Storm Sewer Network Optimization for Urban Flood Mitigation by Coupling Multi-objective Evolutionary Algorithms (MOEAs) and PCSWMM

by

Dexin Chen

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Water Resources Engineering

Department of Civil and Environmental Engineering

University of Alberta

ABSTRACT

Flooding is one of the most frequently met disasters in urban areas in the context of climate change and more intensive anthropogenic activities. Urban drainage system (UDS), defined as surface runoff and sewage collection and transport system, is an essential part of urbanization. The capacity of UDS can substantially influence the flooding levels of urban catchments. However, there are always bottle necks in the complex sewer network that substantially affect the capacity of UDS and thus worsen urban flooding.

To improve the performance of UDS, multi-objective evolutionary algorithms (MOEAs) have been applied to optimize UDS, as they can explore trade-offs between conflicting objectives. However, most previous studies only conducted pipe size optimization in a small-scale area (with less than 100 sewers) without considering pipe slope and engineering criteria. This thesis focuses on urban stormwater drainage system and aims to develop and evaluate a method for simultaneously optimizing sewer size and slope in a large-scale area (with 2930 sewers). The goal is to minimize the sewer rehabilitation/upgrade costs and flood volume in the complex in real-life storm sewer network.

To realize the goal, a new storm sewer network optimization system was proposed that integrated a storm water management model (PCSWMM) with one of MOEAs Preference-inspired coevolutionary algorithm (PICEA-g) using a programming language (MATLAB) as the platform. Specific improvements were made to the PICEA-g algorithm to better tackle the problem, which include new methodologies for initializing candidate solutions, the use of divide and conquer technique, enhanced goal vector boundaries and fitness calculation, and enhanced genetic operators (crossover and mutation).

The new optimization system was tested in a small sample area (16.68 ha and 78 conduits) and applied to the entire study area (777.6 ha and 2930 conduits) based on both 1D and 2D hydraulic modeling of the UDS. Sewer pipe size and slope were used as decision variables, and then the number of flooded nodes (manholes) and cost were taken as objective functions. The results of the storm sewer network before and after the optimization were compared and discussed, indicating a significant improvement by using the optimization system. For the test area, the flooded nodes reduced from 15 to 0 after optimization. For the study area, the flooded manholes decreased from 284 to 115 in the 1D model; when using the 2D model, the flooded nodes dropped from 73 to 30 and 44 in two different modeling scenarios. This comparison indicated that the new optimization system worked effectively using both 1D and 2D modeling. The optimization system can be used as a tool to assist drainage network engineers in developing sewer network optimization strategies and prioritizing detailed rehabilitation/upgrade projects at different scales (single or multiple neighborhood scale, city-wide and regional scale).

Finally, limitations of the optimization system such as practicability, variation range and software were discussed. And future research directions were suggested at the end of the thesis.

ACKNOWLEDGEMENTS

This project would not be possible to finish without the help and support of many people. I would express my thanks to them.

At first, I would like to express my gratitude to Dr. Wenming (William) Zhang, who has been an ideal mentor and supervisor. I really appreciate his help, guidance, and encouragement in my academic work.

Also, I would like to thank all members of the research group, especially Yang Yu. He gave helpful comments and suggestions on coding and modelling. And I appreciate my friends and the people who helped me make my graduate program productive and enjoyable.

At last, I want to express my deepest thanks to my family for their support, encouragement, and love.

Table of Contents

ABSTRACT ii

ACKNOWLEDGEMENTS iv

Introduction 1

1 Literature review 7

1.1 UDS optimization 7

1.1.1 Single objective optimization 8

1.1.2 Multi-objective optimization 9

1.2 Multi-objective evolutionary algorithms (MOEAs) 15

1.2.1 Genetic algorithm (GA) 15

1.2.2 Multi-objective particle swarm optimization (MOPSO) algorithm 17

1.2.3 Non-dominated sorting genetic algorithm II (NSGA-II) 18

1.2.4 Epsilon multi-EA (ε-MOEA) 20

1.2.5 Preference-inspired coevolutionary algorithm (PICEA-g) 21

1.3 Divide and conquer technique 24

2 Knowledge gaps and research objectives 26

2.1 Knowledge gaps 26

2.2 Research objectives 26

3 Methodology 28

3.1 Study area and existing models 28

3.1.1 Study area and PCSWMM model 28

3.1.2 Flooded area 30

3.1.3 1D and 2D models 31

3.1.4 Local design standards 32

3.1.5 Assumptions 32

3.2 Overall optimization structure 33

3.3 Divide and conquer 36

3.3.1 Divide 36

3.3.2 Conquer 37

3.4 Improvement to PICEA-g 40

3.4.1 Initialize candidate solutions 40

3.4.2 Enhanced goal vectors boundaries – cutting plane 42

3.4.3 Enhanced fitness calculation 43

3.4.4 Enhanced genetic operator – crossover and mutation 45

4 Results and discussion 49

4.1 Test area 49

4.2 The entire study area 54

4.2.1 Results based on 1D model 54

4.2.2 Results based on 2D model 64

4.2.3 Limitations 72

5 Conclusion and future research 74

5.1 Conclusion 74

5.2 Future research 76

Reference 78

Appendix 89

List of Tables

Table 1.1 Summary of studies on multi-objective optimization for urban drainage system ... 12

Table 3.1 Cost ($CAD) calculation table with pipe size (mm) and depth ranges in meters (m) (The City of A Water Resources, 2011) ... 38

Table 3.2 Minimum slope for different pipe size (The City of A Water Resources, 2011) 40

Table 4.1 The number of non-dominated solutions after each generation ... 63

Table A1 Before optimization: the selected conduits of the storm sewer trunk in the test area. ... 111

Table A2 After optimization: the selected conduits of the storm sewer trunk in the test area. ... 111

Table A3 Before optimization: the selected conduits of the storm sewer trunk in the study area, using the 1D model. ... 112

Table A4 After optimization: the selected conduits of the storm sewer trunk in the study area, using the 1D model. ... 113

Table A 5 Before optimization: the selected conduits of the storm sewer trunk in the study area, using the 2D model. ... 114

Table A6 After optimization: the selected conduits of the storm sewer trunk in the study area under scenario 1, using the 2D model. ... 114

List of Figures

Figure 1.1 Structure of GA Model (Kebede., 2014) 17

Figure 1.2 Illustration of ε-dominance concept (minimizing f1 and f2) (Deb et al. 2005)..... 21

Figure 1.3 An elitist framework of PICEA-g (Wang et al, 2013). 23

Figure 3.1 Study area with 1D storm sewer network model. 28

Figure 3.2 Location of test area in the entire study area. 29

Figure 3.3 An example of generating one flooded area. 30

Figure 3.4 Structure of optimization system for scenario 1 and scenario 2. 35

Figure 3.5 Process of divide for scenario 1 and scenario 2. 36

Figure 3.6 Process of conquer for scenario 1 and scenario 2. 39

Figure 3.7 An example of generating a solution of pipe slope. 42

Figure 3.8 An example of two objective problem using the cutting plane (Wang et al., 2013). 43

Figure 3.9 Process of evolving (crossover and mutation) for scenario 1 and scenario 2........ 45

Figure 4.1 Before optimization: existing manhole flooding risks and storm sewer conduit (pipe) conditions in the test area. 49

Figure 4.2 After optimization: manhole flooding risks and storm sewer conduit (pipe) conditions in the test area. 50

Figure 4.3 Before optimization: the profile of the selected storm sewer trunk in the test area. 52

Figure 4.4 After optimization: the profile of the selected storm sewer trunk in the test area. 53

Figure 4.5 The objective function values of each solution – the number of flooded nodes (manholes) vs. cost. 53

Figure 4.6 Before optimization: flooded areas in the study area based on the 1D model. 54

Figure 4.7 After optimization: flooded areas in the study area based on the 1D model. 55

Figure 4.8 The comparison of Flooded Area 1 in the study area before (left) and after (right) optimization, using the 1D model (Scenario 1). The selected route (in green color) is used for generating the profile. 56

Figure 4.9 Before optimization: the profile of selected sewer trunk in Flooded Area 1of the study area, using the 1D model. 57

Figure 4.10 After optimization: the profile of selected sewer trunk in Flooded Area 1of the study area, using the 1D model. 57

Figure 4.11 The average number of flooded nodes reduction in each area (1D model). 58

Figure 4.12 The objective function values of solutions (number of flooded nodes vs. cost), obtained by conquer with roulette wheel and by conquer randomly using the 1D model (Scenario 1). 60

Figure 4.13 The objective function values of solutions obtained after conquer technique, 1st generation evolving and 20th generation evolving in Scenario 1. 61

Figure 4.14 Calculation results of $Imdr$ with generations. 62

Figure 4.15 Before optimization: flooded area in the study area by using the 2D model. 64

Figure 4.16 After optimization: flooded area in the study area of Scenario 1 by using the 2D model. 65

Figure 4.17 After optimization: flooded area in the study area of Scenario 2 by using the 2D model. 66

Figure 4.18 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 2D model. The selected sewer trunk route (green color) is used for generating the profile. 67

Figure 4.19 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 2 by using the 2D model. 68

Figure 4.20 Before optimization: the profile of selected storm sewer trunk route in Flooded Area 6 of the study area, using the 2D model. 69

Figure 4.21 After optimization: the profile of selected storm sewer trunk route in Flooded Area 6 of the study area, under Scenario 1 by using the 2D model. 69

Figure 4.22 The comparison between Scenario 1 and Scenario 2 for conquer solutions' objective function values (number of flooded nodes vs. cost). 70

Figure 4.23 The objective function values of solutions obtained after conquer technique, 1st generation of evolving and 20th generation of evolving in scenario 2. 71

Figure A1 An example of how to calculate fitness value in a two-objective problem (Paknejad et al., 2021) 89

Figure A2 The comparison of Flooded Area 1 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 92

Figure A3 The comparison of Flooded Area 2 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 92

Figure A4 The comparison of Flooded Area 3 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 93

Figure A5 The comparison of Flooded Area 4 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 93

Figure A6 The comparison of Flooded Area 5 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 94

Figure A7 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 94

Figure A8 The comparison of Flooded Area 7 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 95

Figure A9 The comparison of Flooded Area 8 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 95

Figure A10 The comparison of Flooded Area 9 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 96

Figure A11 The comparison of Flooded Area 10 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 96

Figure A12 The comparison of Flooded Area 11 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 97

Figure A13 The comparison of Flooded Area 12 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 97

Figure A14 The comparison of Flooded Area 13 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 98

Figure A15 The comparison of Flooded Area 14 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 98

Figure A16 The comparison of Flooded Area 15 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 99

Figure A17 The comparison of Flooded Area 16 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 99

Figure A18 The comparison of Flooded Area 17 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 100

Figure A19 The comparison of Flooded Area 18 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 100

Figure A20 The comparison of Flooded Area 19 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 101

Figure A21 The comparison of Flooded Area 20 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 101

Figure A22 The comparison of Flooded Area 21 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 102

Figure A23 The comparison of Flooded Area 22 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 102

Figure A24 The comparison of Flooded Area 23 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 103

Figure A25 The comparison of Flooded Area 24 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 103

Figure A26 The comparison of Flooded Area 25 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 104

Figure A27 The comparison of Flooded Area 26 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 104

Figure A28 The comparison of Flooded Area 27 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 105

Figure A29 The comparison of Flooded Area 28 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 105

Figure A30 The comparison of Flooded Area 29 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 106

Figure A31 The comparison of Flooded Area 30 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 106

Figure A32 The comparison of Flooded Area 31 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 107

Figure A33 The comparison of Flooded Area 32 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 107

Figure A34 The comparison of Flooded Area 33 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 108

Figure A35 The comparison of Flooded Area 34 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 108

Figure A36 The comparison of Flooded Area 35 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 109

Figure A37 The comparison of Flooded Area 36 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 109

Figure A38 The comparison of Flooded Area 37 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model. 110

Introduction

One of the most frequent natural calamities is flooding. They have the greatest worldwide impact and frequently cause substantial infrastructure damage, traffic disruptions, economic losses and even death of people (Schmitt et al., 2004; Hammond et al., 2015). Urban regions are particularly vulnerable to flooding, which is typically caused by insufficient urban drainage systems (UDSs) (Schmitt et al., 2004). UDSs are made up of a catchment that converts the meteorological input to runoff, constructed drainage system components such as pipelines, reservoirs, treatment, and receiving waters (Salerno et al., 2018). Climate change and increased urbanization are the primary causes of UDS failure (Salerno et al., 2018). Adjusting the current UDSs with minimal investment and optimum system efficiency is important in addressing the difficulties of an uncertain environment and rapid urbanization (Yazdanfar and Sharma 2015).

Urban growth changes the hydrology of the environment and often has a detrimental impact on water quality. Urban land development alters land uses, increasing the quantity of stormwater runoff while generally lowering its quality, which depends on the existence/use of stormwater best management practices (BMPs). Stormwater from a site can be managed in a variety of ways, such as transportation, holding, processing, reusing, infiltration, and evaporation (Ahiablame et al., 2012).

