches}$$

This result supports the claim that our algorithm minimized both parts of the objective function independently as well as the overall objective function value. It is logical that a pipes with larger diameters cost more to manufacture, transport, and install. Only considering initial cost, a 2-inch increase in diameter can increase the cost by $30 per pipe in some instances [7]. This means saving 9 inches in overall diameter can significantly decrease the project cost and future replacement or repair expenses.

Though the p-values from Table 5.1 suggest improvement, looking at graphs of iteration cost reveals our investigations may be inconclusive currently. The graphs in Figures 5-1 and 5-2 illustrate drastic jumps up and down throughout the run and not much of a downward trend, which is unusual for SPSA. Since we are working in a multidimensional space and using a simulation with many internal equations, it is difficult to determine why this is occurring. One option could be that the amount of noise in the system is too large for the system to stabilize, or that some aspect of the randomness is too dynamic. Another option could be that the loss function is fundamentally flat in the range of values we are considering. We also tested out different parameters for α and γ, but other typical values still created results with sharp shifts in the outputs. Next, we tried to cut our a value in half or smaller to decrease the step size in the SPSA process. This can stabilize the system, but creates a tradeoff between making the system unstable and being unable to span the entire range of possible values. We will continue running tests but as of right now we cannot confidently claim that our model performed correctly or improved the system.

Finally we graph the actual demand from our new diameters and the diameters from the 'baseline' network in [4] alongside the desired demand. We use the mean of each base demand distribution for the graphs. When there are no leaks introduced to the system both diameters follow the line for desired demand, as it did in Figure 3-3. When there is leakage, the graphs show that the diameters after 2000 iterations of SPSA create a network that meets more demand than the baseline pipe diameters. Two examples of random leaks are shown in Figure 5-3, and both support the idea that the diameters with simulation-based optimization perform better.



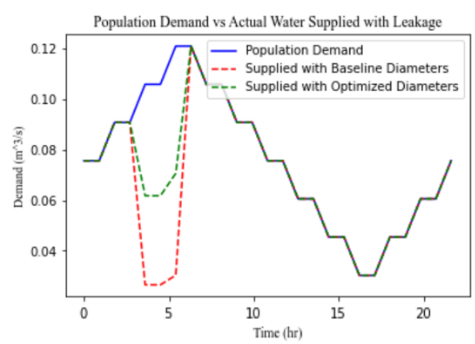


Figure 5-3: Population demand graphs with random leaks introduced showing desired demand and actual demand with the baseline diameters and optimized diameters.

5.1 Generalization to Other Networks

Next, we explore if our system is effective for other water distribution networks. We use a demo network built by EPANET to generalize our methods and run statistical tests to see if constrained SPSA optimization with SQP performs well in other scenarios. In order to use a different network, we first have to build or download an input file with the specifications for pipes, nodes, tanks, etc. In this example we use 'Network 2' from EPANET [24], which is based on an area in Connecticut. Figure 5-4 diagrams the network and shows there is one tank, 40 pipes, and 35 junctions. All pipes in the initial model have diameter of 8 or 12 inches.

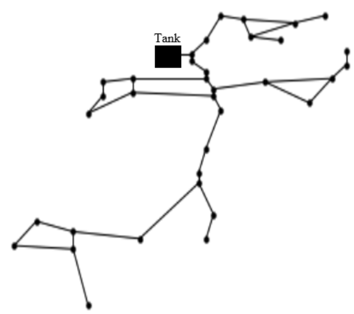


Figure 5-4: Diagram of 'Network 2'

Once the basic network is put into an input file, we set up the random variables and introduce leakage to the system. Similar to the discussion of randomness in the system above, we assign a random number of leaky pipes between zero and twenty-one. We change the mean of our exponential to N hours, where N is the number of leaks, based again on the assumption that almost all leaks can be fixed in 24 hours. We also choose new bounds for our model, using 4 inches as a lower bound and 20 inches an upper bound. These diameters are not far from the range used in the predefined 'Network 2' but still allows values other than the two used initially. In order to keep the two parts of our objective function similar in magnitude but also maintain emphasis on network resiliency, we shift the weightings from our previous model. We use $w_1 = .95$ and $w_2 = .05$ for Network 2.

We can see in Figure 5-5 that without any leakage at the baseline diameters, the water distribution network will meet all desired demand in the population. When leaks are introduced, Figure 5-5 illustrates that population demand is no longer completely met by the water distribution network.

