SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Robust optimal design of urban drainage systems

Submitted by

NG Jia Yi

Thesis Advisor

Stefano GALELLI
Samira FAZLOLLAHI

Engineering Systems and Design

A thesis submitted to the Singapore University of Technology and Design in fulfillment of the requirement for the degree of Doctor of Philosophy

2020

PhD Thesis Examination Committee

TEC Chair:	Prof. Peter Jackson
Main Advisor:	Assoc. Prof. Stefano Galelli
Co-advisor(s):	Dr. Samira Fazlollahi
Internal TEC member 1:	Assoc. Prof. Bikramjit Das
Internal TEC member 2:	Asst. Prof. Nuno Ribeiro
External TEC member 1:	Asst. Prof. Selin Damla Ahipasaoglu

Abstract

Engineering Systems and Design

Doctor of Philosophy

Robust optimal design of urban drainage systems

by NG Jia Yi

Urban drainage systems are critical to cities as they handle large volumes of runoff generated during rainfall events, thereby reducing the risks of floods. Current design practices for drainage systems relies on design storms – critical rainfall events of given intensity and duration corresponding to a given return period. However, rainfall conditions can be varied, in terms of intensity, duration, and profiles (i.e., the time distribution of rainfall). Systems optimized with respect to design storms are expected to perform well under similar and less intense rainfall conditions, but this may not always be guaranteed. Designing robust drainage systems – solutions that perform consistently under a broad range of rainfall events – is therefore a challenging task. In this thesis, we address this task by first identifying the flaws of existing design practices and then proposing a novel framework that tackles these flaws, thus supporting the design of optimal, and robust, drainage systems.

In the first part of the thesis, we inspect existing design practices and identify when and why design storms may fail to produce robust solutions. To do this, we develop a computational framework that evaluates the robustness of drainage systems optimized for a design storm. The framework consists of four building blocks. First, we use sensitivity analysis to identify the most important decision variables [e.g., pipe expansions and low impact development (LID)], thereby reducing the complexity of the design problem. Second, we solve the problem using a multi-objective simulation-optimization scheme, which yields a set of Pareto-efficient solutions optimizing various measures of performance. Following current practice in drainage system design, the first two steps rely on a design storm. Third, we simulate each solution under stochastic rainfall events characterized by different duration, intensity, and profile, and finally evaluate their robustness.The application of this framework to the Nhieu Loc-Thi Nghe basin (Ho Chi Minh City, Vietnam) reveals that that none of the Pareto-efficient solutions are robust across all rainfall events. In particular, we find that the optimized solutions underperform when rainfall intensities, duration, and profiles deviate from those of the design storms. The first part of this thesis thus elucidates the need to include stochastic rainfall events throughout the design process so as to obtain robust drainage solutions.

In the next part of the thesis, we contribute a framework that builds on stochastic rainfall events –instead of design storms– for the entire design process. In particular, the proposed framework begins with the stochastic rainfall generation step and considers multiple rainfall events, representing a wide range of intensity, duration, and profiles, in both sensitivity analysis and optimization steps. To overcome the increase in computational requirements, we use emulation modelling techniques to replace the urban hydraulic simulator in the optimization step. We compare our proposed framework to the design storm based method and demonstrate that the proposed framework is more effective in finding drainage systems that are robust against a broad range of rainfall conditions. We also show that it is more efficient in terms of computational power required, solving the design problem 12 times (approximately 600 hours) faster than the conventional method.

Publications

Journals

Ng, J.Y., Fazlollahi, S., & Galelli, S. (2020). Do Design Storms Yield Robust Drainage Systems? How Rainfall Duration, Intensity, and Profile Can Affect Drainage Performance. *Journal of Water Resources Planning and Management, 146*(3), 04020003. Selected as Editor's Highlight.

Ng, J.Y., Fazlollahi, S., Dechesne, M., Soyeux, E., & Galelli, S. (2020). *Robust optimal design of urban drainage systems using surrogate-assisted optimization.* Manuscript in preparation.

Conferences

Ng, J.Y., Fazlollahi, S., & Galelli, S. (2018). A surrogate-based optimization framework for the deployment of low impact development solutions in large watersheds. *EWRI Congress 2018, Minneapolis, MN, June 3-7, 2018*

Ng, J.Y., Fazlollahi, S., & Galelli, S. (2019). A surrogate-based optimization framework for the robust design of stormwater drainage systems. *EWRI Congress 2019, Pittsburgh, PA, May 19-23, 2019*

Ng, J.Y., Fazlollahi, S., & Galelli, S. (2020). Designing robust stormwater drainage systems with data-driven emulators. *iEMSs 2020, Brussels, Belgium, September 14-18, 2020*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 An overview of urban drainage design - history, practices, challenges, and opportunities

Urban settlements throughout the globe are experiencing growing flood risks, owing to uncontrolled developments, population growth, sea level rise, and climate-driven changes in rainfall patterns. To prevent and minimize flood damages during extreme rainfall events, robust drainage systems are paramount (Butler and Davies, 2000; Bach et al., 2014). These infrastructures consist of drainage networks and, in some cases, low impact development (LID). The former represent a conventional and efficient way to convey large volumes of stormwater, while the latter—also known as sustainable drainage systems (SuDS), water sensitive urban design (WSUD), and green infrastructure (GI) (Fletcher et al., 2015)—manages stormwater by promoting runoff infiltration and retention (Ahiablame, Engel, and Chaubey, 2012). Some common examples of LID include green roofs, pervious pavements, and rainwater harvesting.

The design of drainage systems has evolved during the past two centuries. Before the 1800s, drainage systems were generally developed through uncoordinated efforts rather than formal engineering design (Burian et al., 1999). This changed in 1843, when the first comprehensively designed sewerage system was installed in Hamburg (Butler and Davies, 2000). In the following decades, the success was emulated in other major cities, such as London, Paris, and Chicago (Cain, 1972). It was also in this period that the design of drainage systems started to account for urban hydrology. The initial methods in mid-1800s used empirical equations or look-up tables (based on sewer slope and basin area) to calculate runoff and size the sewer pipes. Rainfall variables were only first explicitly considered when Mulvaney, 1851 developed the Rational method, which linked peak flow to rainfall intensity. Rainfall intensity could finally be associated to rainfall frequency in the 1930s, when intensity-duration-frequency (IDF) curves were derived from rainfall records (Adams and Howard, 1986). The IDF curves estimate the frequency of occurrence (otherwise known as the return period) of rainfall events with given intensity and duration; conversely, given a return period, the IDF curve estimates the corresponding intensity of rainfall events of a specified duration (Sun et al., 2019). With IDF curves, one can derive the so-called design storms – critical rainfall events used to test the endurance of hydraulic structures. IDF curves thus marked the beginnings of ‘design storms’ in urban drainage design.

Design storms remain in use today, although the approaches for modelling and designing urban drainage systems have advanced significantly. The first advancement was the introduction of optimization techniques in the 1960s (Guo, Walters, and Savic, 2008). Prior to this time, design only ensured that drainage systems could satisfy flow constraints, while optimization allowed some objectives—typically the cost—to be minimized, thus generating low-cost solutions that satisfied those constraints. Linear programming, nonlinear programming, and dynamic programming were among the early optimization techniques (Dajani and Hasit, 1974; Mays and Yen, 1975; Mays and Wenzel Jr, 1976). They usually solved for pipe sizes and slopes to ensure that pipes could carry peak flow capacity as specified by design storms, which were uniformly distributed over time. This implies that these optimization techniques neglected the timing of peak flows, an assumption that allowed the problem formulation to take a structure that optimization techniques could handle. Another major advancement was the development of process-based models, such as the Storm Water Management Model (SWMM) (Rossman, 2015), which enabled practitioners to use more sophisticated runoff hydrographs and storm hyetographs. These models can be easily coupled with evolutionary algorithms—or other metaheuristics—to aid drainage systems design (e.g., Wang et al., 2018). The key advantage here is that the simulation-optimization approach allows accounting for some physical characteristics—such as design storm profiles (i.e., how the rainfall depth is distributed over time) or the instantaneous flow within drainage networks—that cannot be easily modelled with traditional optimization techniques. Many studies have used the simulation-optimization approach to optimize pipe sizes (Yu et al., 2017; Ogidan and Giacomoni, 2017), storage capacities (Maharjan et al., 2009), and operations of pumps, weirs, and orifices (Beraud et al., 2010; Le Quiniou, Mandel, and Monier, 2014; Rathnayake, 2015). Yet, the approach is rather computationally intensive (Maier et al., 2014), a feature that limits the design process to a few design storms with specific rainfall duration, intensity, and profile.

Since design storms remain an essential component of the design process in both academia and engineering practice (e.g., *Code of Practice on Surface Water Drainage* 2011; *Urban Stormwater Management Manual for Malaysia* 2012; Kellagher, 2013; *Drainage Design Guide* 2018; *Stormwater Drainage Manual* 2018), it is important to understand whether urban drainage solutions obtained using design storms are robust against different rainfall conditions. A robust system is expected to perform well under a broad range of plausible conditions that could deviate from the conditions for which the system was designed (Herman et al., 2015; McPhail et al., 2018). In the context of drainage design, we would thus expect a robust solution to maintain the same performance (e.g., overflow reduction) for rainfall events that occur as frequently as the design storm, and possibly outperform for smaller and more frequent events. It should also be considered that the design of drainage systems is not necessarily aimed at eliminating floods during a design storm, since there are instances in which a drainage system can only partially reduce the flood extent—a common problem in cities with limited resources. In such case, can its robustness be ensured? This question gains further prominence if one considers that LIDs—which are becoming increasingly popular—are particularly sensitive to rainfall conditions (c.f., Damodaram and Zechman, 2013; Schmitter et al., 2016).

Till date, only a handful of studies have dropped the concept of design storms in favour of a stochastic evaluation of system performance. For example, Yazdi and Neyshabouri, 2014 optimized the height of flood control detention dams by minimizing the expected annual flood damage, which was evaluated using random samples of annual maximum rainfall. Yazdi, Lee, and Kim, 2014 optimized the pipe sizes and number of pumps in a drainage system through a risk-based measure of performance, which was estimated by simulating the behaviour of the drainage system on random samples of rainfall events characterized by different depths and duration. Importantly, the authors found that solutions obtained with the risk-based approach outperformed the ones relying on design storms. While the aforementioned represent successful attempts at designing solutions that do not build on design storms, it may be worthwhile to first evaluate the robustness of drainage systems designed using design storms and understand when and why design storms may fail to produce robust designs.

If design storms are shown to be inadequate, it follows that we could possibly include stochastic rainfall events in the optimization process such that the urban drainage solutions are robust against a wide range of rainfall conditions. However, this poses a new challenge, that is the high computational cost associated with the greater number of simulations needed – the reason why only a few design storms are usually used for optimization in the first place. Fortunately, surrogate modelling is becoming a feasible and popular alternative to simulation models (Razavi, Tolson, and Burn, 2012; Benner, Gugercin, and Willcox, 2015). There are two broad families of surrogate models: one family consists of empirically data-driven models that approximate response of original simulation model by interpolation or regression; the other family consists of simplified physically-based models of the original simulation models. Although surrogate models may sometimes compromise on model accuracy, they are computationally cheaper and faster alternatives to simulation models. When used in settings which require iterated use of simulation models, such as optimization, much computational time could be saved. This make surrogate-based optimization an attractive option for the design of robust urban drainage system. While surrogate models for urban drainage simulator have been explored (Machac, Reichert, and Albert, 2016; Carbajal et al., 2017; Moreno-Rodenas et al., 2018) and application of surrogates in optimization for water resources problems are getting more popular (Razavi, Tolson, and Burn, 2012; Castelletti et al., 2012a), we have yet to find out the feasibility of using surrogates in optimization for urban drainage, its potential advantages and possible flaws. This opens up opportunities to design more effective urban drainage without only relying on design storm: the resulting optimal designs of pipe network and LIDs could possibly be more robust against a wide range of rainfall conditions.

1.2 Contributions of the thesis

In this thesis, we offer a new perspective on the design of urban drainage systems, one that does not necessarily have to rely on design storm and will effectively contribute to the broader goal of designing robust urban drainage systems. To do so, we first examine the current practice in drainage system design, that is one that uses design storms, and investigate its effectiveness in producing robust solutions. Next, we propose a new method of designing robust urban drainage systems, one that uses stochastic rainfall events within a surrogate-based optimization framework. Lastly, we test the scalability of this framework by applying it to a higher dimensional problem. For each step, we apply our work to a real-world case study to understand the practicality of our design framework.

In Chapter 2, we first optimize urban drainage systems using design storms and evaluate the robustness of solutions to find out when and why design storms may fail to produce robust designs. More specifically, we investigate if solutions optimized for design storm are robust against other rainfall events with intensity, duration, and profile that deviates from the design storm. We also determine which component of urban drainage (i.e., LID or pipe network) contributes more to the effectiveness and robustness of urban drainage systems. Lastly, we identify the rainfall characteristics that affect robustness of the solutions.

In Chapter 3, we test out a new approach to urban drainage design. Instead of using the design storm, we include stochastic rainfall events during optimization and seek optimal solutions based on objectives that involve measures of robustness. This allows us to find out if including stochastic rainfall events during the optimization phase can result in solutions that are robust against various rainfall conditions. However, doing so increases computational demand. To overcome this, we construct a Gaussian Process (GP) emulator to approximate the simulation model. We first identify the tradeoff between accuracy and complexity of the GP emulator used to emulate overflow of the urban drainage simulator. Following that, we use the emulator in a surrogate-assisted optimization framework to investigate its feasibility in obtaining robust solutions within a shorter time.

In Chapter 4, we scale up the framework developed in Chapter 3 by applying it to a higher dimensional optimization problem (i.e., greater number of decision variables for the urban drainage design). By doing so, we determine how the emulator accuracy changes with problems of increasing complexity. We also find out if surrogate-assisted optimization can still remain an attractive option to produce robust urban drainage systems.

Finally, we conclude the thesis in Chapter 5 with a summary of our results and a discussion on the future prospects on urban drainage systems.

Chapter 2

Optimizing drainage system using design storms: are solutions robust?

2.1 Introduction

The common practice to urban drainage system design is to rely on design storms; however such a method may not guarantee that solutions are robust against rainfall conditions that deviate from the design storm. While there have been successful attempts to design solutions that do not rely on design storms (Yazdi and Neyshabouri, 2014; Yazdi, Lee, and Kim, 2014), it is worthwhile to first evaluate the robustness of drainage systems designed using design storms and understand when and why design storms may fail to produce robust designs. In this chapter, we aim to improve our understanding of the potential flaws behind the adoption of design storms in drainage systems design. In particular, we aim to answer the following questions: 1) are all solutions optimized for a design storm robust to other rainfall events?; 2) which components of an urban drainage system contribute more to robustness?; and 3) what are the rainfall characteristics (i.e., duration, intensity, and profile) that affect the robustness of a drainage system?

To this goal, we contribute a novel computational framework that could be readily applied to any drainage system. The framework consists of four building blocks, namely sensitivity analysis, simulation-optimization, stochastic generation of rainfall events, and robustness analysis. Sensitivity analysis first identifies the drainage components that contribute most to overflow reduction; then, simulation-optimization searches for the Pareto-efficient configuration of these components. These two steps rely solely on the design storm, as in current practices in urban drainage design. We then simulate the Pareto-efficient solutions under stochastic rainfall events (characterized by different duration, intensity, and profile), and evaluate their robustness across such broad range of rainfall conditions. Using the Nhieu Loc-Thi Nghe basin (Ho Chi Minh City, Vietnam) as case study, we show that the framework can be used not only to identify the most robust solutions, but also to determine how drainage components and rainfall variables influence robustness.

2.2 Case study

2.2.1 Study site: Nhieu Loc-Thi Nghe Canal and Basin

The Nhieu Loc-Thi Nghe (NL-TN) basin is located in the central area of Ho Chi Minh City, Vietnam (Fig. 2.1). Occupying an area of 33 km^2 and stretching across seven city districts, the basin supports a population of 1.2 million people, roughly a fifth of the city's total population. Land use is dominated by residential developments (49.3% of the total area) and commercial, public, and industrial users. Domestic wastewater and stormwater are collected in the NL-TN Canal and eventually discharged into the Saigon River, located in the southeastern part of the city. The NL-TN Canal was heavily polluted when informal settlements developed along its banks in the 1960s. Since 1995, the city has cleaned up the canal, relocated informal settlements, and improved the drainage system (Wust, Bolay, and Du, 2002). However, even with these measures in place, flood is still a common occurrence (*Ho Chi Minh City seriously flooded due to storm Usagi*; *Comprehensive plan needed to prevent flooding in HCMC*). Furthermore, continued sea level rise and increasing rainfall intensity could drive flood risks beyond current levels by mid-century (Lempert et al., 2013).

In our study, the NL-TN basin is simulated with EPA SWMM 5.1 (Rossman, 2015) and represented by 228 sub-catchments and 333 conduits (of which 25 represent the NL-TN canal (Fig. 2.1)). The model calibration is reported in Ho et al., 2015, who compared observed and simulated water depth along the NL-TN canal for a 2-year design storm that produced 90.2 mm of rainfall over a period of 3 hours. In their experiment, Ho et al., 2015 defined the downstream boundary conditions through a typical tidal curve at Phu An station, located at the outlet of the NL-TN canal. In this curve, the tide peaks at 1.48 m at the hour that directly follows the rainfall peak, thereby worsening the consequences of the rainfall event. For this event, SWMM reported reasonable accuracy for both water depth and flood locations.

2.2.2 Rainfall data

Rainfall data are obtained from the Southern Regional Hydrometeorological Center of Vietnam, and are represented by a 10-year time series (with hourly resolution), which was recorded at the Tan Son Hoa meteorological station (Fig. 2.1) from 1 Jan 2008 to 31 December 2017. The average annual precipitation recorded at Tan Son Hoa is 2,075.5 mm, with the monsoon season (May to October) accounting for nearly 80% of the total rainfall.

To proceed with our analysis, we extract rainfall events from the time series. In particular, a rainfall event is defined as a period of time with at least 0.2 mm of rain and separated from preceding and succeeding rainfall events by at least 4 hours. Using this definition, 1,720 rain events were identified from the 10-year rainfall time series. Table 2.1 shows the event-based statistics: the mean duration of rainfall events is 3.4 hours; total rainfall depth can vary widely, ranging from 0.2 mm to 184.8 mm (although 95% of the events have a volume lower than 48.1 mm). The most extreme rainfall events in these 10 years have maximum hourly intensity of 116.7 mm/hr and

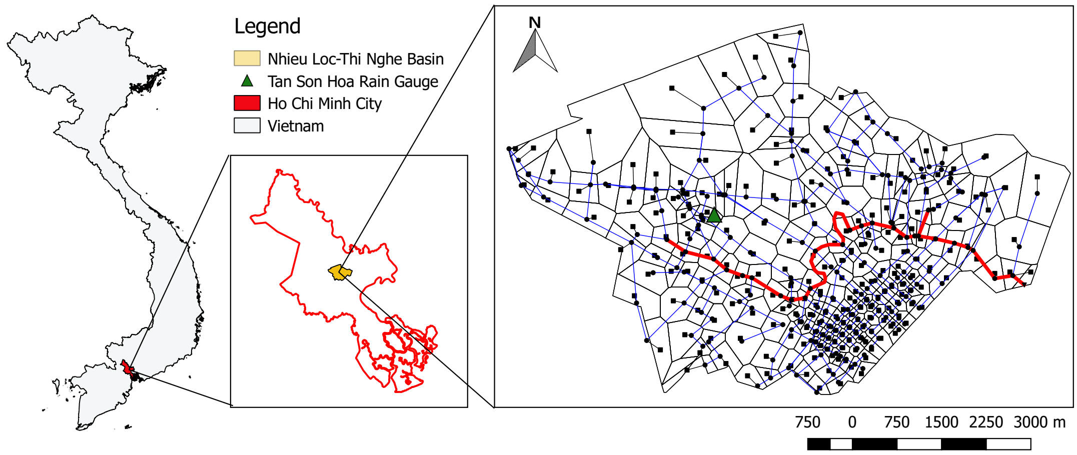


FIGURE 2.1: Map (left) and SWMM representation (right) of the Nhieu Loc-Thi Nghe Canal and Basin. In the SWMM representation obtained from Ho et al., 2015, the NL-TN canal is highlighted in red and the drainage network is highlighted in blue. The sub-catchments and nodes in the drainage network are represented by squares and circles respectively.

TABLE 2.1: Event-based statistics of the rainfall events obtained from the 10-year time series. P5 and P95 correspond to the 5th and 95th percentile.

Parameter	Min	Max	Mean	Median	P5	P95
Event duration (h)	1	30	4.0	3	1	11
Rain duration (h)	1	26	3.4	2	1	9
Dry weather period (h)	4	2903	46.7	21	4.9	145.1
Total rain (mm)	0.2	184.8	12.1	4.3	0.2	48.1
Max rain intensity (mm/h)	0.1	116.7	8.2	3.1	0.2	32.4
Average rain intensity (mm/h)	0.04	41.6	3.2	1.4	0.2	12.5

an average intensity of 41.6 mm/hr. Note that the 3-hour, 2-year return period design storm obtained from Ho et al., 2015 has a rainfall depth of 90.2 mm and a maximum hourly intensity of 67.4 mm/hr.

2.3 Computational framework

The computational framework developed for this study consists of four steps, illustrated in Fig. 2.2. We begin with a sensitivity analysis (Step 1), which is aimed at determining which decision variables (e.g., pipe expansions, implementation of LIDs) should be considered in the design process. Such step is not strictly necessary, but it can help reduce the dimensionality of the design problem, especially when dealing with large drainage systems (such as the one considered here). The selected decision variables are then used to formulate and solve a simulation-optimization problem (Step 2), where SWMM is coupled with a multi-objective evolutionary algorithm (MOEA) to

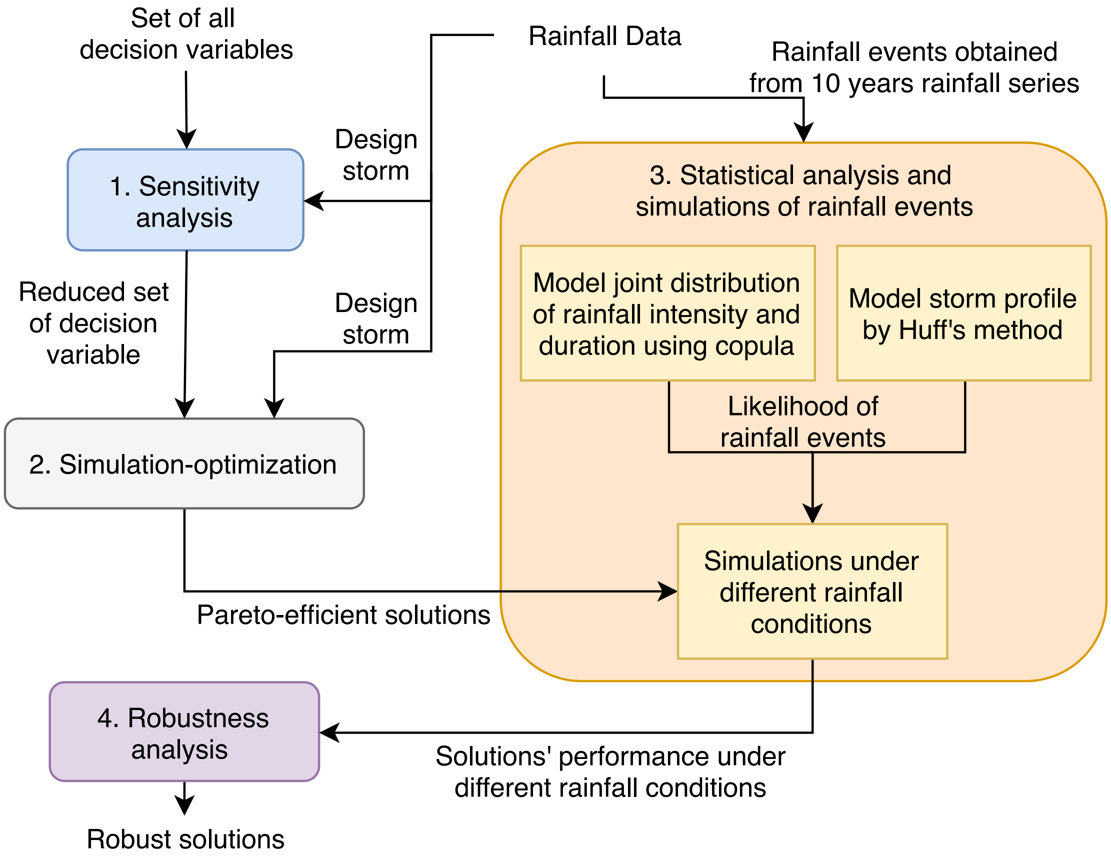

Set of all decision variables
Rainfall Data
Rainfall events obtained from 10 years rainfall series
Design storm
1. Sensitivity analysis
3. Statistical analysis and simulations of rainfall events
Reduced set of decision variable
Design storm
Model joint distribution of rainfall intensity and duration using copula
Model storm profile by Huff's method
Likelihood of rainfall events
2. Simulation-optimization
Simulations under different rainfall conditions
Pareto-efficient solutions
4. Robustness analysis
Solutions' performance under different rainfall conditions
Robust solutions

FIGURE 2.2: Flowchart of the computational framework, consisting of 4 main steps: 1) sensitivity analysis to select the decision variables for the optimization problem; 2) simulation-optimization to obtain Pareto-efficient solutions; 3) statistical analysis and simulation of rainfall events to simulate the Pareto-efficient solutions under multiple stochastic rainfall replicates; 4) robustness analysis to evaluate the capability of the solutions to attain a minimum level of performance.

find a set of Pareto-efficient solutions, or designs, that minimize the investment costs while maximizing the reduction in both overflow volume and flooded nodes during the design storm. Note that, following the current practice in drainage system design, both sensitivity analysis and simulation-optimization rely on a design storm. The solutions so identified are simulated for multiple stochastic rainfall replicates (Step 3), whose duration, intensity, and profile are modeled with the aid of copula functions and Huff's method. The results from the simulation are finally used in the robustness analysis (Step 4), where we determine, for each solution, the capability of attaining the performance it is designed for. This approach is similar to that of Kasprzyk et al., 2013, in which a rich set of alternative solutions is first generated using a MOEA and then their performance assessed under conditions that deviate from those used to evaluate optimality.

2.3.1 Sensitivity analysis

The NL-TN basin is a large watershed, so there are many infrastructural interventions that could be considered to reduce flood-related damages. In particular, the initial decision space includes expanding any of the 308 pipes in the drainage system and, as suggested by Ho et al., 2015, implementing 4 types of LID within 3 of the 228 sub-catchments (these are the only sub-catchments for which detailed information on the land use is available). The LIDs considered are rainwater harvesting, green roofs, urban green spaces, and pervious pavements. Together, pipe expansions and LIDs yield a total of 320 decision variables; a number that would make a simulation-optimization problem intractable. That is because a large number of decision variables would require an extensive number of simulations with SWMM to explore effectively the decision space. Furthermore, one should consider that not all decision variables would necessarily contribute to the reduction of overflow volume (or number of flooded nodes). We thus resort to sensitivity analysis (SA) to identify the decision variables that are potentially most important at reducing overflow for the 2-year, 3-hour design storm. Two SA methods are used: the elementary effect test (EET) and the extended Fourier amplitude sensitivity test (eFAST).

EET is first adopted as a screening tool to discard non-influential variables before using eFAST, which is a more accurate but time consuming SA (Pianosi et al., 2016). The output considered here is the reduction in total overflow volume, while the input factors are the diameters of 308 pipes and area of the 12 LIDs. (Note that in this section we use the term 'input factor', generally adopted in the SA literature, instead of 'decision variable'.) EET perturbs the input factors of the simulation model one at a time from multiple points within the input space. It then measures the global sensitivity by taking the mean of the elementary effects, namely the local derivatives of the output with respect to an input. A total of $r(M+1)$ evaluations is needed, where M is the number of factors and r the number of elementary effects. In this instance, $M = 320$ and $r = 100$—the value of r is chosen by checking that the convergence of the sensitivity estimates is reached (Pianosi et al., 2016). The set of factors that have a positive mean elementary effect on total overflow reduction (78 pipes and 8 LIDs) is then selected for the subsequent SA step.

eFAST (Saltelli, Tarantola, and Chan, 1999) is carried out to further reduce the number of input factors. This variance-based SA assesses the sensitivity of an input factor by determining its contribution to the output variance. For each input factor, eFAST calculates the total-order index, which accounts for the direct effect of the input on the output (in our case, the total overflow reduction) as well as its interactions with the other input factors. This is especially important in our application since interactions amongst pipes (and LIDs) could amplify the individual effects of expanding a single pipe. The only eFAST parameter to set is the number of samples per input factor—equal to 514, as recommended in Saltelli, Tarantola, and Chan, 1999. This number is lower than other variance-based methods due to a tailored sampling strategy employed by eFAST. The factors identified by eFAST are ranked according to their value of the total-order index, yielding a total of 20 input factors (or decision variables), 12 of which are pipe variables and 8 are LID variables. Table 2.2 and 2.3 list the properties of these 20

TABLE 2.2: Description of the pipe variables selected from sensitivity analysis

Decision variable, x_j	Pipe no.	Shape	Length, l_j (m)	Size, d_j (m)	Maximum size (m)
x_1	1	Rectangular closed	546	2.5 x 2.5	4
x_2	2	Circular	360	1.5	2.5
x_3	3	Circular	510	1.2	2.2
x_4	4	Circular	180	1	2
x_5	5	Rectangular closed	840	1.5 x 2.8	4
x_6	6	Rectangular closed	600	1.2 x 2	4
x_7	7	Rectangular closed	727	2 x 4	4
x_8	8	Rectangular closed	540	0.8 x 5	4
x_9	9	Rectangular closed	551	1.2 x 5	4
x_{10}	10	Rectangular open	200	3.2 x 10	4.2
x_{11}	11	Rectangular open	610	2.7 x 10	4
x_{12}	12	Rectangular closed	900	1.8 x 4	4

Maximum size refers to the maximum diameter for circular pipes and maximum depth for rectangular pipes. It corresponds to the maximum size of an existing pipe of the same shape or 1 m larger than the original size, whichever is bigger. Width of rectangular pipes are unchanged during optimization.

decision variables. For more details about the SA results, please refer to Appendix A.

2.3.2 Simulation-optimization

With the reduced set of decision variables selected through the SA, the next step is to identify the optimal solutions with the aid of a simulation-optimization scheme. In our application, we couple SWMM (forced with the same design storm used in the SA) with NSGA-II (Deb et al., 2002), which has been widely used to solve various water resources problems (Nicklow et al., 2009), including the optimization of urban drainage systems (Beraud et al., 2010; Damodaram and Zechman, 2013; Yazdi and Neyshabouri, 2014; Le Quiniou, Mandel, and Monier, 2014; Rathnayake, 2015; Di Matteo, Dandy, and Maier, 2017). Although many advanced Multi-Objective Evolutionary Algorithms (MOEAs) have been recently released (see Reed et al., 2013), we note that NSGA-II performs comparatively well against more recent algorithms when a limited computational budget is available (Wang et al., 2014; Zheng et al., 2016). We also note that the overall goal of our study is to investigate the relation between design storms and the robustness of drainage systems, so the choice of the 'best' MOEA is not the main focus.

The decision variables considered here are the pipe diameters and the number of LID units. Diameters are only allowed to take discrete values, which correspond to existing pipe sizes in the network. The LIDs are assumed to have a fixed unit area, so the overall area of LIDs implemented in the sub-catchments is modified by changing the number of units simulated by SWMM. In particular, green roofs have an area of 60 m^2 per unit, while urban green spaces and pervious pavements an area of 100 m^2 per

TABLE 2.3: Description of the LID variables selected from sensitivity analysis

Decision variable, x_j	LID	Sub-catchment properties			Maximum no. of units
		Sub-catchment no.	Area (ha)	Impervious %	
x_{13}	Green roofs	1	31.45	70	1102
x_{14}	Pervious pavements				900
x_{15}	RWH system				1102
x_{16}	RWH system	2	12.55	70	406
x_{17}	Green roofs	3	102	70	2726
x_{18}	Pervious pavements				1400
x_{19}	Urban green spaces				4000
x_{20}	RWH system				2726

RWH = rainwater harvesting

unit, as modelled in Ho et al., 2015. Rainwater harvesting is represented by a 1 m^2 rain barrel per unit. The maximum number of LID units depends on the number of households and existing roads/parks in the sub-catchments (Ho et al., 2015). All other LIDs settings in SWMM, such as soil thickness, antecedent soil moisture, or porosity, are not modified during the optimization process (further details about the decision variables are reported in Table 2.2 and 2.3).

We consider three objectives, namely the reduction in total overflow volume ($J^{Overflow}$), the reduction in the number of flooded nodes (J^{Node}), which are to be maximized, and the capital cost (J^{Cost}), which is to be minimized. $J^{Overflow}$ accounts for the flood extent (since SWMM does not simulate 2D surface inundation), and is defined as follows:

$$J^{Overflow} = 1 - \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} f_{i,t}(\mathbf{x})}{F_{baseline}} , \tag{2.1}$$

where $\mathbf{x}$ is the vector of decision variables, T the total number of time instances (5 min intervals) in the design storm, N the total number of nodes in the drainage network, $f_{i,t}(\cdot)$ the overflow volume in the i-th node at time t (calculated by SWMM), and $F_{baseline}$ is the total overflow volume during the design storm for the existing drainage system which can be evaluated using the formula in the numerator by substituting $\mathbf{x}$ with the zero vector. The second objective, J^{Node}, accounts for the spatial distribution of a flood event:

$$J^{Node} = 1 - \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\sum_{t=1}^{T} f_{i,t}(\mathbf{x})>0\}}}{N_{baseline}} , \tag{2.2}$$

where $\mathbb{1}_{\{\sum_{t=1}^{T} f_{i,t}(\mathbf{x})>0\}}$ is an indicator function that takes value equal to 1 if the overflow volume $f_{i,t}(\cdot)$ in the i-th node takes a positive value in any of the T time instances (and 0 otherwise), and $N_{baseline}$ is the total number of flooded nodes during the design storm for the existing drainage system which can be evaluated using the formula in the numerator by substituting $\mathbf{x}$ with the zero vector. Finally, the capital cost J^{Cost} is defined as:

$$J^{Cost} = \sum_{j=1}^{M_P} \alpha \times x_j \times l_j \times \mathbb{1}_{\{x_j > d_j\}} + \sum_{k=1}^{M_L} \beta_k \times a_k \times x_{M_P+k}\,, \tag{2.3}$$

where M_P and M_L are the total number of pipe and LID decision variables, respectively, α the cost of a pipe (in USD) per unit length and unit diameter ($0.8 /mm diameter / m length here, independent of the depth and pipe material, but inclusive of excavation cost associated with changing a pipe, estimated from USEPA, 1999), l_j and d_j the length and original diameter of pipe j (original depth if pipe j is rectangular), β_k the cost (in USD) per unit area of LID *k* ($100, $25, $100, $75 per m^2 of green roof, pervious pavement, urban green spaces and rainwater harvesting respectively, estimated from Sample, 2013), a_k the area of a single unit of LID *k*, and $\mathbb{1}_{\{x_j > d_j\}}$ an indicator function that takes value equal to 1 if the j-th pipe is expanded (and 0 otherwise).

The goal of the multi-objective optimization problem is to find the set of non-dominated solutions $\mathbf{x}^*$ that minimizes the vector $\mathbf{J}(\mathbf{x})$, namely:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{J}(\mathbf{x})\,, \tag{2.4}$$

where

$$\mathbf{x} = (x_1, ..., x_{M_P}, x_{M_P+1}, ..., x_{M_P+M_L})\,, \tag{2.5}$$

and

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} -J^{Overflow}(\mathbf{x}) \\ -J^{Node}(\mathbf{x}) \\ J^{Cost}(\mathbf{x}) \end{bmatrix}. \tag{2.6}$$

The problem does not have additional constraints besides the upper and lower bounds on the decision variables (specified in Table 2.2 and 2.3) and the hydraulic constraints that SWMM handles.