Nowadays, separate storm sewer system must be designed, as combined sewer systems are no longer allowed. The typical components of a stormwater drainage systems include minor and major systems. The minor system (underground sewer pipe network) conveys flows from the more frequent, low intensity storm events, offering a minimal degree of service. In general, the hydraulic grade line (HGL) of a storm sewer must be at or below the pipe's obvert to transport design flow (Senior et al., 2018). Extreme rainstorm events that produce more runoff than the minor system can handle are transported by the major system, which includes roads, storage spaces, swales, etc.

Numerous physical processes interact throughout the stormwater runoff process, including the precipitation event, interception and depression storage, evaporation, and infiltration. Although there are numerous ways to estimate runoff volumes and flows, caution should be taken when conducting the analysis because the runoff process is complex. A thorough grasp of the runoff mechanism and the modelling strategy is necessary for the analysis. Runoff can be estimated using a variety of techniques, which consist of the rational method, SCS method, Horton method and deterministic methods. Besides, there are many computer models (SWMHYMO, EPA SWMM, XP-SWMM, etc.) that have been created for the hydraulic and/or hydrological assessments (Barraud et al., 2022).

The United States Environmental Protection Agency (USEPA) developed the Stormwater Management Model (SWMM) in 1971. Since then, SWMM has been widely used to simulate runoff quantity and quality from metropolitan areas for single or multiple storm events. SWMM's hydrologic activities take place on a variety of subcatchment regions that include both pervious and impervious sub-areas. Through subareas, subcatchments, drainage conveyance systems (pipes or channels), storage/treatment facilities, and diversion structures, runoff or overland flow can be transported and routed. At specific nodes from subcatchments, water quality constituents can be modelled using a hydraulic network with optional first order decay and associated pollution removal using BMPs and Low Impact Development (LID) in EPA SWMM (Sutherland and Jelen, 2003). Based on EPA SWMM, there is a spatial decision support system called PC SWMM. It features a cutting-edge, GIS engine that smoothly integrates with the most recent GIS data formats and offers smart tools for speeding up model construction, optimization, and analysis in a wide range of applications. Including the entire EPA SWMM engine, PC SWMM takes into account different hydrologic procedures such as precipitation, evaporation, the accumulation and melting of snow, interflow, overland flow routing in nonlinear reservoirs, infiltration of water into soil layers that are not saturated, and percolation of this water into subsurface water layers (Huber and Roesner, 2012).

A drainage system can be modelled using stormwater computer models for both single and multiple continuous rainfall events. While a continuous model is a simulation that models both dry and wet hydrological processes using a long-term continuous record of atmospheric data, a single event model is described as a simulation of a storm event. In single event modelling, a design single storm event, usually lasting 1 to 24 hours, is used to examine how a drainage system will react. The return frequency of the modelled storm is typically 1:5 year or 1:100 year (Huber et al., 2005). For continuous modelling, a drainage area's runoff is modelled over a set amount of time, e.g., several months or years. The continuous model incorporates a precipitation file obtained from different sources, e.g., the Meteorological Service of Canada. Typically, hourly rainfall totals are included in the precipitation file. Snowmelt modelling can also be done by using temperature files (Maheepala et al., 2001).

An intensive-duration-frequency (IDF) curve, which is related with local rainfall characteristics, is frequently used in the design of UDS (Adams et al., 1986). Although this traditional IDF-based approach such as the rationale method is straightforward and useful, the resulting UDS design is either unreasonable (i.e., the final solution cannot transmit the runoff generated by the design storm event), which will increase the risk of flooding, or the UDS is over-designed, which results in an unnecessary high investment (Lin et al., 2020). The primary problem is that this traditional approach may not accurately represent the actual rainfall process, particularly in the context of climate change (Jato-Espino et al., 2016).

The usage of optimization techniques to resolve UDS issues has increased recently due to their superior computational effectiveness and design correctness compared to more conventional techniques in dealing with complex scenarios (Nicklow et al., 2010). Single objective optimization was used initially, and it only considered the minimum cost of an UDS design (Lowe, 2010). Then, multi-objective evolutionary algorithms (MOEAs) have been created to balance competing UDS objectives (such as cost and flood damages) (Barreto et al., 2010; Mofrad and Yazdi, 2022).

A multi-objective optimization problem (MOP) has goals that may conflict with each other and has a set of Pareto optimal solutions. By letting a population of solutions evolve over time, MOEAs can get close to the Pareto optimal set (a set of solutions that are non-dominated by each other but superior to the rest of the solutions in the search space) in a single run (Shojaeefard et al., 2014). MOEAs were initially primarily employed in the field of water distribution systems, which performed well in serveral case studies (Choi et al., 2017; Wang et al., 2015).

Barreto et al. (2010) used MOEAs to evaluate UDS rehabilitation scenarios, contrasting investment against flood damages. A multi-objective optimization model of greywater reuse was created by Penn et al. (2013) with the goal of quantitatively balancing the costs and potable water demand reduction. To reduce damage costs and intervention expenses, Vojinovic et al. (2014) focused on multi-objective rehabilitation of UDS under uncertainties (climate change, urbanization, population expansion, and ageing of pipes). The application of MOEAs for the rehabilitation of storm sewer pipe networks was studied by Yazdi et al. (2017), who linked the MOEAs to the EPASWMM model. In addition, MOEAs were used by Bayas-Jiménez et al. (2019) to control the flood in UDSs due to climate change. Ngamalieu-Nengoue et al. (2019) conducted research on urban drainage rehabilitation utilizing MOEAs to replace pipes and add storage tanks. Recently, Lin et al. (2020) developed an engineering-based design method (EBDM) to generate approximate solutions to initialize the MOEA's search and improve the effectiveness of multi-objective optimization design.

For MOEAs, one of the most popular evolutionary algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Wang et al., 2018). NSGA-II has been used to address a variety of multi-objective optimization issues in UDS because of its high reliability and optimization effectiveness (Nicklow et al. 2010). It was applied to UDS rehabilitation by Barreto et al. (2010), Vojinovic et al. (2014), and Bayas-Jiménez et al. (2019). Besides, in 2004, the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm was proposed by Coello et al. (2004) to solve MOPs. Yazdi et al. (2017) studied on rehabilitation of storm sewer pipe networks by using both NSGA-II and MOPSO. In addition, Yazdi et al. (2016) used Non-dominated Sorting Differential Evolution (NSDE) to study the rehabilitation of UDSs. Moreover, Ngamalieu-Nengoue et al. (2019) applied both Pseudo Genetic Algorithm (PGA) and NSGA-II to UDS optimization. The majority of earlier studies did not take into account the UDS engineering design criterion that a sewer pipe size should, in general, not be less than that of its upstream pipes (Lin et al., 2020). Additionally, the decision variable used in the literature to constrain the UDS optimization was solely sewer pipe size. Moreover, in most case studies that have been reported, the study areas were small, with less than 100 sewers.

For large-scale MOPs, divide and conquer technique was proposed to divide the original search space of a MOP so that an algorithm only needs to search in one or more subspaces (Hong et al. 2021). With this technique, a complex problem can be divided into several low dimensional subproblems that are easier to deal with (Lv et al., 2021). In UDS optimization, the divide and conquer technique may be used to break up the optimization process for each subcatchment; and then, the best solutions from each subcatchment are put together to make a new set of best solutions (Yu et al., 2022).

In this research, the focus is on storm sewer network optimization for urban flood mitigation. A new storm sewer network optimization system was proposed that integrated a storm water management model (PCSWMM) with one of MOEAs Preference-Inspired Coevolutionary

Algorithm (PICEA-g) using a programming language (MATLAB) as the platform. Both 1D model (i.e., storm sewer pipe network) and 2D model (i.e., both storm sewer network and 2-D surface flow) were used as the base separately to achieve automatic optimization. Both sewer pipe size and pipe slope were used as decision variables, compared to the sole decision variable of pipe size in the literature, and the number of flooded nodes and cost were used as objective functions. Also, divide and conquer technique were applied in this research, which has not been used in previous UDS optimization problems. More importantly, this research expanded the UDS optimization from small-scaled areas (with less than 100 sewers) in the literature to a large-scaled area (with 2930 sewers). The main goal of this study is to develop and evaluate a new MOEAs method for determining the ideal pipe size and slope of a real-life complex storm sewer network to minimize the sewer rehabilitation costs and flood volume. The new optimization system developed from this study will aid in the proper design and rehabilitation/upgrade of urban stormwater drainage systems.

1 Literature review

1.1 UDS optimization

Using specialized mathematical methods, optimization is the process of selecting the best candidate solution or optimum solution from a set of options. A certain function known as the objective function may be maximized or minimized during optimization, depending on the methodology's needed target. Additionally, in order to achieve the best possible solution, the decision variable and an optimization process constraint must be defined prior to formulation. An optimization model is a decision-making model that employs certain programmed mathematical methods (Walski et al., 2003).

After defining the optimization issue, setting the objective functions, and setting the decision variables, the next stage is to continue with the optimization algorithm, which is used to minimize or maximize the objective function depending on the goal of the optimization. And until a good solution is found, this is done by adjusting the decision variables and taking the limitations into account (Wang et al., 2020).

Due to the complexity of UDS, optimization has become more important, and the computational models employed for these purposes have advanced quickly in terms of development and computational capacity. Urban drainage infrastructure optimization aims to reduce intervention costs and flooding while maintaining a certain level of performance (Burch et al., 2010). Also, many conditions must be met in order to achieve the optimum design because of the complicated characteristics of UDS, and these kinds of issues are known as MOPs. Finding a single solution that can optimum all objectives concurrently is unachievable due to the conflict between the many aims. It is crucial for a decision maker to look for the best trade-off solutions, also known as Pareto optimum solutions (Hong et al., 2021), for multi-objective optimization issues.

The number of decision variables and the number of objectives are two important determinants of the issue size in MOPs. The objective function is a mathematical formula that evaluates the fitness of chromosomes and quantifies the degree to which the goals are met. For instance, minimizing the costs and flooding volume. The parameters that may be altered to enhance a system's performance are known as decision variables, such as the pipe's diameter. Constraints are the limits or restrictions that must be taken into account when evaluating networks and solutions while the network is in operation. Constraints may be imposed explicitly on decision variables or implicitly on other system factors (for example, the net head loss around a loop must be zero) (Walski et al., 2003).

The genetic algorithm is one of the most popular methods for resolving optimization issues out of all the strategies that have been devised. Urban drainage system optimization problems have nonlinear objective functions that are unknown analytically and can have a variety of minimum or maximum values (Kebede., 2014).

1.1.1 Single objective optimization

Single objective optimization is the process of using a mathematical formulation to discover the optimal result that satisfies the maximum or minimum values of a single objective function. Single objective optimization was used, with the UDS design's least cost as the only factor (Ahmadi et al., 2018; Lowe, 2010). By integrating different objectives into one or by employing one objective function and using the others as constraints, MOPs can also be reduced to a single objective, allowing all of the objective functions to be quantified as a single function. As a result, prioritizing involves simply combining the objective functions into one objective with the same units (Savic, 2002).

Different optimization issues have addressed a variety of objectives, and single objective optimization is one of these tools that reveals information about the problem's nature. However, this method does not yield a collection of options that give decision makers options.

Consider the following pipe sizing example with two objectives: to minimize cost $f(x)$ and to maximize benefits $f(y)$. Max $f(y)$ is identical to min $-f(y) = minf'(y)$ when $f(y)$ is multiplied by -1. Since $f = minf(x) + f'(y)$, the two goals can be integrated into a single objective function (y) (Kebede, 2014).

1.1.2 Multi-objective optimization

A numerical method called multi-objective optimization was created to obtain a compromise optimal solution for multiple competing objectives and is used to evaluate the outcomes. Each solution in multi-objective optimization is not dominated by another solution to produce a single solution, unlike single objective optimization, thanks to the obtained tradeoffs, which make decision-making simpler. Based on the goals of the objective functions, the objective functions in a multi-objective optimization interact to produce a compromise solution known as the Pareto optimal front (Savic, 2002).

In a multi-objective optimization, one objective must be made worse in order to make the other objective better, and vice versa, to get a compromise solution between the two objective functions. Because the output of multi-objective optimization provides a wider range of possibilities, it is preferable to employ it for decision-making. A Pareto front displaying the compromised outcome of the objective functions is the result of multi-objective optimization.

Urban drainage design and rehabilitation projects require a lot of time-consuming manual trial and error, and it is challenging to find the best solution in a vast drainage network with a variety of pipe diameters and slopes in each area. Optimization associated with urban drainage models offers better performance in these projects. A few network parameters are combined in a standard design procedure (Vélez et al., 2007).

MOEAs were initially primarily applied in the field of water distribution systems and did well in case studies (Choi et al., 2017; Wang et al., 2015). Although MOEA applications in the field of UDS are already widespread, distinct MOEAs' performances still need to be improved (Yazdi et al., 2017). Additionally, there is currently a lack of study on how to find the best solutions using better relevant parameter settings and searching operators.