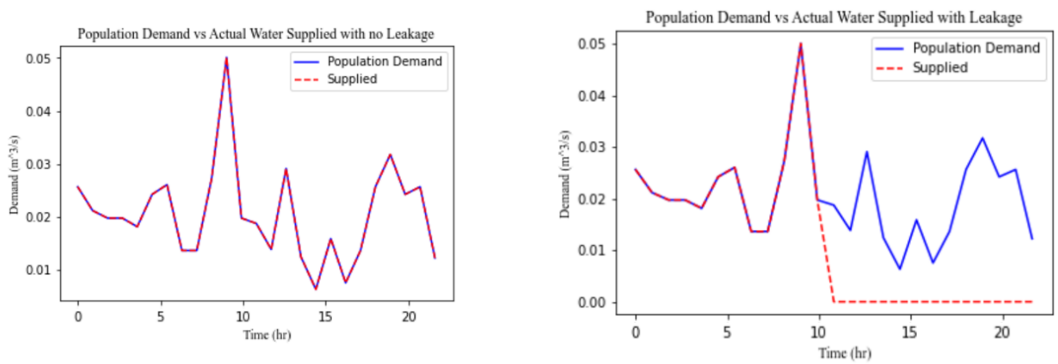


Figure 5-5: Desired demand is met when the network has no leakage but not met by actual demand when leaks are introduced.

Next, we use the simulation to calculate the variance in outputs and use the standard deviation of this to approximate our hyperparameter, c. We use the baseline diameters used in the published 'Network 2' for these calculations. We set $\hat{\boldsymbol{\theta}}_0$ as a vector of 13 inches for each diameter since this is close to the midpoint between our bounds. Using automatic gain calculation as before with our new $\hat{\boldsymbol{\theta}}_0$ and an initial step size of 1, we find values for a. Table 5.2 shows the hyperparameters used in our model for Network 2.

A	γ	α	c	a (500)	a (1000)	a (9000)	a (10000)
.10*M	.101	.602	.185	343	515	1928	2038

Table 5.2: Parameters used for numerical results of Network 2. a is shown for multiple amounts of iterations.

Once the parameters and random variables have been defined and initialized, we are able to apply SPSA with SQP. We check the performance for multiple amounts of iterations, similar to our previous case study. We compare the diameters after optimization to the previously established diameters and display the results in Table 5.3. We show the how different the noisy sample mean is to show whether optimization improves the resilience and cost of the system. We use a matched two-sample test and rely on two-tailed p-values for statistical analysis.

# of Iterations	500	1000	9000	10000
% "Better"	-6.33	-3.79	.0344	3.22
2-tailed p-value	–	–	.922	9.45×10^{-9}

Table 5.3: Statistical tests for Network 2 with different amounts of SPSA iterations. Negative percentages or no value indicates that the baseline diameters are better than those after SPSA.

The table shows that, on average, after $\approx$ 9000 iterations, the diameters from simulation optimization perform better than those previously used for the network. We can also see that it takes more iterations for this to occur than the previous network we explored. This could be because the baseline diameters are a good fit already, or due to the fact that there are almost $4\times$ more design variables in this network. Similar to before, we can analyze one terminal $\hat{\boldsymbol{\theta}}_{9000}$ and discuss the diameters in terms of real world impacts. Compared to the predefined diameters, those after simulation-based optimization could save us over 94 gallons per minute of water and 18 inches in total pipe diameter. Success with a second water distribution network suggests our simulation optimization model with SPSA is broadly applicable.

Chapter 6

Conclusion

In this paper, we combine simulation-based optimization, SPSA, and SQP along with other optimization techniques in order to minimize a constrained objective function. In particular, we use WNTR to build a simulation of a water distribution network based on a design useful for military installations and island locations. We maximize the resilience and minimize the cost of a system by minimizing the total diameter of the pipes used and minimizing the amount of water lost due to pipe leakage. After running numerical tests and analyzing the performance of our model, we see evidence that the diameters returned after optimization may perform significantly better than the initial diameters or the diameters used in a previous study of this water network.

One important benefit of simulation-based optimization is that the model parameters can be easily modified or changed. Since WNTR has options to build any water distribution network with pipes, valves, pumps, tanks, and other pieces, a user can create an existing or future system. After designing the network, the steps from Section 5.1 can be repeated to define the hyperparameters for SPSA, specific constraints for the system, and desired weighting for the objective function. Then, after a large number of SPSA iterations, the program will return optimal diameters for that specific water distribution network. Project managers or engineers can use this information as part of their decision making process for construction design or cost analysis.