As mentioned above, all solutions identified by NSGA-II are simulated by SWMM using the 2-year, 3-hour design storm applied uniformly across the catchment. As for NSGA-II, the distribution indices for crossover and for mutation are 15 and 20, respectively, while the probabilities of crossover and mutation are 0.9 and 0.1 (a setup similar to the one adopted by Di Matteo, Dandy, and Maier, 2017). Since the number of function evaluations (determined, in this case, by the number of generations and population size) can largely influence the results (Wang et al., 2018), we carry out a preliminary analysis aimed at determining the number of function evaluations (50,000) that ensures convergence of the Pareto front (please refer to Appendix B for additional details). Each run has a population size of 200 and is terminated after 250 generations. The optimization problem is solved with 10 different random seeds to account for the variability of the stochastic search process and to ensure a wider Pareto front. The final set of Pareto-efficient solutions thus corresponds to the set of Pareto-efficient solutions identified across all 10 seeds. All experiments were performed on a Dual Intel Xeon

CPU E5-2630 v3 @ 2.40 GHz with 32 GB RAM running Microsoft Windows 8.1. With this setup, the simulation-optimization experiment took a total of 720 computational hours.

2.3.3 Rainfall analysis

To generate the stochastic rainfall events used in the robustness analysis, it is necessary to model the duration, intensity, and profile (or time distribution) of the events at Tan Son Hoa station. We start with duration and intensity, which are dependent on each other (Goel et al., 2000). To estimate the univariate distribution of rainfall duration and intensity, we consider four probability distributions commonly used in hydrology to fit rainfall variables. They are the normal, lognormal, exponential, and gamma distribution. The best distribution is selected based on likelihood maximization and visual inspection of distribution fit. Rainfall duration (in hours) is fitted to a lognormal distribution (mean = 0.961, standard deviation = 0.709) (Fig. 2.3a) and rainfall intensity (in mm/hour) is fitted to a gamma distribution (shape = 0.746, rate = 0.217) (Fig. 2.3b).

Duration and intensity of rainfall events are correlated. Hence, in addition to modeling the marginal distribution of rainfall duration and intensity, copula functions are used to model the joint probability distribution of these two variables (Genest and Favre, 2007; Salvadori and De Michele, 2007). The advantage given by a copula function is that it can capture the dependence structure of two random variables independently of the marginal distributions. This is important, since many variables in hydrology are not modelled using the same type of probability distribution.

The mathematical definition of copulas is as follows. Consider the random vector $(X_1, ..., X_p)$ with continuous marginals $F_1(x_1), ..., F_p(x_p)$. By applying probability integral transform, we obtain the random vector:

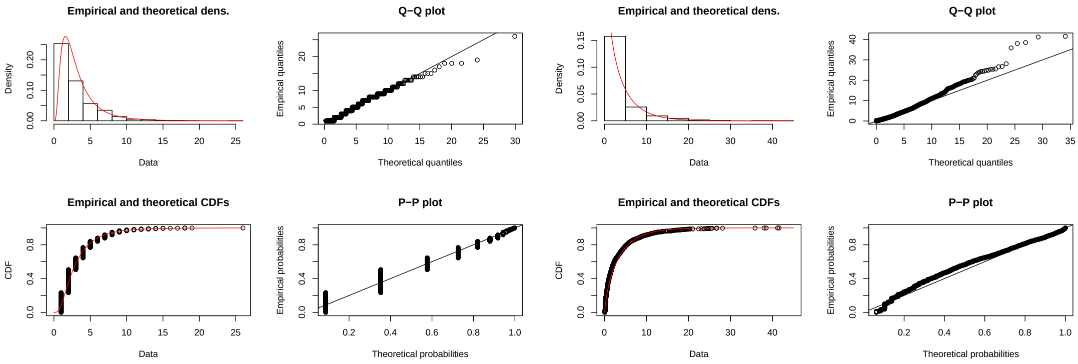
$$(U_1, ..., U_p) = (F_1(X_1), ..., F_p(X_p)), \tag{2.7}$$

which has standard uniform marginals. The copula of $(X_1, ..., X_p)$ is then the joint cumulative distribution function of $(U_1, ..., U_p)$, namely:

$$C(u_1, ..., u_p) = Pr(U_1 \leq u_1, ..., U_p \leq u_p). \tag{2.8}$$

Six copula families – Gaussian, t, Clayton, Gumbel, Frank, and Joe – are considered to model the dependence structure between rainfall duration and intensity. Clayton, Gumbel, Frank, and Joe copulas belong to a class of copulas called Archimedean copulas, which are popular, since they admit an analytical formula (unlike the Gaussian and t-copula) and only require one parameter to govern the strength of dependence in arbitrarily high dimensions. The bivariate Archimedean copula functions are given in Table 2.4. As the Gaussian and t-copula do not have a closed form, the reader is referred to Nadarajah, Afuecheta, and Chan, 2018 for more details.

We select the Frank copula (α = 1.40) based on likelihood maximization to model the joint probability distribution of intensity and duration. The Frank copula is a symmetric Archimedean copula (in contrast with the asymmetric Clayton and Gumbel copulas,



(A) (B)

FIGURE 2.3: Density, CDF, Q-Q, and P-P plot for a) empirical distribution of rainfall duration with fitted lognormal distribution, and b) empirical distribution of rainfall intensity with fitted gamma distribution.

TABLE 2.4: Bivariate Archimedean Copulas

Family	Bivariate Copula $C(u_1, u_2)$	Parameter α
Clayton	$(u_1^{-\alpha} + u_2^{-\alpha} - 1)^{-1/\alpha}$	$\alpha > 0$
Gumbel	$\exp\{-[(-\ln u_1)^\alpha + (-\ln u_2)^\alpha]^{1/\alpha}\}$	$\alpha \geq 1$
Frank	$-\frac{1}{\alpha}\ln(1 + \frac{(e^{-\alpha u_1}-1)(e^{-\alpha u_2}-1)}{e^{-\alpha}-1})$	$\alpha \neq 0$
Joe	$1 - [(1-u_1)^\alpha + (1-u_2)^\alpha - (1-u_1)^\alpha(1-u_2)^\alpha]^{1/\alpha}$	$\alpha \geq 1$

which exhibit greater dependence in the negative tail and positive tail, respectively), which means that the modelled dependence between the rainfall variables does not change with increasing duration or intensity. Random samples of rainfall duration and intensity generated from the Frank copula function gives a Spearman's correlation of 0.213, which is similar to the observed values of 0.235. Fig. 2.4 shows the 1,720 observed rainfall events together with the 4,000 rainfall events randomly generated using the Frank copula and the univariate distributions for intensity and duration.

To model the profile of rainfall events, we use the Huff's method (Huff, 1990), which has found successful application in previous studies on urban drainage (Yazdi, Lee, and Kim, 2014; Yu et al., 2017). To derive a curve that describes the cumulative fraction of storm rainfall as a function of time, the method follows three steps. First, it groups rainfall events into 1st, 2nd, 3rd, and 4th quartile storms according to whether the heaviest rainfall occurs in the 1st, 2nd, 3rd, or 4th quarter of the event. Second, it estimates four representative curves (one for each group) by averaging the profile of all events within the group. Finally, it estimates the empirical probability associated to each group. Since the available rainfall time series has a hourly resolution, the Huff's method is only applied to rainfall events with a duration of at least 4 hours. The profile of the 4 rainfall groups estimated for the study area are illustrated in Fig. 2.5a; the corresponding probabilities (for the 1st, 2nd, 3rd, and 4th quartile storms) are 45.8%, 24.9%, 18.1%, and 11.2%. Shorter rainfall events (<4 hours) are assumed to follow the same

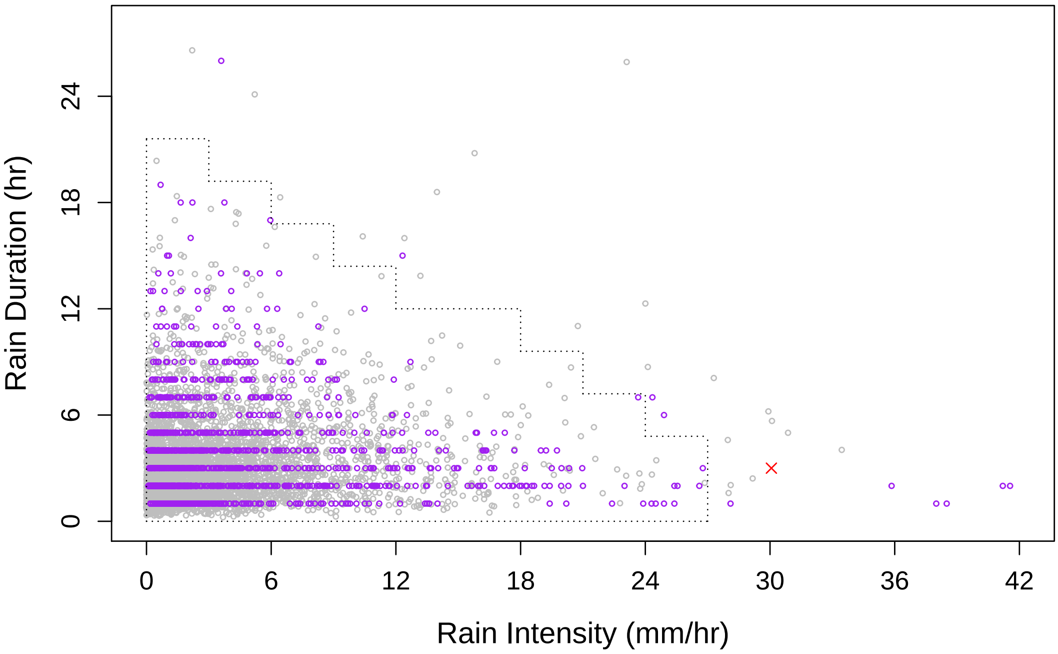


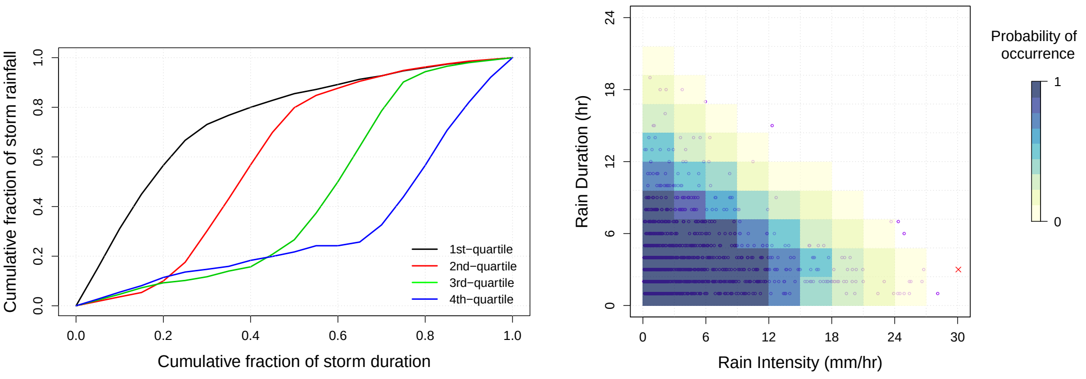
FIGURE 2.4: Grey circles are the stochastic rainfall replicates generated using the multivariate distribution based on Frank copula and the univariate marginals. Purple circles are rainfall events observed from 2008-2017. The red cross is the design storm used for both SA and simulation-optimization. Black dotted lines bound the rainfall conditions considered for the robustness analysis.

probability distribution of the 4 Huff's types.

The models described above allow generating stochastic rainfall events. Here, the challenge lies in representing a wide range of rainfall conditions, so as to meaningfully support the robustness analysis. Instead of using Monte Carlo simulation, which tends to be computationally intensive (Yazdi, Lee, and Kim, 2014), we approached the generation process by dividing the range of rainfall duration and intensity into intervals of equal size; events with the 'same' duration and intensity can then be further divided into 4 different profiles. As we shall see later in the results, this approach also helps understand how sensitive each optimized drainage system is to rainfall duration, intensity, and profile. The result of this process is illustrated in Fig. 2.5b: the ranges for average rainfall intensity and duration are set equal to $[0, 30]$ (mm/hr) and $[0, 24]$ (hours), with the two upper bounds (i.e., 30 and 24) falling at the 99.9th percentile of the respective univariate marginal distribution. These ranges are then divided into 10 equal intervals, resulting in a 10 by 10 gridded area. The choice of 10 intervals strikes a balance between two factors: first, the intervals must be wide enough to limit the number of grids and hence the number of SWMM simulations; second, the intervals need to be narrow to ensure that events falling within a grid would yield similar simulation outcome. Fig. 2.5b shows the probability associated with each grid, calculated using the marginal distributions and the Frank copula. Note that 49 grids have probability of at least 5% (shaded grids), meaning that the chance that within a given year a rainfall event occurs with duration and intensity within the range specified by the grid is at least 5%. The center for each of the 49 grids is then taken to be the representative rain event for the grid, with the underlying assumption that the SWMM simulation outcome does not differ greatly across different values of rainfall variables within a grid. The total rain volume was then distributed across time using a 5 minutes time step according to the 4 Huff's profile groups. Therefore, for each of the 49 grids, 4 time series of rain events with the same duration and intensity, but a different profile, are generated, yielding a total of 196 rainfall events.

2.3.4 Robustness analysis

Many robustness metrics have been proposed in the water resources community (Herman et al., 2015; Giuliani and Castelletti, 2016; McPhail et al., 2018), with each of them reflecting different levels of risk aversion. The robustness metric used here is a 'satisficing metric'—also known as Starr's domain criterion (Starr, 1963; Schneller and Sphicas, 1983)—that is based on the idea of counting the number of times a solution satisfies a pre-defined performance threshold. This metric has lately found successful application in problems such as reservoir operations (Quinn et al., 2018; Libisch-Lehner et al., 2019) and design of water supply portfolios (Herman et al., 2014). In our study, we use such metric to add up the weights corresponding to the rainfall events in which a drainage system satisfies the performance thresholds. To calculate the robustness R_s of the s-solution (returned by the simulation-optimization) across all rainfall events, we proceed in two steps. First, we calculate the score $S_{s,q}$, which takes value 1 if the performance $\mathbf{P_{s,q}}$ of the s-solution on the q-th event is element-wise greater than the



(A)

(B)

FIGURE 2.5: (a) Median time distribution curves for the four Huff's rainfall types (b) Probability of occurrence of the rainfall events calculated using the fitted marginal distributions and copula. Color of the grid represents the probability that a rainfall event with characteristics falling within the grid will occur in a year. The purple circles are the 1,720 rainfall events observed from 2008 to 2017. The red cross is the design storm used for the sensitivity analysis and simulation-optimization.

threshold $\mathbf{T_s}$ (and 0 otherwise). The score is formally defined as follows:

$$S_{s,q} = \begin{cases} 1 & \mathbf{P_{s,q}} \geq \mathbf{T_s} \\ 0 & \text{otherwise} \end{cases} , \tag{2.9}$$

where the performance $\mathbf{P_{s,q}}$ is the reduction in overflow volume and number of flooded nodes (calculated with Eq. 2.1 and 2.2), and the threshold $\mathbf{T_s}$ is the performance of solution s on the design storm, namely:

$$\mathbf{T_s} = \begin{bmatrix} J^{Overflow}(\mathbf{x_s}) \\ J^{Node}(\mathbf{x_s}) \end{bmatrix} . \tag{2.10}$$

The robustness metric R_s is then calculated by taking the weighted sum of the score $S_{s,q}$ across the N_E rainfall events that caused an overflow in the existing drainage system, that is:

$$R_s = \sum_{q=1}^{N_E} S_{s,q} \times w_q , \tag{2.11}$$

where w_q is the weight associated to the q-th rainfall event. The latter is proportional to the risk of the q-th rainfall event, which is the product of Pr_q (the probability of occurrence of the q-th event) and F_q (the overflow volume in the existing drainage system for the q-th event calculated similarly to $F_{baseline}$ in Eq. 2.1):

$$w_q = \frac{Pr_q \times F_q}{\sum_{l=1}^{N_E} Pr_l \times F_l} . \tag{2.12}$$

With the satisficing metric R_s, we thus seek to understand whether a solution meets the pre-defined level of performance for most rainfall events. The performance threshold defined here sets the expectation that a robust drainage solution should perform as good as what it is designed for when simulated under other rainfall events. The definition of satisficing metric adopted here has two key features. First, satisficing metrics generally give equal weight to all events because such metrics are commonly used for decision-making problems under deep uncertainty (McPhail et al., 2018), where one knows the range of variability of the stochastic disturbances but not their underlying probability distribution function. In this case, however, such distribution is available, so we consider weights that are proportional to the risk. Second, the threshold is usually a fixed value applied to all design alternatives, however it is solution-dependent here. The reason for this choice is that design alternatives have different investment costs, hence it would be disadvantageous to less expensive solution if the same threshold was applied to all solutions. In other words, the solution-dependant threshold implicitly accounts for the investment costs and accounts for a scenario in which one expects a performance comparable to the one attained on the design storm. Note that the multivariate performance threshold can be modified to be univariate by considering only one objective at a time. In the event that only flooded nodes are considered, the weights should also be changed accordingly to be proportional to the number of flooded nodes instead of overflow volume. Also, note that the reduction in the number of flooded nodes is always rounded down to the nearest integer (e.g., if the flooded nodes reduction threshold is 10% and there are 11 flooded nodes in the existing drainage system, a solution needs to reduce only 1 flooded node).

2.4 Case study results

To illustrate the results yielded by our computational framework, we begin by describing the output of the simulation-optimization step, namely the set of Pareto-efficient solutions. We then continue with the robustness analysis, in which we also show how duration, intensity, and profile of rainfall events affect the robustness of drainage systems.

2.4.1 Performance of the Pareto-efficient solutions

The 10 independent optimization experiments returned a total of 669 Pareto-efficient solutions, which are illustrated in Fig. 2.6. As expected, results show that the reduction of overflow volume and number of flooded nodes (for the design storm) improve with the investment costs, with the most expensive alternatives (about \$76 millions) guaranteeing a reduction of overflow volume and number of flooded nodes up to 60% and 26%, respectively. Results also show that the reduction in the number of flooded nodes is smaller than the reduction in the overflow volume—even if these two variables are positively correlated ($r = 0.87$). This is because of three reasons. First, overflow can be reduced but not completely eliminated at some nodes. Second, a few flooded nodes are far from the selected sites for pipe expansions and LID implementations, so flooding issues at these nodes cannot be solved. Third, increasing flow capacity tends to transfer peak flows downstream, and new overflows are created in other segments of the network. In this last case, the total overflow is reduced, but is distributed over more nodes.

An interesting feature of the optimization results is the kink located in the middle of the Pareto front (where the overflow reduction is about 50% and investment costs $13 millions), after which the Pareto front becomes much steeper. This suggests that, for investments lower than $13 millions, significant improvements in overflow reduction (and number of flooded nodes reduction) can be obtained with marginal investment increases; beyond $13 millions, additional investments have diminishing returns in overflow reduction (and number of flooded nodes). This feature can be explained by Fig. 2.7, which illustrates for each solution the relation between the relative investments in pipe expansions and LIDs (for a given solution, the relative investment in pipe expansions or LIDs is simply the ratio between the investment made and the maximum allowable investment). The figure shows that all solutions with limited investments are only characterized by pipe expansions; a pattern observed up to until 75% of the maximum allowable cost for pipe expansion, which corresponds to about $13 millions. Beyond this point, overflow volume can no longer be reduced by further pipe expansions, so the investments are suddenly oriented towards LIDs, which provide an additional 10% overflow volume reduction. In other words, these results suggest that pipe expansion is a more effective strategy for controlling pluvial floods in this case study, since additional capacity can convey a larger amount of stormwater and also act as storage. Naturally, the implementation of LIDs to a larger number of subcatchments would further decrease the overflow volume, although one might expect that pipe expansions would still be selected first, owing to their lower costs and greater effectiveness in transferring peak flows. In addition, it should also be noted that LIDs have more pronounced effects in a temperate climate with less intense rainfall rather than in Ho Chi Minh City, which has a tropical climate (Carpenter and Kaluvakolanu, 2011). Finally, note that LIDs provide additional benefits beyond runoff reduction, such as increased groundwater recharge or mitigation of the urban heat island (Fletcher, Andrieu, and Hamel, 2013), which are not included in our problem formulation.

2.4.2 Robustness analysis

Fig. 2.8a shows the robustness of each Pareto-efficient solution, as defined in Eq. 2.11. Robustness lies between 0.4 to 0.8 for most Pareto-efficient solutions, meaning that they attain a desired level of performance of 40 to 80% (in terms of weights) on the stochastic rainfall events. The variance in robustness decreases with increasing performance (and cost); thus, the expensive Pareto-efficient solutions that attain more than 50% overflow reduction (for the design storm) have robustness that ranges from 0.5 to 0.7. Fig. 2.8b and 2.8c show the cumulative distribution of performance for the two objectives, overflow reduction and flooded nodes reduction, for all Pareto-efficient solutions across the simulated rainfall events. Note that the cumulative distribution here refers to weights, defined in Eq. 2.12, and not probability, such that rainfall events with larger flood or higher frequency have higher weights. In particular, a point at (50%, 0.6) can be interpreted as the solution achieving reduction greater than or equal to 50% for 0.6 (in terms of accumulated weights) of the stochastic rainfall events. The markers in these two figures correspond to the robustness if only one objective is used as the satisficing

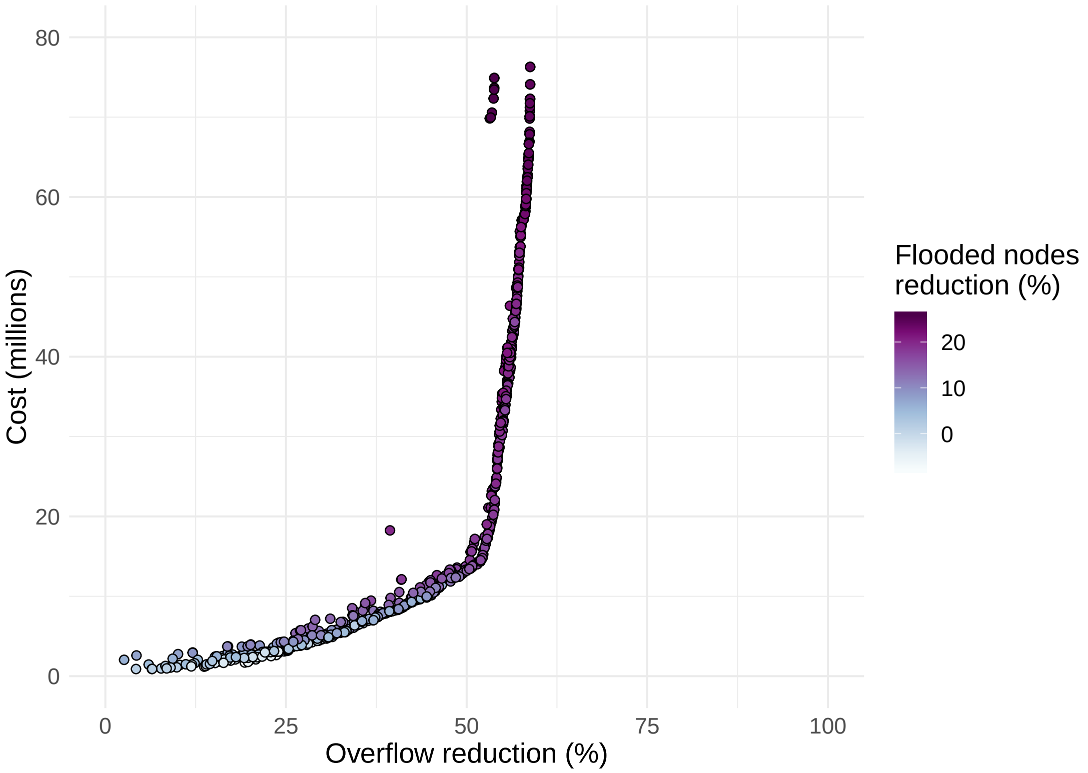


FIGURE 2.6: 669 Pareto-efficient solutions from the simulation-optimization step. The three objectives are represented by the axes (Overflow reduction and Cost) and color (Reduction in the number of flooded nodes). The kink located in the middle of the Pareto front indicates that for investments lower than $13 millions, significant improvements in the reduction of overflow and number of flooded nodes can be obtained with marginal investments.

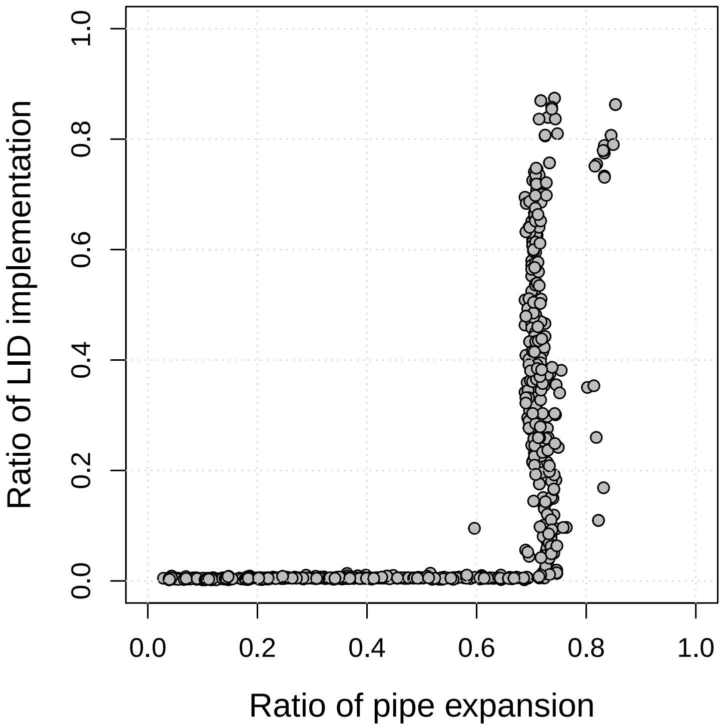


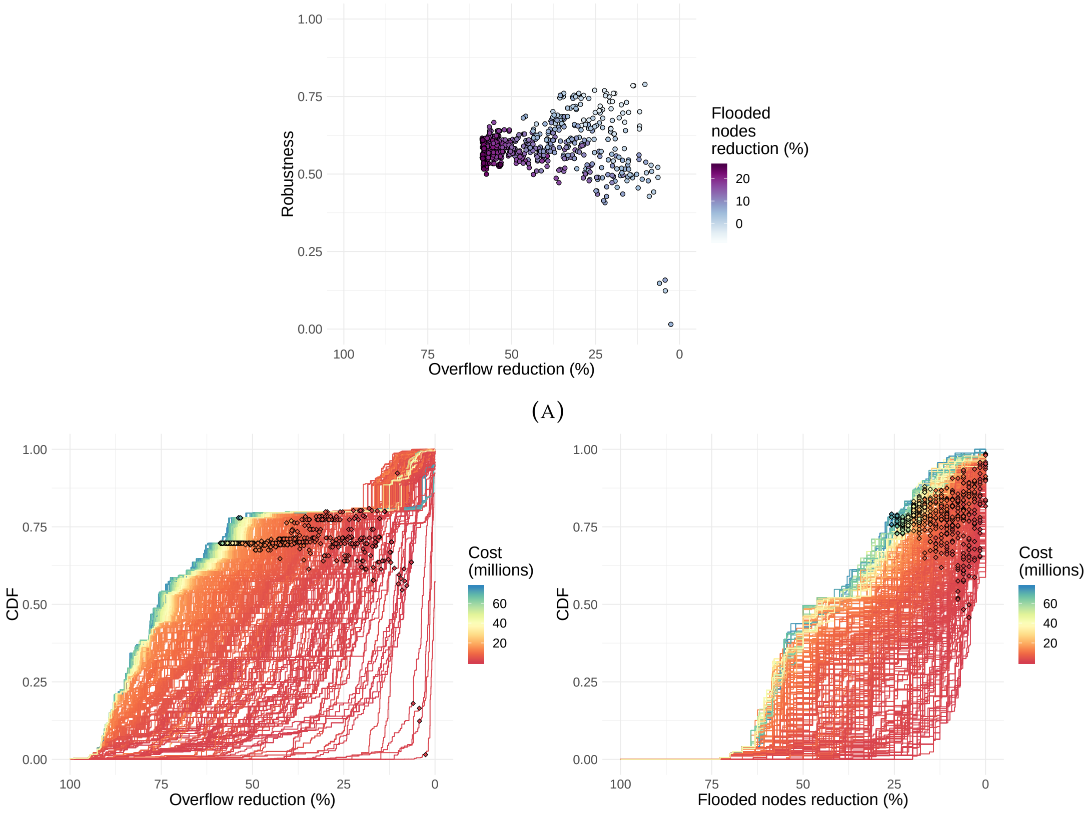
FIGURE 2.7: Relative investment in pipes and LIDs (ratio between the investment made and the maximum allowable investment). Maximum investments on pipes and LIDs are $20 millions and $69 millions, respectively. Each point represents one of the 669 Pareto-efficient solution. Note that, for a limited budget availability, all investment goes towards expanding pipes; investment in LID only begins when it is no longer beneficial to invest in pipe expansion.

threshold in Eq. 2.10. In the ideal case (in which all solutions were robust), we should expect the markers to be at 1.0, which means that the solutions perform equal to or better than what they are designed for during the simulation-optimization for all simulated rainfall events. None of the solutions are robust against all rainfall events when considering a univariate satisficing threshold, although bigger investments generally translate to better performance in both overflow and flooded nodes reduction as can be seen from the higher cumulative distribution of the most expensive solutions. Both Fig. 2.8b and 2.8c also show the same trend of decreasing variance in robustness as the cost of solutions increases. However, this is more apparent in Fig. 2.8b, where robustness stagnates at 0.7 for drainage solutions above $13 millions.

Three main points are highlighted in Fig. 2.8. First, alternative drainage designs of similar cost can have varied performance in different rainfall events, and consequently achieve different robustness. This implies that selecting a solution from the Pareto-efficient set is not straightforward, as one solution can outperform another solution in many other rainfall events even if they perform similarly for the design storm. In addition, even less costly solutions can outperform the more costly ones in some rainfall events. This is indicated by the intersecting lines in Fig. 2.8b and 2.8c. Second, LIDs play a smaller role than pipes in influencing robustness. The costly drainage solutions characterized by different LID deployments show little variance in robustness, whereas less costly solutions characterized by different pipes expansion can attain higher robustness. This is likely because costly drainage solutions all have similar pipe arrangements (that work well for the design storm), and when this arrangement is not adequate for a rainfall event, different LIDs arrangement are unable to improve robustness. Robustness is primarily determined by pipe arrangements, whereas LIDs play a more complementary role, contributing additional overflow or flooded nodes reduction but not enough to increase robustness. Third, all solutions underperform in some rainfall events. Even though alternative pipe designs allow some solution to attain higher robustness, no solution is completely robust. The horizontal line in Fig. 2.8b at $y = 0.8$ illustrates the sudden drop in performance across all solutions, where a maximum of only 20% overflow reduction can be achieved on the stochastic rainfall events regardless of the cost of the drainage solutions.

To better understand the robustness of the Pareto-efficient solutions, we investigate the relationship between the characteristics of the rainfall events (i.e., duration, intensity, and profile) and the performance of the drainage systems (i.e., reduction of overflow volume and number of flooded nodes), on which robustness depends. Fig. 2.9a shows the number of successes (i.e., number of solutions meeting the multivariate performance threshold) for each rainfall event. For rainfall events with moderate intensity (between 6 to 24 mm/hr), most, if not all, solutions meet the threshold. This explains why the robustness for most solutions is above 0.4, since these moderate events represent 40% (in terms of weights) of all rainfall events. For intense rainfall events (i.e., the most intense events with duration ranging between 3 and 12 hours), less than half of the solutions meet the performance thresholds. This is even more apparent for rainfall events that are not of profile 1. Since these solutions are designed for the design storm, which has rainfall profile 1, they underperformed when simulated under another profile. These intense rainfall events are responsible for the gap in robustness between the more costly and less costly solutions. The underperformance in these events also points out that using a single design storm during the optimization process is insufficient. Rainfall events of longer duration, although less intense, may have a greater depth and result in greater flood than the design storm. Therefore, a robust drainage solution may only be found with an optimization problem that considers multiple rainfall events. Finally, the most interesting results illustrated in Fig. 2.9a is the fact that the drainage systems do not perform well for the smallest rainfall events with intensity lower than 6 mm/hr and duration shorter than 2.4 hours, which explains the sudden drop in performance shown in Fig. 2.8b. The cause for the poor performance in these small events is that the overflow occurs near the catchment outlet when high tide coincides with rainfall; yet, the expansion of pipes near the catchment outlet is not selected as a decision variable by the sensitivity analysis (recall that the sensitivity analysis is also based on the design storm). Overflow near the catchment outlet is observed in the design storm; however, overflow at other areas of the NL-TN basin is more severe, so the sensitivity analysis selects pipe variables in other areas of the catchment. The broader implication is that for a large watershed, overflow can occur at different locations for different rainfall events; hence, if the design is made with respect to a design storm, flood may persist in events that are different from the design storm.

Fig. 2.9b and 2.9c show how mean overflow reduction and flooded nodes reduction (across all solutions) vary as a function of rainfall duration, intensity, and profile. For moderate intensity events, the mean overflow reduction and mean flooded nodes reduction largely exceed the value of 43% and 13%, respectively, achieved for the design storm (reaching more than 90% for some events). In general, performance improves as rain intensity decreases and for events more intense than 15 mm/hr, performance improves with decreasing duration. For the intense rainfall events similar to the design storm, the mean overflow reduction is about 35–50% and the mean flooded nodes reduction is 0–15%—values comparable to the performance during the design storm. Once again, performance for rainfall profile 1, especially for the most intense 3-hour event, which is rather similar to the design storm, is better than the performance for other rainfall profile types. In particular, performance for intense events of profile 3 is the poorest. For the small but frequent rainfall events, mean overflow reduction is lower than 10%. Fig. 2.9 highlights that all three rainfall characteristics could affect the performance of solutions. Overall, performance (and consequently robustness) appears to be more sensitive to rainfall depth and intensity, as the division between different levels of overflow or flooded nodes reduction is mostly vertical or diagonal. This also means that broader categories of rainfall events could be considered in the future so that fewer simulations would be needed to find a robust solution.