In order to compare investment against flood damages in UDS rehabilitation scenarios, Barreto et al. (2010) employed MOEAs. Using a genetic algorithm, Sun et al. (2011) developed a risk-based strategy and determined the ideal pipe diameters and slopes in a UDS by using genetic algorithm. Delelegn et al. (2011) used the Non-dominated Sorting Genetic Algorithm II (NSGA II) with a 1D-2D hydraulic model to create detention ponds in metropolitan areas using the return period approach. Similar to this, Park et al. (2012) used GA to optimize the storage capacity, outflow structure diameter, and quantity of detention ponds in an urban region in South Korea. The first multi-objective optimization model with the goal of quantitatively balancing the cost of on-site and local graywater was created by Penn et al. (2013). To reduce damage costs and intervention expenses, Vojinovic et al. (2014) concentrated on multi-objective rehabilitation of UDS under uncertainty (climate change, urbanization, population expansion, and ageing of pipes). J. examined the use of MOEAs for the restoration of storm sewer pipe networks. In order to improve the rehabilitation of UDSs under a fixed design storm, Yazdi et al. (2017) integrate the EPA-SWMM hydraulic model with three distinct MOEAs, including NSGA-II, Multi-Objective Particle Swarm Optimization (MOPSO), and Non-dominated Sorting Harmony Search NSHS. The EPASWMM model was connected to the algorithms by Yazdi et al. (2018). Additionally, Bayas-Jiménez et al. (2019) used MOEAs to manage the flood in metropolitan area drainage networks brought on by climate change. Ngamalieu-Nengoue et al. (2019) conducted research on urban drainage rehabilitation utilizing MOEAs to replace pipes and add storage tanks. In order to begin the MOEA's search and increase the efficacy of multi-objective optimization design, Lin et al. created an engineering-based design technique in 2020. Table 1.1 summarizes the studies on multi-objective optimization for urban drainage systems.

Even though MOEA-based methods have been successfully used to find the best ways to design UDSs, they are not always easy to use in real life. One important problem is the low convergence rate (Lin et al., 2020), which affects how well the MOEA-based optimization methods use computers (Fu et al., 2009; Wang et al., 2018). This is because the MOEAs use physically based UDS simulation models for fitness evaluation and search, which can be hard to run on a computer (Maier et al., 2014). For example, in Hadka and Reed 2013s' study, a preliminary test was done where a UDS with 53 pipes was optimized by a benchmark MOEA algorithm Borg and SWMM. Simulations showed that if 1,000,000 evaluations were allowed, this optimization would take about 15 days on a 2.9GHz Dell PC (Inter R) and about 360 hours. In real-world applications, MOEAs often require such a large number of evaluations (Wang et al., 2015). This extra work on the computer can take up a lot of the time that is usually available for the UDS design, which is usually a few hours to a few days (Wang et al., 2015). Even though techniques like metamodels (Fu et al., 2009) and the GAHP model (Hassan et al., 2018) have been made to improve the computational efficiency of MOEA optimization for UDS designs, their performance is still not good, especially when dealing with large real-world UDSs (Maier et al., 2014).

Table 1.1 Summary of studies on multi-objective optimization for urban drainage system

Study	Algorithm/ Method	Objectives	Decision variables	Case study	Limitations
Seyedashraf et al. (2021)	NSGA-II	To expand the capacity of an existing drainage network within a region of interest using sustainable drainage assets	The types of sustainable drainage assets and surface area	Case study 1 is an 84 ha urban catchment, with 64 subcatchments, 566 manholes, and 511conduits. Case study 2 is a 66 ha urban catchment area in Windsor, Canada	Focus on sustainable drainage assets
Lin et al. (2020)	EBDM	To enhance the multi-objective optimization efficiency	Pipe size and slope	Case Study 1 is a university campus in Hanghzhou, China, with a drainage area of 8.1 ha and 19 subcatchments. Case Study 2 has a drainage area of 29 ha consisting of 53 subcatchments	Small-scale; hydraulic model not used
Ngamalieu-Nengoue et al. (2019)	NSGA-II and PGA	To optimize the rehabilitation process and reduce search space	Pipe size	In Ayurá district, Medellin city, Colombia, with a drainage area over 22.5 ha,73 nodes.	Small-scale; engineering criteria and pipe slope not considered
Ngamalieu-Nengoue et al. (2019)	NSGA-II	To urban drainage networks rehabilitation and reduce investment costs.	Pipe size	In E-Chicó district, Bogotá city, Colombia, with a drainage area over 51ha, 35nodes.	Small-scale; engineering criteria and pipe slope not considered
Bayas-Jiménez et al. (2019)	NSGA-II	To reduce the cost of rehabilitation of drainage networks and provide resilience to cities in climate change scenarios.	Pipe size	In E-Chicó district, Bogotá city, Colombia, with a drainage area over 51ha, 35nodes.	Small-scale; engineering criteria and pipe slope not considered

Study	Algorithm/ Method	Objectives	Decision variables	Case study	Limitations
Yazdi et al. (2018)	NSGA-II and MOPSO	To assign optimal rehabilitation plans for sewer pipe network	Pipe size	In Seoul, South Korea with a drainage area of 44 ha, 32 pipes, nodes and subcatchments	Small-scale; engineering criteria and pipe slope not considered
Yazdi et al. (2017)	NSGA-II, NSDE and NSHS	To compare rehabilitation plans of UDS network by using different algorithm	Pipe size	In Yongdub, South Korea with a drainage area of 101 pipes and nodes, and 102 subcatchments	Small-scale; engineering criteria and pipe slope not considered
Vojinovic et al. (2014)	NSGA-II	To accommodate the effects of uncertainties into the design and rehabilitation of UDS and minimize damage costs and intervention costs.	Pipe size	In Dhaka, Bangladesh with a drainage area of 830 ha and 88 sewer links with a total length of 13,635 m.	Engineering criteria and pipe slope not considered; hydraulic model not used
Penn et al. (2013)	NSGA-II	To quantitatively trade off the cost of graywater and the total amount of flow discharged into the municipal sewer system	Pipe size	In central Israel, with a sewer system of 154 nodes and 153 links	Only consider graywater
Delelegn et al. (2011)	NSGA-II	To evaluate sewer pipe design scenarios contrasting investment against flood damages	Storage volumes	In West Garforth, UK	Only consider storage reservoirs as the intervention
Barreto et al. (2010)	NSGA-II and ε-MOEA	To evaluate urban drainage rehabilitation scenarios contrasting investment against flood damages	Cost	A drainage system of 12 pipes, 13 manholes, and 11 subcatchments	Small-scale; hydraulic model not used

In addition to not being very effective, one problem with MOEA-based approaches is that the UDS design solutions that are found are often not practical (Lin et al., 2020). In the multi-objective optimization framework, this issue has often been left out (Walski, 2001). For example, most of the studies that have already done are about fixing up urban drainage systems by changing the size of the pipes. The size of the pipe was used as a decision variable, but the slope of the pipe, which is another important factor, was not considered. (Bayas-Jiménez et al., 2019, Yazdi, 2017, Vojinovic et al., 2014). This leads to the limitation of UDSs optimization. At the same time, most of the MOEA-based optimization methods used didn't take into account the engineering criterion of the UDS design that a pipe's size shouldn't be smaller than the size of the pipes upstream (Ngamalieu-Nengoue et al., 2019; Vojinovic et al., 2014; Yazdi, 2018). Also, many of the methods used in the past only used simple statistics, such as the average or sum of the flood volumes or peak flows in the pipes, to show how reliable the system was (Liu et al., 2018; Mohammadiun et al., 2018). But these simple statistics can be less useful in real life because they can't show how the flood volumes and peak flows are spread out across the entire UDS. This can make the risk of flooding high in some places with high flood volumes or pipe peak flows (Lin et al., 2020).

In 2011, Delelegn et al. used a 1D2D coupled model of SWMM5 linked with NSGA-II to do multi-objective optimization of the cost-benefit of urban flood management. By using SIMBA6 (an extended version of EPA SWMM), Penn et al. (2013) made the first multi-objective optimization model that tried to figure out how to trade off the cost of local graywater. Rathnayake and Tanyimboh (2015) used SWMM 5.0 to study multi-objective optimization of combined sewer systems which goal was to minimize the bad effects of combined sewer overflows on the environment. J. Yazdi et al. (2016) came up with a way to fix up urban drainage systems that is based on resilience and combines NSDE and the EPA-SWMM simulation model. In the paper by Ngamalieu-Nengoue et al. (2019), SWMM is first connected to PGA for the search space reduction. This search space reduction method is used to reduce the number in decision variables of urban drainage network optimization. Also, the SWMM model was used to simulate how the drainage system works so that Seyedashraf et al. could use sustainable drainage assets to increase the capacity of an existing drainage network in an area of interest (2020).

In order to make pipe optimization simpler and more efficient, most studies were limited to small study areas. For instance, the case study 1 in Lin et al. (2019) is a university campus in Hangzhou, China, with a drainage area of 8.1 ha and 19 subcatchments and case study 2 has a drainage area of 29 ha consisting of 53 subcatchments. In Ngamalieu-Nengoue et al. (2019), a subcatchment of the drainage network of Ayurá district, Medellin city Colombia was used, and this network was extended over 22.5 ha (73 nodes). The case study used by Yazidi et al. (2017) is a storm sewer pipe system in the south part of Seoul, South Korea, with an area of 44 ha and including 32 pipes. In 2010, Wilmer Barreto et al. conducted research on a network system consisting of 12 pipes, 13 manholes, and 11 subcatchments.

1.2 Multi-objective evolutionary algorithms (MOEAs)

1.2.1 Genetic algorithm (GA)

Genetic algorithm (GA) was first introduced by John Holland in 1975, and he continued to research it with his team in order to advance it. However, over the past ten years, a lot of work has been put into researching and developing genetic algorithms. A genetic algorithm employs a numerical model to tackle issues related to the evolution of individuals through the processes of selection and reproduction. In a genetic algorithm, a population’s survival is determined by how well it adapts to its environment. To maintain a population of individuals that survives the process, specific rules of selection cross-over and mutation are used. A genetic algorithm bases its selection criterion on people with high fitness values, and as a result, chromosomes in a population are either deleted or preserved depending on fitness at the time of evaluation. By combining two or more chromosomes and altering certain parameters, a genetic algorithm creates new children through a process of cross-over and mutation. Individuals' genetic information is represented by arrays of binary data or genes, so simple maneuvering causes mutation and cross-over to occur (Murphy et al., 1993).

Murphy et al. (1993) developed and used an enhanced evolutionary method for pipe network optimization, and decision variables were represented using grey coding as opposed to binary coding. The performance of improved genetic algorithms was shown to be superior to that of traditional optimization methods when compared in their solutions. Vairavamoorthy and Ali (2000) developed an enhanced genetic algorithm, also known as a real coded algorithm, for a water distribution system. The project's goal was to reduce the capital cost of the network by ensuring that each node had sufficient pressure. Tests on the performance of various networks were used to determine the method's effectiveness and robustness.

In contrast to traditional algorithms, genetic algorithms deal with populations rather than a single answer. A population's individual chromosome is represented by the value chosen for the decision variable, and each generation selects the chromosomes that match the following generation's needs the best. The chromosomes may persist for many generations or be replaced by others in the process to select the greatest match depending on their intended function and fitness (Liong et al., 1995). After establishing the fitness of the chromosomes and representing the genes, the cycle proceeds by initializing a random population. The needed solution will be improved by repeating the mutation, crossover, and selection operators until the termination requirement is met. On Figure 1.1, the structure of GA is depicted (Kebede., 2014).

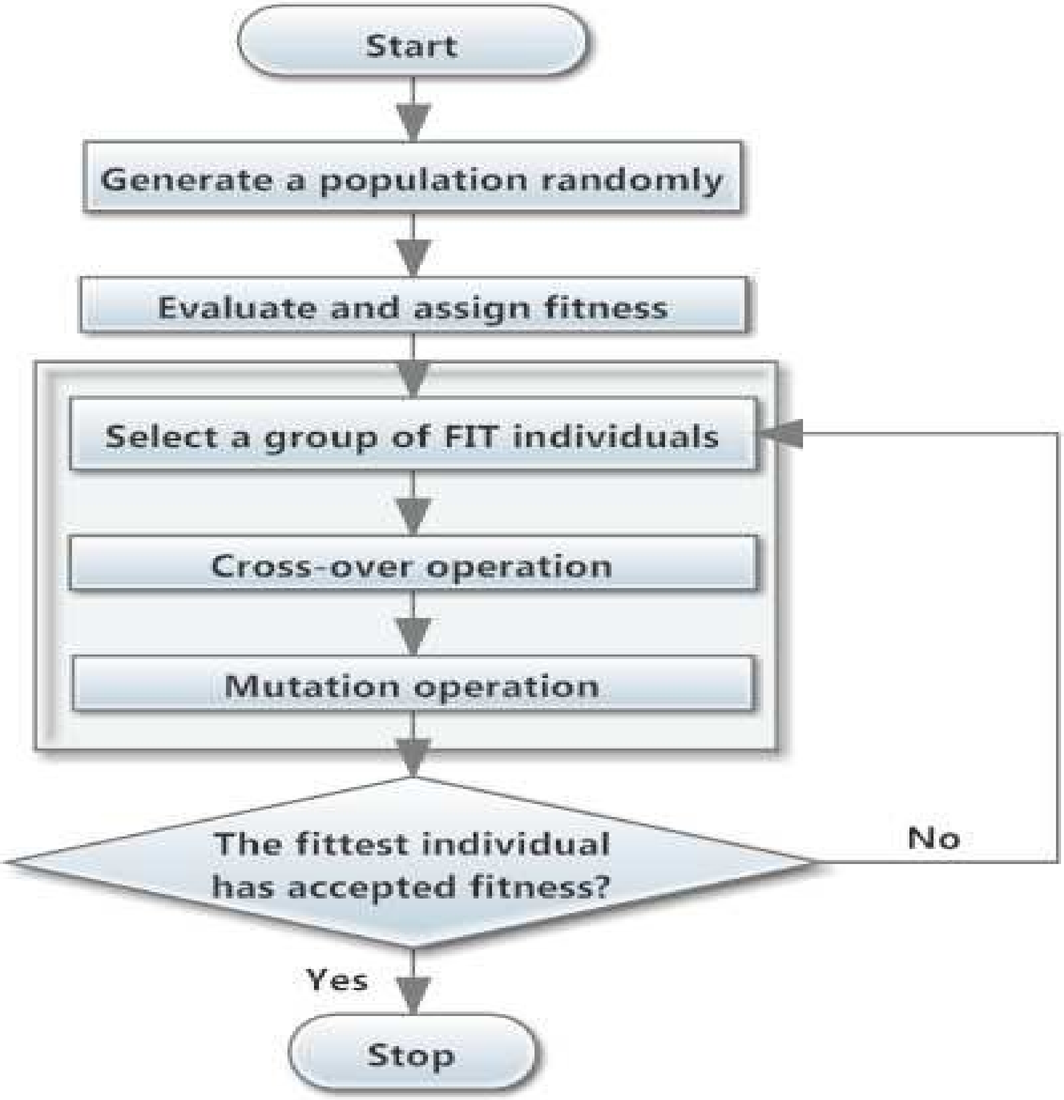

Start
Generate a population randomly
Evaluate and assign fitness
Select a group of FIT individuals
Cross-over operation
Mutation operation
The fittest individual has accepted fitness?
No
Yes
Stop