The versatility and applicability of this model suggests it can be helpful for any system, but especially relevant for the military. The United States military is frequently building temporary and permanent bases. When building a base in a foreign country, the military may not want to rely on the host nation for water supply or may be far from industrialized cities. The military also works on a budget, so reducing the cost of adding water distribution networks to bases allows reallocation of money to other areas of operation. Water on military bases is often used in training scenarios or fire-fighting practice, so a resilient system is integral to military preparedness. Additionally, having a simulation that returns specific diameters to install prior to building a base is helpful when construction parts have to be shipped or flown across the world to discreet locations.

In this paper, the objective function is based on two metrics: resiliency and cost. This function could be extended or changed for future studies of this model. As discussed above, the algorithm we used could also be applied to many other water distribution network designs. Other aspects of the simulation could be changed or studied further, such as the elements of randomness or the pipe leakage calculations. There could also be physical experiments done if a small model or real-life water network system was built or observed. Then the results in this paper could be used for comparison against a tangible system. As simulations continue to improve and are more widely used, the optimization methods explored in this paper can be applied to countless models across all fields of study.

Bibliography

[1] Amaran, S. et al., Simulation optimization: a review of algorithms and applications, *4OR-Q Journal of Operations Research, vol. 12, pp. 301–333*, DOI:10.1007/s10288-014-0275-2, 2014.

[2] Ardent Utilities, How Long Does it Take to Repair Water Pipes?, https://www.ardentutilities.co.uk/how-long-does-it-take-to-repair-water-pipes, 2022.

[3] Avilia, Carlos Andres Macias et al., Leakage Management and Pipe System Efficiency. Its Influence in the Improvement of the Efficiency Indexes, *Water Vol. 13*, DOI:10.3390/w13141909, 2021.

[4] Bunn, Brendan, An Operational Model of Interdependent Water and Power Distribution Infrastructure Systems, *Naval Postgraduate School, M.S. Thesis*, DOI:https://apps.dtic.mil/sti/pdfs/AD1065269, 2018.

[5] Caron, J., Fong V., and Brion V., On the Use of Simulation and Optimization for Mission Modules Selection in a Maritime Context, *Military Operations Research, Vol. 24, No. 1, pp. 41-56*, 2019.

[6] Checco, John, Berg, Bjorn, and Loerch, Andrew, Optimizing US Army Force Size Under Uncertainty Through Stochastic Programming, *Military Operations Research, Vol. 22, No. 2, pp. 19-38*, 2017.

[7] City of Rockville, Standard Prices for Cost Estimating, *Department of Public Works*, https://www.rockvillemd.gov/DocumentCenter/View/1121/Standard_Prices_Cost_Est_Permit, 2010.

[8] CrossFit, The ASA Statement on P-Values: Context, Process, and Purpose, https://www.crossfit.com/health/the-asa-statement-on-p-values-context-process-and-purpose, 2020.

[9] DTK Hydronet Solutions, Leakages in Water Distribution Networks, https://www.dtkhydronet.com/post/leakages-in-water-distribution-networks, 2019.

[10] Hale, David et al., Revisiting the Application of Simultaneous Perturbation Stochastic Approximation towards Signal Timing Optimization, https://transit.ce.unlv.edu/apaz/wp-content/uploads/2017/09/Revisiting-the-Application-of-Simultaneous-Perturbation-Stochastic-Approximation-towards-Signal-Timing-Optimization.pdf.

[11] Hansen, Jennifer, Water Flow: The Average GPM Per Household, *Water Filter Spot*, https://waterfilterspot.com/water-flow-the-average-gpm-per-household/, 2021.

[12] Jeng Wang, I. and Spall James, Stochastic optimisation with inequality constraints using simultaneous perturbations and penalty functions, *International Journal of Control, Vol. 81 No. 8 pp. 1232–1238*, DOI:10.1080/00207170701611123, 2008.

[13] Jian, Nanjing and Henderson, Shane, An Introduction to Simulation Optimization, *Cornell University*, https://people.orie.cornell.edu/shane/pubs/WSC2015TutSlides, 2015.

[14] Kleinman, Nathan, Spall, James, and Naiman, Daniel, Simulation-Based Optimization with Stochastic Approximation Using Common Random Numbers, *INFORMS Vol. 45 No. 11 pp. 1570–1578*, DOI:0025-1909/99/4511/1570.

[15] Klise, Katherine et al., An Overview of the Water Network Tool for Resilience (WNTR), *CCWI 2018 Joint Conference, Kingston, Ontario, Canada*, https://www.osti.gov/servlets/purl/1510389, 2018.

[16] Klise, Katherine et al., A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study, Environmental Modelling and Software, pp. 420–431, DOI:10.1016/j.envsoft.2017.06.022, 2017.