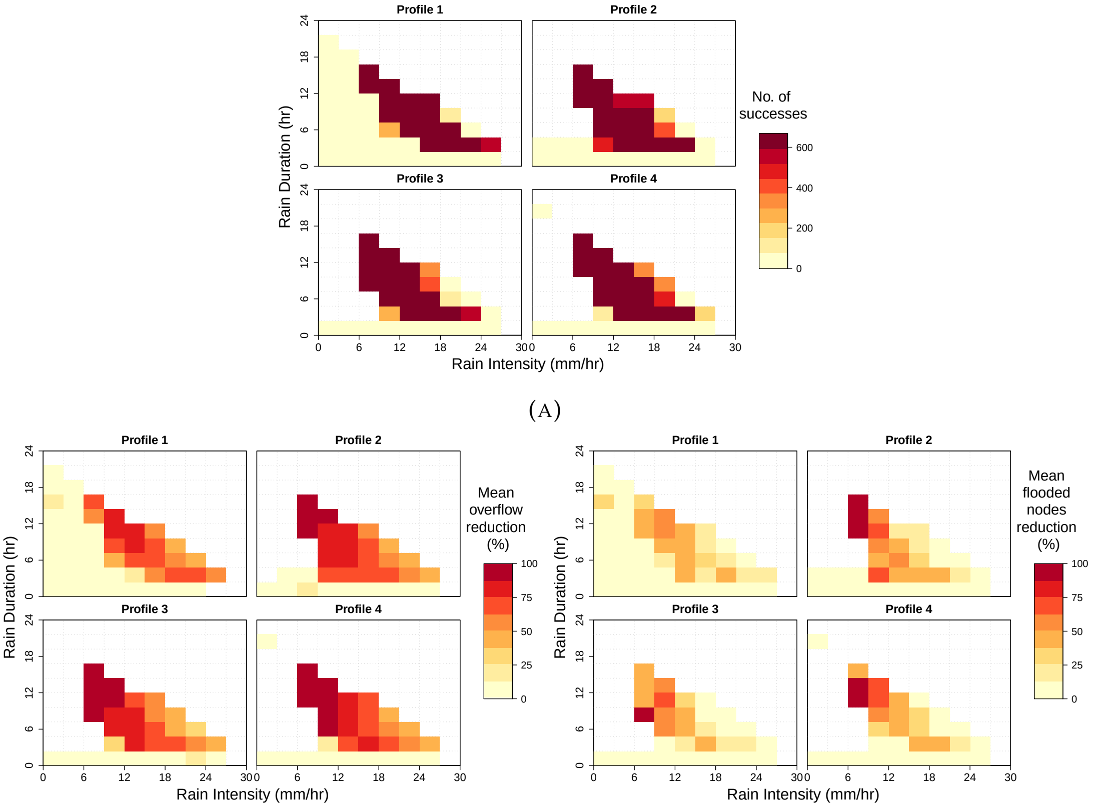


(A)

(B)

(C)

FIGURE 2.8: (a) Robustness of Pareto-efficient solutions when both reduction in overflow volume and number of flooded nodes are used as performance thresholds. The x-axis and color represent the performance (i.e., reduction in overflow volume and number of flooded nodes) of each solution during the design storm. Cumulative distribution for (b) overflow reduction and (c) flooded nodes reduction for each solution across the stochastic rainfall events. Only events with 10 or more flooded nodes in the existing drainage system are considered for (c). Markers correspond to performance during the design storm (on the x-axis) and robustness of the solution (on the y-axis) if only a single objective (overflow reduction or flooded nodes reduction) is used as performance threshold. Note that the x-axis is inverted so that it proceeds in the direction of degrading performance from left to right.



(A)

(B)

(C)

FIGURE 2.9: (a) Number of successes (i.e., solutions that meet the performance thresholds) for each stochastic rainfall event characterized by different duration, intensity, and profile. (b) Mean overflow reduction (%) and (c) mean flooded nodes reduction (%) of all solutions in each rainfall event. Solutions generally performed better than designed in events with moderate intensity, underperformed in events with high rainfall depth (the most intense events ranging from 3 to 12 hours) and performed poorly in events much smaller than the design storm.

2.5 Conclusion

Design storms have dominated the planning of urban drainage systems for multiple decades. However, the solutions achieved by optimizing the design of such infrastructure with respect to a few, representative rainfall events may not be robust to events characterized by different duration, intensity, or profile. In this work, we present a novel computational framework specifically conceived to assess the robustness of drainage systems with respect to multiple stochastic rainfall replicates. Its application to the Nhieu Loc-Thi Nghe basin (Ho Chi Minh City) revealed a few interesting insights:

- None of the drainage solutions are completely robust, since their performance in the design storm cannot be replicated for all rainfall events. In particular, we found that alternative designs with similar performance during the design storm can perform differently for other rainfall events. Hence, the results obtained during the optimization process may provide a myopic and biased picture of the systems' real performance;

- Between pipes and LIDs, investments aimed at expanding pipes (i.e., the conveyance capacity) have a greater effect on reducing flood and potentially increasing robustness of the drainage system. Investments in LIDs can alleviate flood extent during rainfall events but are secondary to pipe expansions in determining robustness. In this respect, it should also be noted that LIDs can possibly function better in less intense storms in temperate climates (in contrast to the tropical climate experienced in Ho Chi Minh City). Also, the deployment of LIDs was considered for only a limited number of sub-catchments—a broader deployment of LIDs in the catchment may bring greater performance;

- Solutions optimized with respect to a design storm are not robust for two types of rainfall events: 1) less intense but longer rainfall events, which have greater depth and cause more severe floods than the design storm, and 2) small, yet frequent, rainfall events. For the first type, costly solutions tend to underperform as the performance during design storm cannot be replicated. For the second type, all solutions underperform significantly, due to the poor choice of the decision variables (recall that such choice is based on the design storm). Small, frequent floods happen in the area close to the catchment outlet; yet, no major modifications to the drainage network in this area are made.

Overall, this first part of our work suggests that the use of design storms can limit the robustness of drainage systems. Naturally, different design storms lead to different solutions, but it is unlikely that any of these solutions can be robust against rainfall conditions that deviate from the adopted design storm. To overcome this problem, stochastic rainfall replicates should be considered throughout the design process, namely when selecting the decision variables, optimizing their values, and evaluating ex-post the solutions' performance. Yet, this approach would largely impact the computational requirements, especially for large basins. This problem could be tackled in two complementary ways. First, one could consider the idea of reducing the number of rainfall events used in the analysis. Our results show that the performance of the drainage systems is particularly sensitive to rainfall intensity, so this could be leveraged to classify rainfall events into broader groups. Second, surrogate models could replace SWMM in the most computationally-expensive parts of the design process (i.e., sensitivity analysis and optimization). Surrogates have been applied successfully in the water resources literature (Razavi, Tolson, and Burn, 2012; Castelletti et al., 2012a), including urban drainage (Mahmoodian et al., 2018a) and flood risks assessment (Löwe et al., 2018; Yazdi and Neyshabouri, 2014).

In the next chapter, we look into surrogate-assisted optimization and explore a new approach to urban drainage design, one that moves away from design storms but considers stochastic rainfall event, which gives greater focus on robustness during the optimization step.

Chapter 3

Designing robust urban drainage systems: a data-driven approach

3.1 Introduction

In the previous chapter, we established that optimizing drainage systems using the design storm alone may give solutions that are not robust against rainfall events with characteristics deviating from the design storm. This prompts the need to consider stochastic rainfall events during optimization, which in turn increases the number of simulations needed and places large demands on computational resources. To counter this issue, we look into an alternative to process-based models (such as SWMM), which is surrogate modelling.

Surrogate models are computationally faster and cheaper alternatives to simulation models and have been used for search (optimization) or sampling purposes – tasks that require repeated use of simulations. Surrogate models have been extensively applied to mechanical and aerospace engineering problems and its use has later extended into the water resources community, with application in groundwater remediation, water distribution networks, and urban drainage systems (see the survey by Razavi, Tolson, and Burn, 2012). While surrogate models for urban drainage simulator have been developed (Machac, Reichert, and Albert, 2016; Carbajal et al., 2017; Moreno-Rodenas et al., 2018), embedding these models in an optimization framework has not yet been explored. Yet, the successful applications of surrogate-based optimization in a wide range of water resources problems indicate that it could be a viable method for the design of robust urban drainage system. In this chapter, we aim to find out if a surrogate-assisted optimization that considers stochastic rainfall events can aid the design of robust drainage systems. In particular, we aim to answer the following questions: (1) Does including stochastic rainfall events during the optimization phase result in solutions that are robust against various rainfall conditions?; (2) To what extent can a surrogate-assisted optimization allow us to obtain robust solutions in a shorter time?; (3) What is the tradeoff between accuracy and complexity of surrogate models used to emulate overflow of an urban drainage simulator?

To do this, we first stochastically generate rainfall events of varying duration, intensity, and profile and then select representative rainfall events that cover a wide range of overflow conditions. Next, we construct data-driven Gaussian process (GP)-based emulators to predict the overflow conditions during the representative rainfall events.

We then use three optimization schemes to search for Pareto-efficient configurations of urban drainage. The optimization schemes are surrogate-assisted optimization, which uses the emulators constructed in the previous step, simulation-optimization using representative rainfall events, and simulation-optimization using design storm only. We finally evaluate robustness of the Pareto-efficient solutions obtained from the three optimization schemes using an unseen set of stochastic rainfall events. Using the Nhieu Loc-Thi Nghe basin (Ho Chi Minh, Vietnam) as a case study, we demonstrate the value of this computational framework in deriving robust urban drainage design.

The chapter is organized as follows: We start with a literature review on surrogate modelling, then we describe the data-driven GP emulation approach adopted in our framework. Next, we introduce the new computational framework that aims to derive robust drainage systems within limited computational budget. Finally, we present the results of applying this framework to the Nhieu Loc-Thi Nghe basin and conclude the chapter.

3.2 Literature review on surrogate modelling

Surrogate modelling is an umbrella term for a wide range of methods that develop and utilize computationally efficient alternatives to higher-fidelity simulation models (such as SWMM). As simulation models are developed to represent rigorously the physical laws of the real-world, they are computationally intensive. The repeated use of simulations in some settings adds on to the computational burden. Surrogates models are faster and cheaper to run, although they may compromise on, the sometimes unneeded, model accuracy. Surrogate modelling has found significance over the last two decades in design, control, parameter calibration, design space exploration, optimization, sensitivity or uncertainty analysis, all of which involve iterative model evaluations over a potentially large parameter space (Razavi, Tolson, and Burn, 2012; Benner, Gugercin, and Willcox, 2015).

Although classification of surrogate modelling may differ, it can be largely divided into two broad families: response surface modelling and lower-fidelity modelling (Razavi, Tolson, and Burn, 2012; Castelletti et al., 2012a). In the first family, response surface surrogates, also known as emulators, metamodels, model emulation, or proxy models (terms which are used interchangeably henceforth), are empirically data-driven models that use interpolation or regression to approximate, for different values of explanatory variables, the response of the original simulation model. The response is usually an aggregated output of the simulation model and a different response surface has to be fitted for each model response (with the exception of neural network). Common techniques employed in response surface surrogates include polynomials, Gaussian processes (GP) (or kriging), radial basis function, polynomial chaos expansion, support vector machines, and neural networks (Forrester, Sobester, and Keane, 2008; Razavi, Tolson, and Burn, 2012). One main advantage of response surface modelling is that it is non-intrusive; it solely relies on the input-output pairs generated from the simulations while the simulation model can remain as a "black-box" function. This can be useful when the underlying equations of the simulation model are not easily accessible and the data-driven approach allows it to be readily adaptable to other problems.

In the second family, lower-fidelity surrogates are simplified physically-based models of the original higher-fidelity models. Unlike response surfaces which predicts model response, lower-fidelity surrogates represent system dynamics at a lower order. There are two main classes of lower-fidelity models, one is the projection-based reduced model (also called model reduction) and the other is hierarchical model (Eldred and Dunlavy, 2006). A projection-based reduced model projects system operators onto a reduced subspace, retaining the system state representation and underlying structure of the original model (Antoulas, 2005; Benner, Gugercin, and Willcox, 2015). Hierarchical model are like the original models but with reduced numerical accuracy, relaxed physical assumptions, or coarser temporal or spatial grid size. Lower-fidelity models can be hard to construct owing to its intrusive nature and the need for domain-specific knowledge. However, once the model is constructed, it is able to simulate different condition or give multiple model responses since it has retained the physically-based characteristics of the original model. Finally, the two families of surrogates are not mutually exclusive because both data-driven response surface surrogate and model-driven lower-fidelity surrogate can be employed in hybrid approaches (Bermúdez et al., 2018; Carbajal et al., 2017; Castelletti et al., 2012b; Machac, Reichert, and Albert, 2016; Mahmoodian et al., 2018b).

The use of surrogate models has extended into the water resources community, with application in groundwater remediation, water distribution networks, and urban drainage systems (Razavi, Tolson, and Burn, 2012). Surrogates for urban drainage simulators are varied in terms of the response variable emulated and also in terms of the technique, which can range from purely data-driven to purely model-driven. Xu et al., 2010 and Xu, Overloop, and Giesen, 2013 employ a model-driven approach, applying model reduction technique to the St Venant equations to predict water quantity and quality in real time control of open channel systems. On the other hand, Nagel, Rieckermann, and Sudret, 2020 use a data-driven approach using sparse polynomial chaos expansion (PCE) to predict discharge from a catchment. More often, we see hybrid surrogates being used. Bermúdez et al., 2018 develops a hybrid surrogate model that combines artificial neural networks (ANN) with a conceptual sewer and flood model to predict surface flood volume based on rainfall inputs. Moreno-Rodenas et al., 2018 combines hydrology knowledge with PCE to emulate flow under various rainfall and parametric scenarios. Carbajal et al., 2017 compares a naïve data-driven GP emulator with a mechanistic emulator to approximate water level and inflow in SWMM. They find that including mechanistic knowledge of the simulation model is only advantageous when training data are sparse and unevenly sampled. Machac, Reichert, and Albert, 2016 and Machac et al., 2016 come to a similar conclusion that including mechanistic description in an emulator for urban drainage modelling is a reasonable tradeoff between computational complexity and the size of design data set.

More often than not, surrogates are built so that they can be utilized within a metamodel-enabled analysis framework which can be used for search (optimization) or sampling purposes. Razavi, Tolson, and Burn, 2012 classified metamodel-enabled analysis framework into 4 types. The first type is the basic sequential framework whereby users fit a metamodel offline through a formal design of experiment (DoE), fully substitutes the original function with the metamodel, and perform analysis (search or sampling) on the metamodel directly (Broad, Dandy, and Maier, 2005; Broad, Maier, and Dandy, 2010; Galelli, Castelletti, and Goedbloed, 2015). Next is the adaptive-recursive framework whereby users first fit a metamodel through a formal DoE, use the metamodel to identify region of interest in the input space, and evaluate new points(s) using the original function. The metamodel is updated with the new input-output pair(s) and the updating steps are repeated until convergence or stopping criteria are reached (Tabatabaei et al., 2015; Fen, Chan, and Cheng, 2009; Zou, Lung, and Wu, 2009; Castelletti et al., 2010). The third type is the metamodel-embedded evolution framework which is limited to analyses dependent on evolutionary optimization algorithms (Jin, 2011; Yazdi and Neyshabouri, 2014; Akhtar and Shoemaker, 2016; Zhang et al., 2017). No formal DoE is needed initially, the population-based evolutionary algorithm proceeds for a few generations using the original function. A metamodel is fitted based on the first generations and updated in the course of optimization since individuals are selectively evaluated with either the metamodel or original function. The final type is the approximation uncertainty based framework, which considers the uncertainty associated with the approximation instead of assuming approximated values as true, thus it balances exploration with exploitation during the search (see algorithms like EGO (Jones, Schonlau, and Welch, 1998), and MSRBF (Regis and Shoemaker, 2007), and applications in Di Pierro et al., 2009; Regis and Shoemaker, 2009).

When using surrogates for optimization, the choice of framework and related design considerations depend on our understanding (if any) of the complexity or smoothness of the original function and also the computational budget, which represents a tradeoff between speed of convergence and global accuracy of the model. For multiobjective optimization, as is often the case for water resources problems, all four above-mentioned frameworks are applicable. Research to extend single objective surrogate optimization to a multiobjective setting is well established (see ParEGO which uses EGO iteratively to optimize aggregated objective functions (Knowles, 2006) or SMS-EGO which uses multiple surrogates for multiple objectives (Ponweiser et al., 2008)). It is also common to simply use a multiobjective evolutionary algorithm and embed a metamodel within. Akhtar and Shoemaker, 2016 use a parallel response surface-assisted multiobjective evolutionary algorithm to solve a groundwater remediation problem. They evaluate pumping decisions to minimize cost and contamination and apply radial basis functions to approximate each objective. Regis and Shoemaker, 2009 solve a ground remediation problem using a parallel stochastic RBF algorithm for global optimization. The metamodel dynamically evolves during the optimization using new design points selected based on their approximated function value and distance from previously evaluated points (a measure of approximation uncertainty). Zhang et al., 2017 derives large-scale reservoir operating rules using weighted multiobjective adaptive surrogate model optimization. Galelli, Castelletti, and Goedbloed, 2015 determine the optimal control of a multi-purpose reservoir by first performing model reduction on a 3-D hydrodynamic model and then integrating the low-order, dynamic emulator into the real-time reservoir operation model. Broad, Dandy, and Maier, 2005 and Broad, Maier, and Dandy, 2010 apply ANN to a water distribution system and perform genetic algorithm directly on the trained metamodels. Di Pierro et al., 2009 minimizes cost and head deficit of a water distribution network by using ParEGO and LEMMO, a hybrid algorithm which uses machine learning techniques together with a multiobjective evolutionary algorithm. Yazdi and Neyshabouri, 2014 couple a periodically updated ANN model with NSGA-II to determine detention dam heights for flood control under rainfall uncertainties.

Since surrogates have been effectively employed in optimization to solve water resources problem but have not been explored for urban drainage planning, we thus employ an emulation approach for urban drainage model and use the emulator within our optimization framework to investigate its potential in designing robust urban drainage systems.

3.3 Data-driven GP emulator

We first describe the emulation approach used to approximate the SWMM model. We will use the terms 'inputs', 'parameters', or 'decision variables' to refer to the inputs of the simulator and the terms 'outputs' or 'response' to refer to the outputs of the simulator. Also note that the inputs to the simulator are also the decision variables that will be used for optimization in the later step. In our case, the inputs are the sewer pipe sizes and LID areas and the outputs are the the overflow volume and number of flooded nodes during each rainfall event of interest. To approximate this, we adopt the data-driven GP-based method developed by Carbajal et al., 2017. We select this emulator due to the followings reasons: (1) the emulator output is a time series instead of an aggregated output; (2) the emulator is purely data-driven and hence could be readily applied to any catchment model. As we are interested in the overflow conditions, which can be highly nonlinear due to surcharge flows in pipes (Rossman, 2015), it becomes difficult to include mechanistic knowledge in the emulator. Furthermore, Carbajal et al., 2017 notes that it is advantageous to use data-driven emulators when we lack good prior knowledge about the simulator and when the parameter space can be evenly well sampled. Since our primary aim is to design a robust drainage system, we opt for a data-driven emulator that can be easily applied and incorporated into our optimization schemes.

The data-driven GP emulator should be able to predict the time series of the response of interest, $\vec{y}(\vec{\gamma})$, given the parameters or inputs of the model, $\vec{\gamma}$. The emulator does so by first extracting the most significant dynamic features of the system response (also known as the time varying basis) using matrix factorization. In this study, we use singular vector decomposition (SVD) but other techniques like nonnegative matrix factorization (NMF) can be used too. It then uses GP (Rasmussen and Williams, 2006) as a mapping between the parameters and the coefficients of the linear combinations of the dynamic features. Given any new parameters, the unseen response is then approximated by the linear combination of the features. Since a time series of model response is predicted by the emulator, there is added flexibility in deriving different aggregated output from the model such as the sum, mean, or maximum value of the time series.

The emulator is constructed with the following steps:

(i) Perform a design of experiment to select n points in the parameter space (denoted by parameters $\vec{\gamma}$) and run the simulation model for each point. Each simulation output is a time series of the variables of interest of length N. Allocate n_{trn} input-output pairs as the training set and n_{tst} input-output pairs as the test set, where $n_{trn} + n_{tst} = n$.

(ii) Given the output matrix $\mathbf{Y} \in \mathbb{R}^{N \times n_{trn}}$, perform SVD. This is the step where we extract the dynamic features of the model response. We obtain the orthogonal matrices $\Phi \in \mathbb{R}^{N \times n_{trn}}$, $\mathbf{W} \in \mathbb{R}^{n_{trn} \times n_{trn}}$, and nonnegative diagonal matrix $\Sigma \in \mathbb{R}^{n_{trn} \times n_{trn}}$ that fulfills

$$\mathbf{Y} = \Phi \Sigma \mathbf{W}^T. \tag{3.1}$$

$\mathbf{Y}$ can be decomposed into

$$\mathbf{Y} = \Phi \mathbf{B}\,, \tag{3.2}$$

where

$$\mathbf{B} = \Sigma \mathbf{W}^T \tag{3.3}$$

and Φ are the eigenvectors (or principal components) of the covariance matrix $\mathbf{Y}\mathbf{Y}^T$. The columns of Φ are the dynamic features, or time varying basis, of the model response. The columns of $\mathbf{B}$ are the coefficients of the linear combinations of the features for each training point.

(iii) Extract only the first $q \leq n_{trn}$ features Φ' from Φ (i.e., the first q columns). Since approximation degrades rapidly with decreasing number of principal components, this is the time varying basis that contains the most significant features of the system response. We also obtain matrix $\mathbf{B}' \in \mathbb{R}^{q \times n_{trn}}$ (i.e., first q rows of $\mathbf{B}$) of coefficients where there are q coefficients for each training point.

(iv) Interpolate using GP to obtain a function of the inputs $\mathbf{B}' = f(\vec{\gamma})$.

(v) Finally, given any new points in the test set, we can now evaluate $f(\vec{\gamma})$, and predict the output time series using

$$\vec{y}(\vec{\gamma}) = \Phi' f(\vec{\gamma})\,. \tag{3.4}$$

3.4 Computational framework

The computational framework developed for this study consists of 4 steps, illustrated in Fig. 3.1. We begin with stochastic rainfall generation to generate and select rainfall events that can be representative of overflow conditions caused by events of varying rainfall intensities, duration, and profiles. Next, we construct GP emulators to approximate the overflow conditions under these selected rainfall events given different pipe sizes or LID implementations in the catchment. Multiple emulators are constructed using different size of design data set (n) and different number of features of system response selected (q), which allows us to study how these parameters affect the accuracy of the emulator. We then use the emulators in our optimization problem where we seek a set of Pareto-efficient solutions that minimize the investment costs while maximizing the reduction in both overflow volume and flooded nodes during the representative rainfall events. We compare this surrogate-assisted optimization scheme with the computationally expensive robust simulation-optimization scheme and also the design storm simulation-optimization scheme. Finally, we evaluate the robustness of the solutions derived from the three optimization schemes by simulating them under random samples of rainfall events.

Note that this framework is similar to the framework introduced in Section 2.3 with the following key differences. First, rainfall generation step is brought earlier ahead of the optimization step so that we can account for various rainfall conditions during optimization. Second, emulators are used to replace simulation models during optimization to manage the additional computational demands brought about by the inclusion of more rainfall events. The framework introduced can be readily applied to any catchment model. In the following sections, we demonstrate its application to the Nhieu Loc-Thi Nghe Canal and Basin, a 33 km^2 catchment in the central area of Ho Chi Minh City, whose data are described in Section 2.2.

3.4.1 Stochastic rainfall generation

Using the 10-year hourly rainfall time series described in Section 2.2.2, we model the marginal distribution of rainfall duration and intensity using lognormal and gamma distribution respectively, and model the dependence structure between the two variables using a Frank copula (refer to Section 2.3.3). We now sample from these distributions rainfall events with various duration and intensities for a 25-year period. While we note that the recommended design return period is largely region specific, we choose a 25-year period because urban drainage for a large catchment (like the Nhieu Loc-Thi Nghe basin) is commonly designed for a return period of 25 years or more (*Code of Practice on Surface Water Drainage* 2011; *Stormwater Drainage Manual* 2018). Although the design storm used in Chapter 2 is a 2-year design storm, we want to consider an extensive range of rainfall events, including events smaller and larger than the design storm, to test our framework.

By sampling from the distributions, we select 259 rainfall events after accounting for rounding differences, such that each event has intensity and duration that differs by at least 1mm/hr and 1 hour, respectively. For instance, events with intensity ranging from 25 mm/hr to 26 mm/hr and duration ranging from 1 hour to 2 hour are considered as one event represented by intensity of 25.5mm/hr and duration of 1.5 hour. Considering that there are four Huff profile types (Section 2.3.3) to describe the distribution of rain over time, this gives us a total of 1,036 (259 × 4) rainfall events with varying intensity, duration, and profile.

It would be prohibitively expensive computationally to seek solutions robust against all thousand rainfall events, especially when sufficiently similar rainfall events would

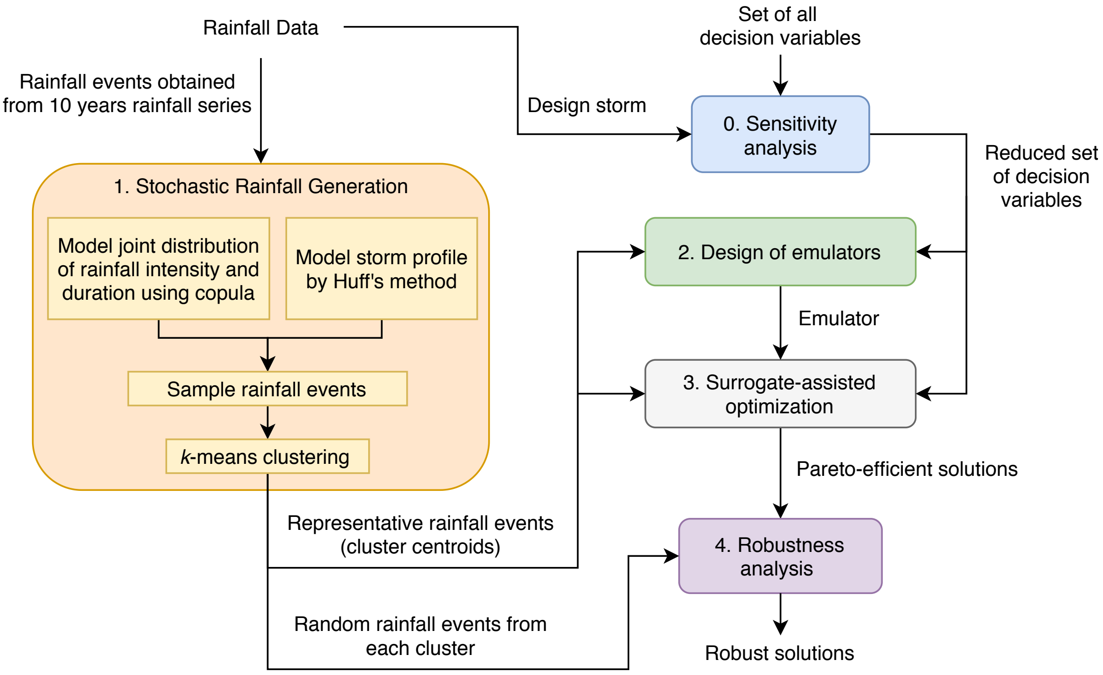
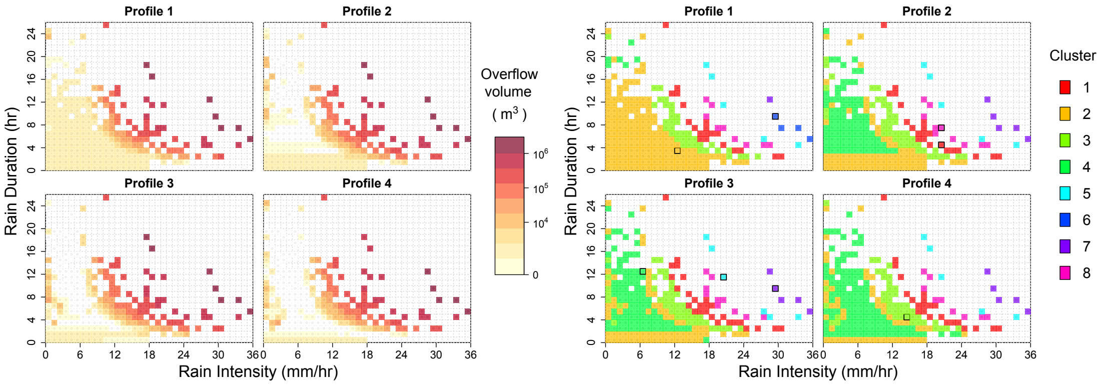

Rainfall Data
Set of all decision variables
Rainfall events obtained from 10 years rainfall series
Design storm
0. Sensitivity analysis
Reduced set of decision variables
1. Stochastic Rainfall Generation
Model joint distribution of rainfall intensity and duration using copula
Model storm profile by Huff's method
2. Design of emulators
Emulator
Sample rainfall events
3. Surrogate-assisted optimization
k-means clustering
Pareto-efficient solutions
Representative rainfall events (cluster centroids)
4. Robustness analysis
Random rainfall events from each cluster
Robust solutions

FIGURE 3.1: Flowchart of the computational framework, consisting of 4 main steps: 1) stochastic rainfall generation to select rainfall events that are representative of a wide range of rainfall intensities, duration, and profiles; 2) design of emulators to construct GP emulator that approximate SWMM; 3) surrogate-assisted optimization to obtain Pareto-efficient solutions that performs well across the representative rainfall events; 4) robustness analysis to evaluate the performance of the solutions for random rainfall events. Sensitivity analysis to select decision variables for the emulator and optimization problem depends on the design storm and is a step performed in the previous chapter.



(A)

(B)

FIGURE 3.2: (a) Overflow volume (in logarithmic scale) of stochastic rainfall events generated for a 25-year period. (b) 8 rainfall clusters obtained using k-means clustering in which overflow conditions during rainfall events in the same cluster are similar. The highlighted events for each cluster is the event closest to the centroid of the cluster.

produce similar overflow conditions (as shown in Section 2.4.2). Thus, we first simulate the existing drainage system under these 1,036 events. The overflow volume for each event is illustrated in Fig. 3.2a. Next, we cluster similar rainfall events using k-means clustering (Berkhin, 2006). Each rainfall event is a data point represented by a vector consisting of the logarithm of total overflow volume, the total number of flooded nodes, and the overflow volume over each of the 137 nodes. 90 nodes in the drainage system are not included here since they do not have overflow under all rainfall events. The vectors of overflow conditions are normalized across all rainfall events. Fig. 3.2b shows the 8 rainfall clusters selected through k-means clustering, where each cluster consists of rainfall events that produce similar overflow conditions under the existing drainage system. The highlighted rainfall event represents the rainfall event that is closest to the centroid of the cluster. These 8 representative rainfall events are used in the later steps of the computational framework under the hypothesis that a drainage solution can reduce overflow for the clusters of rainfall events if it can do so for these rainfall events.

3.4.2 Design of emulators

We now construct GP emulators to approximate the SWMM simulation model using the method described in Section 3.3. Although 20 decision variables are selected in Section 2.3.1, we only select the top 5 decision variables here (listed in Table 3.1 and 3.2) because of two reasons. First, simulation-optimization using 20 decision variables and multiple rainfall events is too computationally demanding. Second, high dimensionality of the problem can cause surrogate model to be less accurate or even infeasible. Since Carbajal et al., 2017 built emulators for models with 1 to 8 parameters, we decide to select a value within this range. Using 5 decision variables is large enough to observe significant overflow reduction for the catchment and also small enough for us to test the emulator. The main aim in this chapter is to find out if surrogate-assisted optimization using stochastic rainfall events can assist us in identifying robust solutions, thus we do not wish to introduce high dimensionality at this stage. Scaling this approach to a higher dimension will be addressed in the next chapter.

The emulators are constructed using different size of design data set (n = 50, 100, 150, 200, 250, 300, 400, 500) and different number of features of the system response (q = 1, 2, 3, 4, 5, 7, 10, 15, 20, 25) to investigate the effect of these parameters on the accuracy of the emulators. The inputs to the simulation model are the pipe sizes and LID areas. We select the design points in the parameter space using Latin hypercube sampling (LHS) starting with n = 50, augmenting the existing LHS by adding new design points while maintaining the LHS properties until n = 500. The outputs of the simulation model are the overflow time series and number of flooded nodes time series. Each time series has 996 data points, recorded at 5 minutes interval. The time series is formed by concatenating the overflow (or flood nodes) time series from each of the 8 representative rainfall events obtained from Section 3.4.1. Time series for individual rainfall events begin when the rainfall event starts and ends 2.5 hours after rain has ceased, ensuring all overflow are captured within the timeframe. The input and output pairs of the simulator are used to construct the simulator, with a different emulator constructed for a different output (i.e., one for overflow volume and one for flooded nodes). Since the emulator predicts a time series of overflow or number of flooded nodes, single or aggregated outputs that can be drawn from the time series include total overflow volume, peak overflow, peak number of flooded nodes, and overflow duration. We perform a k-fold cross validation (k = 10), partitioning the data randomly into 10 equal size subsamples and always leaving 10% of the dataset out as test set.

We assess the accuracy of the emulators using the following measures:

(i) Kling-Gupta efficiency (KGE)

The KGE (Gupta et al., 2009) compares the correlation, variability, and bias between the predicted time series (from emulator) and observed time series (from simulator). It is defined as follows

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{3.5}$$

where r is the correlation coefficient, α the ratio between standard deviation of predicted and observed values (σ_p/σ_o) and β the ratio between the mean of predicted and observed values (μ_p/μ_o). A value of 1 for KGE means that the predicted time series matches the observed one perfectly.

(ii) Mean bias error (MBE) or Mean absolute error (MAE)

$$MBE = \frac{1}{N}\sum_{j=1}^{N}(\hat{y}(\gamma_j) - y(\gamma_j)) \tag{3.6}$$

TABLE 3.1: Description of the pipe variables used as input parameters for emulator

Decision variable, x_j	Pipe no.	Shape	Length, l_j (m)	Size, d_j (m)	Maximum size (m)
x_1	1	Rectangular closed	546	2.5 x 2.5	4
x_2	4	Circular	180	1	2
x_3	9	Rectangular closed	551	1.2 x 5	4
x_4	12	Rectangular closed	900	1.8 x 4	4

TABLE 3.2: Description of the LID variable used as input parameters for emulator

Decision variable, x_j	LID	Sub-catchment properties			Maximum no. of units
		Sub-catchment no.	Area (ha)	Impervious %	
x_5	Urban green spaces	3	102	70	4000

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|\hat{y}(\gamma_j) - y(\gamma_j)| \tag{3.7}$$

Where $\hat{y}$ and y are the emulated and simulated output respectively and N the number of data points in the test set.

(iii) Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{y}(\gamma_j) - y(\gamma_j))^2} \tag{3.8}$$

MBE captures the bias of prediction which is important when the emulators are used in optimization, since a positive bias means that the solutions obtained are not performing as well as expected. MAE captures the magnitude of error whereas RMSE penalizes large deviation more.

3.4.3 Surrogate-assisted optimization

With the emulators constructed, we then proceed to identify optimal drainage solutions through surrogate-assisted optimization. In our application, we adopt a basic sequential framework similar to that of Broad, Maier, and Dandy, 2010, by directly replacing the simulation model with the emulator and coupling it with a multi-objective evolutionary algorithm – that is NSGA-II. The decision variables are the pipe diameters for the 4 pipes and number of LID units for the urban green space as listed in Table 3.1 and 3.2 and with details described in Section 2.3.2.

We consider three objectives, which are similar to Eq. 2.1-2.3 in Section 2.3.2. They are slightly modified now because we are seeking robust solutions that are optimal for 8 representative rainfall events rather than just the design storm. We wish to maximize the weighted reduction in total overflow volume $J_w^{Overflow}$, and the weighted reduction in peak number of flooded nodes J_w^{Node}, and to minimize the capital cost J^{Cost}. $J_w^{Overflow}$ accounts for the flood extent and is defined as follows:

$$J_w^{Overflow} = \sum_{q=1}^{Q} \left(w_q \times \left(1 - \frac{\sum_{t=1}^{T_q} g_{1,t,q}(\mathbf{x})}{F_{baseline,q}} \right) \right) , \tag{3.9}$$

where $\mathbf{x}$ is the vector of decision variables, Q the total number of rainfall events used, which is 8 here; w_q the weight of rainfall event q; T_q the total number of time instances (5 min intervals) in rainfall event q, N the total number of nodes in the drainage network, $g_{1,t,q}(\cdot)$ the overflow volume in the catchment at time t in rainfall event q predicted by the emulator, and $F_{baseline,q}$ is the total overflow volume during rainfall event q for the existing drainage system evaluated using SWMM.

The weight of rainfall event q,

$$w_q = \frac{\sum_{l \in C_q} Pr_l \times F_{baseline,l}}{\sum_{l=1}^{N_E} Pr_l \times F_{baseline,l}} , \tag{3.10}$$

is associated to the rainfall cluster that rainfall event q represents (C_q is the set of rainfall events in the cluster) and is proportional to the total risk risk (probability of occurrence Pr_l multiplied by overflow volume $F_baseline, l$) of rainfall events within the cluster, which is also the definition used in Eq. 2.12 in Section 2.3.4.