Figure 1.1 Structure of GA Model (Kebede., 2014)

1.2.2 Multi-objective particle swarm optimization (MOPSO) algorithm

A social search technique called particle swarm optimization (PSO) was motivated by the motion of a flock of birds (Kennedy and Eberhart, 1995). PSO has been proven effective in a number of different optimization projects. This algorithm is based on the idea that each particle determines where it will be at any given time based on where it has been so far in the group and where it will be at the best spot in its immediate vicinity. The great speed of convergence that PSO's method exhibits for single-objective optimization makes it seem particularly well suited for multi-objective optimization (Coello et al., 2004). For managing multiple objectives, several different algorithms have been proposed to expand PSO (Parsopoulos and Vrahatis, 2002; Ray and Liew, 2002). This multi-objective problem was solved using the multi-objective particle swarm optimization (MOPSO) technique, which was developed by Coello et al. (2004) and has been successfully used to solve a variety of engineering challenges. Water engineering applications include managing irrigation and drain water (Noory et al., 2010), parameter calibration of flood routing models (Azadnia and Zahraie, 2010), optimal reservoir operation (Baltar and Fontane, 2008), auto calibration of reservoir water quality modelling (Afshar et al., 2013) and water allocation (Liu et al., 2013). A secondary repository of particles is employed by the MOPSO algorithm, and later, other particles use this repository to direct their own flight. The non-dominance criterion is used to choose the particles that can enter the repository. Members of the repository are scattered among the grid boxes, or hypercubes, created from the objective space of non-dominated solutions that was discovered. First, a grid or hypercube based on the roulette wheel approach is chosen to choose the leader. This makes it more likely that a hypercube with fewer non-dominated members will be chosen. The leader of the hypercube is then chosen at random from among its members. Additionally, MOPSO has a unique mutation operator that enhances the algorithm's exploratory capabilities (Coello et al., 2004).

In addition to the MOPSO algorithm, the idea of nondominance has been applied in many well-known multi-objective meta-heuristic algorithms for sorting individuals according to the values of the objective function, including microGA, Niched Pareto Genetic Algorithm (NPGA), Strength Pareto Evolutionary Algorithm (SPEA), and NSGA-II (Yazdi et al., 2015).

1.2.3 Non-dominated sorting genetic algorithm II (NSGA-II)

One of the most well-known elitist multi-objective optimization methods, NSGA-II ensures the performance of a solution through generations, where the most adaptable solution advances to the following generation and requires no hydraulic parameters to function. The advantage of NSGA-II is its quick, selective solution of multi-objective optimization algorithms (Deb, 2002).

Because NSGA-II compares more than one objective and uses the non-dominated technique of ranking to assess the fitness of the chromosomes within a generation, the search process for basic genetic algorithms differs from NSGA-II. The dominance of one solution is examined using the fitness of each optimization objective. Additionally, the crowding distance approach will be utilized to assess two chromosomes with the same rank of dominance, and this results in a uniformly distributed Pareto front with the best possible outcome (Deb, 2002).

Kebede (2014) used a hydrodynamic simulation model integrated with NSGA-II, which increased the significance of NSGA-II. In his study, one of the integrated model's objectives was to reduce flooding in the drainage network during design or restoration, so the main focus should be on solutions with little to no flooding. Zare et al. (2012) found the best places for LID practices in Tehran by looking at costs, improvement of quality indices, and amount of surface runoff. They used two widely used MOEAs: NSGA-II and MOPSO. They reported that NSGA-II does better than MOPSO based on several residual metrics.

Yazdi et al. (2017) examined MOPSO and NSGA-II for determining the best restoration strategies for sewer pipe networks. In order to find the most effective rehabilitation plans for pipe replacements, these algorithms were then connected to the EPA SWMM hydraulic model and applied to a storm sewer pipe network case study in Seoul, South Korea. The outcomes demonstrated that the algorithms under consideration had various tendencies when tackling the benchmark tests and rehabilitation issues. The sorting approach used by the NSGA-II and MOPSO algorithms differs significantly. While the MOPSO method simply selects non-dominated members as the external repository and does not explicitly sort members in the repository or in the population, the NSGA-II algorithm sorts all population members based on the two mentioned criteria. And the NSGA-II approach offers a simpler process for creating the following generation.

1.2.4 Epsilon multi-EA (ε-MOEA)

The foundation of ε-MOEA, as proposed by Laumanns et al. (2002) and Deb et al. (2003), is the idea of ε-dominance, which forbids the existence of two solutions for any objective function that differ by less than. In Lausannes's study, they proposed the approximate Pareto set as a solution concept for evolutionary multi-objective optimization. It is theoretically appealing because it helps to build algorithms with the desired convergence and distribution properties, and it is practically important because it works with a solution set with a limited size and a predetermined resolution.

By providing a suitable value for ε, it is possible to keep this algorithm's convergence and diversity properties. The objective space is broken up into a grid of hyperboxes, the size of which can be changed by the choice of ε. The dominance of the hyperbox for each hyperbox containing a solution is examined. One solution is kept for each nondominated hyperbox in ε-MOEA using the archive technique proposed in Deb et al. (2003). The following details the unique dominance checking procedure (Figure 1.2). The dominated archive members (D) are first discarded if the hyperbox of a new solution (C) dominates another hyperbox (D) in the archive. Second, the dominating solutions are eliminated when there are many solutions in the same hyperbox (A,B) (B). Third, if a hyperbox (E, F) contains more than one nondominated solution, one of them is chosen at random. Deb et al. (2005) recommended picking the solution (E) that is most near the hyperbox's origin for the third phase.

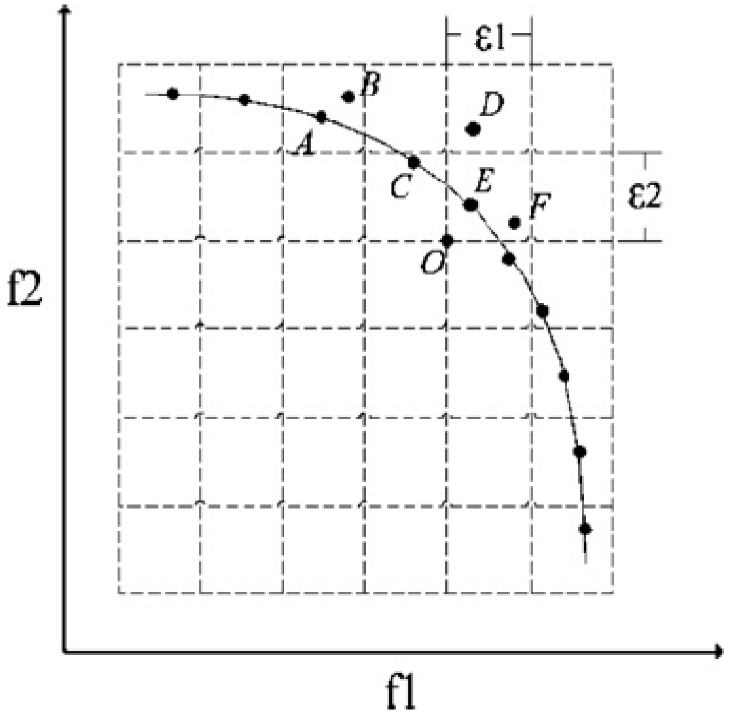


Figure 1.2 Illustration of ε-dominance concept (minimizing f1 and f2) (Deb et al. 2005)

Both NSGA-II and ε-MOEA were put to the test by Barreto et al. (2010) for the best rehabilitation of urban drainage systems. It is discovered that the NSGA-II algorithm performs better with smaller population sizes, while it performs worse with larger population sizes. It is discovered that the MOEA algorithm is less sensitive to population size. Additionally, when processing time is crucial, the lower cardinality of ε-MOEA is a valuable characteristic for optimizing urban drainage networks. When the population size is raised, the diversity of the generated sets of solutions, however, is not as good as that of NSGA-II.

1.2.5 Preference-inspired coevolutionary algorithm (PICEA-g)

The Pareto-dominance relation's diminished capacity to provide comparison between various solutions has been noted as one of the primary issues for multi-objective optimization (Deb et al., 2002). Due to this lack of comparability, Pareto-dominance algorithms have difficulty directing the search toward the Pareto front. But it is able to compare other incomparable options by considering the preferences of the decision-maker (Purshouse and Fleming, 2007). Coevolving the preference family with the typical population of candidate solutions is one technique to potentially keep the family relevant as the search advances. The preferences would gain fitness by providing comparability between solutions, and the solutions would earn fitness by performing well against the preferences. This method was initially presented as PICEA-g by Wang et al. (2013), which was a more effective approach for MOPs. In Wang et al. (2013) s' research on first realization of PICEA-g, Lohn et al. (2002) s' method of fitness assignment was retained. According to Lohn et al. (2002), increasing variety along the Pareto front might be accomplished by coevolving a family of target vectors.

By satisfying a certain set of goal vectors in objective space, candidate solutions earn fitness according to Lohn et al (2002) s' fitness assignment method, but the fitness contribution must be split among other solutions that also meet those targets. Targets can only become fit by being fulfilled by a candidate solution, and the more frequently other solutions in the population satisfy the targets, the less fit they become. Overall, the objective is for the targets to adaptively direct the solution population in the direction of the Pareto front. In other words, the target population and the population of candidate solutions coevolve toward the Pareto front.

By using PICEA-g, a family of preferences was coevolved with candidate solutions, the preferences gain higher fitness by being satisfied by fewer candidate solutions, and the candidate solutions gain fitness by meeting as many preferences as possible. The PICEA-g algorithm can optimize three or more goals. According to the empirical results, PICEA-g showed highly competitive performance and can therefore make a strong claim for use on many-objective problems (Wang et al., 2013). By providing a family of goal vectors in PICEA-g, candidate solutions receive a new fitness assignment and are guided towards the Pareto optimal front. PICEA-g overcomes the issue of poor performance of NSGA-II when the objective number is large (Wang et al., 2013). In addition, PICEA-g was applied to optimize LID spatial allocation in previous studies (Men et al., 2020; Yu et al., 2022).

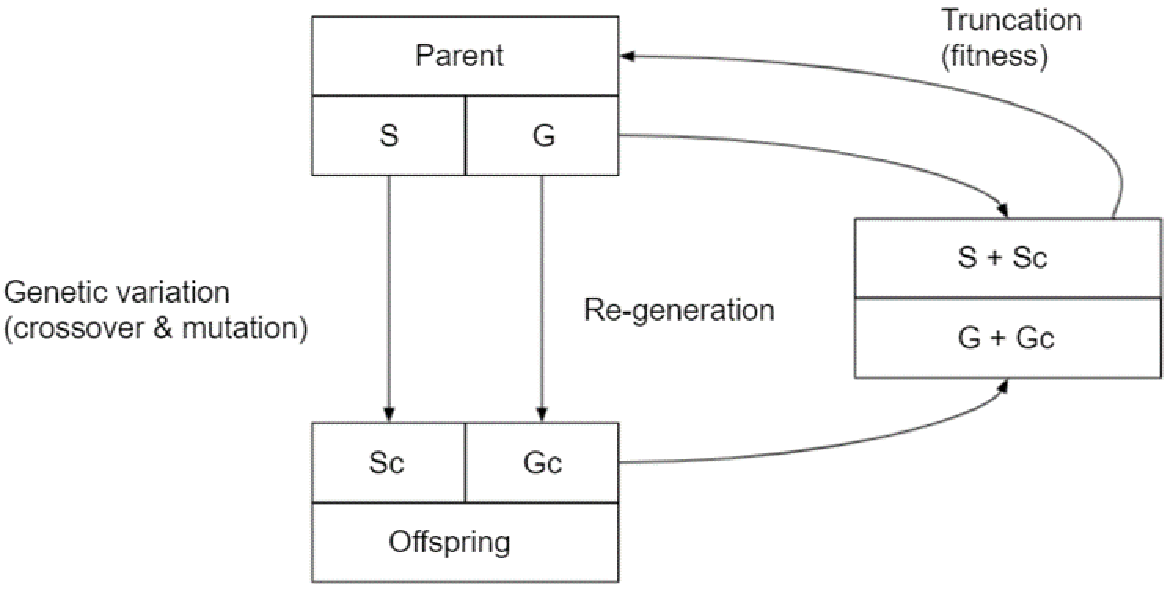


Figure 1.3 An elitist framework of PICEA-g (Wang et al, 2013).