[17] Klise, Katherine et al., Water Network for Resilience (WNTR) User Manual, *United States Environmental Protection Agency and Sandia National Laboratories*, DOI:EPA/600/R-20/185, 2020.

[18] Li, Zewei and Spall, James, Discrete Stochastic Optimization for Public Health Interventions with Constraints, https://arxiv.org/pdf/2206.13634.pdf, To appear in *Operations Research Forum*.

[19] Mabrok, Mohamen et al., Modeling and simulations of Water Network Distribution to Assess Water Quality: Kuwait as a case study, *Alexandria Engineering Journal pp. 11859–11877*, DOI:10.1016.j.aej.2022.05.038, 2022.

[20] Maskit, M. and Ostfeld, A., Leakage Calibration of Water Distribution Networks, *Procedia Engineering Vol. 89 pp. 664–671*, 2014.

[21] National Technology and Engineering Solutions of Sandia, Software quality assurance, https://wntr.readthedocs.io/en/stable/developers.html, 2019.

[22] Nguyen, Anh-Tuan, Reiter, Sigrid, and Rigo, Philippe, A review on simulation-based optimization methods applied to building performance analysis, DOI:10.1016/j.apenergy.2013.08.061, 2013.

[23] Oregon Residential Specialty Code, Sizing of Water Piping System, http://ecodes.biz/ecodes_support/free_resources/Oregon/11_Residential/PDFs/Appendix, 2011.

[24] Rossman, L, L A, EPANET USERS MANUAL. *U.S. Environmental Protection Agency, Washington, DC*, DOI:EPA/600/R-94/057 (NTIS PB94-165610), https://uknowledge.uky.edu/wdst_manuals/1/, 1994.

[25] Rubinstein, Reuven and Kroese, Dirk, Simulation and the Monte Carlo Method, Third Edition, *John Wiley and Sons, Inc.*, DOI:10.1002/9781118631980, 2017.

[26] Shao, Yu et al., Real-Time Water Distribution System Hydraulic Modeling Using Prior Demand Information by Formal Bayesian Approach, *Journal of Water Resource Plan Management*, DOI:10.1061/(asce)wr.1943-5452.0001137, 2021.

[27] Shi, Jiahao and Spall, James, SQP-based Projection SPSA Algorithm for Stochastic Optimization with Inequality Constraints, *American Control Conference*, DOI:978-1-6654-4197-1, 2021.

[28] Soliman, Karim et al., The resilience of water supply systems, *ResearchGate*, DOI:10.13140/RG.2.2.25556.55689, 2020.

[29] South West Water FAQs, https://www.southwestwater.co.uk/frequently-asked-questions/leaks/how-long-does-it-take-to-repair-a-leak.

[30] Spall, James, An Overview of the Simultaneous Perturbation Method for Efficient Optimization, *Johns Hopkins APL Technical Digest, Vol. 19, No. 4*, 1998.

[31] Spall, James, Implementation for the Simultaneous Perturbation Algorithm for Stochastic Optimization, *IEEE Transactions on Aerospace and Electronic Systems, Vol. 34 No. 3 pp. 817–823*, 1998.

[32] Spall, James C., Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control, *John Wiley and Sons, Inc.*, DOI:10.1002/0471722138, 2003.

[33] Spall, James, Simultaneous Perturbation Stochastic Approximation, https://www.jhuapl.edu/spsa/.

[34] Spall, James, Stochastic Optimization, *Handbook of Computational Statistics: Concepts and Methods (2nd ed.)*, Springer-Verlag, Heidelberg, pp.173-201, 10.1007/978-3-642-21551-3_7, 2012.

[35] Stanford, Penalty Functions, https://web.stanford.edu/group/sisl/k12/optimization/MO-unit5-pdfs/5.6penaltyfunctions.pdf.

[36] Suribabu, C.R., Resilience-based optimal design of water distribution network, *Springer*, DOI:10.1007/s13201-017-0560-2, 2017.

[37] Wang, Long, Zhu, Jingyi, and Spall, James, Mixed Simultaneous Perturbation Stochastic Approximation for Gradient-Free Optimization with Noisy Measurements, *Annual American Control Conference pp. 27–29*, 2018.

[38] Whitesides, Randall W., Selecting the Optimum Pipe Size, *PDHonline*, https://www.pdhonline.com/courses/m270/m270content.pdf, 2015.

[39] Wikipedia, Hazen-Williams Equation, Last updated 22 October 2022.

[40] Zais, Mark and Laguna, Manuel, A Simulation-Optimization Approach to Estimate Workforce Requirements, *Military Operations Research, Vol. 22, No. 1, pp. 19-38*, 2017.