The second objective J_w^{Node} accounts for the spatial extent of the flood and is defined as follows:

$$J^{Node_w} = \sum_{q=1}^{Q} \left(w_q \times \left(1 - \frac{\max_t g_{2,t,q}(\mathbf{x})}{N'_{baseline,q}} \right) \right) , \tag{3.11}$$

where $g_{2,t,q}(\cdot)$ is the number of flooded nodes in the catchment at time t in rainfall event q predicted by the emulator; and $N'_{baseline,q}$ the peak number of flooded nodes during rainfall event q for the existing drainage system evaluated using SWMM.

Note that here we are maximizing weighted reduction in peak flooded nodes instead of reduction in total flooded nodes as in Section 2.3.2. This is because our emulation approach produces a time series of flooded nodes and it is possible to obtain peak flooded nodes instead of total flooded nodes from the time series.

Finally J^{Cost} is the capital cost involved in pipe expansion and LID implementation. It can be computed without an emulator and is defined the same way as in Eq. 2.3. The goal of the optimization problem is to find the set of nondominated solutions $\mathbf{x}^*$ that minimizes the vector $\mathbf{J}(\mathbf{x})$, where

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} -J_w^{Overflow}(\mathbf{x}) \\ -J_w^{Node}(\mathbf{x}) \\ J^{Cost}(\mathbf{x}) \end{bmatrix} . \tag{3.12}$$

Since the solutions are evaluated by emulators during the optimization, we finally evaluate the solutions obtained at the final generation and some intermediate generations using the SWMM simulation model. For NSGA-II, the distribution indices for crossover and mutation are 0.9 and 0.1 respectively. The optimization problem is initiated with 10 independent random seed and each of the 10 runs has a population size of 200 and is terminated after 100 generations (a drop from 250 generations used in Section 2.3.2 because we are handling fewer decision variables here).

We first perform surrogate-assisted optimization using the 40 different emulators constructed (8 values for size of design data set (n = 50, 100, 150, 200, 250, 300, 400, 500) and 5 values for number of features of the system response (q =3, 4, 5, 7, 10)). In other words, we solve for 40 optimization problems using a different emulator in each problem. The purpose here is to investigate how the choice of emulators affects error propagation during the search and the quality of solutions obtained from the optimization. Due to the large computation demand, we only initiate the optimization search with 5 independent random seed (instead of 10) for this set of experiments. We finally select one emulator and run the surrogate-assisted optimization for 10 independent random seed.

We then compare the surrogate-assisted optimization (R-EO for 'robust emulation-optimization') scheme with two other optimization schemes: robust simulation-optimization scheme (R-SO) and the design storm simulation-optimization scheme (DS-SO). R-SO has the same set-up as the surrogate-assisted optimization with the exception that it uses the simulator instead of the emulator, hence the increased accuracy comes at the expense of greater computational demands. It replaces $g_{1,t,q}(\mathbf{x})$ in Eq. 3.9 with $\sum_{i=1}^{N} f_{i,t,q}(\mathbf{x})$, the total overflow volume in the catchment at time t in rainfall event q, where $f_{i,t,q}(\cdot)$ is the overflow volume in the i-th node at time t for rainfall event q calculated by SWMM and N is the total number of nodes in the drainage network. It replaces $g_{2,t,q}(\mathbf{x})$ in Eq. 3.11 with $\sum_{i=1}^{N} \mathbb{1}_{\{f_{i,t,q}(\mathbf{x})>0\}}$, the number of flooded nodes at time t in rainfall event q, where the indicator function takes value of 1 if the overflow volume $f_{i,t,q}(\cdot)$ in the i-th node is positive at time t for rainfall event q (and 0 otherwise).

In contrast to R-SO, which maximizes the weighted reduction in overflow and peak flooded nodes and aims to obtain optimal solutions robust against various rainfall conditions, DS-SO uses only design storm in the optimization step, equivalent to the one introduced in Section 2.3.2. The solutions obtained from DS-SO are evaluated using the 8 rainfall events subsequently for comparison purposes. The settings for NSGA-II are the same for all three optimization schemes. All experiments were run on either a Dual Intel Xeon CPU E5-2630 v3 @ 2.40 GHz with 32 GB RAM running Microsoft Windows 8.1 (16 cores) or a Intel Xeon @ 2.4Ghz with 96 GB RAM running Microsoft Windows 10 (24 cores).

To evaluate the search quality of R-EO, R-SO, and DS-SO, three performance measures are used. They are the hypervolume indicator, generational distance, and additive ϵ-indicator (Reed et al., 2013). These metrics capture the convergence of the solution set (found by the optimization) with respect to a reference set, which is usually

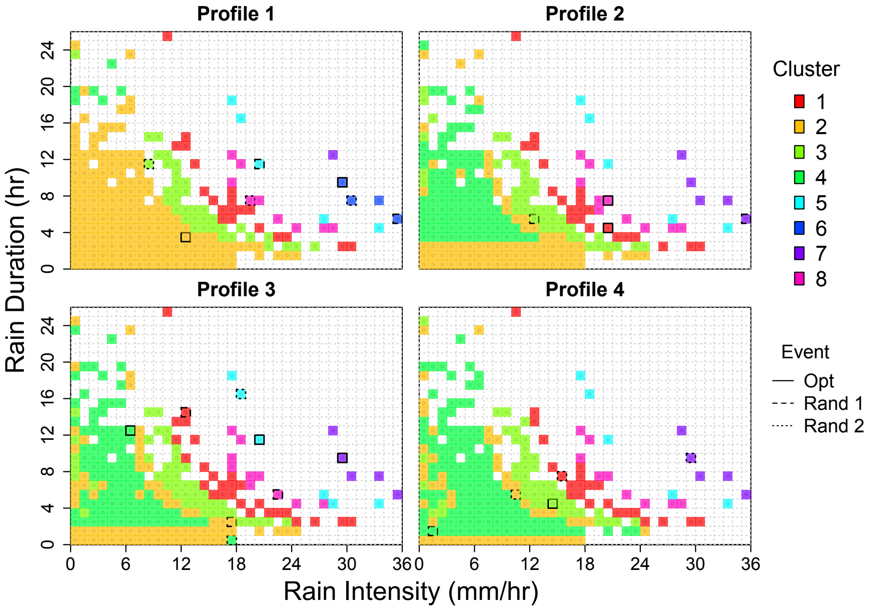


FIGURE 3.3: 16 rainfall events selected from the 8 rainfall clusters for robustness analysis. The first and second set of 8 random rainfall events are bordered by dashed and dotted lines respectively. Rainfall events used within optimization are bordered by solid lines.

the true Pareto front. When the true Pareto front is unknown, the best known solution set found across the three optimization schemes is used as the reference set. Hypervolume measures the volume of objective space dominated by a solution set. Generational distance measures the average Euclidean distance of each point in the solution set to its closest point in the reference set. Additive ϵ-indicator measures the worst case distance (in contrast to the average distance) between the solution and reference set, and is especially sensitive to gaps in the solution set.

3.4.4 Robustness analysis

To assess the robustness of solutions obtained from optimization, we randomly sample two other rainfall events from each of the 8 rainfall clusters. The rainfall properties for the events (including the events used in optimization) are given in Fig. 3.3 and the overflow conditions for these events are given in Table 3.3. We run the Pareto-efficient solutions obtained from the three optimization schemes (R-EO, R-SO, and DS-SO) under these two set of 8 rainfall events each using the simulator and calculate the overflow and peak flooded nodes reduction for each events as well as the weighted reduction similar to Eq. 3.9 and 3.11.

3.5 Case study results

We begin by explaining how the accuracy of emulators is influenced by the number of design data points and number of basis functions, and also explain how this accuracy

TABLE 3.3: Overflow volume and peak number of flooded nodes during each rainfall event for the existing drainage system, together with the weights of each rainfall cluster

Cluster	Total overflow			Peak number of flooded nodes			Weights
q	Opt	Rand 1	Rand 2	Opt	Rand 1	Rand 2	w_q
1	194,959	271,430	178,463	40	20	21	0.229
2	1,357	3,448	3,628	2	6	3	0.309
3	25,758	13,336	8,279	11	5	2	0.294
4	334	330	602	1	1	1	0.024
5	1,062,597	964,427	1,216,971	55	60	47	0.025
6	1,980983	1,459,750	1,579,932	97	107	101	0.006
7	2,085,457	1,542,374	1,888,831	92	99	91	0.009
8	560,134	483,482	410,660	47	59	47	0.104

changes during the search in surrogate-assisted optimization. We then compare the Pareto-efficient solutions obtained from surrogate-assisted optimization (R-EO) with the Pareto-efficient solutions obtained from robust simulation-optimization (R-SO) and design storm simulation-optimization (DS-SO). We finally look at how these solutions perform under different rainfall events, thereby showing that our method is a viable method to produce robust drainage systems.

3.5.1 Performance assessment of emulators

For each point in the test set, we obtain the overflow reduction time series (measured in absolute volume reduction) for each rainfall event predicted by the emulator. We compare this to the time series given by the simulator and calculate the KGE for each test point and for each rainfall event. This is done for each setup of the emulator (i.e., for each value of n and q). The first quartile, median, and third quartile KGE for each setup of the emulator are illustrated in Fig. 3.4a. In general, a greater n or a greater q leads to a better KGE although improvements in KGE are diminishing for higher values of n and q. Beyond n = 150 and q = 7, the improvement in KGE is small. Improvements in KGE are the most obvious for the first quartile values and not so for the median or third quartile values, which means that some errors cannot be reduced simply by increasing n or q.

The violin plot in Fig. 3.4b shows the distribution of KGE for the 8 rainfall events, for different values of n, and for a fixed q = 7. For rainfall events 2, 3, and 4, KGE is lower and shows greater variance. These are smaller rainfall events that result in small overflow volume and few flooded nodes, as shown in Table 3.3. For the other rainfall events, we observe decreasing variance with increasing n from n = 50 to n = 250, but extreme values for KGE still exist for n greater than 250. The extreme values could also be a result of the larger test set for larger n since the test set is always 10% of the dataset. The difference in performance for the different rainfall events suggest that the emulator may not be suitable when values vary widely across rainfall events. Using the first few significant basis functions (up to $q = 7$) can capture system response for the larger values in the time series but may not do so adequately for the smaller values.

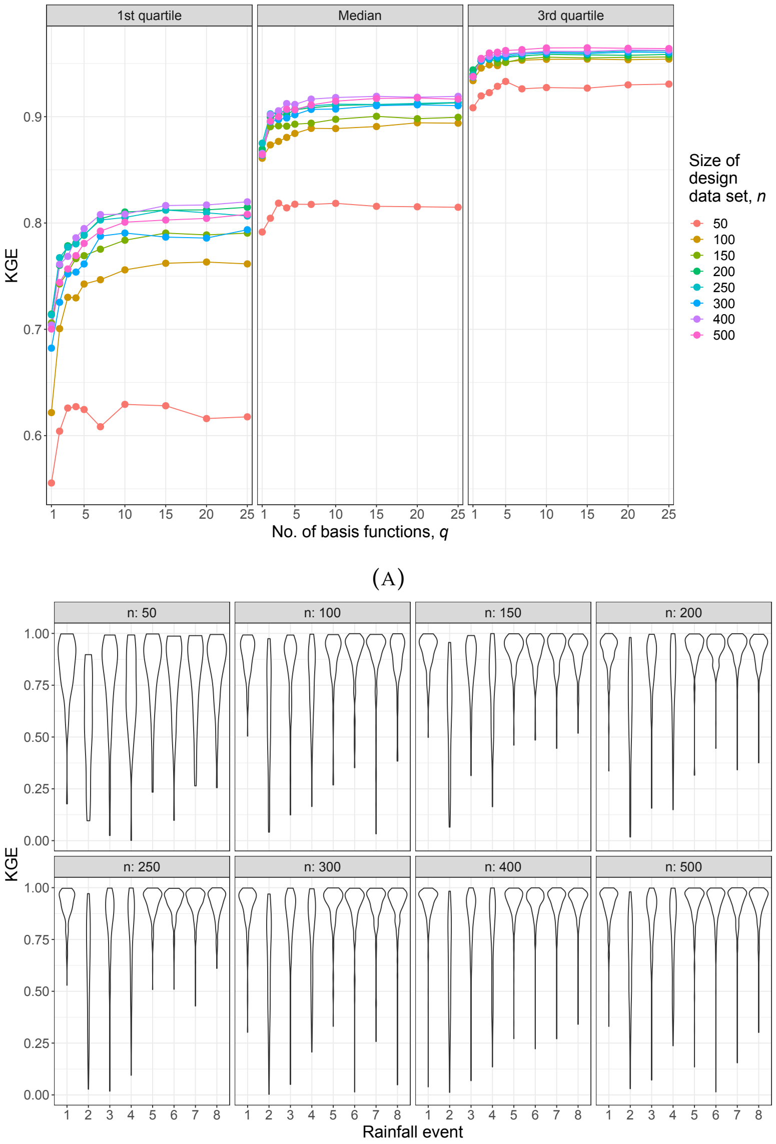
Fig. 3.5a shows the distribution of error for percentage overflow reduction (i.e., emulated percentage overflow reduction – simulated percentage overflow reduction) for the 8 rainfall events and the different values of n, and for a fixed $q = 7$. We observe again decrease in variance with increasing n for the larger rainfall events (5-8). There is also no bias in the predicted values. We also plot MAE and RMSE for the various values of n and q in Fig. 3.5b and 3.5c. We see again the improvements in prediction with increasing n or q. The RMSE values are close to the MAE values which indicate the absence of particularly large errors. Overall, the errors of emulators appear acceptable. Although using a larger n or q decreases emulation error, an emulator with larger n requires a longer time to construct and using an emulator with larger n or q for surrogate-assisted optimization is more time consuming. Hence, having $n \geq 150$ or $q \geq 7$ is possibly sufficient.

Performance of emulators in surrogate-assisted optimization

Although we construct emulators with q = 1, 2, 3, 4, 5, 7, 10, 15, 20, 25, we only use emulators with q = 3, 4, 5, 7, 10 in the surrogate-assisted optimization. This is a balance between computing time and accuracy, since if the emulator uses too few basis functions, it is inaccurate, and if there are too many basis functions, it slows down the optimization. Fig. 3.6 shows the RMSE of the percentage overflow reduction for the solution set evaluated at the 1st, 10th, 50th, and 100th generation of the optimization. For most rainfall events and also the weighted objective, RMSE tends to increase as the search progresses. Interestingly, RMSE decreases for rainfall event 4 (and also for rainfall event 2 and 3 in some instances). These are the small rainfall events for which the emulator is unable to predict flood conditions accurately (Fig. 3.5a) for the test set, which contains data points that are distributed across the decision space. However, during the optimization, the solutions converge locally in the decision space to a region where the emulator has a higher accuracy, hence the error decreases.

RMSE changes steeply in early generations and more gradually in later generations. This is more noticeable for each individual rainfall events rather than the weighted objectives, since the decrease in RMSE for some rainfall events cancels out the increase in RMSE for the other rainfall events. This suggests that emulator update in the surrogate-assisted optimization can take place early on (within 10 generations of the evolutionary algorithm), although it is also important to ensure that the updated emulator are not just fitted locally to allow exploration.

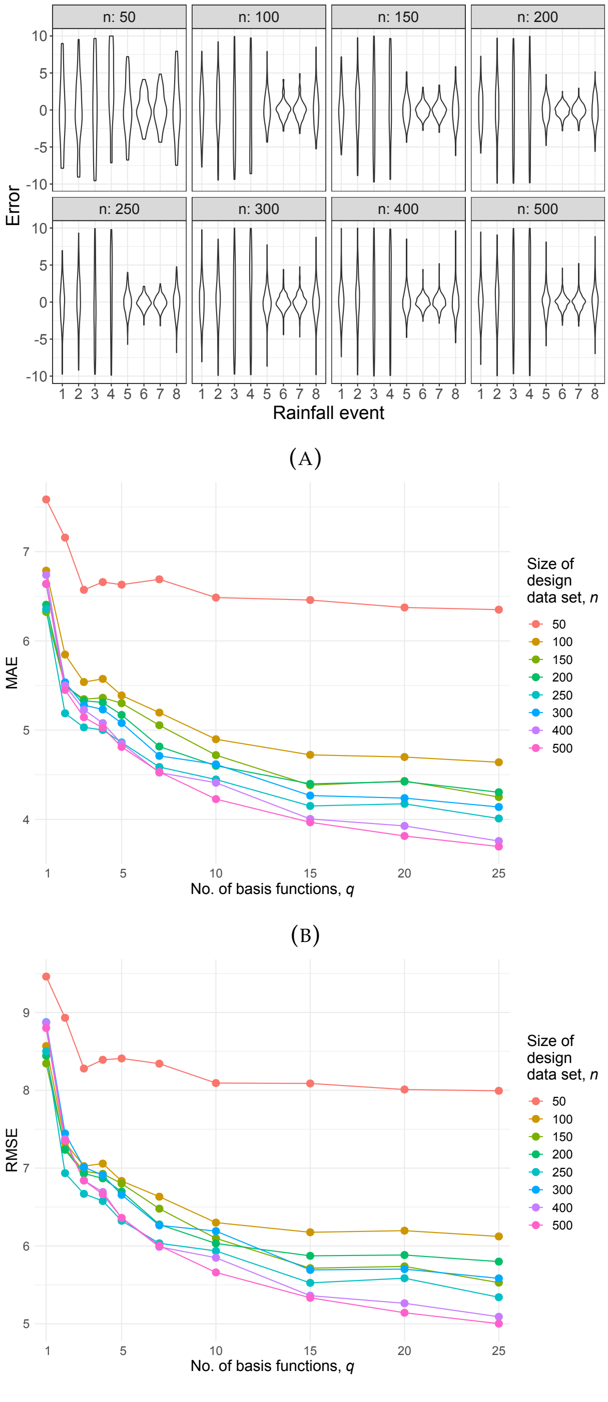
Fig. 3.7 shows the MBE of the percentage overflow reduction. There is an increasing bias as search progresses which is undesirable since it means that the Pareto-efficient solutions obtained from the surrogate-assisted optimization are underperforming when evaluated by the simulator. However, the bias can be mitigated by using more basis functions in the emulators. As we see from Fig. 3.6b, 3.6c, 3.7b, and 3.7c, especially in the weighted objective panel, increasing q improves prediction of the emulator more



(A)

(B)

FIGURE 3.4: (a) Statistics of KGE of overflow reduction (by volume) time series predicted by emulators with different n and q. (b) Distribution of KGE of overflow reduction time series for the 8 rainfall events predicted by emulators with different n and $q = 7$.



(A)

(B)

(C)

FIGURE 3.5: (a) Distribution of errors of percentage overflow reduction for the 8 rainfall events predicted by emulators with different n and q = 7. (b) Mean absolute error (MAE) and (c) root mean square error (RMSE) of percentage overflow reduction predicted by emulators with different n and q.

than increasing n does.

Finally, we look at the quality of the solutions obtained from the surrogate-assisted optimization by calculating the hypervolume, ϵ-indicator and generational distance of the Pareto front obtained at the end of 100 generations. Fig. 3.8 shows the performance metrics against the time taken for optimization for each setup of emulator. This time does not count the time used to build the emulator, which we know scales linearly with n and is not affected by q. Time taken for optimization scales linearly with q and exponentially with n. There appears to be no clear relation between complexity of the emulator and quality of solutions obtained. Even though emulators with higher q (and to a lesser extent, higher n) tend to produce lower error in predictions, they do not necessarily give more optimal solutions (i.e., when evaluated using the simulator, the solutions with high prediction error can still outperform a solution with low error). This also suggests that the search process itself is more important than the emulator used. Modifying the search process, such as accounting for the uncertainty in prediction to emphasize greater exploration, might give greater benefits than trying to minimize emulation error. Given the same optimization algorithm, it may not be worthwhile to use a more complex emulator (larger n or q), hence we proceed with our analysis using a single emulator (n = 250, q = 7) that produces relatively low error.

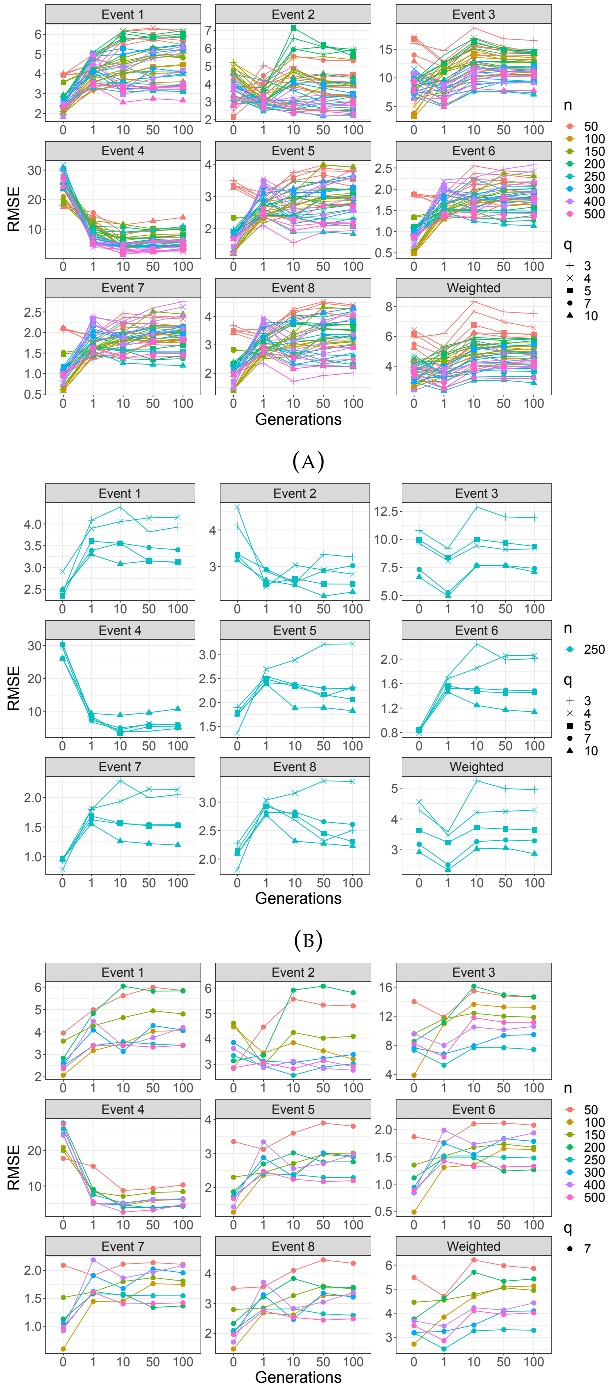
3.5.2 Performance of the Pareto-efficient solutions

In the following results, we perform surrogate-assisted optimization using the emulator that gives the highest average KGE during the modelling phase (n = 250, q = 7) and compare the solutions with those obtained from simulation-optimization.

Solutions from surrogate-assisted optimization

The 10 independent optimization runs of R-EO returned a total of 1,292 Pareto-efficient solutions (as predicted by emulator). The solutions were then evaluated by SWMM and the results are plotted in Fig. 3.9. The cost of solutions ranges from $0.2 million to $45 million with the most expensive alternatives guaranteeing a reduction in weighted overflow volume and weighted peak flooded nodes up to 45% and 21% respectively. Although the emulator predicts overflow reduction in the range of 20% to 49%, when evaluated by the simulator this objective ranges from 16% to 45%. For flooded nodes reduction, the emulator predicts values from 8% to 20% but the simulator gives values from 0.2% to 21%.

The positive bias between the emulated and simulated objective values is small for low cost solutions but increases as the cost of solutions increase and stays fairly constant beyond the kink in the Pareto front at around $5 million. This kink in the Pareto front is similar to the results in Fig. 2.6. Before this point, with a limited budget, investment goes towards pipe expansions and beyond this point, investment goes towards LID implementation. The increasing bias in solutions below this kink means that it becomes increasingly harder for the emulator to predict accurately when there are more changes in pipe inputs. Similarly, the constant bias observed in solutions above the

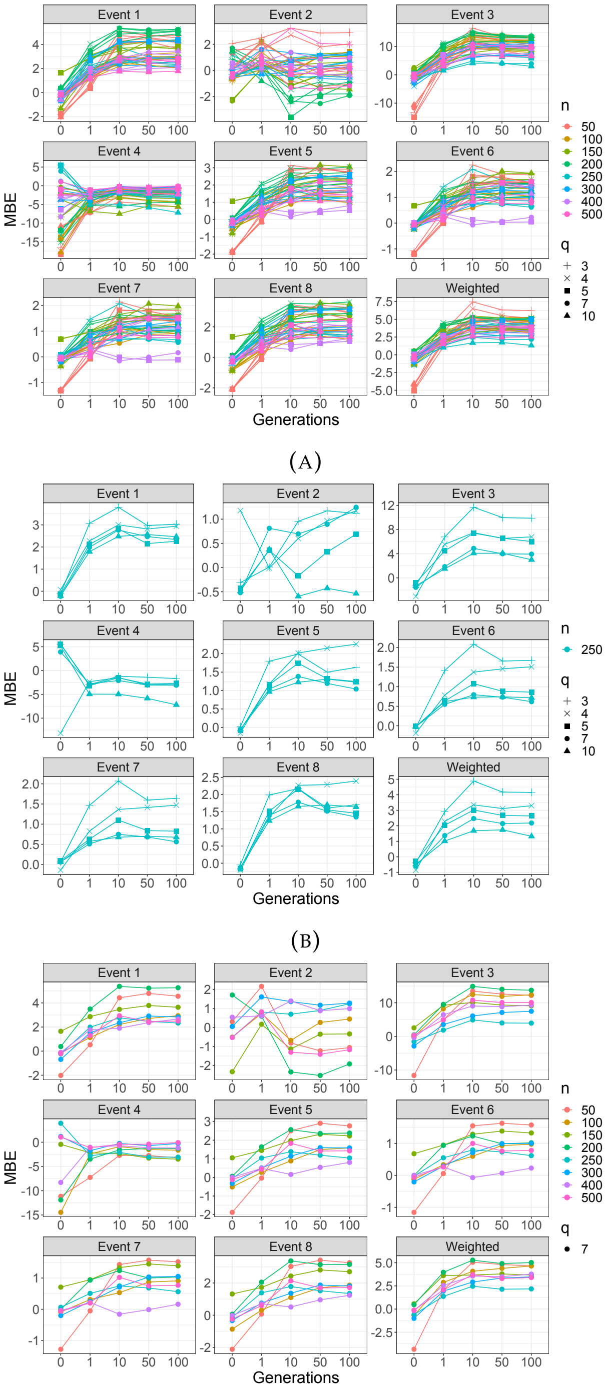


(A)

(B)

(C)

FIGURE 3.6: Root mean square error (RMSE) of percentage overflow reduction for test set (0th generation) and solutions obtained during 1st, 10th, 50th, and 100th generation of surrogate-assisted optimization (a) using emulators with different n and q; (b) using emulators with n = 250; and (c) using emulators with q = 7.



(A)

(B)

(C)

FIGURE 3.7: Mean bias error (MBE) of percentage overflow reduction for test set (0th generation) and solutions obtained during 1st, 10th, 50th, and 100th generation of surrogate-assisted optimization (a) using emulators with different n and q; (b) using emulators with n = 250; and (c) using emulators with $q = 7$.

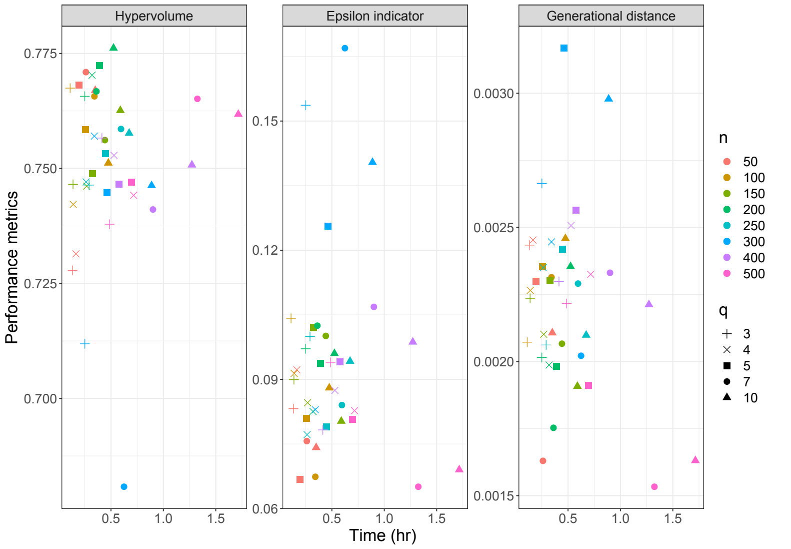


FIGURE 3.8: Performance metrics against time taken for surrogate-assisted optimization using emulators of different n and q

kink suggests that it is relatively easier for the emulator to predict changes in system response due to increasing LID implementation.

Comparison with simulation-optimization

Fig. 3.10 shows the Pareto-efficient solutions obtained from the three optimization schemes: R-EO, R-SO, and DS-SO. The best performing optimization scheme is R-SO. This is an expected result since it evaluates solution performance under 8 rainfall events using the simulator during optimization. Pareto front obtained from R-EO and DS-SO are similar but differ in two main ways. First, for flooded nodes reduction (Fig 3.10b), no solution from DS-SO is able to attain flooded nodes reduction up to 20% unlike solutions from R-EO or R-SO. Second, some solutions from DS-SO, especially the low cost one, deteriorate in performance when evaluated using the 8 rainfall events that have intensities, duration, and profiles differing from the design storm. When evaluated using the design storm, the Pareto front does not have any gaps as shown in Fig. 3.11, but because these solutions are designed specifically for the design storm, some of them are not robust against other rainfall events, hence resulting in gaps in the Pareto front (see Fig. 3.10a and 3.10b). For solutions obtained from R-EO, although evaluating with simulator causes performance to deteriorate, none of the solutions have emulated objective values with exceptionally high positive bias, thus there is no major gaps in the Pareto front obtained from R-EO.

Fig. 3.12 compares the three optimization schemes according to hypervolume, ϵ-indicator and generational distance. The individual trial for R-SO or DS-SO refers to each run that is initiated using a different random seed while the combined trial refers to the combined Pareto front obtained from the 10 individual trials. Only the combined trial is available for R-EO since the Pareto front is computed using the emulated

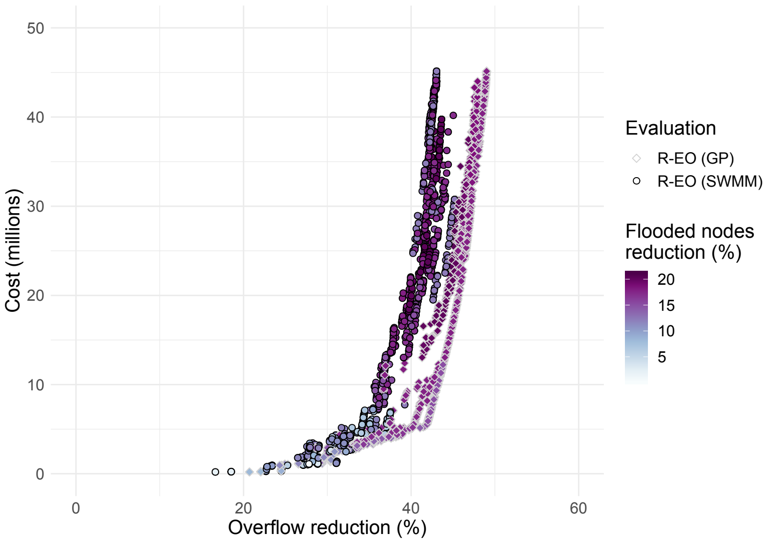


FIGURE 3.9: 1,292 Pareto-efficient solutions from R-EO. The three objectives are represented by the axes (weighted overflow reduction and cost) and color (weighted reduction in the number of peak flooded nodes). Solutions are evaluated using the emulator (grey border points) during optimization and evaluated using the SWMM simulator (black border points) post optimization.

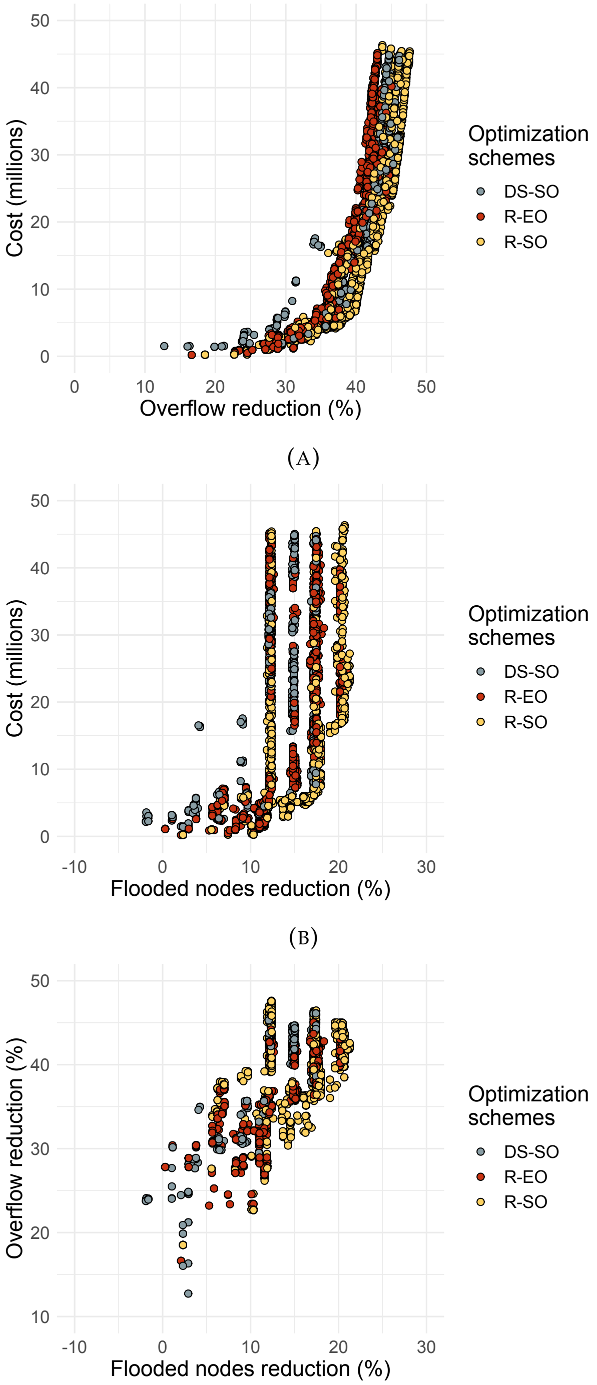
objectives before evaluation using SWMM. Here, we see that R-SO is the best performing, followed by R-EO, and followed lastly by DS-SO. The gaps in the Pareto front obtained using DS-SO result in the lower performance which is particularly prominent when measured by ϵ-indicator and hypervolume as these two metrics are more easily affected by gaps. Interestingly, there is better performance in the early generations for individual trials of DS-SO in terms of ϵ-indicator and hypervolume. This is because in the early generations, there are more 'random' solutions that are not designed for the design storm and happen to perform well for other rainfall events. This is also why the early generations for the combined trial does not show the same trend, since it only includes the Pareto-efficient solutions from the individual trials. This trend can also be observed for R-EO for the ϵ-indicator metric, also for the reason that in early generations there can be more 'random' solutions that actually perform well in the simulator but are later on eliminated as they are predicted to perform poorly by the emulator. We also note that the optimization search converges early for R-EO, and is close in performance to individual trials of R-SO only in early generations (<15). This suggests that an update of emulator could potentially reinvigorate the search such that performance of R-EO will not stagnate.