A flowchart of PICEA-g elitist framework is depicted in Figure 1.3. A population of candidate solutions S (fixed size of N) and goal vectors G (fixed size of Ng) are evolved for a fixed number of max generations. They will co-evolve for a preset number of generations. The fixed size of N is determined as 100 in this study. In each generation, parent solutions S goes through genetic variation operators (crossover and mutation) and generate offspring solutions Sc (fixed size of N). Meanwhile, new goal vectors Gc (fixed size of Ng) are generated randomly in a defined space. Fitness is then calculated for each population of S, Sc, G, and Gc. After sorting the fitness of individuals in the combined population of $(S + Sc)$ and $(G + Gc)$, the best N solutions and Ng goal vectors will become the new population of S and G in the next generation (Wang et al., 2013).

1.3 Divide and conquer technique

As practical needs led to research on designing MOEAs with high scalability, ideas like simplifying large-scale MOPs with the divide and conquer technique and dimensionality reduction and improving the search ability of MOEAs by rebalancing exploration and exploitation were put forward (Antonio and Coello, 2013; Qian and Yu, 2017). With the divide and conquer method, a hard problem can be broken down into a number of smaller problems that are easier to solve (Lv et al., 2016). In UDS optimization, the divide and conquer technique is used to break up the optimization process for each subcatchment. Then, the best solutions from each subcatchment are put together to make a new set of best solutions (Yu et al., 2022).

The goal of the divide and conquer technique is to divide or change the original search space of a MOP so that an algorithm only needs to search in one or more subspaces. Since the subspaces are usually low-dimensional, these methods can help get rid of the curse of dimensionality that comes from adding more decision variables. On the other hand, to avoid the problem that some Pareto-optimal solutions fall outside of the subspaces made by space division or transformation, which would lead to poor performance, the enhanced search-based approach directly explores the original search space, but with better search abilities (Hong et al. 2021).

In Hong et al. 2021, it was shown that divide and conquer based large scale MOEAs work well to solve most large-scale MOPs where there are no or weak correlations between the decision variables. In 2016, Lv et al. applied the divide and conquer strategy with the cooperative co-evolution algorithm to manage air traffic flow. Cooper et al. (2014) used the divide and conquer method to solve the problem of how to design a large-scale public transportation network. And Antonio and Coello (2013) came up with a plan for cooperative coevolution that used the divide and conquer strategy. Besides, divide and conquer technique was widely applied to vehicle routing problems (Watanabe et al. 2015), resource allocation problems (Friese et al. 2016) and engineering design problems (Gaur et al. 2017).

2 Knowledge gaps and research objectives

2.1 Knowledge gaps

Based on the above literature review on solving MOPs by using MOEAs for UDS, the following knowledge gaps were identified:

1) Existing studies on MOEAs only used sewer pipe size as the sole decision variable in UDS optimization, while the other important factor, sewer pipe slope, has not been considered. Also, the connection of pipes is still an issue for practical engineering applications because the decision variable in original MOEAs is independent, and the variation is random.

2) Most UDS optimization solutions obtained by MOEA-based approaches are found not to be practical. The engineering criteria of the UDS design have been barely considered in the optimization, e.g., a pipe size generally should not be smaller than the upstream pipe sizes.

3) Physically based UDS simulation models are used in MOEAs for fitness evaluation and search, which can be challenging to run on a computer. Even though some techniques have been developed to improve the computational efficiency of MOEA optimization for UDSs, their performance is still not satisfactory.

4) In order to make sewer pipe optimization simpler and more efficient, most studies are limited to small study areas that include less than 100 pipes.

5) When using hydraulic models to solve MOPs, most studies are limited to 1D models. 2D models have not been applied due to the efficiency of optimization, and the results have been compared with 1D modeling results.

2.2 Research objectives

The overall research objective is to fill the above-mentioned knowledge gaps for UDS optimization, focusing on a large, complex, real-life, urban storm sewer network for flood mitigation purpose. The specific objectives of this study are as follows:

1) To create and evaluate a method for determining the ideal sewer pipe size and slope of stormwater drainage systems to minimize the costs of sewer system rehabilitation/upgrade and flood volumes.

2) To ensure the connection of sewer pipe system and consider the UDS engineering design criterion by setting variation conditions for candidate solutions.

3) To improve the computational efficiency of MOEA optimization for UDS designs by using parallel computing in Matlab.

4) To simplify large-scale MOPs with the divide and conquer technique so that the best solutions from each subcatchment are put together to make a new set of best solutions.

5) To apply both 1D and 2D hydraulic models in the sewer network optimization and compare their results to examine and improve the efficiency of the optimization.

3 Methodology

3.1 Study area and existing models

3.1.1 Study area and PCSWMM model

In this study, the study area is a drainage area of 777.6 ha consisting of 2200 subcatchments, 2927 junctions and 2932 conduits (Figure 3.1). The main land use of this study area is residential with a mix of commercial and industrial and the average impervious of the study area is 54.14%. City A 1 hour 10 years design storm is set as the time series of this model. In addition, there is a test area that is part of the study area located in city A (Figure 3.2). The test area is a 16.68 ha area which contains 74 subcatchments, 78 junctions and conduits.

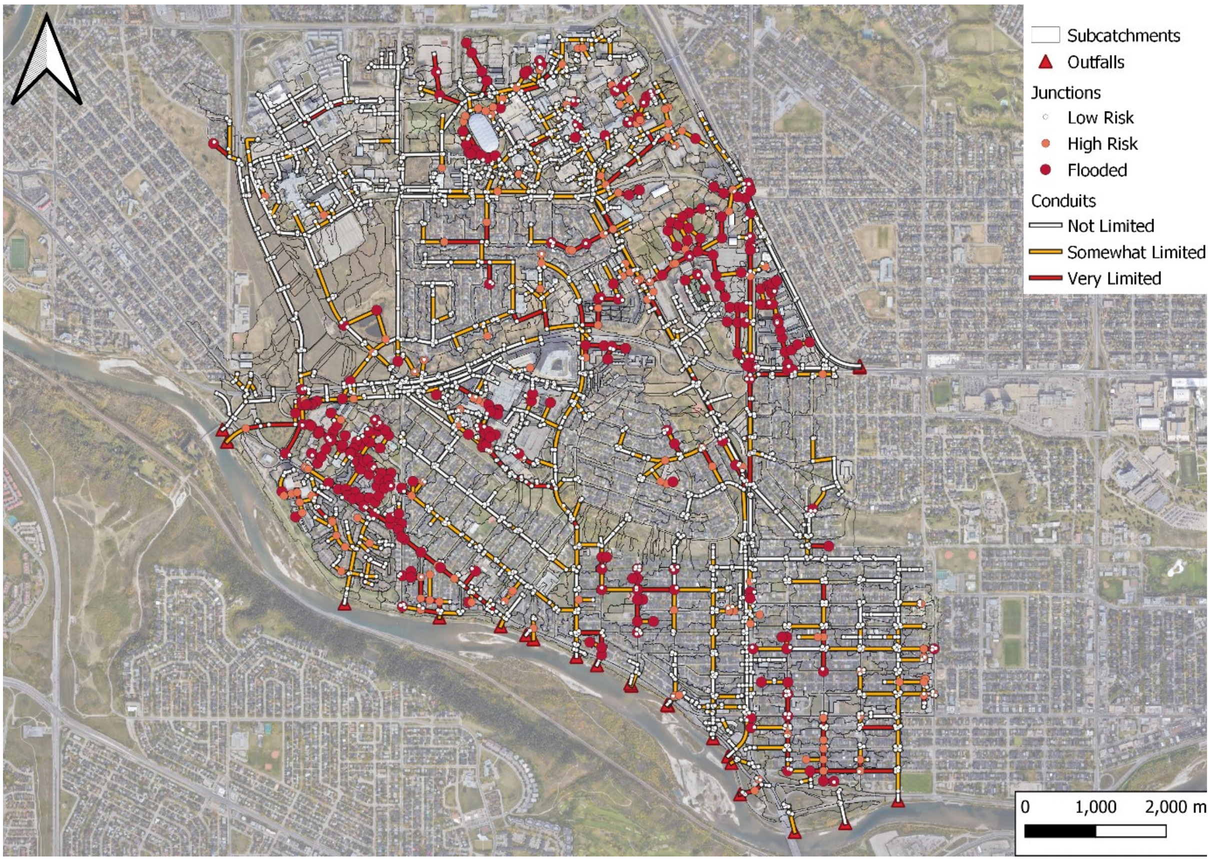


Figure 3.4 Study area with 1D storm sewer network model.

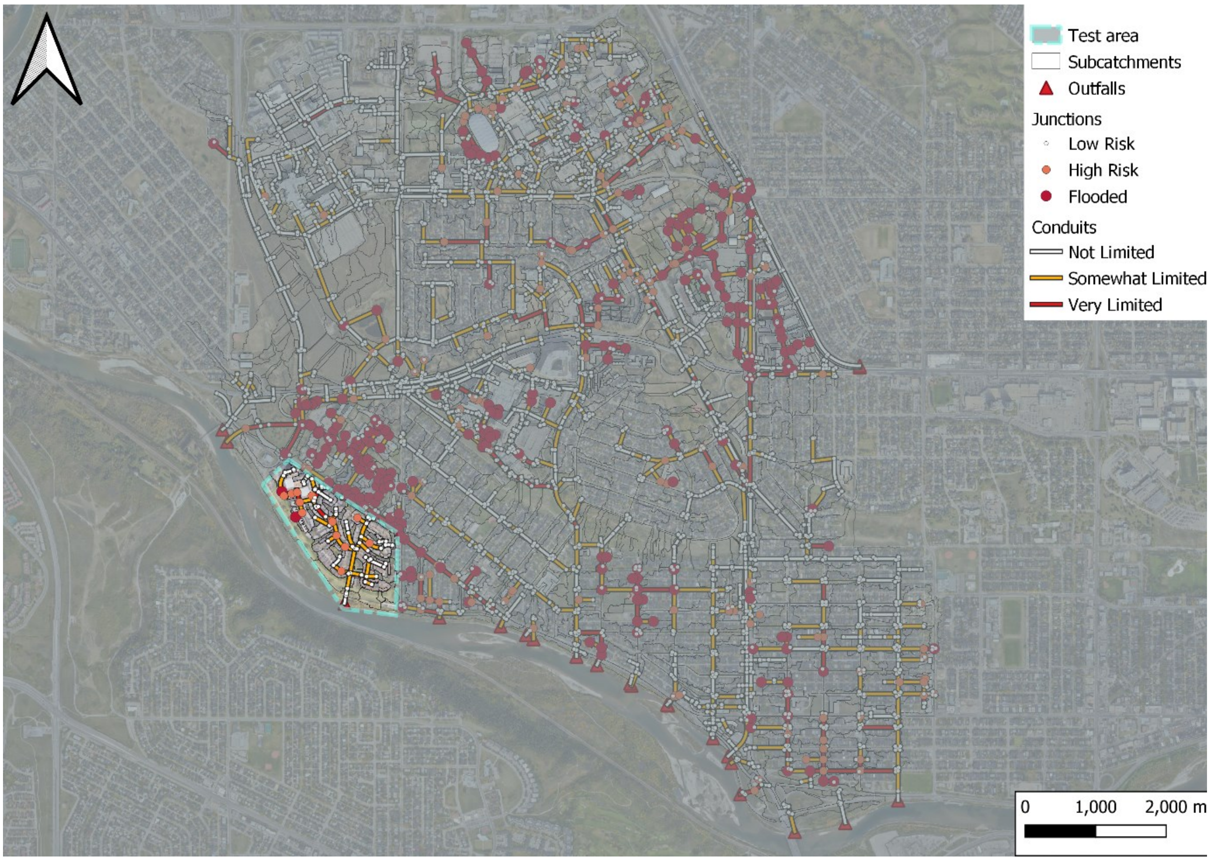


Figure 3.5 Location of test area in the entire study area.

Junctions were categorized depending on the amount of freeboard in PCSWMM model. Freeboard was defined as maximum HGL (Hydraulic Grade Line) - ground elevation. The junction was defined as flooded when freeboard > 0, high risk when freeboard <= 0 and > -0.35, and low risk when freeboard <= -0.35. Conduits were also categorized into three types based on capacity limited (CL) which is the number of hours during the simulation in which the flow through the conduit was limited by its capacity. The CL is a computational measure in SWMM for evaluating sewer pipe capacity limitation. A pipe is flagged as limited during a timestep if both pipe ends are full and the hydraulic gradient is greater than the slope of the pipe (SWMM User’s Manual, 2015). This metric was very useful for pipe upgrades optimization. The conduit was defined as very limited when CL >= 0.1 hr, somewhat limited when CL >= 0.01 and < 0.1hr, and not limited when CL < 0.01hr.