Finally, we compare the time taken to perform the three optimizations, including the time needed for simulations to construct the emulator prior to optimization and also to evaluate solutions using the simulator post optimization for R-EO (Fig. 3.13). Simulation time using SWMM takes about 4-10 seconds depending on the rainfall duration for one solution and for a single rainfall event. A single trial of DS-SO takes around 32 computation hours and thus the time needed for 10 trials is around 320 hours. In comparison, 10 trials of R-EO take only 4.4 computation hours. The time consuming component of R-EO is not the optimization but the simulations required to construct the emulator (7 hours) and to evaluate the solutions (30 hours). R-SO on the other hand requires 536 hours just for a single trial since it evaluates each solution through 8 different rainfall events and has to perform 20,000 function evaluations during optimization. The advantage of R-EO over DS-SO is evident here, since it provides time savings by a factor of 7 and also yields better solution. The comparison between R-EO and R-SO is harder to make since the quality of solutions obtained for the two schemes is only comparable in the early generations. While a single run of R-SO to roughly 13 generations takes about the same time as R-EO, the potential that R-EO has is that computation resources can be reallocated to make this method more efficient. Since the current R-EO adopts a basic sequential framework, all resources are spent on the initial DoE to construct the emulator, thus if some computation resources can be used to update the emulator during optimization, better solutions can be obtained using the same time needed for emulation and optimization (note that the post evaluation will always take roughly the same time).

3.5.3 Robustness analysis

Fig. 3.14 shows the cumulative distribution of performance (i.e., overflow reduction and peak flooded nodes reduction) across the 16 random rainfall events used for robustness analysis for all Pareto-efficient solutions obtained from R-EO, R-SO, and DS-SO. Note that the cumulative distribution here refers to weights, defined in Eq. 3.10, and not probability, such that rainfall events with larger flood or higher frequency have higher weights. This is consistent with the earlier cumulative distribution plots in Fig. 2.8. Each line in the plot represents a solution and a solution has better performance if it lies as far to the left as possible; this is also why we see expensive solutions (in green-blue) lying closer to the left. For example, if a line passes through the point (50%, 0.6), it can be interpreted as a solution achieving a reduction greater than or equal to 50% for 0.6 (in terms of accumulated weights) of the random rainfall events.

From Fig. 3.14a, we see that some solutions obtained from DS-SO are less robust against the wide range of rainfall conditions represented by the random set of rainfall events. Consider the horizontal line $y = 0.5$, solutions obtained from R-EO or R-SO intersect this line at 50-70% overflow reduction but there are solutions from DS-SO that attain less than 40% overflow reduction. For R-EO and R-SO, from $y = 0$ to $y = 0.6$, the solutions lie close together and difference in performance is a result of different cost of the solutions. Beyond $y = 0.6$, solutions are more scattered with some solutions deviating to the right, underperforming compared to solutions of similar cost. This is most apparent in the plot for DS-SO, since the solutions are designed for the design storm and some of them are unable to perform well in the rainfall events that are sampled. Similar insights can be drawn from Fig. 3.14b although it may be less apparent since peak number of flooded nodes takes on discrete values and solutions have overlapping lines.



(A)

(B)

(C)

FIGURE 3.10: Comparison of Pareto-efficient solutions from DS-SO, R-EO, and R-SO in terms of (a) cost against weighted overflow reduction; (b) cost against weighted peak flooded nodes reduction; and (c) weighted overflow reduction against weighted peak flooded nodes reduction. For DS-SO, solutions are Pareto-efficient for the design storm and are not necessarily optimal when evaluated using the weighted objectives.

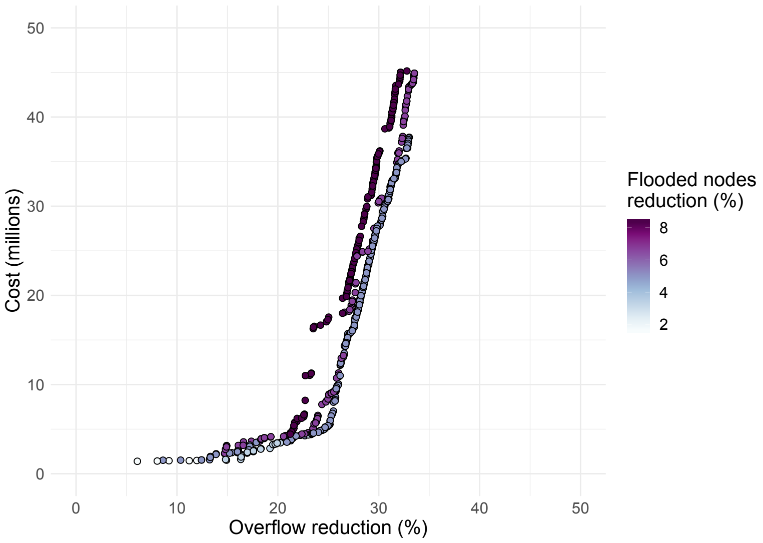
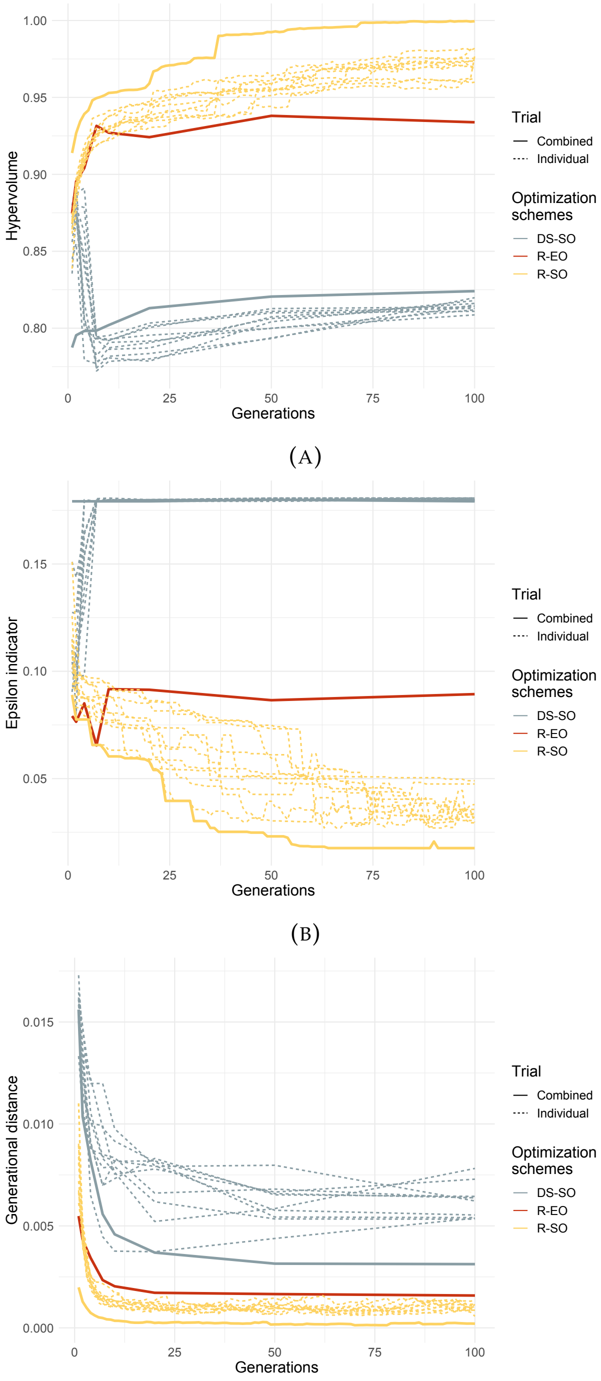


FIGURE 3.11: 627 Pareto-efficient solutions from R-SO. The three objectives for DS-SO are represented by the axes (overflow reduction and cost) and color (reduction in the number of peak flooded nodes). These solutions are designed specifically for the design storm.

Table 3.4 shows the hypervolume of the Pareto front for the three optimization schemes for the two set of random rainfall events sampled. The hypervolume is calculated for each of the 8 rainfall events and also for the weighted objectives. The rainfall events are arranged in order of increasing severity of overflow caused. We note that although DS-SO attain a higher hypervolume for the weighted objectives, this is largely due to the poor performance for Rainfall Event 2 for both R-EO and R-SO. In fact, the overflow or flooded nodes reduction attained by all solutions for Rainfall Event 2 is quite similar. Another issue is that the decision variables selected for optimization are not suited to reduce overflow in a small rainfall event as highlighted in Section 2.4.2. We shall resolve this issue in the next chapter when we include the stochastic rainfall events into sensitivity analysis so that we select decision variables that target a wide range of rainfall conditions prior to optimization. If we look at the larger rainfall events (i.e., Rainfall Event 5-8), we notice that solutions from DS-SO underperforms. This can be expected for the largest rainfall events (Rainfall Event 6 and 7) since they are more intense than the design storm. However for Rainfall Event 5 and 8, they are less intense than the design storm but have a longer duration. Solutions from DS-SO underperform for such events, consistent with what is illustrated in Fig. 2.9 in the previous chapter. For the even smaller rainfall events (Rainfall Event 1 and 3), we also see that solutions from DS-SO do not perform as well as those from R-EO or R-SO although the margin is smaller. Lastly, we note that solutions from R-EO does not perform as well as R-SO. This is expected due to the emulation error but the inclusion of stochastic rainfall events during optimization does allow us to obtain solutions that perform well across a wide range of rainfall conditions.



(A)

(B)

(C)

FIGURE 3.12: (a) Hypervolume, (b) ϵ-indicator, and (c) generational distance of solution set as optimization progresses for DS-SO, R-EO, and R-SO. Individual trial refers to each run that is initiated using a different random seed while the combined trial refers to the combined Pareto front obtained from the 10 individual trials.

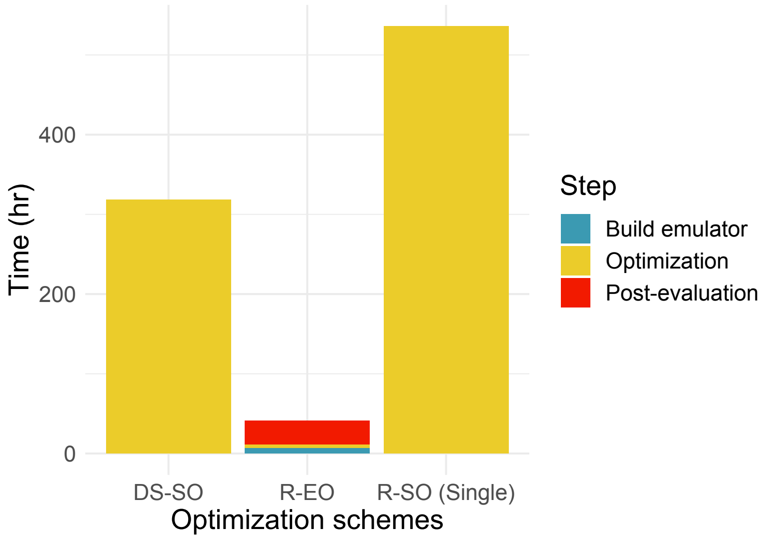


FIGURE 3.13: Comparison of computation time required for DS-SO, R-EO, and R-SO. Post-evaluation refers to evaluating the Pareto-efficient solutions obtained from R-EO using the simulator.

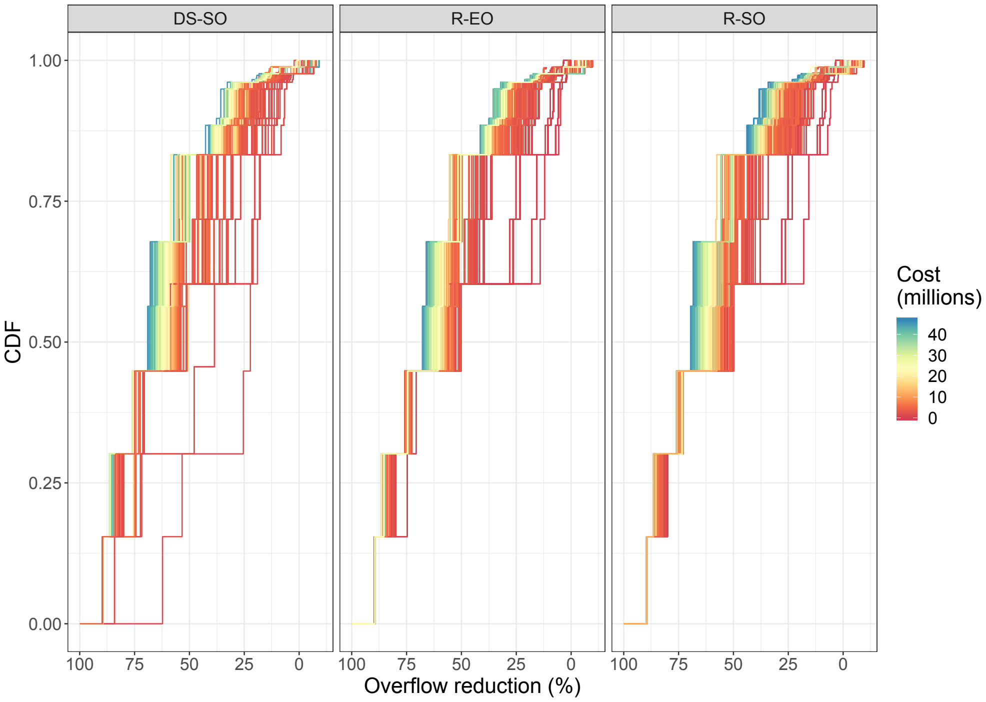
TABLE 3.4: Hypervolume of solutions from DS-SO, R-EO, and R-SO for random rainfall event set 1 (left) and set 2 (right)

Set 1	DS-SO	R-EO	R-SO
Event 4	0.97	1	0.99
Event 2	0.99	0.58	0.87
Event 3	1	1.02	1.03
Event 1	1	1.03	1.03
Event 8	0.75	1.09	1.12
Event 5	0.96	1.09	1.1
Event 6	0.75	1.06	1.17
Event 7	0.47	0.93	1.08
Weighted	0.97	0.8	0.95

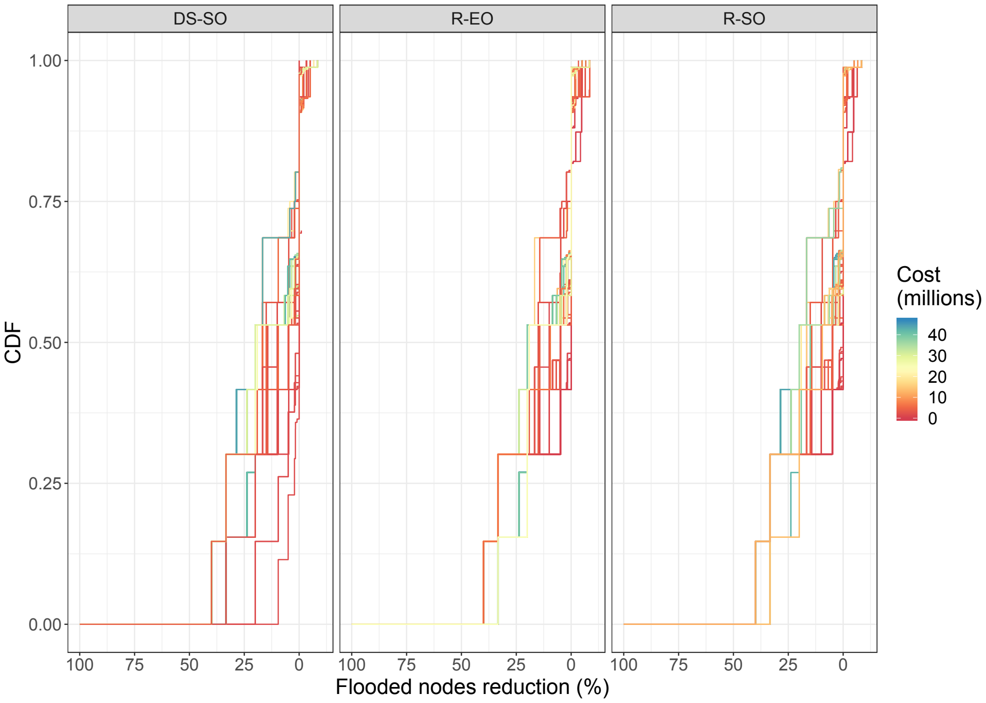
Set 2	DS-SO	R-EO	R-SO
Event 4	0.94	0.89	1
Event 2	0.97	1	0.98
Event 3	0.9	0.95	1
Event 1	0.97	0.95	1
Event 8	0.98	0.99	1.01
Event 5	0.96	1.03	1.05
Event 6	0.76	1.01	1.07
Event 7	0.6	0.96	1.04
Weighted	0.59	0.46	0.54

Note that 2 random rainfall events are selected from each of the 8 clusters, which give rise to the 2 set of events.

Note that the reference set to calculate hypervolume is the best known solution set found across the three optimization schemes for the weighted objectives. Hence for the individual rainfall events, the solution set from the different optimization scheme may be better than this reference set, hence hypervolume can exceed 1 in such cases.



(A)



(B)

FIGURE 3.14: Cumulative distribution for (a) overflow reduction and (b) flooded nodes reduction for the 16 random rainfall events selected. Note that the x-axis is inverted so that it proceeds in the direction of degrading performance from left to right.

3.6 Conclusion

Designing urban drainage systems robust against various rainfall conditions is critical for cities. The use of design storms in the planning of urban drainage systems, although widely used, may not always be effective in producing robust drainage design. Our work introduces stochastic rainfall events into the optimization phase and uses surrogate-assisted optimization to derive solutions robust against these events using limited computational resources. The application of our computational framework to the Nhieu Loc-Thi Nghe basin reveals a few insights that answer the questions we set out with:

- Solutions optimized for stochastic rainfall events can perform better in a wider range of rainfall conditions compared to solutions optimized only for the design storm. This is more apparent in rainfall events that have large rainfall volumes even if the rain intensity could be smaller than the design storm. Although solutions optimized for stochastic rainfall events can also underperform under small rainfall events, this can be explained by the inadequate selection of decision variables for optimization. This prompts the use of stochastic rainfall events prior to optimization in the selection of decision variables.

- By including stochastic rainfall events into the optimization process, computational demand multiplies but surrogate-assisted optimization provides a feasible alternative to simulation optimization by reducing time needed at the expense of some accuracy. Surrogate-assisted optimization is 7 times (approximately 250 hours) faster than simulation optimization using design storm and 129 times (approximately 5000 hours) faster than the robust simulation optimization setup using stochastic rainfall events. However, its performance is only comparable to robust simulation optimization in the early generations of the evolutionary algorithm, thus suggesting that computational resources can be reallocated to update the surrogate model during optimization.

- The emulator used in our work here shows increasing accuracy with larger number of design data points (n) and greater number of basis functions (q) but the increase in accuracy is diminishing for larger values of n and q. More importantly, when the emulator is used within surrogate-assisted optimization, the accuracy of the emulator does not appear to correlate highly with the quality of the designed solutions.

Overall, this part of the thesis suggests that optimizing using stochastic rainfall events rather than just design storms can yield urban drainage systems that are more robust against a wide range of rainfall conditions. To deal with the increased computational demand, surrogate-assisted optimization is a viable option. While the results here are specific to our case study, the computational framework is not catchment specific and can be readily applied to other catchments. We expect similar conclusions for other catchment models although one might expect varying degrees of success. The effectiveness of adding stochastic rainfall events largely depends on how effective (or ineffective) the design storms are for the catchment model and the time savings promised by surrogate-assisted optimization depend on the time required for a single simulation (i.e more time savings expected if the original function is more time consuming).

Although surrogate models for urban drainage simulator have been developed, they have not been used for optimization before. The work here is a first attempt at using surrogate-assisted optimization to design for robust urban drainage. Further research in this direction can also benefit developing cities in the planning of robust drainage design under climate change or population growth. In this chapter, since we only experiment on an emulator with 5 inputs, it is easier to interpret our findings but more difficult to realize a greater extent of overflow reduction. In the next chapter, we will explore how our framework can be scaled to solve problems of higher dimensions.

Chapter 4

Scalability of emulator for the robust design of urban drainage systems

4.1 Introduction

In the previous chapter, we set out to design robust urban drainage systems by replacing the design storm with stochastic rainfall events during the optimization process (and with the aid of emulators). Since the main goal was to test out the methodology, we limited the problem to a 5-dimensional decision space. In this chapter, we explore the possibility of scaling our computational framework to a higher dimensional problem (i.e., greater number of decision variables). While we acknowledge the caveat that the emulator may be less accurate for a higher dimension problem, we want to find out the extent to which surrogate-assisted optimization can allow us to design robust urban drainage systems in more practical setting. In this chapter, we thus aim to answer the following questions: (1) How does the accuracy of the emulators change with problems of increasing complexity?; and (2) How effective is surrogate-assisted optimization in producing robust urban drainage solutions for higher dimensional problems?

To do this, we adopt the computational framework in Chapter 3 but introduce two main modifications. First, instead of the design storm, we use a set of representative rainfall events generated from the stochastic rainfall analysis in the sensitivity analysis. In this way, the sensitivity analysis identifies variables that reduce overflow in various rainfall events rather than just the design storm. Second, we select a new set of 20 decision variables from the aforementioned sensitivity analysis. We then build the emulator for the higher dimensional decision space using the new inputs and use surrogate-assisted optimization to obtain the optimal configuration of these new 20 decision variables.

The computational framework presented in this chapter represents a culmination of efforts to improve the methodology for the design of robust urban drainage systems. Here, we deviate from the common practice of using design storms, relying on stochastic rainfall events in both sensitivity analysis and optimization. To deal with the increased computational demands, surrogate-assisted optimization is used instead of simulation-optimization. To evaluate how our proposed methodology holds up against current design practices, we applied the framework to the Nhieu Loc-Thi Nghe basin and compare the solutions obtained from this proposed framework to the solutions obtained in Chapter 2, in which the design process is solely based on design storms.

4.2 Computational framework

The computational framework for this chapter consists of 5 steps, illustrated in Fig. 4.1. We begin with the stochastic rainfall analysis, with which we generate a set of rainfall events representative of the range of rainfall intensities, duration, and profiles of rainfall events occurring in the catchment. Next, we conduct sensitivity analysis to select a set of decision variables that targets overflow reduction for the representative rainfall events found in the previous step. To reduce the additional computational demands brought about by the inclusion of stochastic rainfall events, we modify the sensitivity analysis by partitioning and screening the variables. We then build an emulator based on the selected set of decision variables, and use the emulator within the surrogate-assisted optimization. The optimization seeks Pareto-efficient solutions that minimize the investment costs while maximizing the reduction in both overflow volume and flooded nodes during the representative rainfall events. We compare our solutions obtained from surrogate-assisted optimization with the solutions obtained from the design storm simulation-optimization performed in Chapter 2. We finally evaluate the robustness of solutions obtained from the two optimization schemes by simulating these solutions under unobserved stochastic rainfall events.

With the exception of Step 2 (modified sensitivity analysis), the other steps have been explained in Section 3.4. We next explain the modified sensitivity analysis in detail and describe the setup for the emulation and optimization step for this framework.

4.2.1 Modified sensitivity analysis

From Chapter 2 and 3, we found that the Pareto-efficient solutions failed to reduce overflow adequately even for small rainfall events and one of the reasons was due to the inadequate selection of decision variables for optimization. In those optimization setup, decision variables were selected for their performance at reducing overflow during the design storm using sensitivity analysis (Section 2.3.1), hence they tend to be in the areas that experience severe overflow during the design storm. However, for a large watershed, such as the Nhieu Loc-Thi Nghe basin, overflow can occur at different areas during different rainfall events, which may not necessarily coincide with the severe overflow areas during the design storm. Thus, flood may persist in different areas if decision variables are selected only based on the design storm. Therefore, stochastic rainfall events have to be included during sensitivity analysis for a comprehensive selection of decision variables.

We thus use the 8 representative rainfall events identified in Section 3.4.1 in the sensitivity analysis. As a result, computational demands will increase by around 8 times because it scales linearly with the number of rainfall scenarios. However, it is inefficient to consider the entire original decision space, which consists of 308 pipe variables and 12 LID variables, since pipe expansion or LID implementation at a one location cannot

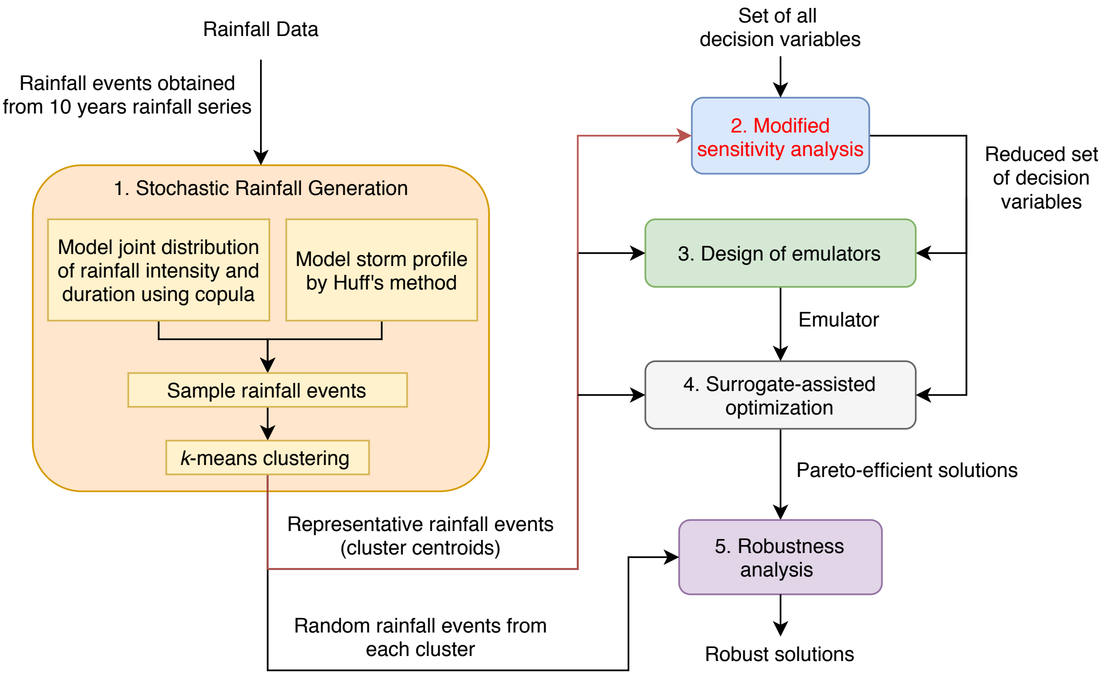

Rainfall Data
Rainfall events obtained from 10 years rainfall series
1. Stochastic Rainfall Generation
Model joint distribution of rainfall intensity and duration using copula
Model storm profile by Huff's method
Sample rainfall events
k-means clustering
Representative rainfall events (cluster centroids)
Random rainfall events from each cluster
Set of all decision variables
2. Modified sensitivity analysis
Reduced set of decision variables
3. Design of emulators
Emulator
4. Surrogate-assisted optimization
Pareto-efficient solutions
5. Robustness analysis
Robust solutions

FIGURE 4.1: Flowchart of the computational framework, consisting of 5 main steps: 1) stochastic rainfall generation to select rainfall events that are representative of a wide range of rainfall intensities, duration, and profiles; 2) sensitivity analysis to select decision variables for emulation and optimization; 3) design of emulators to construct GP emulator that approximate SWMM; 4) surrogate-assisted optimization to obtain Pareto-efficient solutions that performs well across the representative rainfall events; 5) robustness analysis to evaluate the performance of the solutions for random rainfall events. The key difference between this framework and the one in Fig. 3.1 is highlighted in red. Multiple rainfall events, instead of the design storm, are used in the modified sensitivity analysis.

alleviate the overflow at a distant location. This is especially true for the smaller rainfall events that cause overflow in a limited number of nodes. We thus implement a preprocessing step before the sensitivity analysis in which we first cluster the nodes (i.e., the manhole of the drainage system) of the catchment according to the flow conditions during each rainfall event, and perform sensitivity analysis using decision variables in clusters that account for large overflow volumes.

For each rainfall event, we want to cluster the 227 nodes in the catchment model according to the flow conditions of each node. We first simulate the catchment model using SWMM under the existing drainage system. Each node is then represented by the concatenated time series of overflow (as a ratio of total overflow in the catchment at that time instant) and depth (as a ratio of maximum depth of the node). We then use hierarchical time series clustering (Legendre and Legendre, 2012) to form the clusters of nodes. Here, we use an agglomerative clustering technique that treats each node as a separate cluster initially and merges clusters that are the most similar until a single cluster is formed. An advantage of this clustering technique is that we can perform an ex-post selection of the number of clusters instead of having to define it a priori. Another advantage is that we can easily add additional conditions to the clustering such that only clusters that are adjacent (i.e., the clusters of nodes are linked by at least one common pipe) can merge. This is done out of theoretical considerations such that the resulting cluster of nodes have similar flow conditions during the rainfall event because they are physically connected. Distance between each cluster is measured by the Euclidean distance between the time series vector. Clusters are merged according to the complete linkage criteria, which defines the distance between cluster using the furthest distance between a pair of nodes in different clusters. We select the number of clusters by considering the size of the largest cluster and also the Davies-Bouldin Index (DBI) (Davies and Bouldin, 1979), which favors high within-cluster similarities and high between-clusters differences. Fig. 4.2 illustrates the clusters obtained by using hierarchical clustering on Rainfall Event 1. For the clusters obtained in the remaining 7 rainfall events, please refer to Appendix C.

We carry out elementary effect test (EET) (Pianosi et al., 2016) for each of the representative rainfall events independently to identify the most influential variables that can reduce overflow volume for the specific event. Similar to Section 2.3.1, the output considered is the reduction in total overflow volume. Instead of considering all the diameters of 308 pipes and area of the 12 LIDs as the input factors (as in Section 2.3.1), we only consider the pipes and LID variables within 'important' clusters. The 'important' clusters are found by ranking the clusters obtained from hierarchical clustering according to the overflow volume per node in decreasing order, and selecting the clusters until they cover at least 90% of the overflow volume for that rainfall event. In this way, we discard clusters that are distant from overflow locations or are less likely to contribute to overflow reduction. Taking advantage of the clusters, we then implement a slightly modified version of EET. In the original EET, inputs factors of the simulation model are perturbed one at a time from multiple points within the input space. Global sensitivity is measured by taking the mean of the elementary effects, namely the local derivatives of the output with respect to an input. In our modified version, given N

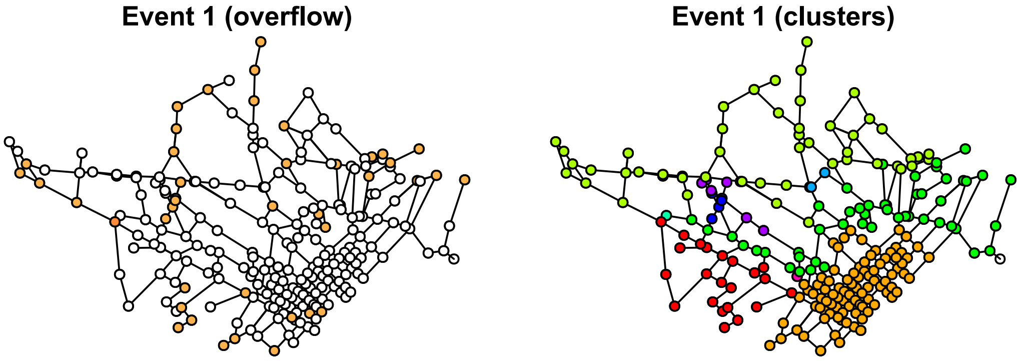


FIGURE 4.2: Nodes with overflow (represented by colored nodes in the left plot) and the resulting 9 clusters obtained using hierarchical clustering (right) for Rainfall Event 1. Each cluster is represented by a different color in the plot on the right. The nodes and pipes represent the manholes and pipes of the drainage system.

'important clusters, we perturb N inputs factors at a time (one from each cluster). We then attribute overflow changes within each cluster to the input factor that is perturbed within that cluster. The assumption here is that the effect of inputs on a different cluster is limited. For EET, a total of $r(M+1)$ evaluations is needed, where M is the number of factors and r the number of elementary effects. In this instance, r = 100 — a choice made in Section 2.3.1. M is then the size of the largest 'important' cluster, which in our case, corresponds to 80, 34, 81, 31, 31, 45, 44, and 44 for the 8 rainfall events respectively. This marks a reduction in simulations required since M = 320 if all inputs are included.

Finally, for each rainfall event, we rank the inputs according to the mean elementary effect, selecting the top inputs that are most influential in reducing overflow. We then take the union of the set of inputs such that we have a total of 12 pipe variables selected for optimization. Table 4.1 lists the properties of the 12 pipe variables. Compared to Table 2.2, 3 of these variables are different from the variables identified using sensitivity analysis with the design storm only. 8 LID variables are also selected for optimization, as we want to keep the same number of LID and pipe variables for optimization as in Chapter 2. Properties of the LID variables are listed in listed in 2.3. For additional figures and details of the sensitivity analysis, please refer to Appendix C.

4.2.2 Design of emulators

We now construct the urban drainage emulator in the same way as described in Section 3.3 and 3.4.2 with the key difference that we are now using 20 inputs identified from sensitivity analysis instead of 5. This may have an impact on the emulator accuracy as the emulator has not been tested on problems with 20 inputs. Carbajal et al., 2017, who developed the emulation approach, has applied the emulator to problems up to 8 inputs. As a result, we need to reconsider the design of experiment and ensure that the accuracy of the emulator is acceptable.

TABLE 4.1: Description of the pipe variables selected from modified sensitivity analysis

Decision variable, x_j	Pipe no.	Shape	Length, l_j (m)	Size, d_j (m)	Maximum size (m)
x_1	1	Rectangular closed	546	2.5 x 2.5	4
x_2	3	Circular	510	1.2	2.2
x_3	4	Circular	180	1	2
x_4	5	Rectangular closed	840	1.5 x 2.8	4
x_5	6	Rectangular closed	600	1.2 x 2	4
x_6	8	Rectangular closed	540	0.8 x 5	4
x_7	9	Rectangular closed	551	1.2 x 5	4
x_8	11	Rectangular open	610	2.7 x 10	4
x_9	12	Rectangular closed	900	1.8 x 4	4
x_{10}	13	Circular	450	0.6	2
x_{11}	14	Rectangular closed	827	2 x 6	4
x_{12}	15	Rectangular open	1250	3.5 x 20	4.5

Maximum size refers to the maximum diameter for circular pipes and maximum depth for rectangular pipes. It corresponds to the maximum size of an existing pipe of the same shape or 1 m larger than the original size, whichever is bigger. Width of rectangular pipes are unchanged during optimization.

Pipe no. 13-15 are the new additions; they were not selected when sensitivity analysis was performed using the design storm.

For Gaussian process (GP) regression, Jones, Schonlau, and Welch, 1998 suggest that a general rule of thumb for the size of the initial design of experiment (DoE) is to have number of design points, $n = 10D$, where D is the dimension of the input space. Loeppky, Sacks, and Welch, 2009 find that n is dependent on the sensitivity (τ) and sparsity (ψ) of the inputs in a GP model. Sensitivity is defined as $\tau = \sum \theta_j$, and sparsity is defined as $\psi = \sum \theta_j^2$, where θ_j is the fitted parameter of the GP model for input j where $j = 1, ..., D$. A high θ indicates that the input is active and strongly influences the response, so a high sensitivity means that there are many active inputs. A high sparsity means that the the sensitivity does not spread equally across the D inputs, such that some are inactive inputs (low θ) and some are active inputs. Experimenting with D ranging from 4 to 20, Loeppky, Sacks, and Welch, 2009 find that if sensitivity is small ($\tau < 10$), then $n = 10D$ is possible. Even if sensitivity is high (τ > 20), if sparsity is also high, then it is still sufficient to use $n = 10D$. In other words, for high dimensional problems, if only a few inputs are active and strongly influence the response, then the problem can still be tractable with $n = 10D$ simulation runs to construct the GP model.

We investigate the feasibility of building emulator for our 20-D problem. We construct emulators using different size of design data set (n = 50, 100, 150, 200, 250, 300, 400, 500) and different number of features of the system response (q = 1, 2, 3, 4, 5, 7, 10, 15, 20, 25). We measure the accuracy of our emulators using KGE, MBE, MAE, and RMSE, defined in Eq. 3.5 to 3.8.