3.1.2 Flooded area

In this study, only manholes were considering as flooded nodes and only conduits were considered for upsizing (no orifice, weir, etc.). Then, Manhole-HGL matrix was created by finding position of node depth summary from PCSWMM report file. At last, the flooded nodes were determined based on conditions that freeboard> 0.

Flooded nodes were integrated into different flooded areas. To determine the flooded range of one flooded node, different numbers of connected conduits were tested in this study. For instance, four connected conduits both upstream and downstream were used to determine the flooded range at first and two connected conduits from both upstream and downstream were also tested. The results of flooded area by using different numbers of connected conduits were compared, and three connected conduits from upstream and four connected conduits from downstream were finally selected in this study. If there is any flooded node or its flooded range intersects another junction's flooded range, the flooded range of these flooded nodes would be combined to one flooded area. If there is no intersection of two flooded ranges, the flooded area would only be the flooded range of the flooded node.

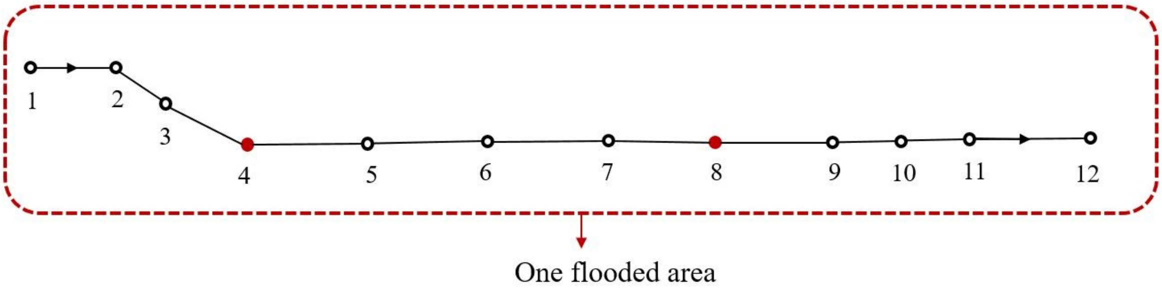


Figure 3.6 An example of generating one flooded area.

An example of generating a flooded area is shown in Figure 3.3. Node 4 and node 8 were two flooded nodes on the route 1-12. For node 4, considering three connected conduits from upstream and four connected conduits from downstream, the flooded range was from node 1 to node 8. Meanwhile, the flooded range of node 8 was node 5-12. Due to the intersection of these two flooded ranges, node 4 and node 8 combined to form one flooded area that included nodes 1-12.

3.1.3 1D and 2D models

Due to the presence of roads, buildings, and other infrastructure, it is sometimes difficult to depict the flow patterns effectively using 1D modeling. A distinction between a 1D model and a 2D model reveals that while performing a 1D model, the water flow is assumed to run in a longitudinal direction in a flood plain or water channels (Rungo and Olesen 2003). Also, the 1D model equations are derived using the law of conservation of mass and energy. In contrast, 2D modeling is performed by integrating the water flow equations over the depth of water flow to find the average velocities of the flowing water at different depths using finite element methods and other relevant numerical methods (Sanders 2007).

In this study, a 1D model of the subject region was converted into a 2D model because 1D models are sometimes inadequate for modeling urban and complex systems. And 2D model was run in the end with optimized solution to see more representative and realistic results. The direct connection method was used to generate the 2D model, which allowed the 1D conveyance network to be directly connected to the 2D floodplain. A boundary layer was necessary as the initial step in constructing a 2D model in order to determine the extent of the 2D model domain. In the model setup, a 2D boundary layer with a 15 m resolution hexagonal mesh and a roughness coefficient of 0.033 was added, while a point layer or so-called 2D node layer was created using the DEM elevation data to represent the floodplain topography. Based on the features of the previously established layers, a 2D mesh layer with 52176 junction points and 2D cells was generated. Using PCSWMM's connection tool, each 1D junction was then connected to the closest 2D junction point for 2D simulation. This connection enables the free transfer of flow from the 1D drainage model to the 2D model for estimating flood extents.

3.1.4 Local design standards

Without prior approval, sewer pipelines should not be surcharged for design flows (e.g., 1:100 year). The maximum 1:100 year HGL shall be at least 1.2 m below the ground surface in cases where surcharge cannot be avoided to preserve ensure catchbasin interception (Senior et al., 2018). Typically, the actual velocity corresponding to the design flow must be greater than 0.60 m/s and less than 3.0 m/s in storm sewers. Supercritical flow should not happen unless considerations for structural stability and longevity are provided in the design. The minimum depth of cover from the pipe obvert to the finished road grade for public storm sewers is 1.20 m. And the minimum depth of cover from the pipe obvert to finished grade for private property connections is 1.00 m (The City of A Water Resources, 2011).

Sewer pipe changes in size, grade, or direction must be made in manholes (MHs), unless the sewer is curved. The maximum distance between MHs must be 185 m, and an MH is always required at the upper end of a sewer for maintenance reasons. If the drop is equal to the difference in pipe diameter, the elevation of the obverts should be kept continuous to maintain the energy gradient. For manhole drops, at MHs where the downstream pipe has a larger diameter than the upstream ones, the drop must be equal to or greater than the difference in pipe diameter. Where no change in pipe diameter occurs, a minimum drop of 30 mm is required in a through MH, and a minimum drop of 60 mm is required in a bend. In general, large drops are discouraged because of hydraulic considerations. For drops greater than 1.0 m, a specially designed drop MH might be necessary to address hydraulic requirements due to the elevation change (The City of A Water Resources, 2011).

3.1.5 Assumptions

This study assumed that the downstream pipe size should not be less than the upstream pipe size in most cases. And the invert at junction and inlet outlet elevation were assumed the same. In addition, the maximum depth of pipe was set as 15m to avoid high construction costs.

3.2 Overall optimization structure

The three essential components of this optimization system were the hydrologic process computation module PCSWMM, the optimization algorithm PICEA-g, and the platform MATLAB which interacts with and integrates all the system's components. The system started with establishing the objective functions (cost and flooded nodes) and decision variables (pipe size and slope). On the basis of the determined range of each decision variable, a user-specified number of candidate solutions were generated.

Because of the large size of the study area, the simulation process took a long time. To improve the computational efficiency of MOEA optimization for UDS designs, parallel computing in MATLAB was applied to this study. Without parallel computing, the original running time for the study area in MATLAB was around 10 days. This period can be shorten to 1-2 days when parallel computing was applied. In addition, divide and conquer technique was used to solve the large-scale model. By downscaling the optimization issue, this technique can achieve better optimization results in a more efficient way and achieve the optimization that would be impossible in a large-scale (when decision variables are too much). Moreover, this study took into account the UDS engineering criterion that a pipe's size should not be less than that of its upstream pipes in most cases.

Scenario 1 uses the PCSWMM 1D modelling results of the study area for optimization and scenario 2 is the condition uses the PCSWMM 2D modelling results of the study area for optimization. These two scenarios are different in junction depth and pipe capacity limited which are used in MATLAB to do pipe optimization. In scenario 1, there are 284 flooded manholes and 342 very limited conduits while the numbers reduce to 73 and 294 respectively in scenario 2. Because in 1D scenario, the flood is only considered in junctions. However, in 2D scenario, the water will spread to the ground surface when the junction is flooded.

Figure 3.4 presents the overall structure of the optimization system for Scenario 1 and Scenario 2. Divide results for each flooded area were obtained from candidate solutions in each scenario. Then results for the whole study area were generated by applying conquer technique. The conquer results would go through the process of evolving (crossover and mutation) and finally optimal solutions were created in 1D model, and the processes were completed in the platform MATLAB. In this study, the final optimal solutions were utilized to run 2D model to compare the results of different scenarios in 2D model. The details of process of divide, conquer and evolving are shown in Figure 3.5, 3.6 and 3.9 respectively.

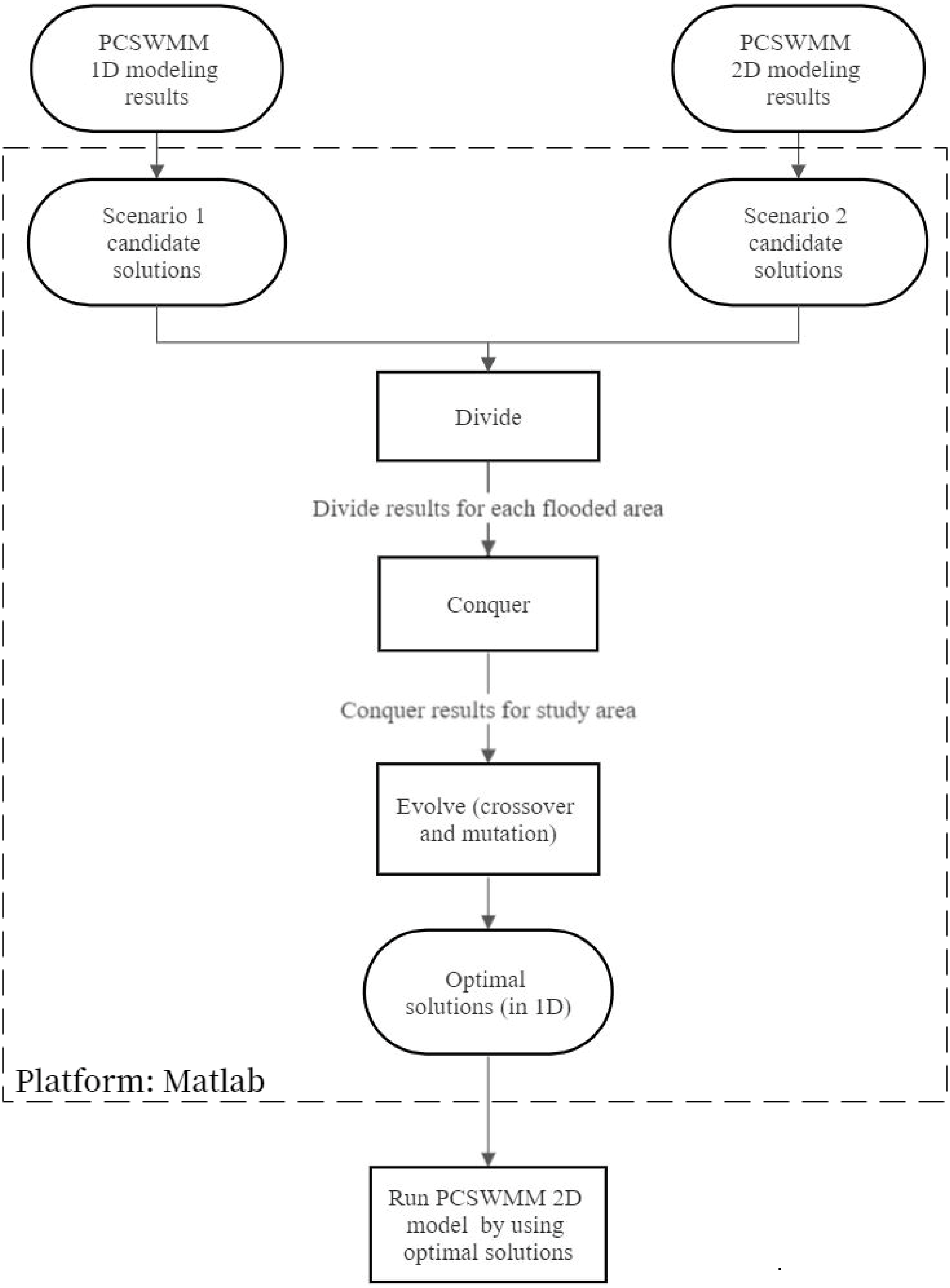

PCSWMM 1D modeling results
PCSWMM 2D modeling results
Scenario 1 candidate solutions
Scenario 2 candidate solutions
Divide
Divide results for each flooded area
Conquer
Conquer results for study area
Evolve (crossover and mutation)
Optimal solutions (in 1D)
Platform: Matlab
Run PCSWMM 2D model by using optimal solutions

Figure 3.7 Structure of optimization system for scenario 1 and scenario 2.

3.3 Divide and conquer technique

3.3.1 Divide

In this study, divide and conquer technique was applied to evolve the optimization results. divide and conquer technique can decompose the optimization process for each individual flooded area, and then combine the divide solutions from each flooded area into new sets of optimal solutions. For the optimization process, there are 37 flooded areas in the study area. For each flooded area, the model simulated 20 divide solutions by changing the pipe slope and size while the rest of the 36 areas remained the same. So, there would be initial 740 divide solutions in total.