4.2.3 Surrogate-assisted optimization

After constructing the emulator, we use it in the surrogate-assisted optimization to identify the optimal drainage configurations for the 20 decision variables . The setup is the same as the one described in Section 3.4.3 except that the number of generations is increased from 100 to 250. This is the same number of generations used for the design storm simulation-optimization with 20 decision variables in Section 2.3.2. To make a distinction between this setup and the one in Section 3.4.3, we refer to this surrogate-assisted optimization as ’R-EO-20’. We compare solutions obtained from R-EO-20 with the Pareto-efficient solutions obtained from the simulation-optimization in Section 2.3.2, which we will refer to as ’DS-SO-20’. Lastly, we compare the robustness of the solutions obtained from R-EO-20 and DS-SO-20 by simulating the solutions under 16 different random rainfall events identified in Section 3.4.4 and listed in Table 3.4, which are unobserved during the optimization process.

4.3 Case study results

We begin by assessing the accuracy of emulators built for a 20-dimensional input space. Next, we describe the Pareto-efficient solutions obtained from the surrogate-assisted optimization (R-EO-20) and compare them with the solutions obtained from simulation-optimization (DS-SO-20). Finally, we evaluate the robustness of these solutions by comparing their performance under a wide range of rainfall events to determine if our proposed framework (represented by R-EO-20) can be more effective than current practices (represented by DS-SO-20) in deriving robust urban drainage systems.

4.3.1 Performance assessment of emulators

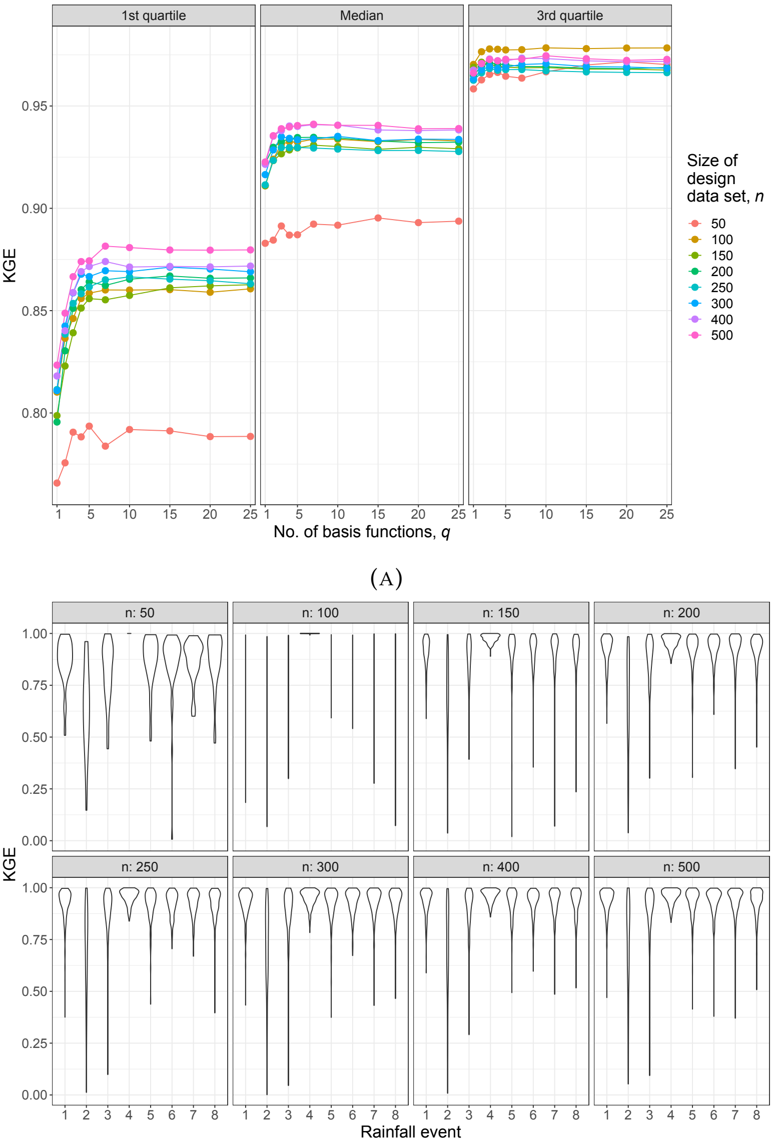
Fig. 4.3 shows the KGE of the overflow reduction time series predicted by the emulators for different values of n and q. Compared to the emulators constructed for the 5-D input space (Fig. 3.4), we notice the similarity that higher n and q give higher KGE. However, there are three differences between the emulators constructed for the 20-D and 5-D problems. First, the improvement in KGE for increasing n is not as obvious for the 20-D emulator as can be seen by the overlapping lines for $n \geq 100$. Second, the KGE is higher and the variance in KGE is smaller (Fig. 4.3b). However, this can be partially attributed to the larger absolute value in overflow reduction for the 20-D problem. Third, the 20-D emulator yields significantly better performance for Rainfall Event 4. This is likely because in the 5-D emulator, none of the inputs are close to the overflow location for this small rainfall event, but for the 20-D emulator, pipe expansion near the overflow location is possible. Hence, it becomes easier to predict overflow reduction now, since changes in some of the 20 inputs strongly influence the output response. This means that emulators performance may not necessarily degrade with increasing dimension of input space. Given the same size of design data set, the response surface (i.e., the relationship between the inputs and output) largely determines the performance of the emulator.

Fig 4.4 illustrates the error distribution, MAE, and RMSE of percentage overflow reduction predicted by the 20-D emulator. This is an aggregated output derived from the overflow time series that is predicted. There are two main differences with the 5-D emulator. Firstly, the error is higher and secondly, the error decreases more rapidly with increasing number of basis function, q. Although some regions of the response surface might be easier to predict now (e.g. Rainfall Event 4), the overall accuracy of emulator for the aggregated output is lower due to the higher dimension.

We eventually choose to use the emulator with n = 250 and q = 7 in the surrogate-assisted optimization. This is the same choice as in Chapter 3 because given the error plots, there is no strong reason to deviate from this choice. Table 4.2 gives the sensitivity and sparsity of the seven GP models (one for each basis function) used to build the emulator that predicts overflow time series. Recall that in this emulation approach, the time series to be predicted is first decomposed into q basis functions, and one GP model is needed for each basis function. For q = 7, seven GP models are needed. For each GP model, either sensitivity is low or sparsity is high, meaning that there are always a few strong inputs that influence the output, making prediction with GP regression easier. According to the recommendation in Loeppky, Sacks, and Welch, 2009, this makes a design data set of $n = 10D$, suitable even for a high dimension problem like the one considered here.

Performance of emulators in surrogate-assisted optimization

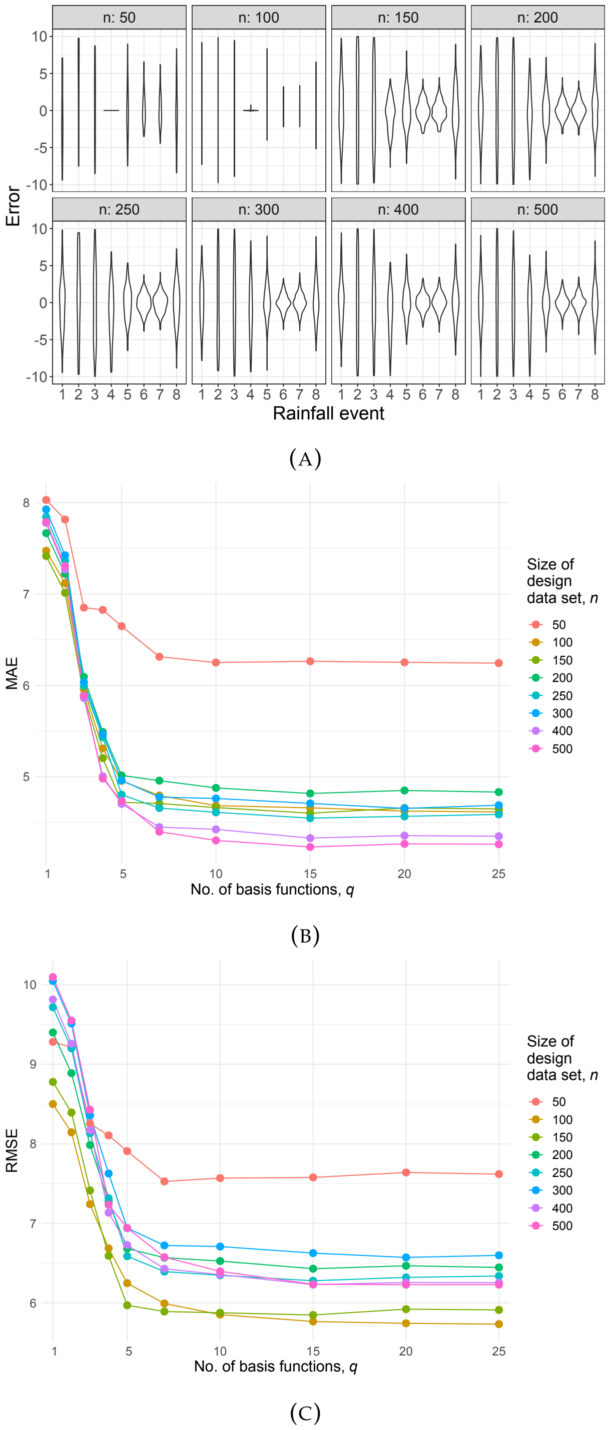
Fig. 4.5 shows how the error of the emulator propagates as the surrogate-assisted optimization search progresses for 250 generations. It shares similarities with the error plots for the 5-D surrogate-assisted optimization (Fig. 3.6 and 3.7): one, there is an increasing bias as search progresses for all events, except Rainfall Event 4; and two,



(A)

(B)

FIGURE 4.3: (a) Statistics of KGE of overflow reduction (by volume) time series predicted by emulators with different n and q. (b) Distribution of KGE of overflow reduction time series for the 8 rainfall events predicted by emulators with different n and $q = 7$. Emulators are constructed for 20 input variables.



(A)

(B)

(C)

FIGURE 4.4: (a) Distribution of errors of percentage overflow reduction for the 8 rainfall events predicted by emulators with different n and q = 7. (b) Mean absolute error (MAE) and (c) root mean square error (RMSE) of percentage overflow reduction predicted by emulators with different n and q. Emulators are constructed for 20 input variables.

TABLE 4.2: Sensitivity and sparsity of emulator with $n = 250$ and $q = 7$

Basis function	1	2	3	4	5	6	7
Sensitivity (τ)	5.59	10.87	11.3	12.43	13.47	13.83	20.06
Min. sparsity (τ^2/D)	1.56	5.9	6.38	7.73	9.07	9.56	20.13
Max. sparsity (τ^2)	31.24	118.09	127.58	154.6	181.47	191.24	402.55
Sparsity (ψ)	5.64	20.9	21.04	22.8	42.9	25.23	178.16

Minimum sparsity is calculated by assuming sensitivity is spread equally among all inputs. Maximum sparsity is calculated by assuming all but one input are inactive.

error increases rapidly in the earlier generations. The difference between the 5-D and 20-D surrogate-assisted optimization is that error for the higher dimensional problem is larger (12.5% for RMSE and 10.2% for MBE compared to the range of 3-8% for RMSE and 1-7% for MBE obtained by the 5-D surrogate-assisted optimization). This can be explained by the larger decision space of the optimization problem, and hence a higher likelihood for the search to be directed to a region where the emulator wrongly predicts that solutions are optimal. This indicates that even when the emulator may not have a high error for a high dimensional input space, caution must be taken during surrogate-assisted optimization as the error can increase more during the search process for a high dimensional problem compared to a lower dimensional one.

4.3.2 Performance of Pareto-efficient solutions

We now look at the Pareto-efficient solutions obtained using surrogate-assisted optimization (R-EO-20) and compare them to the solutions obtained from simulation-optimization (DS-SO-20).

Solutions from surrogate-assisted optimization

The 10 independent optimization runs of R-EO-20 returned a total of 765 Pareto-efficient solutions (as predicted by the emulator). Fig. 4.6 shows the Pareto front when evaluated by the simulator (SWMM) and emulator. The solutions cost up to $80 million and they attain weighted overflow reduction up to 84% and weighted peak number of flooded nodes reduction up to 42%.

We notice a kink in the Pareto front at around the $10 million mark, where solutions exhibit smaller increase in overflow reduction for increasing cost beyond this point. This is similar to the plots observed for other optimization setup in Fig. 2.6 and 3.10a. The kink is attributed to the switch in investing in LID instead of pipe expansion. However, when evaluated using the simulator, a gap appears such that no solution has overflow reduction between 51% and 60%. Furthermore, we see that the positive bias between the emulated and simulated objective values is larger for the less expensive solutions on the left of the gap. This runs contrary to the 5-D surrogate-assisted optimization (R-EO) (Fig. 3.9), where the bias is larger for cheaper solution. For R-EO,

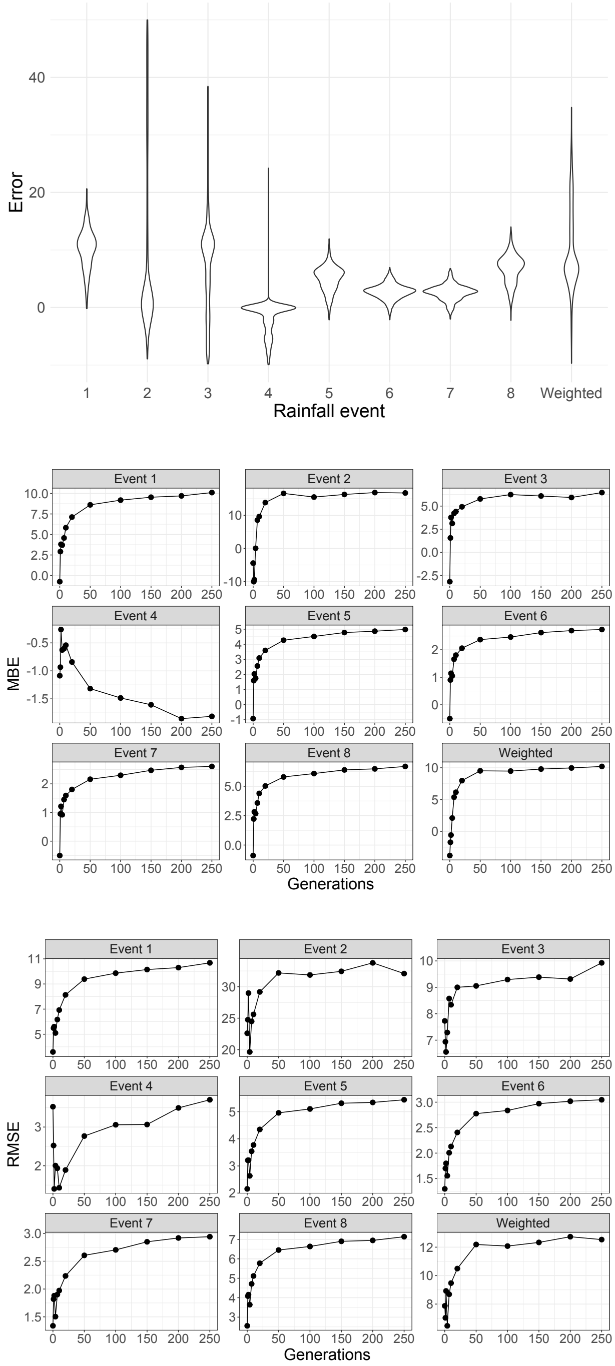


FIGURE 4.5: (a) Distribution of errors of percentage overflow reduction for the 8 rainfall events for the solutions obtained at the 250th generation. (b) Mean bias error (MBE) and (c) root mean square error (RMSE) of percentage overflow reduction for test set (0th generation) and solutions obtained during 1st, 2nd, 4th, 7th, 10th, 20th, 50th, 100th, 150th, 200th and 250th generation of surrogate-assisted optimization (R-EO-20) using emulator with n = 250 and q = 7.

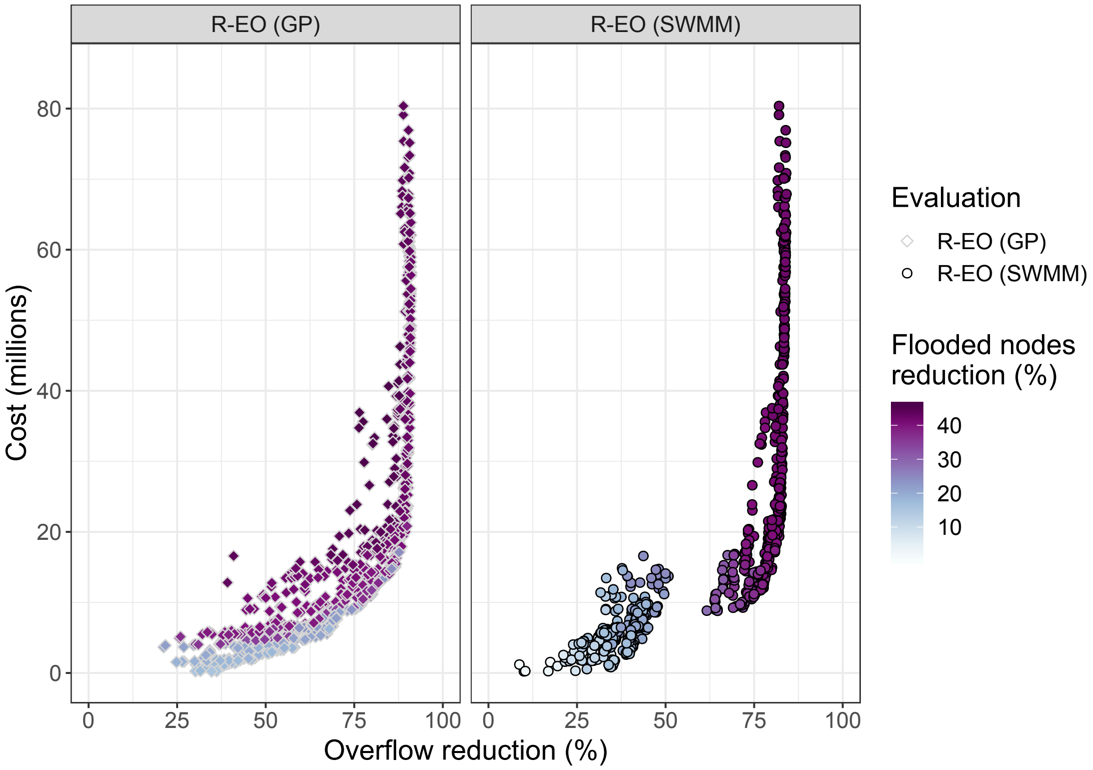


FIGURE 4.6: 765 Pareto-efficient solutions from R-EO. The three objectives are represented by the axes (weighted overflow reduction and cost) and color (weighted reduction in the number of peak flooded nodes). Solutions are evaluated using the emulator (grey border points, left) during optimization and evaluated using the SWMM simulator (black border points, right) post optimization.

solutions less than $5 million (where the kink is) has mean bias error (MBE) of 2.3% and solutions more than $5 million has MBE of 4.8%. For R-EO-20, the less expensive solutions to the left of the gap have MBE of 15.6% and the expensive solutions have MBE 5.7%. This suggests that the high dimensionality of the optimization problem has partially prevented the emulator to find low cost solutions that are truly Pareto-efficient. It is likely that solutions to the right of the gap are closer to the true Pareto front. Solutions between $10 to $18 million that have overflow reduction greater than 61% are the most important to decision makers since they are low cost solutions that have markedly better performance at alleviating overflow.

Comparison with simulation-optimization

Fig. 4.7 illustrates the Pareto-efficient solutions obtained from R-EO-20 and DS-SO-20, that are evaluated using SWMM. We observe that solutions from R-EO-20 clearly outperforms solutions from DS-SO-20 when the cost of solutions is above $10 million. This is due to the better selection and optimization of decision variables as multiple rainfall events are included during the sensitivity analysis and optimization steps for

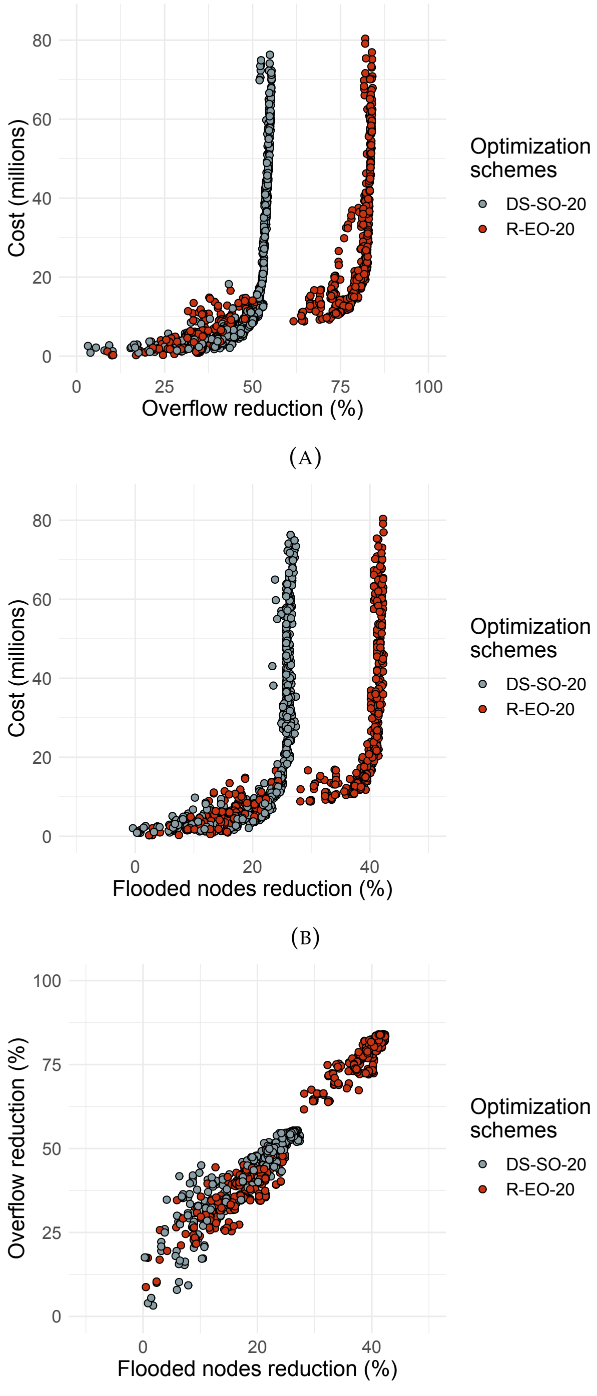
R-EO-20. Notice that above $10 million, the gap between R-EO-20 and DS-SO-20 remains constant. Since investments are directed to LID implementation when cost goes beyond $10 million, it means that LID does not contribute to the advantage that R-EO-20 solutions have. The optimal pipe configuration attained at the $10 million mark for R-EO-20 is responsible for the gap we see between the the two sets of solutions. When investment cost falls below $10 million, solutions from both optimization schemes are comparable. Although one reason for this observation is due to the emulator inaccuracy, it also indicates that when investment budget is limited and expansion of pipes must be more selective, it is harder to manage overflow reduction for all the different rainfall events since overflow occurs at different locations when rainfall characteristics differ.

Fig. 4.8 compares R-EO-20 and DS-SO-20 according to the three performance metrics, hypervolume, ϵ-indicator and generational distance. For all three measures, R-EO-20 outperforms DS-SO-20. We also see that the metrics improve rapidly at earlier generations and become roughly constant in later generations, similar to Fig. 3.12 for the 5-D problem. However, for the 5-D problem, improvement stops at around the 15th generation, but for the 20-D problem, improvement stops at around the 50th generation. This suggests that for a surrogate-assisted optimization, the frequency of emulator update should depend on the number of dimensions of the problem.

Finally, we compare the time needed for DS-SO-20 and R-EO-20 in terms of simulation runs required. Table 4.3 shows the number of simulation runs required for the different steps of the framework. Note that for sensitivity analysis, there can potentially be fewer simulation runs for DS-SO-20 if we apply the clustering algorithm on the drainage system introduced in Section 4.2.1. Although we do not run simulation-optimization on multiple rainfall events for the 20 newly selected decision variables (R-SO-20) due to the high computation time required, we still present the number of simulations that would be required. The values are calculated assuming that the same setup as R-EO-20 for the evolutionary algorithm is used for the simulation-optimization. We see that R-EO-20 calls the simulator 12 times less than DS-SO-20 and 91 times less than R-SO-20. While one simulation run in SWMM takes about 5 to 10 seconds, given the number of iterations required for this framework, this translate to high time savings of at least 600 hours compared to DS-SO-20. For more complex catchment model, simulation for a single rainfall event can take up to minutes, computation savings promised by surrogate models will be even greater in such cases.

4.3.3 Robustness analysis

Fig. 4.9 shows the cumulative distribution of performance (i.e., overflow reduction and peak flooded nodes reduction) across the 16 random rainfall events used for robustness analysis for all Pareto-efficient solutions obtained from R-EO-20 and DS-SO-20. Note that the cumulative distribution here refers to weights, defined in Eq. 3.10, and not probability, such that rainfall events with larger flood or higher frequency have higher weights. Each line in the plot represents a solution and a solution has better performance if it lies as far to the left as possible; this is also why we see expensive solutions (in green-blue) lying closer to the left. For example, if a line passes through the point



(A)

(B)

(C)

FIGURE 4.7: Comparison of Pareto-efficient solutions from DS-SO-20 and R-EO-20 in terms of (a) cost against weighted overflow reduction; (b) cost against weighted peak flooded nodes reduction; and (c) weighted overflow reduction against weighted peak flooded nodes reduction.

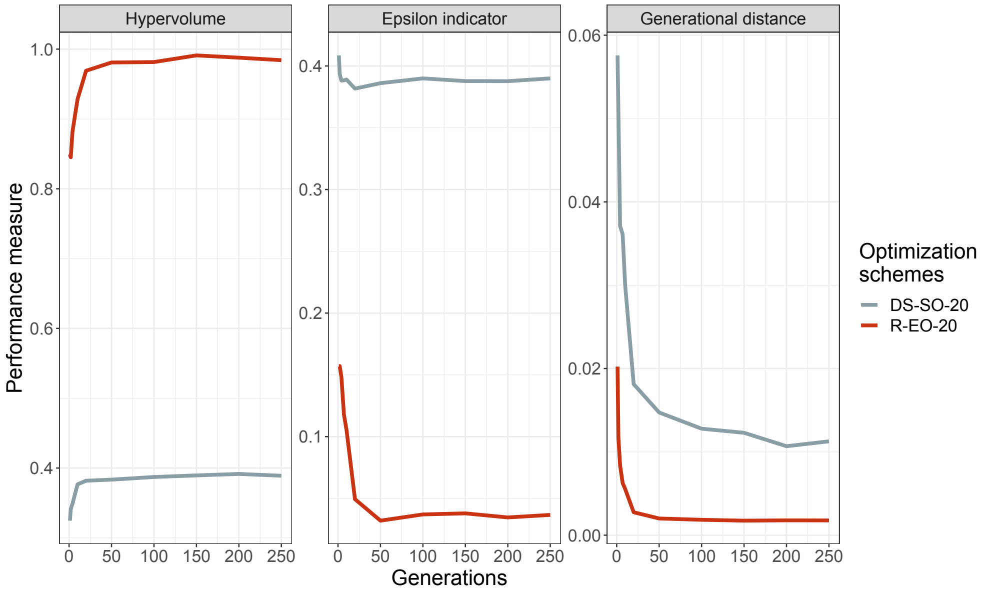


FIGURE 4.8: Hypervolume, ϵ-indicator, and generational distance of solution set as optimization progresses for DS-SO-20 and R-EO-20.

TABLE 4.3: Comparison of simulations required for DS-SO-20, R-EO-20, and R-SO-20

	DS-SO-20	R-EO-20	R-SO-20
Sensitivity analysis	32,000	39,000	39,000
Emulation	0	250	0
Optimization	500,000	0	4,000,000
Post-evaluation	0	6,120	0
Total	532,000	44,370	4,039,000

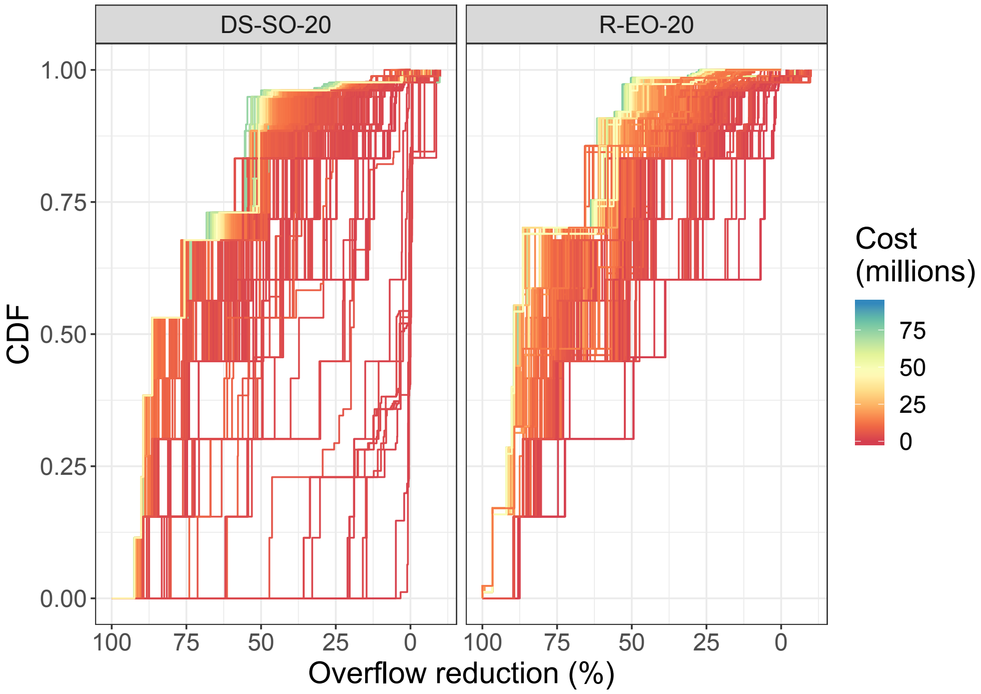
(50%, 0.6), it can be interpreted as a solution achieving a reduction greater than or equal to 50% for 0.6 (in terms of accumulated weights) of the random rainfall events.

Solutions obtained from R-EO-20 are more robust against a wide range of rainfall characteristics as seen from the lines in Fig. 4.9 lying closer to the left compared to solutions from DS-SO-20. There are solutions from DS-SO-20 that are Pareto-efficient for the design storm but underperform when evaluated for random rainfall events. These solutions corresponds to the lines lying in the lower right diagonal of the CDF plots. Such lines are not present for the CDF plots for R-EO-20, indicating that using multiple rainfall events in both sensitivity analysis and optimization step improves robustness of solutions.

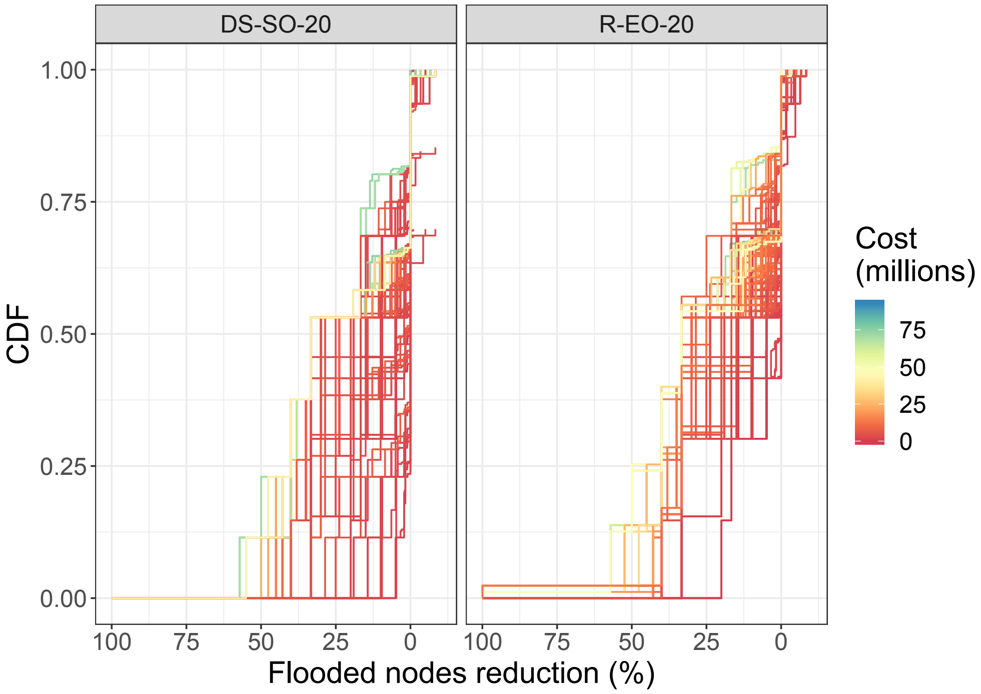
While the CDF plots provide an overview of how solutions perform for various rainfall events, they do not reflect the type of rainfall events that solutions underperform in. Table 4.4 shows the hypervolume of the Pareto front for DS-SO-20 and R-EO-20 for the two sets of random rainfall events sampled. The hypervolume is calculated for each of the 8 rainfall events and also for the weighted objectives. The rainfall events are arranged in order of increasing severity of overflow caused, hence allowing us to find out in which rainfall events solutions underperform in. We see that hypervolume of solutions obtained by R-EO-20 is higher for all cases, with the exception of Rainfall Event 1 which is the event most similar to the design storm. We also observe that solutions obtained from DS-SO-20 perform better in rainfall events similar to the design storm (e.g. Event 3 and 8) but performance degrades as rainfall characteristics deviates from characteristics of the design storm. Solutions from DS-SO-20 performs the worst for Rainfall Event 4, which is the smallest rainfall event, as they fail to eliminate the small overflow, unlike solutions from R-EO-20. This highlights the importance of considering not just the design storm but also rainfall events of a range of intensities, duration, and profiles in the design of robust urban drainage systems.

4.4 Conclusion

Current practice of using design storms may not necessarily yield robust urban drainage systems. While including stochastic rainfall events into the design process and tackling the increased computational demand with surrogate-assisted optimization can be a feasible alternative, the scalability of emulator comes into question when the problem is a high dimensional one. In this chapter, we investigate the suitability of an urban drainage emulator and its application in surrogate-assisted optimization for a high dimensional problem. In particular, we compare between two design frameworks: the design storm method and our proposed robust design method. The design storm method reflects current design practices. It relies on the design storm for sensitivity analysis and optimization and uses simulation-optimization to obtain optimal drainage designs. Our proposed method considers stochastic rainfall events in both sensitivity analysis and optimization and uses surrogate-assisted optimization to obtain optimal drainage design within reasonable computational time. The application of the framework on the Nhieu Loc-Thi Nghe basin reveals the following insights:



(A)



(B)

FIGURE 4.9: Cumulative distribution for (a) overflow reduction and (b) flooded nodes reduction for the 16 random rainfall events selected. Note that the x-axis is inverted so that it proceeds in the direction of degrading performance from left to right.

TABLE 4.4: Hypervolume of solutions from DS-SO-20 and R-EO-20 for random rainfall event set 1 (left) and set 2 (right)

Set 1	DS-SO-20	R-EO-20
Event 4	0	1.01
Event 2	0.63	1
Event 3	0.99	1
Event 1	1.01	0.96
Event 8	0.88	1.11
Event 5	0.82	1.1
Event 6	0.51	1.04
Event 7	0.71	1.12
Weighted	0.7	1

Set 2	DS-SO-20	R-EO-20
Event 4	0	1.01
Event 2	0.99	1
Event 3	0.19	1
Event 1	0.93	0.99
Event 8	0.91	1.01
Event 5	0.84	1.09
Event 6	0.63	1.06
Event 7	0.66	1.01
Weighted	0.52	0.7

Note that 2 random rainfall events are selected from each of the 8 clusters, which give rise to the 2 set of events.