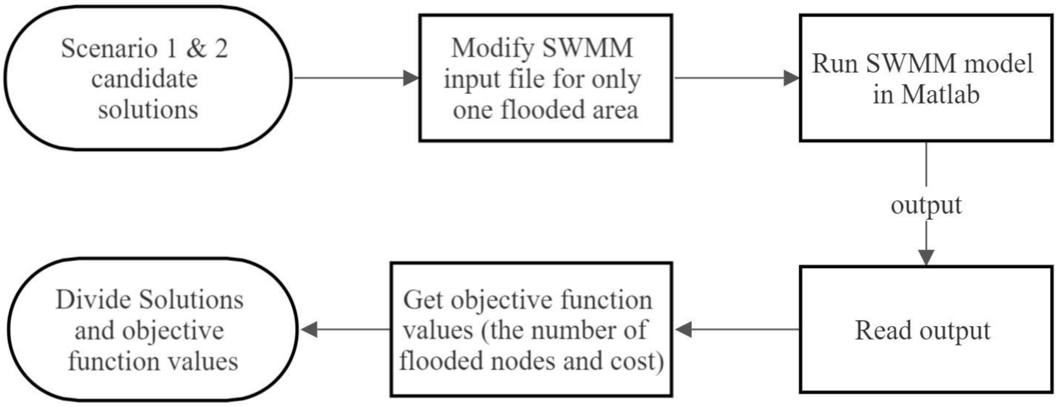

Scenario 1 & 2 candidate solutions
Modify SWMM input file for only one flooded area
Run SWMM model in Matlab
output
Read output
Get objective function values (the number of flooded nodes and cost)
Divide Solutions and objective function values

Figure 3.8 Process of divide for scenario 1 and scenario 2.

The process of divide is shown in Figure 3.5. With the candidate solutions generated in scenario 1 and scenario 2, the total cost of pipe system changing was determined using Table 3.1. The subsequent step was to alter the SWMM input file based on the decision variables of each candidate solution and start SWMM simulation by using the dynamic link library files of SWMM (run one SWMM simulation for each candidate solution). PCSWMM is not an open-source application, however its engine is EPASWMM, therefore its input file can be edited in

MATLAB.

Once the simulation was complete, the objective function value (the number of flooded nodes and cost) was obtained from the SWMM output file. Each candidate solution linked to a distinct SWMM simulation and a distinct set of goal function values. Thus, divide solutions and related objective function values were obtained.

3.3.2 Conquer

The roulette wheel was applied in divide and conquer technique for selecting one divide solution from the entire solution population. To determine the possibility of selecting each area, the equations are written as follows:

$$Avg\_FRR_{DS_{i,m}} = \frac{\max(Avg\_FR_{DS_{i,m}},0)}{Avg\_C_{DS_{i,m}}} \tag{3.1}$$

where DS is the divide solution; m is the number of each area; i is the number of each divide solution. $Avg\_FR_{DS_{i,m}}$ is the average number of flooded node reduction; $Avg\_C_{DS_{i,m}}$ is the average cost for each divide solution and $Avg\_FRR_{DS_{i,m}}$ is the average of flooded node reduction ratio for each divide solution.

$$P(Area_m) = \frac{Avg\_FRR_{DS_{i,m}}}{\sum_{m=1}^{M} Avg\_FRR_{DS_{i,m}}} \tag{3.2}$$

where $P(Area_m)$ is the possibility of selecting each area; M is the maximum number of flooded areas which is 37 in this study area. $\sum_{m=1}^{M} Avg_FRR_{DS_{i,m}}$is the sum of the average flooded node reduction ration in 37 flooded area.

In this study, cost is calculated by Table 3.1 based on pipe size and depth range.

Table 3.2 Cost ($CAD) calculation table with pipe size (mm) and depth ranges in meters (m) (The City of A Water Resources, 2011)

Pipe size (mm)	Depth ranges in meters (m)									
	0-2.5	**2.5-3.0**	**3.0-3.5**	**3.5-4.0**	**4.0-4.5**	**4.5-5.0**	**5.0-5.5**	**5.5-6.0**	**6.0-6.5**	**6.5-7.0**
150	145	149	154	158	163	168	176	185	195	204
200	160	170	180	191	202	214	236	259	285	313
250	160	170	180	191	202	214	236	259	285	313
300	168	181	196	212	229	247	272	299	329	361
375	193	209	225	243	263	284	312	343	378	416
450	213	230	248	268	289	312	343	378	416	457
525	234	255	278	303	330	360	388	420	453	489
600	257	288	323	361	405	453	489	529	571	617
675	283	317	355	397	445	481	519	561	606	654
750	311	342	376	414	456	501	551	606	667	734
900	358	397	441	489	543	603	669	743	825	915
1050	429	494	568	653	751	864	933	1007	1088	1175
1200	515	593	681	784	901	1036	1140	1254	1379	1517
1350	618	711	818	940	1081	1244	1368	1505	1655	1821
1500	742	853	981	1128	1298	1492	1671	1872	2097	2348
1650	890	1024	1178	1354	1557	1791	1970	2167	2384	2622
1800	1068	1282	1539	1539	2216	2659	2925	3217	3539	3893

To determine the possibility of selecting each divide solution DS_i in one flooded area, the equations are:

$$FRR_{DS_i} = \frac{\max\,(FR_{DS_i}, 0)}{C_{DS_i}} \tag{3.3}$$

where FR_{DS_i} is the number of flooded node reduction and C_{DS_i} is the cost for each divide solution. FRR_{DS_i} is the flooded node reduction ratio for each divide solution.

$$P(DS_i) = \frac{FRR_{DS_i}}{\sum_{n=1}^{N} FRR_{DS_i}} \tag{3.4}$$

where $P(DS_i)$ is the possibility of selecting each divide solution DS_i in one flooded area, n is the number of DS in one area and N is the total number of DS in one area which is 20. $\sum_{n=1}^{N} FRR_{DS_i}$is the sum of flooded node reduction ration in 20 divide solutions.

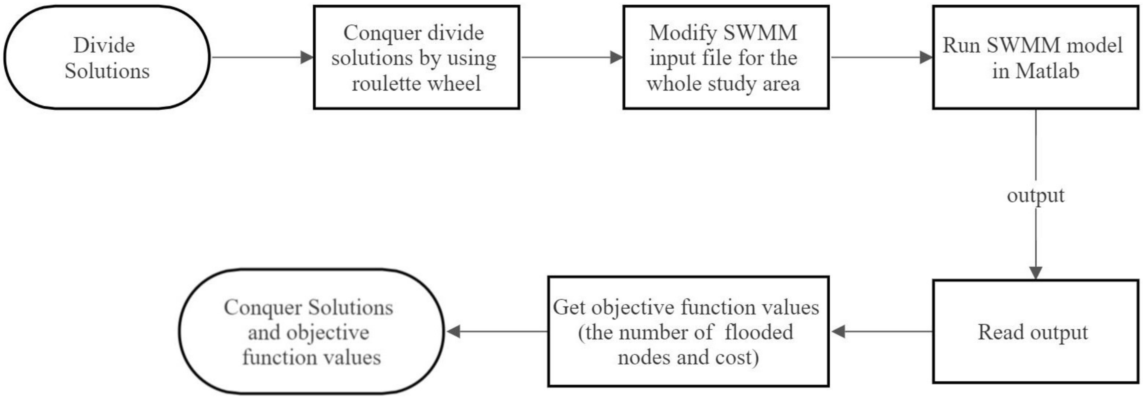

Divide Solutions
Conquer divide solutions by using roulette wheel
Modify SWMM input file for the whole study area
Run SWMM model in Matlab
output
Read output
Get objective function values (the number of flooded nodes and cost)
Conquer Solutions and objective function values

Figure 3.9 Process of conquer for scenario 1 and scenario 2.

As a result, areas and divide solutions with higher ratio of flooded nodes have higher chance of being selected. These selected areas and divide solutions can generate candidate solutions. To some extent, this method ensures that the better divide solutions are more likely to be selected.

Process of conquer is shown in Figure 3.6. The divide solutions were selected by conquer technique using roulette wheel. Then SWMM input file was modified based on conquer solutions for the whole study area. After all, conquer solutions and objective function values were gained through output file of SWMM simulation in MATLAB.

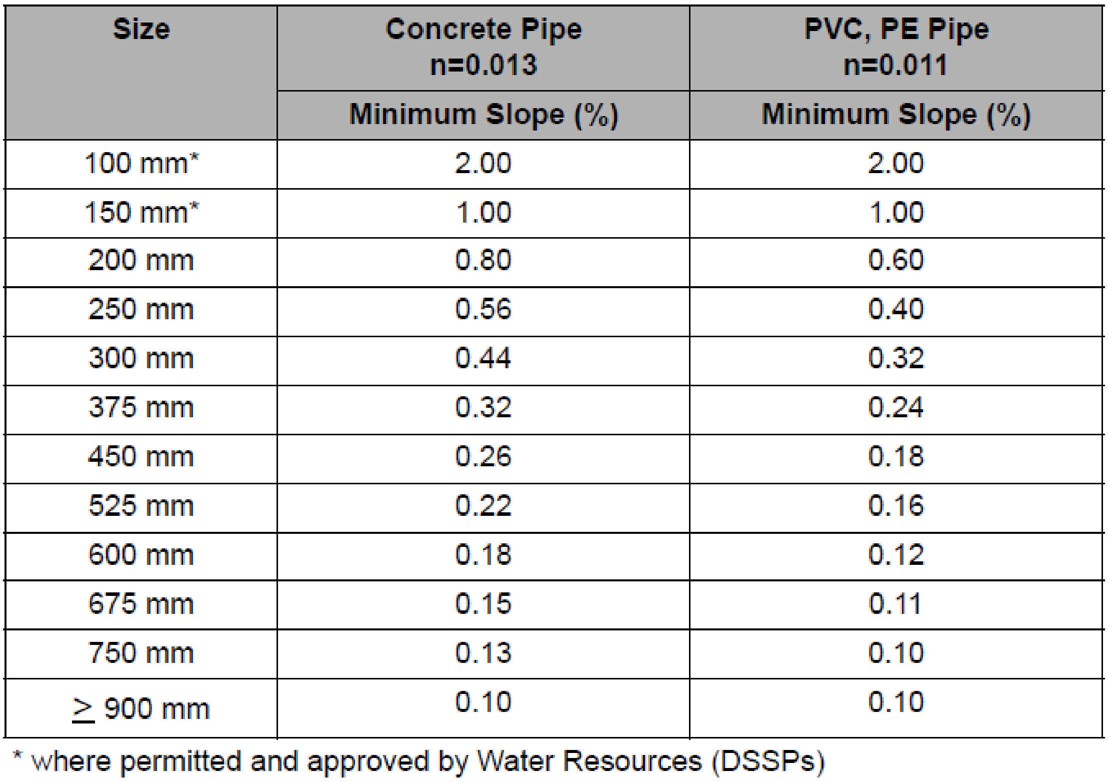
In addition, the method of conquer by selecting results randomly which means each divide solution has the same probability of being selected was also applied in this study. The results obtained by these two different methods are compared in this thesis.

3.4 Improvement to PICEA-g

3.4.1 Initialize candidate solutions

In the original PICEA-g algorithm, the value of each decision variable in the initial set of candidate solutions is created arbitrarily within the decision variable constraints. Diversity and unpredictability of candidate solutions can be obtained when the number of decision variables is small. But when the number is great, the total of decision variables approaches the mean of the sum of choice variable bounds. In this study, the creation of candidate solutions is not as random as in the original algorithm, as random pipe size and slope would result in pipe disconnection and break the technical criteria of the UDS design.

Table 3.3 Minimum slope for different pipe size (The City of A Water Resources, 2011)



Size	Concrete Pipe n=0.013	PVC, PE Pipe n=0.011
	Minimum Slope (%)	Minimum Slope (%)
100 mm*	2.00	2.00
150 mm*	1.00	1.00
200 mm	0.80	0.60
250 mm	0.56	0.40
300 mm	0.44	0.32
375 mm	0.32	0.24
450 mm	0.26	0.18
525 mm	0.22	0.16
600 mm	0.18	0.12
675 mm	0.15	0.11
750 mm	0.13	0.10
≥ 900 mm	0.10	0.10

* where permitted and approved by Water Resources (DSSPs)

To create pipe diameter solutions, the pipe CL was checked in advance. one increment is the increment in pipe size modification in Table 3.2. The probabilities and number of increment were finally decided by the author after testing different numbers.

- If CL >0.01hr, a 40% probability increase for 2 increments and a 60% probability increase for 4 increments were established.
- If CL > 0 and ≤ 0.01 hr, there was a 70% chance of increasing 2 increments and a 30% chance of increasing 3 increments would be set.
- If CL equaled 0 hour, 80% would remain the same, 20% would raise 2 increments.

Looking at all the routes contain the flooded junctions, from downstream, the maximum diameter was figured out, from upstream, the minimum diameter was figured out, meanwhile from the most upstream, the absolute minimum diameter was generated. Then new pipe diameter was created, and it would be checked with the min and max diameter values. Also, downstream pipe size would be equal or larger than upstream pipe size.

To create the slope of pipe solution, minimum slope was used to gain the minimum invert elevation at each junction from downstream and then minimum slope was utilized to determine the maximum invert elevation at each junction from upstream by applying new pipe diameters and Table 3.2. Depending on pipe CL, new elevations at each node could be set. Meanwhile, the elevation of the most upstream node and the most downstream node would not be changed. Also, the elevation range was defined as the difference between node elevation and minimum elevation when slope increased. To pick a random value within the elevation range, the increment = (random (0-1))^2 ×elevation range.

- If the pipe CL > 0.01hr, there was an 80% chance that the slope would grow by 1 increment and a 20% chance that it would remain the same.
- If CL > 0 and ≤ 0.01 hr, a 40% chance of increasing the slope by 0.5 increment and a 60% chance of maintaining the slope would be set.
- If CL equaled 0 hours, 40% would remain the same, 30% would decrease by 0.25

increment, and 30% would increase by 0.25 increment.