Note that the reference set to calculate hypervolume is the best known solution set found across the three optimization schemes for the weighted objectives. Hence for the individual rainfall events, the solution set from the different optimization scheme may be better than this reference set, hence hypervolume can exceed 1 in such cases.

- The urban drainage emulator designed for the higher dimensional input space has higher error overall than its counterpart for the 5-D problem but its prediction of overflow for some rainfall events has improved. This means that while the entire response surface is harder to fit for the higher dimensional space, some regions of the response surface may become easier to fit. Emulation error increases by a greater extent during surrogate-assisted optimization for the higher dimensional problem, suggesting the greater potential of updating emulators for such problems.

- For the higher dimensional problem, expensive solutions produced by the proposed method outperforms the solutions from the design storm method but the less expensive solutions from the two methods are comparable. One reason is that prediction error is high for low cost solutions due to the high dimensionality but it also suggests that a minimum investment budget is needed to effectively curb overflow in different rainfall events.

- The proposed method yields urban drainage systems that are more robust than the design storm method. When evaluated under rainfall events with different intensities, duration, and profiles, the performance gap between solutions obtained from the two methods extends further as conditions deviate from those of the design storm. This is true in both directions of deviation, when rainfall events are smaller or larger than the design storm.

- The computational requirements of the proposed method is sharply reduced by the use of emulators. It calls the simulator 12 times less than design storm based method, translating to high time savings of at least 600 hours.

Overall, this study demonstrates that our proposed method is a feasible way to design robust urban drainage systems. Considering stochastic rainfall events during sensitivity analysis allows for a more appropriate selection of decision variables that target overflow reduction for different rainfall events. Including these rainfall events during optimization allows the search to identify optimal solutions that perform well across these events. While computation time is increased as a result of the additional rainfall events considered, surrogate-assisted optimization proves to be a viable alternative to simulation-optimization. Since our surrogate-assisted optimization adopts the basic sequential framework, emulation error increases during the optimization search. An area for future research is to adopt other frameworks, such as the metamodel-embedded evolution framework or approximation uncertainty based framework, so that through emulator update or greater exploration of decision space, solutions obtained can be closer to the true Pareto front. To provide greater computation time savings, we can also look into the use of emulators in sensitivity analysis, which has already been demonstrated in literature (Borgonovo, Castaings, and Tarantola, 2012; Qian et al., 2018; Nagel, Rieckermann, and Sudret, 2020). Last but not least, climate change can drive rainfall behaviours to become more extreme than what is considered here. It will also be worthwhile to adapt the proposed framework to design urban drainage systems that are robust under the deep uncertainty of climate change and other external drivers, such as urban development.

Chapter 5

Conclusions

5.1 Summary

In this thesis, we set out to develop a methodology that is able to design robust optimal urban drainage systems. To do so, we first identify the potential flaws of existing urban drainage design methods that rely on design storms (Chapter 2). Then, we propose our method that introduces stochastic rainfall events and surrogate-assisted optimization in the design process. We test the new method on a smaller dimensional problem in Chapter 3. Finally, we scale up our proposed framework to a higher dimensional problem and study its feasibility in Chapter 4. Applying our design framework to a real world case study, the Nhieu Loc-Thi Nghe basin in Ho Chi Minh City, Vietnam, allows us to gather insights on the practicality of our methods.

In Chapter 2, we begin by using a 4-step framework to identify optimal urban drainage solutions and evaluate the robustness of solutions. The first two steps of the framework consists of sensitivity analysis to identify decision variables for optimization and then simulation-optimization to seek Pareto-efficient configurations of those decision variables. These two steps depend on a design storm, which follows current design practices. Solutions obtained are hence optimal for the design storm. In the second half of the framework, we then evaluate these solutions under rainfall events of various intensities, duration, and profiles and then assess the robustness of solutions against different rainfall conditions. From our experiments, we find that none of the drainage solutions obtained are robust, as their performance for the design storm cannot always be replicated in other rainfall events. Solutions underperform in two types of rainfall events: 1) less intense but longer rainfall events that cause more severe floods, and 2) small, yet frequent rainfall events. In both cases, rainfall conditions deviate from those of the design storm, and the drainage components chosen and optimized for the design storm are no longer appropriate to alleviate flood in these different rainfall events. We also find that between the two components of urban drainage (pipes and LIDs), expanding pipes is more effective at reducing overflow and potentially increasing robustness of a drainage system. From the work in this chapter, we understand the potential flaws that design based on design storm may have and we thus propose a method to overcome the challenges in the subsequent chapter.

In Chapter 3, we propose our method to design robust optimal urban drainage systems by making two major changes to the framework in Chapter 2. First, instead of depending on design storm, we introduce stochastic rainfall events into the optimization step. However, this translates to an increase in computational demand as more simulation runs are needed. To overcome this, we thus construct a GP-based emulator to model and predict the overflow conditions and then employ the emulator in the surrogate-assisted optimization to obtain drainage solutions that perform well in multiple rainfall events. Since this is our first attempt at using an emulator, we consider a smaller optimization problem with a 5-dimensional decision space. We find that the emulator is generally more accurate with increasing training points and basis functions, but accuracy does not appear to correlate with the optimality of the Pareto front. Comparing our proposed framework with the design storm method, we realize that including stochastic rainfall events during optimization improves robustness but the extent of improvement is limited by the choice of decision variables. This prompts the use of stochastic rainfall events in the sensitivity analysis step when choosing decision variables for optimization. Although the consideration of multiple rainfall events increase computational demand, we show that surrogate-assisted optimization is a feasible alternative as it is 7 times faster than simulation optimization that uses design storm and 129 times faster than the robust simulation optimization setup that uses stochastic rainfall events.

In Chapter 4, we scale up the framework in Chapter 3 by applying it to a 20-dimensional problem. We find that emulator error increases slightly for the higher dimensional problem, but this error can become larger as the search progresses during the surrogate-assisted optimization. In the framework in this chapter, we also introduce stochastic rainfall events into sensitivity analysis, an earlier step of the design framework, so that decision variables selected are aimed at overflow reduction for multiple rainfall events. The proposed framework here relies on stochastic rainfall events for the entire design process and is compared to the method that relies on design storm only. We successfully demonstrate that our proposed method yields urban drainage systems that are more robust than the design storm method. When evaluated under rainfall events with different intensities, duration, and profiles, the performance gap between solutions obtained from the two methods extends further as conditions deviate from those of the design storm.

5.2 Future works

In this thesis, we contribute a design framework that is capable of finding robust optimal drainage systems. Although we demonstrate its effectiveness, we also notice many areas for future improvements. First, exploring other emulation approaches for the urban drainage model may yield different results. Here, we adopt a GP-based emulator, but there are also other approaches, such as radial basis functions, artificial neural networks, and polynomial chaos expansion, which can be applied (Razavi, Tolson, and Burn, 2012). Besides data-driven approaches, including mechanistic knowledge, based on the sets of ordinary or partial differential equations of the simulation model, into the emulator is also a possibility, albeit an onerous one (Machac, Reichert, and Albert, 2016; Carbajal et al., 2017; Bermúdez et al., 2018). Second, it will also be interesting to include the emulator into sensitivity analysis or perhaps to make use of simulations runs during sensitivity analysis to construct the emulator. Sensitivity analysis, like optimization, is also a time-consuming process that requires iterated calls of the simulator, hence using an emulator at this stage could provide further computational savings. Third, improving the surrogate-assisted optimization is perhaps the most likely way to yield better solutions. Here, we adopt the basic sequential framework for surrogate-assisted optimization. Updating the emulator or making use of the uncertainty of emulation for greater exploration are two ways to improve the optimization (Di Pierro et al., 2009; Yazdi and Neyshabouri, 2014; Akhtar and Shoemaker, 2016). However, they come along with difficult choices such as choosing the frequency and method of updating the emulator, or choosing which solution to be evaluated by the original function during the optimization.

In terms of application, an interesting research avenue is to design urban drainage systems robust under changing climate and population growth. In our work, we only consider the range of current rainfall conditions. Since drainage systems usually last more than decades, it will be worthwhile to consider climate-driven changes in rainfall behaviour and adapt our proposed framework to factor in such changes. Our framework could be extended to account for deep uncertainty (McPhail et al., 2018), so as to find urban drainage designs that are robust to changing rainfall conditions and other external drivers. Another application is in the real time control of drainage systems, which is also an computationally intensive task requiring iterated simulation runs that can benefit from the use of an emulator.

Appendix A

Additional details on sensitivity analysis

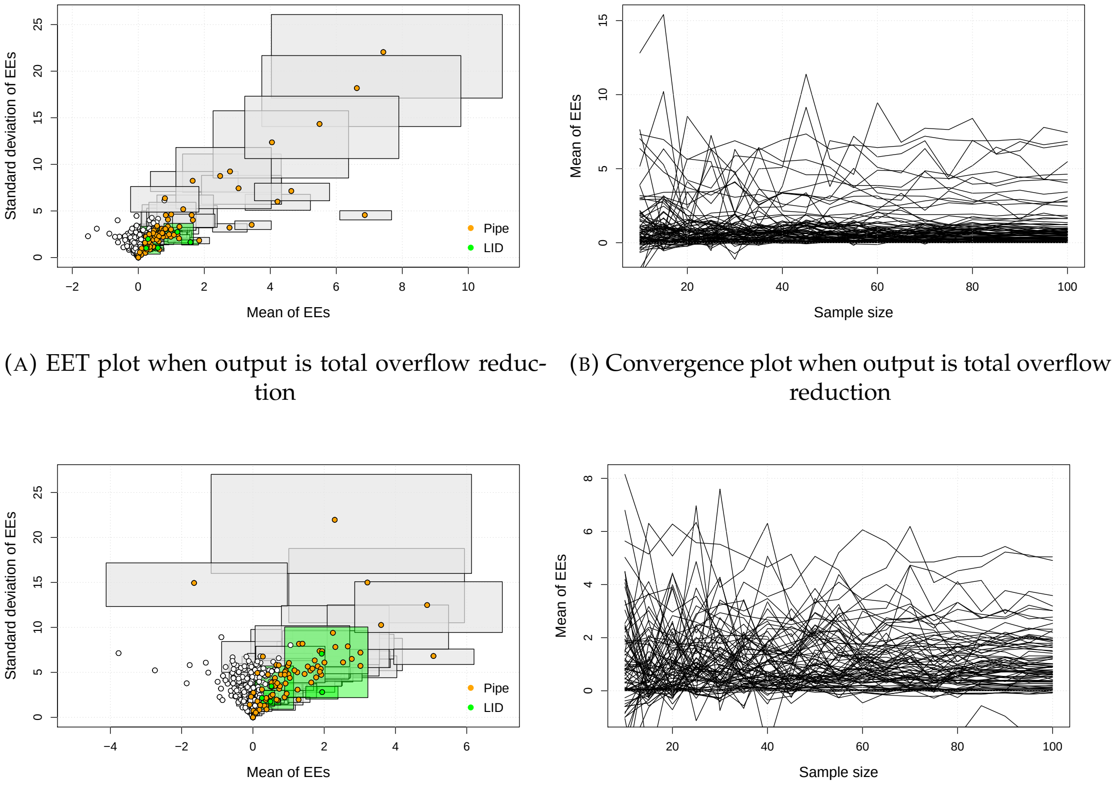
We use two sensitivity analysis (SA) methods to determine which decision variables should be considered in the design process. The two methods are elementary effect test (EET) and the extended Fourier amplitude sensitivity test (eFAST). We originally considered 3 outputs for each SA method: reduction in total overflow (sum of overflow over all nodes and across all time instances), reduction in peak overflow (maximum overflow over all nodes at a single instance), and reduction in local overflow (maximum total overflow over a single node). However, we eventually found that the output local overflow reduction is not very informative since most of the time it just refers to the reduction in total overflow at one particular node. Hence, we only use total overflow reduction and peak overflow reduction in selecting decision variables during SA.

EET is first performed on the initial 320 input factors—308 pipe variables and 12 LID variables. The sampling strategy of EET builds r trajectories in the input space, each consisting of $M+1$ points, such that one EE per factor is evaluated for each trajectory (Morris, 1991). Here, r = 100 and M = 320. Since the EET randomly selects the starting point of a trajectory, the pipes and LIDs are expanded or implemented to 50% of the maximum value on average. As a result, overflow is evaluated over a catchment that has a highly expanded drainage network and high LID implementation. This is unrealistic, since it will be very costly to apply such a drainage solution. To overcome this issue, we first generate 1,000 random points in the input space, and then select 10% of the cheapest solutions to be the starting points of the trajectories. Fig. A.1 shows the results for this round of EET, including the convergence plot of the sensitivity estimates, calculated using the R version of the SAFE toolbox (Pianosi, Sarrazin, and Wagener, 2015). EET is the most suitable for screening to discard non-influential input factors. Hence, we only select inputs that have a positive 95% lower one-sided bound for mean of EEs for either output (total overflow reduction or peak overflow reduction). This ensures that we are 95% confident that these inputs will reduce either peak overflow or total overflow. The 78 selected pipes and 8 selected LIDs are highlighted in the plots.

After screening, both EET and eFAST are performed on this reduced set of input factors (as it is advisable to apply more than one SA, when possible, to reinforce the conclusions drawn from individual SA). Similarly, only 10% of the cheapest solutions are chosen as starting points for the trajectories in EET. Fig. A.2 shows the eFAST results and Fig. A.3 shows the EET results. For each output, factors are ranked according to their contribution to the overflow reduction (based on mean of EE for EET and on total order index for eFAST). A factor is selected if it is ranked in the top 20 in both eFAST and EET for the same output as this shows consistency across different SA. 12 pipes are selected from this criterion (they are highlighted in Fig. A.2 and Fig. A.3). Only 1 of the LIDs is selected using this criterion; yet, all 8 LIDs are included eventually in the optimization, because we want to maintain the ratio between pipe and LID variables so as to find out if their performances change for different rain events.

Fig. A.4 shows the 12 selected pipes in the drainage network and also indicates the 3 sub-catchments in which the LIDs are applied to. Please refer to Table 2.2 and 2.3 for description of these selected pipes and LIDs. Upon inspecting these selected variables, we noticed that these pipes that contribute most to overflow reduction are near overflow regions or are connected to pipes of larger diameters. Of the 12 original LID variables, those that are not selected mainly lie at the eastern subcatchment of the Nhieu Loc-Thi Nghe (NL-TN) basin. This is because pipe expansion around that subcatchment can more effectively reduce overflow compared to reducing runoff from that subcatchment using LIDs.

From the SA, we also notice that total overflow reduction and peak overflow reduction are correlated, hence only total overflow reduction is used as an objective in the simulation-optimization. We also use the reduction in total number of flooded nodes, rather than local overflow reduction, as an objective to get an indication of the spatial distribution of the flood events.



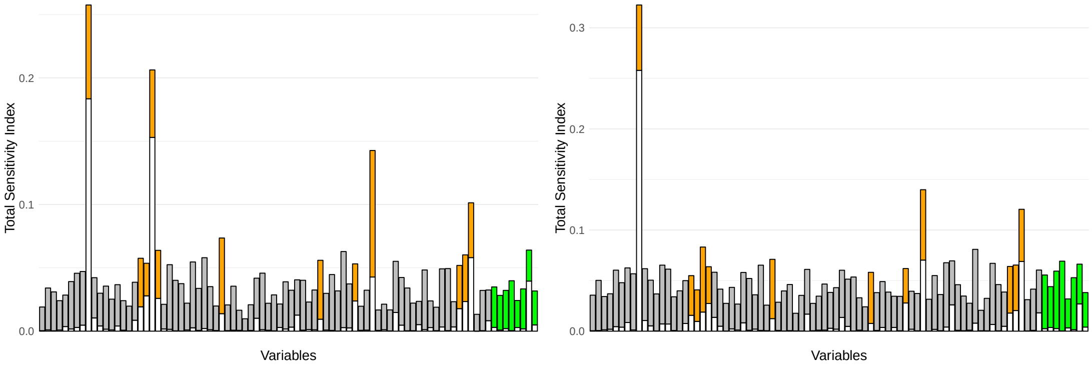
(A) EET plot when output is total overflow reduction

(B) Convergence plot when output is total overflow reduction

(C) EET plot when output is peak overflow reduction

(D) Convergence plot when output is peak overflow reduction

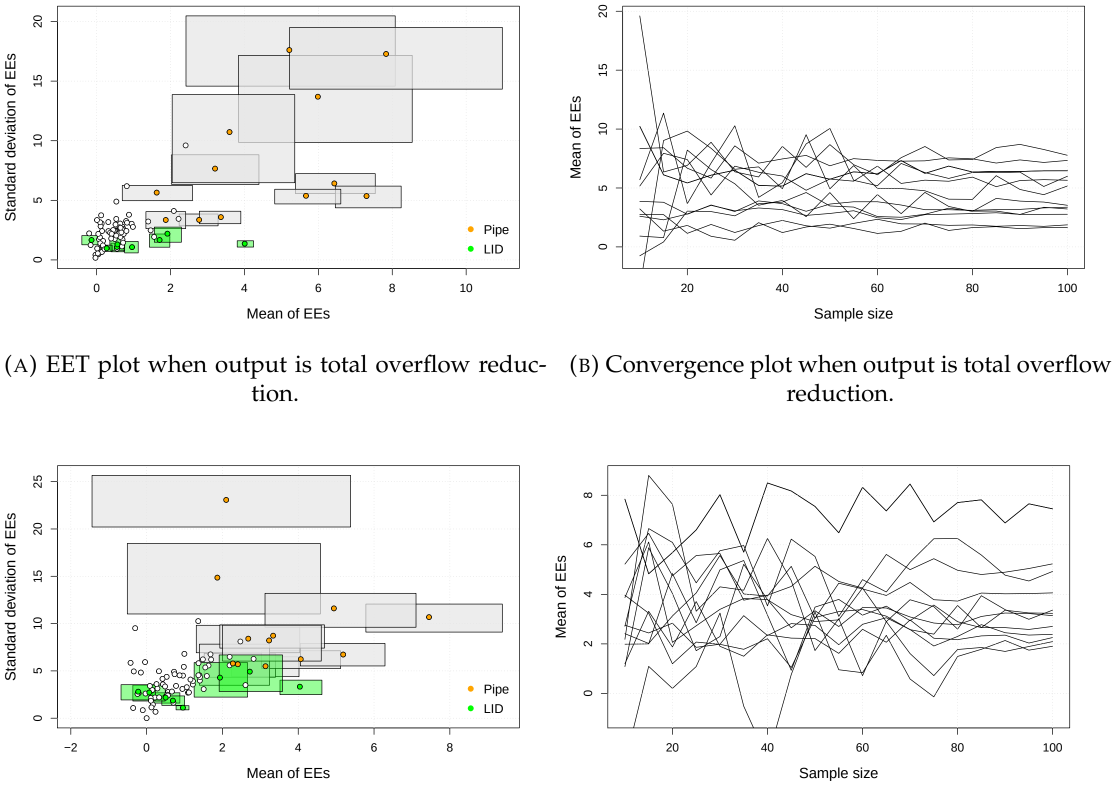
FIGURE A.1: Average of EEs against their standard deviation when output is (a) total overflow reduction and (c) peak overflow reduction. Each point represents one input factor. The larger the mean EE, the more influential the factor is. The larger the standard deviation of EE, the more interactions that input has with other inputs. The confidence bounds derived via bootstrapping is represented by the boxes around the point. The confidence bounds is only shown for the selected variables. Convergence plot of mean of EEs against sample size when output is (b) total overflow reduction and (d) peak overflow reduction. Mean of EEs should converge when sample size is large to indicate that the sensitivity estimate is independent of sample size. For clarity, only the selected pipes are shown in the convergence plot. Since the mean EEs still fluctuate at large sample size, it is less suitable to rank the inputs at this stage, hence only screening is done to remove inputs that are most certainly not able to reduce overflow. From this first round of EET, 78 pipes (in orange) and 8 LIDs (in green) are selected.



(A) Sensitivity of total overflow

(B) Sensitivity of peak overflow

FIGURE A.2: Bar plots of total sensitivity indices for each of the 78 pipe inputs and 8 LID inputs. White bar represents the main effect of the input on the output while the colored bar represents the effect on the output due to interaction with other inputs. 12 selected pipes are highlighted in orange while the 8 LIDs are highlighted in green.



(A) EET plot when output is total overflow reduction.

(B) Convergence plot when output is total overflow reduction.

(C) EET plot when output is peak overflow reduction.

(D) Convergence plot when output is peak overflow reduction.

FIGURE A.3: EET and convergence plots for the second round of EET that uses a reduced set of variables. Mean of EEs tend to converge at large sample size in contrast to the first EET performed. 12 selected pipes are highlighted in orange while the 8 LIDs are highlighted in green.

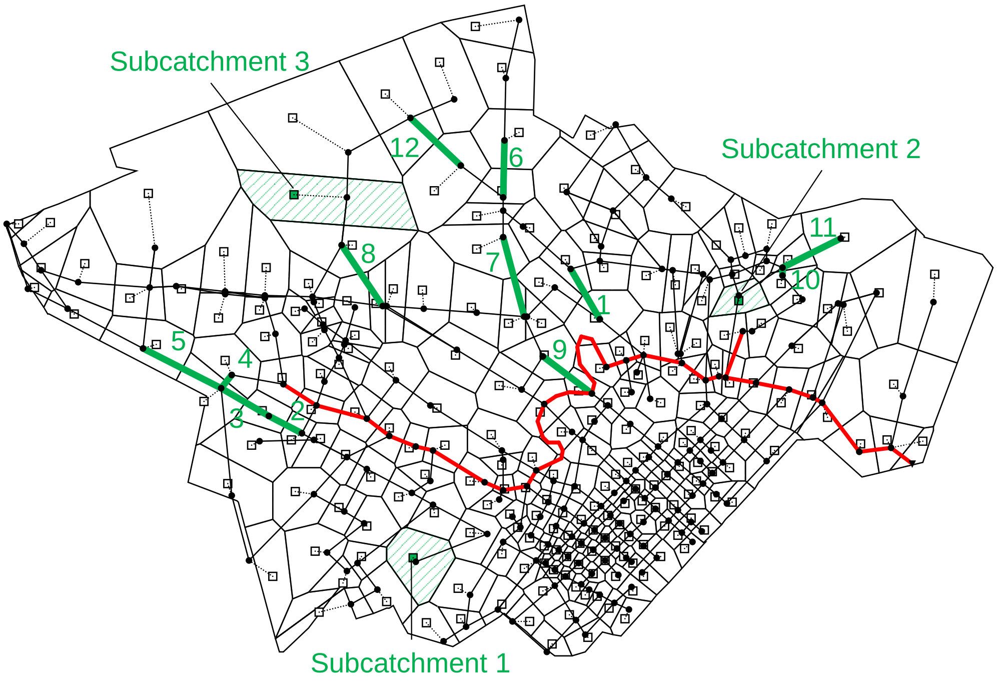


FIGURE A.4: Decision variables selected by the sensitivity analysis.

Appendix B

Additional details on preliminary simulation-optimization

A preliminary simulation-optimization is performed prior to the one described in Section 2.3.2. The main difference stands in the decision variables used. In this preliminary round, decision variables are not selected using sensitivity analysis. Pipe variables are instead selected based on the duration with which they reach full capacity. We select 16 pipe variables following this rule as they are at full capacity for more than 3 hours since the beginning of the design storm. Optimization experiments are performed independently using 10 initial random seeds: 5 experiments use 250 generations and population of 200 individuals; the other 5 experiments use 500 generations and a population of 100 individuals—totalling 50,000 function evaluations. The objectives and other parameters of the optimization are the same as those described in Section 2.3.2.

After obtaining the results for this preliminary simulation-optimization, we realize that there may be a better way to select decision variables for optimization. This is when we use SA as described in the Section 2.3.1. To compare the effects of using a different set of decision variables, optimization is performed again on the set of decision variables selected through SA with the same settings as in the preliminary simulation-optimization. Fig. B.1 shows the Pareto front obtained using the two different sets of decision variables. Using the set of decision variables selected from SA, the maximum overflow reduction is 58.8%. However, using the preliminary set of decision variables only gives a maximum of 29.6% overflow reduction, despite the similar investment costs. This demonstrates the importance of SA in choosing the right variables for the optimization. This is especially important in large watersheds, where there could be many variables that could be used in the optimization process. We learn that the variables chosen from SA are not necessarily those that reach full capacity during the storm, because overflow can occur when only one end of the pipe is full. Thus, it is more beneficial to expand these pipes than those in the preliminary simulation-optimization.

To evaluate the search quality and progress of the simulation-optimization, three performance measures are used. They are the hypervolume indicator, generational distance, and additive ϵ-indicator (Reed et al., 2013). These metrics capture the convergence of the solution set (found by the optimization) with respect to a reference set, which is usually the true Pareto front. When the true Pareto front is unknown, the best known solution set found across all trials of the optimization is used as the reference set. Hypervolume measures the volume of objective space dominated by a solution set.

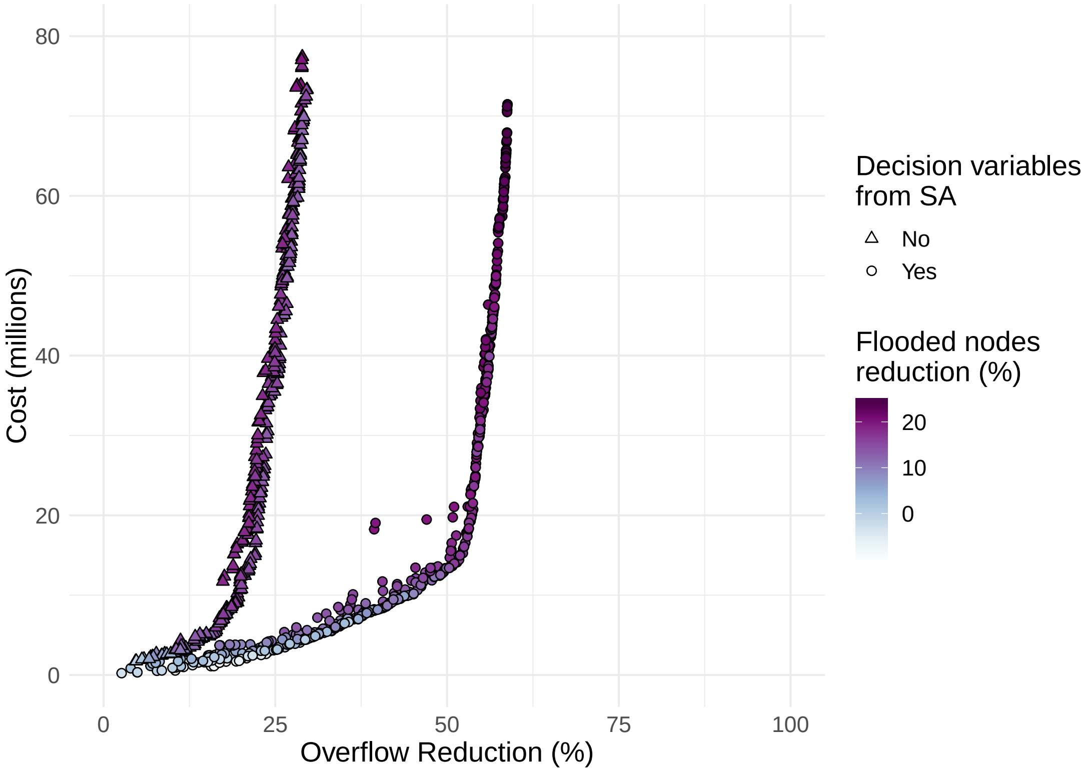
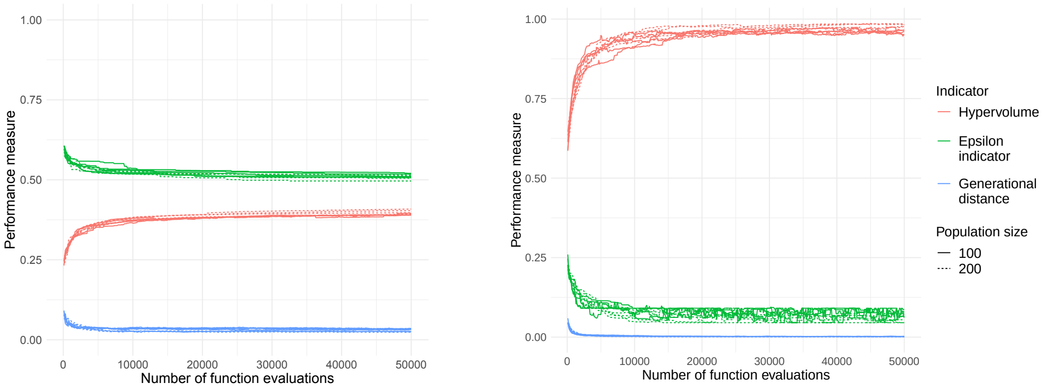


FIGURE B.1: Pareto front of simulation-optimization using the preliminary set of decision variables (triangles) and using the decision variables selected by SA (circles). Using decision variables selected by SA roughly doubles the overflow reduction for the same level of investments, indicating the importance of SA to select variables prior to optimization.

Generational distance measures the average Euclidean distance of each point in the solution set to its closest point in the reference set. Additive ϵ-indictor measures the worst case distance (in contrast to the average distance) between the solution and reference set, and is thus especially sensitive to gaps in the solution set. Fig. B.2 shows how the three performance measures evolve as the optimization algorithm progresses for each generation. The optimization which uses decision variables selected from SA performs better than the preliminary optimization across all metrics, once again demonstrating the importance of the SA. The optimization experiments using a population size of 200 also outperform the experiments with a population size of 100 in most instances. The values of the three metrics also do not fluctuate when nearing 50,000 function evaluations, implying that no further improvements can be made in the solution set even if the algorithm proceeds further. Therefore, in the eventual setup described in the main paper, all 10 experiments have population size 200 and 50,000 function evaluations.



(A)

(B)

FIGURE B.2: Hypervolume, generational distance, and additive ϵ-indicator of solution set as optimization progresses for (a) preliminary simulation-optimization and (b) simulation-optimization using decision variables selected from SA. Preliminary simulation-optimization experiments perform poorly, especially in terms of hypervolume and additive ϵ-indicator, which are more sensitive to gaps in the solution set.

Appendix C

Additional results for modified sensitivity analysis

Here, we present additional results and figures for the modified sensitivity analysis described in Section 4.2.1. After performing hierarchical clustering on the nodes of the drainage system for each rainfall event, we need to select the number of clusters. The key is that if there are too few clusters, then we are less able to reduce the number of simulations in the modified sensitivity analysis. Conversely, if there are too many clusters, then the assumption that pipe changes in one cluster do not affect overflow in another cluster is harder to hold true. We first select the number of clusters such that the maximize size of a cluster does not exceed 20% of all nodes. Following which, we find the dendrogram height (i.e. the distance metric between the clusters for the particular agglomeration) associated with that number of clusters. We then use the maximum dendrogram height across the 8 rainfall events to be the cutoff point to merge additional clusters. We also look at the Davies-Bouldin Index (DBI; a lower DBI is more ideal) and ensure that DBI is low for the selected number of clusters (Fig. C.1). The number of clusters for the 8 rainfall events are 9, 7, 9, 21, 22, 16, 17, and 14 respectively, shown in Fig. C.2 and C.3.

We then perform modified sensitivity analysis using only inputs from the ’important’ clusters identified (Section 4.2.1). We also perform a full sensitivity analysis using all 320 input factors for the 8 rainfall events to compare our results. Fig. C.4 and C.5 show the results for the full and modified sensitivity analysis. We see that in most instances, the top ranking variables are the same for both sensitivity analyses, showing that the hierarchical clustering method together with the modified sensitivity analysis is a feasible approach to reduce simulations runs.

Finally, we take the union of the set of best performing pipe variables for each rainfall events to obtain the 12 pipe variables used for optimization. These pipe variables are shown in the map in Fig. C.6.

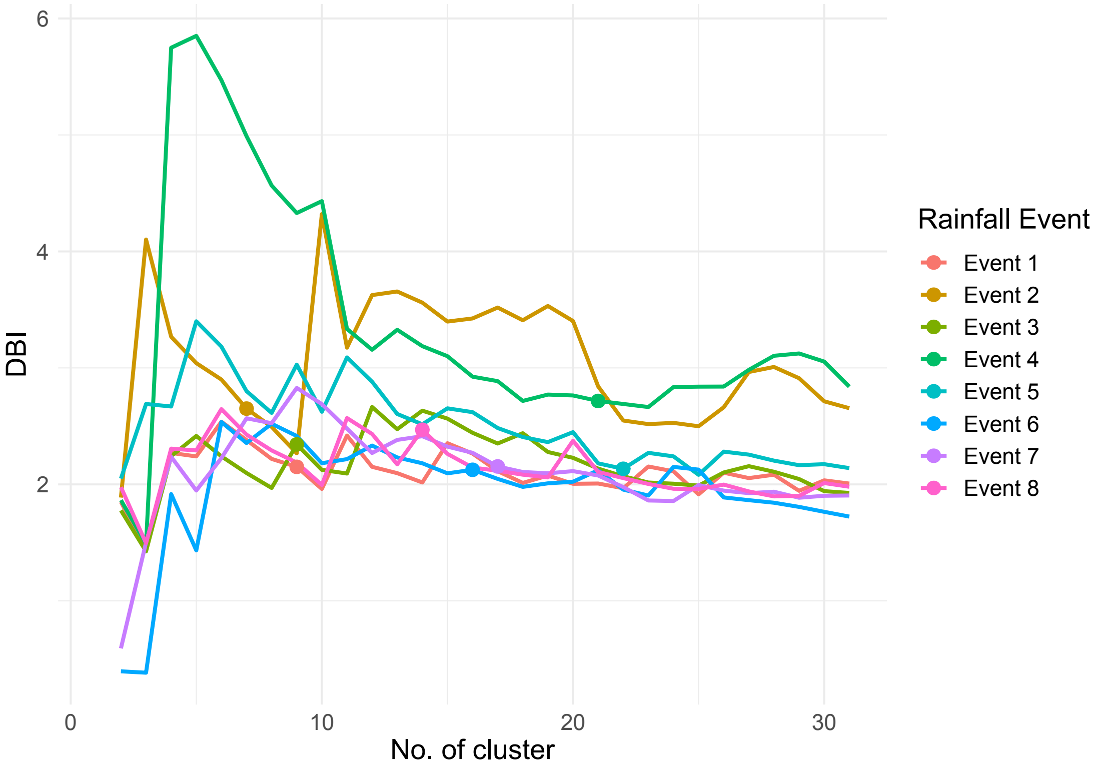


FIGURE C.1: Davies-Bouldin Index (DBI) for hierachical clustering of drainage system for the 8 rainfall events. Selected number of clusters for each event is indicated by the circle on the line.

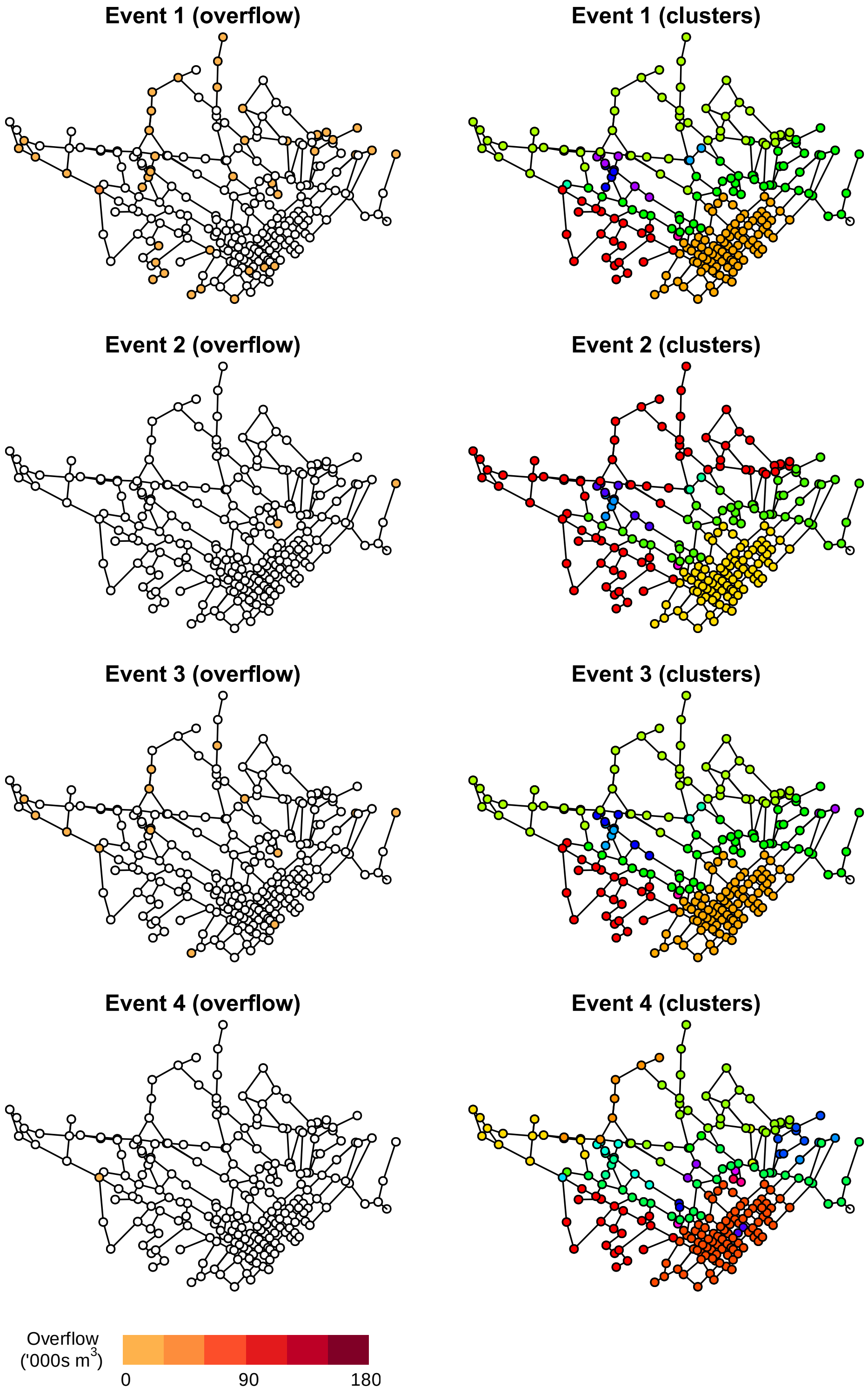


FIGURE C.2: Overflow volume (left) and hierarchical clustering of nodes (right) for Rainfall Events 1-4.

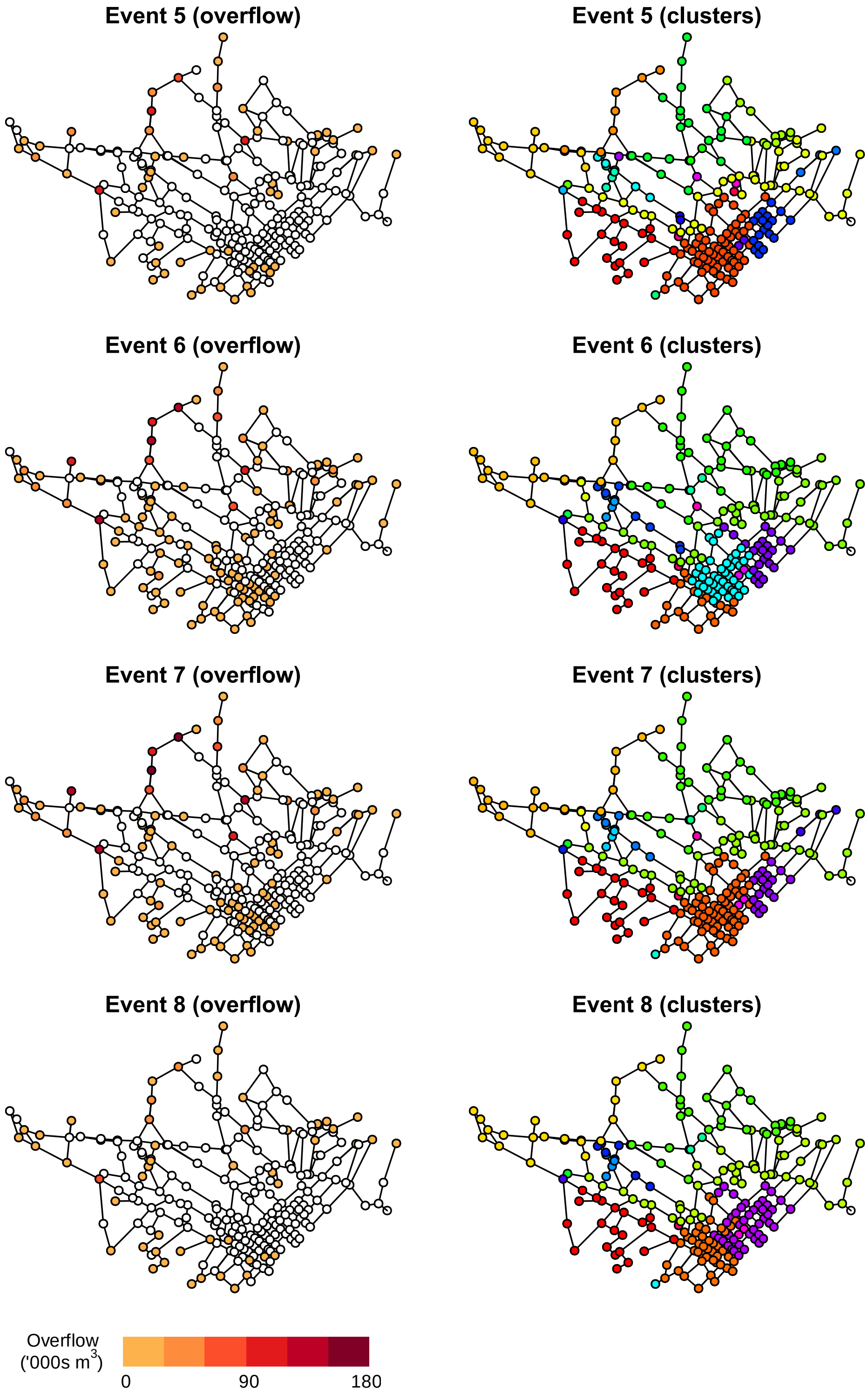


FIGURE C.3: Overflow volume (left) and hierarchical clustering of nodes (right) for Rainfall Events 5-8.

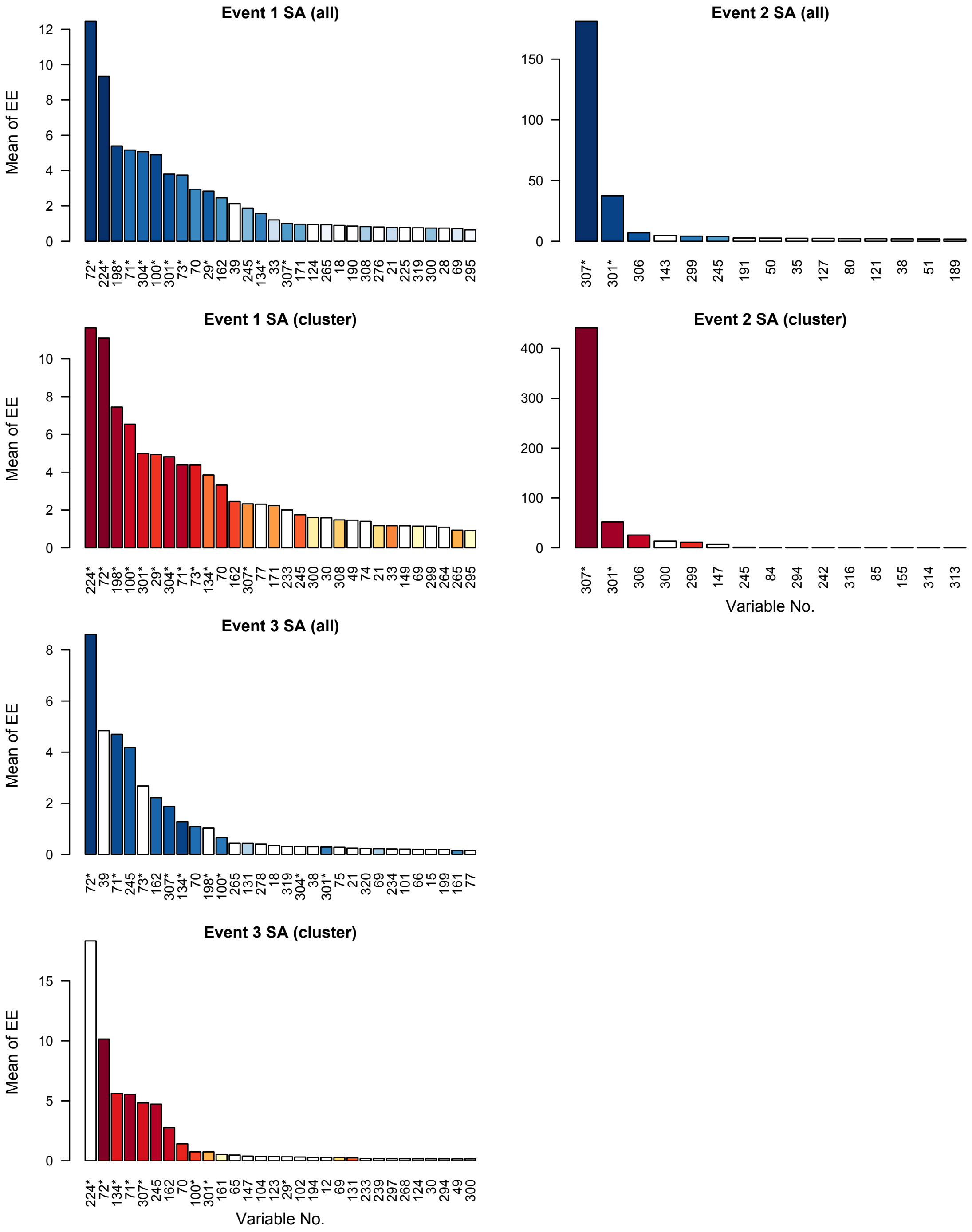


FIGURE C.4: Mean elementary effects (EE) of variables shown in descending order for top 30 variables (or for variables with mean EE > 0) for Rainfall Event 1-3. Plots in blue refer to the full sensitivity analysis performed using all 320 input factors while plots in red refer to the modified sensitivity analysis. A darker shade of the blue (red) colored bar represents a higher rank of the variable in the modified (full) sensitivity analysis. Selected variables are marked by an asterisk.

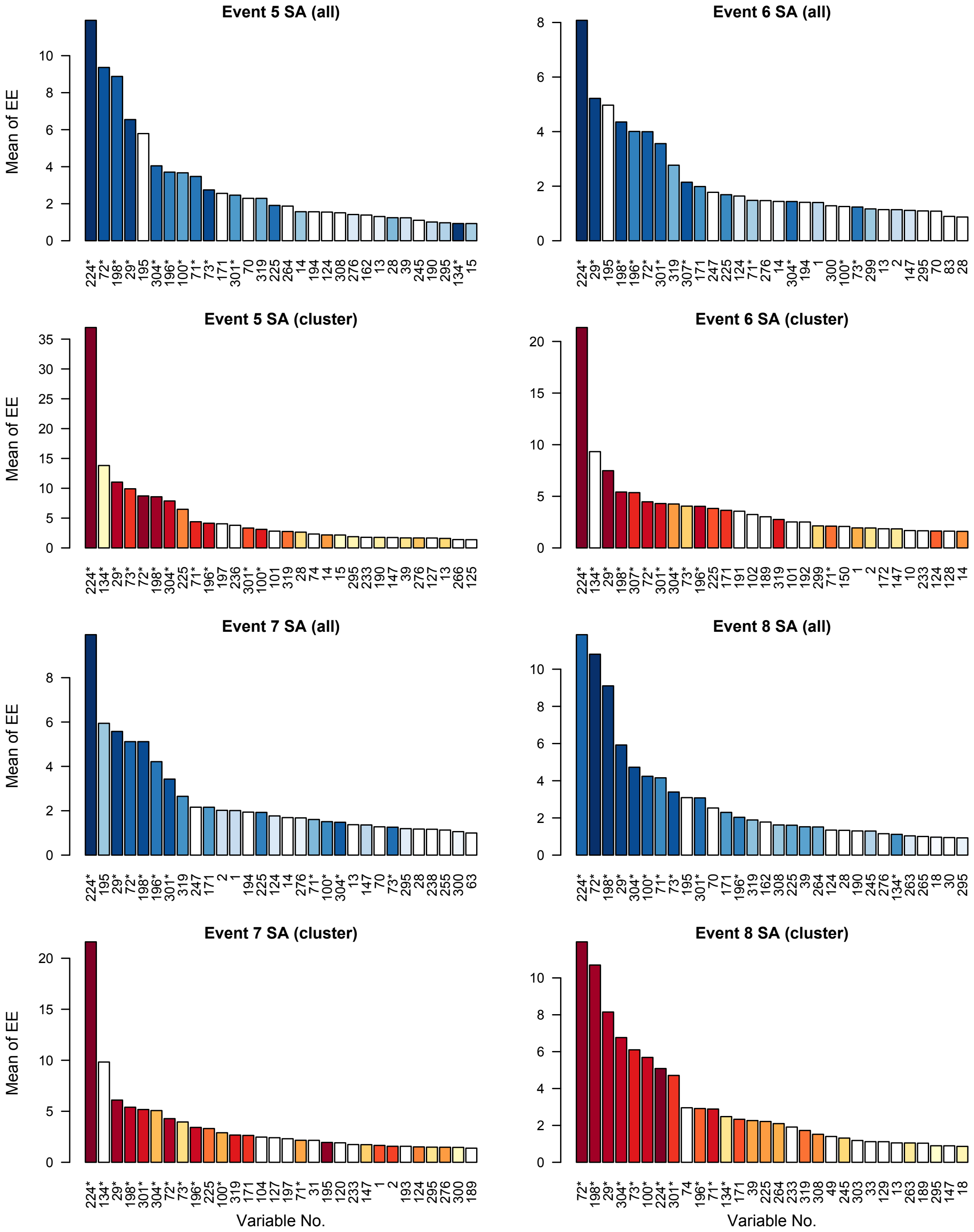


FIGURE C.5: Mean elementary effects (EE) of variables shown in descending order for top 30 variables for Rainfall Event 5-8. Results for Rainfall Event 4 is not plotted since only one variable gives positive mean EE.

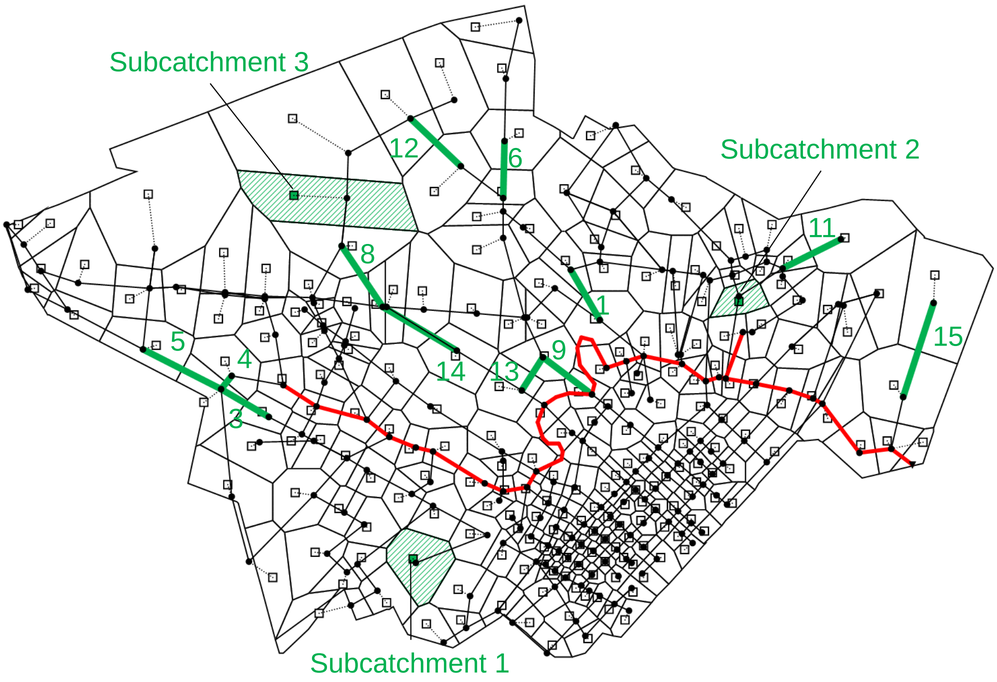


FIGURE C.6: Decision variables selected by the modified sensitivity analysis. Pipes 13-15 are new additions which are not selected previously when sensitivity analysis was performed using design storm only.

Bibliography

Adams, Barry J and Charles DD Howard (1986). “Design storm pathology”. In: *Canadian Water Resources Journal* 11.3, pp. 49–55.

Ahiablame, Laurent M, Bernard A Engel, and Indrajeet Chaubey (2012). “Effectiveness of low impact development practices: literature review and suggestions for future research”. In: *Water, Air, & Soil Pollution* 223.7, pp. 4253–4273.

Akhtar, Taimoor and Christine A Shoemaker (2016). “Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection”. In: *Journal of Global Optimization* 64.1, pp. 17–32.

Antoulas, Athanasios C (2005). “An overview of approximation methods for large-scale dynamical systems”. In: *Annual reviews in Control* 29.2, pp. 181–190.

Bach, Peter M et al. (2014). “A critical review of integrated urban water modelling–Urban drainage and beyond”. In: *Environmental Modelling & Software* 54, pp. 88–107.

Benner, Peter, Serkan Gugercin, and Karen Willcox (2015). “A survey of projection-based model reduction methods for parametric dynamical systems”. In: *SIAM review* 57.4, pp. 483–531.

Beraud, Benoit et al. (2010). “Optimisation of sewer networks hydraulic behaviour during wet weather: coupling genetic algorithms with two sewer networks modelling tools”. In: *NOVATECH 2010*.

Berkhin, Pavel (2006). “A survey of clustering data mining techniques”. In: *Grouping multidimensional data*. Springer, pp. 25–71.

Bermúdez, María et al. (2018). “Development and comparison of two fast surrogate models for urban pluvial flood simulations”. In: *Water Resources Management* 32.8, pp. 2801–2815.

Borgonovo, Emanuele, William Castaings, and Stefano Tarantola (2012). “Model emulation and moment-independent sensitivity analysis: An application to environmental modelling”. In: *Environmental Modelling & Software* 34, pp. 105–115.

Broad, Darren Ross, Graeme Clyde Dandy, and Holger R Maier (2005). “Water distribution system optimization using metamodels”. In: *Journal of Water Resources Planning and Management* 131.3, pp. 172–180.

Broad, DR, Holger R Maier, and Graeme Clyde Dandy (2010). “Optimal operation of complex water distribution systems using metamodels”. In: *Journal of Water Resources Planning and Management* 136.4, pp. 433–443.

Burian, Steven J et al. (1999). “Historical development of wet-weather flow management”. In: *Journal of Water Resources Planning and Management* 125.1, pp. 3–13.

Butler, David and John Davies (2000). *Urban drainage*. London: CRC Press.

Cain, Louis P (1972). “Raising and watering a city: Ellis Sylvester Chesbrough and Chicago’s first sanitation system”. In: *Technology and Culture* 13.3, pp. 353–372.

Carbajal, Juan Pablo et al. (2017). “Appraisal of data-driven and mechanistic emulators of nonlinear simulators: The case of hydrodynamic urban drainage models”. In: *Environmental Modelling & Software* 92, pp. 17–27.

Carpenter, Donald D and Preethi Kaluvakolanu (2011). “Effect of roof surface type on storm-water runoff from full-scale roofs in a temperate climate”. In: *Journal of Irrigation and Drainage Engineering* 137.3, pp. 161–169.

Castelletti, A et al. (2010). “A multiobjective response surface approach for improved water quality planning in lakes and reservoirs”. In: *Water Resources Research* 46.6.

Castelletti, Andrea et al. (2012a). “A general framework for dynamic emulation modelling in environmental problems”. In: *Environmental Modelling & Software* 34, pp. 5–18.

Castelletti, Andrea et al. (2012b). “Data-driven dynamic emulation modelling for the optimal management of environmental systems”. In: *Environmental Modelling & Software* 34, pp. 30–43.

Code of Practice on Surface Water Drainage (2011). 6th. Public Utilities Board Singapore. Singapore. URL: `$https://www.pub.gov.sg/Documents/COP_Final.pdf$`.

Dajani, Jarir S and Yakir Hasit (1974). “Capital cost minimization of drainage networks”. In: *Journal of the Environmental Engineering Division* 100.2, pp. 325–337.

Damodaram, Chandana and Emily M Zechman (2013). “Simulation-optimization approach to design low impact development for managing peak flow alterations in urbanizing watersheds”. In: *Journal of Water Resources Planning and Management* 139.3, pp. 290–298.

Davies, David L and Donald W Bouldin (1979). “A cluster separation measure”. In: *IEEE transactions on pattern analysis and machine intelligence* 2, pp. 224–227.

Deb, Kalyanmoy et al. (2002). “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197.

Di Matteo, Michael, Graeme C Dandy, and Holger R Maier (2017). “Multiobjective optimization of distributed stormwater harvesting systems”. In: *Journal of Water Resources Planning and Management* 143.6, p. 04017010.

Di Pierro, Francesco et al. (2009). “Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms”. In: *Environmental Modelling & Software* 24.2, pp. 202–213.

Drainage Design Guide (2018). Florida Department of Transportation. Tallahassee, FL.

Eldred, Michael and Daniel Dunlavy (2006). “Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models”. In: *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 7117.

Fen, Chiu-Shia, Chencha Chan, and Hsien-Chie Cheng (2009). “Assessing a response surface-based optimization approach for soil vapor extraction system design”. In: *Journal of Water Resources Planning and Management* 135.3, pp. 198–207.

Fletcher, Tim D, Herve Andrieu, and Perrine Hamel (2013). “Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art”. In: *Advances in Water Resources* 51, pp. 261–279.

Fletcher, Tim D et al. (2015). “SUDS, LID, BMPs, WSUD and more–The evolution and application of terminology surrounding urban drainage”. In: *Urban Water Journal* 12.7, pp. 525–542.

Forrester, Alexander, Andras Sobester, and Andy Keane (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

Galelli, Stefano, Andrea Castelletti, and Albert Goedbloed (2015). "High-Performance Integrated Control of water quality and quantity in urban water reservoirs". In: *Water Resources Research* 51.11, pp. 9053–9072.

Genest, Christian and Anne-Catherine Favre (2007). "Everything you always wanted to know about copula modeling but were afraid to ask". In: *Journal of Hydrologic Engineering* 12.4, pp. 347–368.

Giuliani, Matteo and Andrea Castelletti (2016). "Is robustness really robust? How different definitions of robustness impact decision-making under climate change". In: *Climatic change* 135.3-4, pp. 409–424.

Goel, NK et al. (2000). "A derived flood frequency distribution for correlated rainfall intensity and duration". In: *Journal of Hydrology* 228.1-2, pp. 56–67.

Guo, Yufeng, Godfrey Walters, and Dragan Savic (2008). "Optimal design of storm sewer networks: Past, present and future". In: *Proceedings of the 11th International Conference on Urban Drainage, Edinburgh, Scotland, UK*. Vol. 31.

Gupta, Hoshin V et al. (2009). "Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling". In: *Journal of hydrology* 377.1-2, pp. 80–91.

Herman, Jonathan D et al. (2014). "Beyond optimality: Multistakeholder robustness tradeoffs for regional water portfolio planning under deep uncertainty". In: *Water Resources Research* 50.10, pp. 7692–7713.

Herman, Jonathan D et al. (2015). "How should robustness be defined for water systems planning under change?" In: *Journal of Water Resources Planning and Management* 141.10, p. 04015012.

Ho, Huu Loc et al. (2015). "Exploratory Assessment of SUDS Feasibility in Nhieu Loc-Thi Nghe Basin, Ho Chi Minh City, Vietnam". In: *British Journal of Environment & Climate Change* 5.2, pp. 91–103.

Huff, Floyd A (1990). *Time distributions of heavy rainstorms in Illinois*. Circular no. 173. Champaign, IL: Illinois State Water Survey.

Jin, Yaochu (2011). "Surrogate-assisted evolutionary computation: Recent advances and future challenges". In: *Swarm and Evolutionary Computation* 1.2, pp. 61–70.

Jones, Donald R, Matthias Schonlau, and William J Welch (1998). "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4, pp. 455–492.

Kasprzyk, Joseph R et al. (2013). "Many objective robust decision making for complex environmental systems undergoing change". In: *Environmental Modelling & Software* 42, pp. 55–71.

Kellagher, R (2013). *Rainfall runoff management for developments*. Report SC30219. Bristol: Environment Agency.

Knowles, Joshua (2006). "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems". In: *IEEE Transactions on Evolutionary Computation* 10.1, pp. 50–66.

Le Quiniou, M, P Mandel, and L Monier (2014). "Optimization of drinking water and sewer hydraulic management: Coupling of a genetic algorithm and two network hydraulic tools". In: *Procedia Engineering* 89, pp. 710–718.

Legendre, Pierre and Loic FJ Legendre (2012). *Numerical ecology*. 3rd English Edition. Amsterdam: Elsevier.

Lempert, Robert et al. (2013). *Ensuring robust flood risk management in Ho Chi Minh City*. The World Bank.

Libisch-Lehner, C P et al. (2019). "On the value of ENSO state for urban water supply system operators: opportunities, trade-offs, and challenges". In: *Water Resources Research* 55.4, pp. 2856–2875.

Loeppky, Jason L, Jerome Sacks, and William J Welch (2009). "Choosing the sample size of a computer experiment: A practical guide". In: *Technometrics* 51.4, pp. 366–376.

Löwe, Roland et al. (2018). "Simulating flood risk under non-stationary climate and urban development conditions Experimental setup for multiple hazards and a variety of scenarios". In: *Environmental Modelling & Software* 102.C, pp. 155–171.

Machac, David, Peter Reichert, and Carlo Albert (2016). "Emulation of dynamic simulators with application to hydrology". In: *Journal of Computational Physics* 313, pp. 352–366.

Machac, David et al. (2016). "Fast mechanism-based emulator of a slow urban hydrodynamic drainage simulator". In: *Environmental Modelling & Software* 78, pp. 54–67.

Maharjan, M et al. (2009). "Staged cost optimization of urban storm drainage systems based on hydraulic performance in a changing environment". In: *Hydrology and Earth System Sciences Discussions* 5.3, pp. 1479–1509.

Mahmoodian, Mahmood et al. (2018a). "A Data-Driven Surrogate Modelling Approach for Acceleration of Short-Term Simulations of a Dynamic Urban Drainage Simulator". In: *Water* 10.12, p. 1849.

Mahmoodian, Mahmood et al. (2018b). "A hybrid surrogate modelling strategy for simplification of detailed urban drainage simulators". In: *Water resources management* 32.15, pp. 5241–5256.

Maier, Holger R et al. (2014). "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions". In: *Environmental Modelling & Software* 62, pp. 271–299.

Mays, Larry W and Harry G Wenzel Jr (1976). "Optimal design of multilevel branching sewer systems". In: *Water Resources Research* 12.5, pp. 913–917.

Mays, Larry W and Ben Chie Yen (1975). "Optimal cost design of branched sewer systems". In: *Water Resources Research* 11.1, pp. 37–47.

McPhail, C et al. (2018). "Robustness Metrics: How Are They Calculated, When Should They Be Used and Why Do They Give Different Results?" In: *Earth's Future* 6.2, pp. 169–191.

Moreno-Rodenas, Antonio M et al. (2018). "A dynamic emulator for physically based flow simulators under varying rainfall and parametric conditions". In: *Water research* 142, pp. 512–527.

Morris, Max D (1991). "Factorial sampling plans for preliminary computational experiments". In: *Technometrics* 33.2, pp. 161–174.

Mulvaney, T J (1851). "On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment". In: *Proceedings of the institution of Civil Engineers of Ireland* 4, pp. 19–31.

Nadarajah, Saralees, Emmanuel Afuecheta, and Stephen Chan (2018). "A compendium of copulas". In: *Statistica* 77.4, pp. 279–328.

Nagel, Joseph B, Jörg Rieckermann, and Bruno Sudret (2020). “Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: Application to urban drainage simulation”. In: *Reliability Engineering & System Safety* 195, p. 106737.

Nicklow, John et al. (2009). “State of the art for genetic algorithms and beyond in water resources planning and management”. In: *Journal of Water Resources Planning and Management* 136.4, pp. 412–432.

Ogidan, Olufunso S and Marcio Giacomoni (2017). “Enhancing the Performance of a Multiobjective Evolutionary Algorithm for Sanitary Sewer Overflow Reduction”. In: *Journal of Water Resources Planning and Management* 143.7, p. 04017023.

Pianosi, Francesca, Fanny Sarrazin, and Thorsten Wagener (2015). “A Matlab toolbox for global sensitivity analysis”. In: *Environmental Modelling & Software* 70, pp. 80–85.

Pianosi, Francesca et al. (2016). “Sensitivity analysis of environmental models: A systematic review with practical workflow”. In: *Environmental Modelling & Software* 79, pp. 214–232.

Ponweiser, Wolfgang et al. (2008). “Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection”. In: *International Conference on Parallel Problem Solving from Nature*. Springer, pp. 784–794.

Qian, Elizabeth et al. (2018). “Multifidelity Monte Carlo estimation of variance and sensitivity indices”. In: *SIAM/ASA Journal on Uncertainty Quantification* 6.2, pp. 683–706.

Quinn, Julianne D et al. (2018). “Exploring How Changing Monsoonal Dynamics and Human Pressures Challenge Multireservoir Management for Flood Protection, Hydropower Production, and Agricultural Water Supply”. In: *Water Resources Research* 54.7, pp. 4638–4662.

Rasmussen, Carl Edward and Christopher KI Williams (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA.

Rathnayake, Upaka (2015). “Enhanced water quality modelling for optimal control of drainage systems under SWMM constraint handling approach”. In: *Asian Journal of Water, Environment and Pollution* 12.2, pp. 81–85.

Razavi, Saman, Bryan A Tolson, and Donald H Burn (2012). “Review of surrogate modeling in water resources”. In: *Water Resources Research* 48.7.

Reed, Patrick M et al. (2013). “Evolutionary multiobjective optimization in water resources: The past, present, and future”. In: *Advances in Water Resources* 51, pp. 438–456.

Regis, Rommel G and Christine A Shoemaker (2007). “A stochastic radial basis function method for the global optimization of expensive functions”. In: *INFORMS Journal on Computing* 19.4, pp. 497–509.

— (2009). “Parallel stochastic global optimization using radial basis functions”. In: *INFORMS Journal on Computing* 21.3, pp. 411–426.

Rossman, Lewis A (2015). *Storm water management model user's manual, version 5.1*. National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency Cincinnati.

Saltelli, Andrea, Stefano Tarantola, and KP-S Chan (1999). “A quantitative model-independent method for global sensitivity analysis of model output”. In: *Technometrics* 41.1, pp. 39–56.

Salvadori, G and C De Michele (2007). “On the use of copulas in hydrology: theory and practice”. In: *Journal of Hydrologic Engineering* 12.4, pp. 369–380.

Sample, David J (2013). *Best Management Practice, Fact Sheet 1-15.* Virginia Cooperative Extension.

Schmitter, Petra et al. (2016). “Effect of catchment-scale green roof deployment on stormwater generation and reuse in a tropical city”. In: *Journal of Water Resources Planning and Management* 142.7, p. 05016002.

Schneller, GO and GP Sphicas (1983). “Decision making under uncertainty: Starr’s domain criterion”. In: *Theory and Decision* 15.4, pp. 321–336.

Starr, Martin Kenneth (1963). *Product design and decision theory*. Prentice-Hall.

Stormwater Drainage Manual (2018). 5th. Drainage Services Department HKSAR. Hong Kong.

Sun, Yabin et al. (2019). “Deriving intensity–duration–frequency (IDF) curves using downscaled in situ rainfall assimilated with remote sensing data”. In: *Geoscience Letters* 6.1, p. 17.

Tabatabaei, Mohammad et al. (2015). “A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods”. In: *Structural and Multidisciplinary Optimization* 52.1, pp. 1–25.

Urban Stormwater Management Manual for Malaysia (2012). 2nd. Department of Irrigation and Drainage Malaysia. Kuala Lumpur.

USEPA (1999). *Collection Systems O&M Fact Sheet. Trenchless Sewer Rehabilitation.* EPA 832-F-99-032. United States Environmental Protection Agency.

Viet Nam News. *Comprehensive plan needed to prevent flooding in HCMC*. Last accessed on December 2018. URL: `https://vietnamnews.vn/society/481504/comprehensive-plan-needed-to-prevent-flooding-inhcmc.html`.

VietnamPlus. *Ho Chi Minh City seriously flooded due to storm Usagi*. Last accessed on November 2018. URL: `https://en.vietnamplus.vn/ho-chi-minh-city-seriously-flooded-due-to-storm-usagi/142506.vnp`.

Wang, Qi et al. (2014). “Two-objective design of benchmark problems of a water distribution system via MOEAs: Towards the best-known approximation of the true Pareto front”. In: *Journal of Water Resources Planning and Management* 141.3, p. 04014060.

Wang, Qi et al. (2018). “Comparison of Multiobjective Optimization Methods Applied to Urban Drainage Adaptation Problems”. In: *Journal of Water Resources Planning and Management* 144.11, p. 04018070.

Wust, Sébastien, Jean-Claude Bolay, and Thai Thi Ngoc Du (2002). “Metropolization and the ecological crisis: precarious settlements in Ho Chi Minh City, Vietnam”. In: *Environment and Urbanization* 14.2, pp. 211–224.

Xu, M et al. (2010). “Real-time control of combined surface water quantity and quality: polder flushing”. In: *Water science and technology* 61.4, pp. 869–878.

Xu, Min, Peter-Jules van Overloop, and NC Van de Giesen (2013). “Model reduction in model predictive control of combined water quantity and quality in open channels”. In: *Environmental modelling & software* 42, pp. 72–87.

Yazdi, J, EH Lee, and JH Kim (2014). “Stochastic multiobjective optimization model for urban drainage network rehabilitation”. In: *Journal of Water Resources Planning and Management* 141.8, p. 04014091.

Yazdi, J and SAA Salehi Neyshabouri (2014). “Adaptive surrogate modeling for optimization of flood control detention dams”. In: *Environmental Modelling & Software* 61, pp. 106–120.

Yu, Jianjun et al. (2017). “Stochastic Optimization Model for Supporting Urban Drainage Design under Complexity”. In: *Journal of Water Resources Planning and Management* 143.9, p. 05017008.

Zhang, Jingwen et al. (2017). “Assessing the weighted multi-objective adaptive surrogate model optimization to derive large-scale reservoir operating rules with sensitivity analysis”. In: *Journal of hydrology* 544, pp. 613–627.

Zheng, Feifei et al. (2016). “Comparison of the searching behavior of NSGA-II, SAMODE, and Borg MOEAs applied to water distribution system design problems”. In: *Journal of Water Resources Planning and Management* 142.7, p. 04016017.

Zou, Rui, Wu-Seng Lung, and Jing Wu (2009). “Multiple-pattern parameter identification and uncertainty analysis approach for water quality modeling”. In: *Ecological Modelling* 220.5, pp. 621–629.