Figure 3.7 presents an example of generating a solution of pipe slope. From the most downstream junction 6, the minimum elevation is determined by applying the minimum slope of new pipes while the minimum elevation is determined from the most upstream junction 1. Then, the change of pipe slope would within the maximum and minimum elevations depending on pipe capacity when the elevations of node 1 and node 6 remain stable.

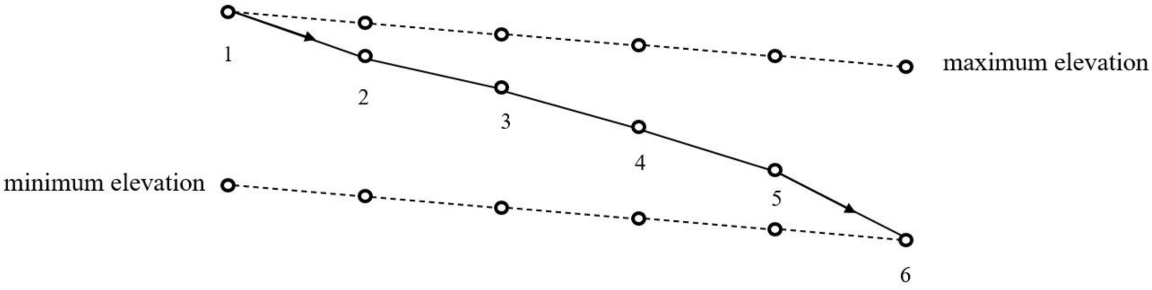


Figure 3.10 An example of generating a solution of pipe slope.

3.4.2 Enhanced goal vectors boundaries – cutting plane

Wang (2013) developed an improved technique termed cutting plane to aid in the development of goal vectors that are more beneficial in directing candidate solutions toward the Pareto optimal front as PICEA-g's performance is influenced by goal vectors, particularly the bounded space of goal vectors.

In this study, the cutting plane was utilized to redefine the bound space of goal vectors, ensuring that no single goal vector either dominated or was dominated by all candidate solutions. Figure 3.8 illustrates an example of an optimization problem involving the minimization of two objectives in order to explain the cutting plane. Goal vectors are created at random within the bound space. In the original PICEA-g algorithm, the bound space of goal vectors is OKHL. The cutting plane divides the bound space from OKHL to DAGBCFM$_{4-1}$E (gray area), where E and F are the extreme points and M$_{1-4}$ are the non-dominated solutions. In the original bound space, any goal vectors that fall within the region OKDEM$_{1\text{-}4}$FL cannot be dominated by any solutions, whereas any goal vectors that fall within the region AHNG could be dominated by all solutions. Therefore, any goal vectors that fall outside the region DAGBCFM$_{4\text{-}1}$E (gray area) are wasted and ineffective. Meantime, objective $f2$ is far from the Pareto optimal front rather than object $f1$ (pick any point along with EF). For example, in this study, $f1$ and $f2$ are determined as the number of flooded node and the cost. Because GBCF's area is larger than DAGE's, more goal vectors fall inside the region GBCF rather than DAGE when using the cutting plane approach, resulting in a greater emphasis on steering solutions towards target $f2$ in the following generation.

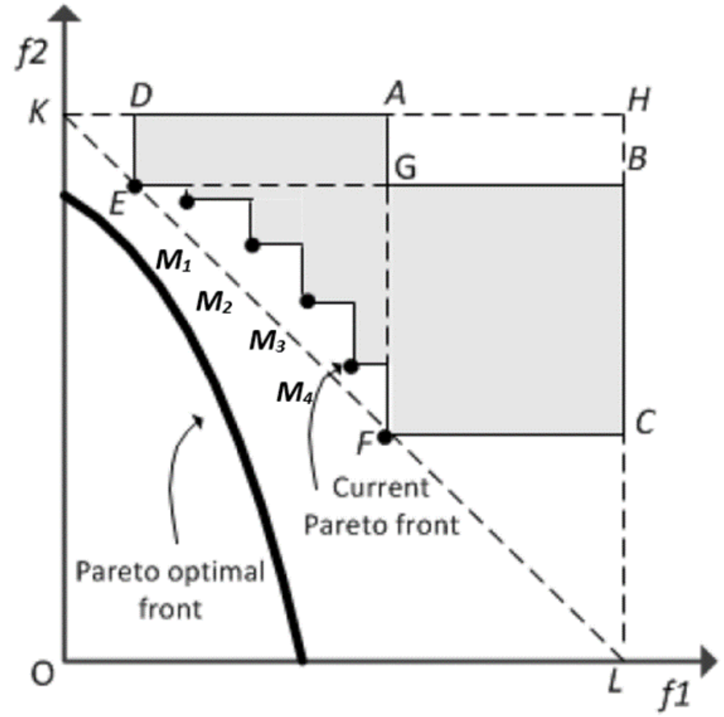


Figure 3.11 An example of two objective problem using the cutting plane (Wang et al., 2013).

3.4.3 Enhanced fitness calculation

The fitness value represents how individual of candidate solution population performs in achieving the Pareto optimal front (Wang et al., 2013). The fitness value (*Fit*) of candidate solutions (*cs*) and goal vectors (*gv*) are calculated as Eq. 3.5-3.7 (Wang et al., 2013):

$$Fit_{cs} = \sum_{g \in G \uplus G_C | s \preccurlyeq g} \frac{1}{n_{gv}} \tag{3.5}$$

$$Fit_{gv} = \frac{1}{1 + \gamma} \tag{3.6}$$

$$\gamma = \begin{cases} 1 & n_{gv} = 0 \\ \dfrac{n_{gv} - 1}{2N - 1} & otherwise \end{cases} \tag{3.7}$$

where *g* and *G* represents goal vectors; G_C is the offspring goal vectors; *s* represents solutions; n_{gv} is the number of solutions that dominate the goal vector *gv*; and *N* is the population size of candidate solutions. One point dominates another point means the value of one point's position in all dimensions are smaller than those of another point when all objectives are minimized. An example of Eq. 3.5-3.7 is shown in Example A1.

Two candidate solutions could have the same fitness value when one of them dominates the other. Therefore, the original approach for calculating fitness cannot sort candidate solutions effectively without taking into account the dominance connections among candidate solutions themselves. Paknejad et al. (2021) developed a new method to improve fitness value calculation (Eq. 3.8-3.9), which was implemented in this study.

$$Fit_{cs} = \sum_{g \in G \uplus G_C | s \preccurlyeq g} \frac{1}{n_{gv}} \times Fit_{gv} + \frac{1}{rank_{cs}} \tag{3.8}$$

$$Rank_{cs} = 1 + P \tag{3.9}$$

where *P* is how many individuals dominate CS_i at the current population. Then, the new fitness values can sort the domination relationships between the candidate solutions. An example of

Eq. 3.8-3.9 is shown in Example A2.

3.4.4 Enhanced genetic operator – crossover and mutation

The evolving process of optimization system is consisting of crossover and mutation. There are three types of crossovers used by PICEA-g (single-point crossover, uniform crossover, and simulated binary crossover) (Deb and Agrawal, 1994). And polynomial mutation is used as genetic operators (Deb et al., 2002) which are crucial for the generation of offspring solutions and the evolution of solutions. Adjustments or the acquisition of various genetic operators are necessary to increase the algorithm's performance for diverse issues and settings (Srinivas and Patnaik, 1994). Utilizing a logistic map and a roulette wheel in crossover and mutation operators was proposed by Paknejad et al. (2021), and it was also applied and adjusted in this study to improve the allocation solutions.

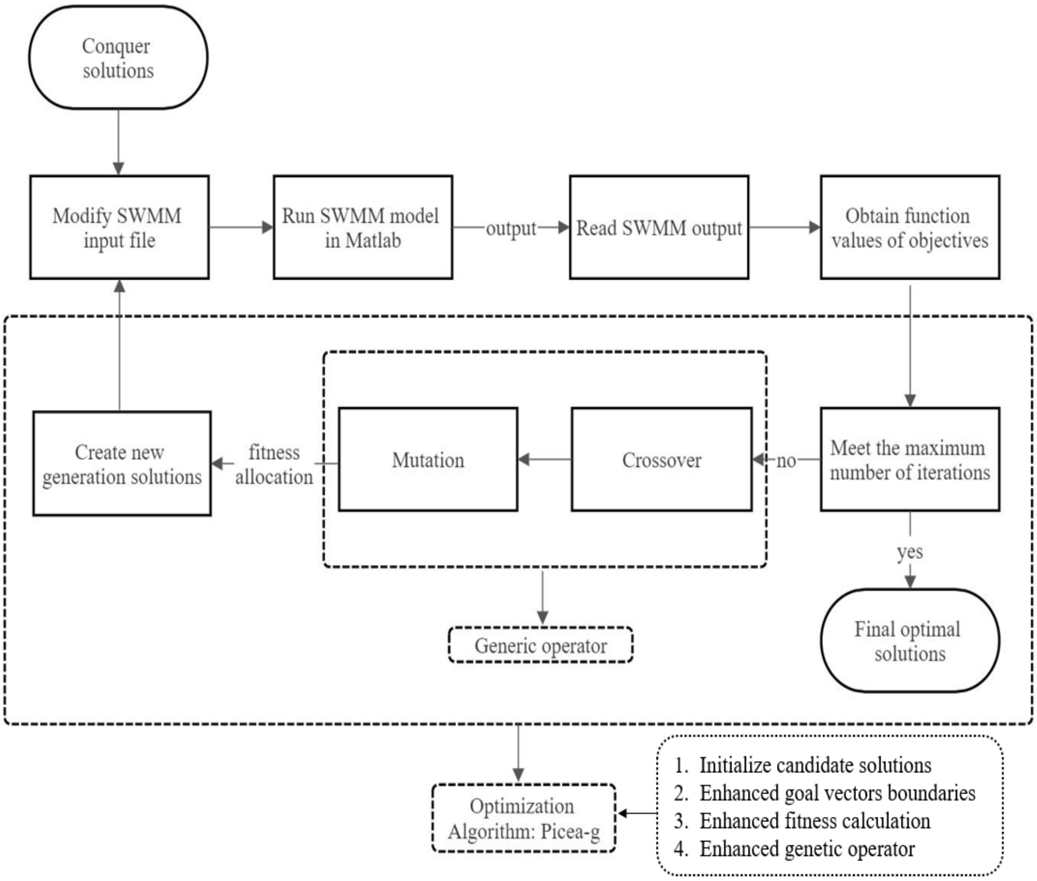

Conquer solutions
Modify SWMM input file
Run SWMM model in Matlab
output
Read SWMM output
Obtain function values of objectives
Create new generation solutions
fitness allocation
Mutation
Crossover
no
Meet the maximum number of iterations
yes
Final optimal solutions
Generic operator
Optimization Algorithm: Picea-g
1. Initialize candidate solutions
2. Enhanced goal vectors boundaries
3. Enhanced fitness calculation
4. Enhanced genetic operator

Figure 3.12 Process of evolving (crossover and mutation) for scenario 1 and scenario 2.

Figure 3.9 shows the evolving process. In the SWMM simulation, the PICEA-g optimization algorithm would generate a set of goal vectors that were utilized to determine the fitness value of the present population of solutions. After sorting the fitness of each individual solution, the algorithm would use genetic operators (crossover and mutation) to evolve the current population of solutions into a better population of solutions in the next generation, which were closer to the Pareto optimal solutions than the initial candidate solutions. The subsequent steps were identical to the preceding steps and created a loop consisting of modifying SWMM inputs, running SWMM simulation, receiving objective function values, and generating new solutions. The loop would end when the maximum number of iterations given in the user's configuration was reached (maximum number of generations in this study was set to 20).

3.4.4.1 Logistic map

May (2004) introduced the logistic map to generate chaotic sequences, and its mathematical equation is as follows:

$$X_{n+1} = rX_n(1 - X_n) \tag{3.10}$$

where X_n is a number bounded on [0, 1], and r is a parameter in the range of [0, 4]. The r value changes the behavior of the logistic sequence, e.g., when $3.57 < r \leq 4$, the logistic sequence develops chaotic behavior. In this study, the value of r is set as 3.75. The logistic map was applied in crossover and mutation operators to increase the solution sets variation. (Paknejad et al., 2021).

3.4.4.2 Roulette wheel

The roulette wheel was applied in crossover operator and divide and conquer technique for selecting one solution set from the entire solution population. Higher fitness value of one solution set means higher chance of being selected. The probability of selecting one CS_i is:

$$P(CS_i) = \frac{Fit_{CS_i}}{\sum_{j=1}^{N} Fit_{CS_j}} \tag{3.11}$$

3.4.4.3 Crossover

At each index, the crossover operator extracts one solution from each of the two selected sets (one pair) of parent solutions and combines them into a single set of offspring solutions. For each set of parent solutions, its paired set of parent solutions were selected by using roulette wheel. When a random number was less than a predetermined crossover probability, this pair of parent solutions would undergo crossover. In this study, the probability to do crossover is 0.75. For each individual index of the new offspring solution, if the generated chaotic sequence by the logistic map was $\leqslant 0.75$, this individual offspring solution was extracted from the first parent solution at the same index. Otherwise, it was removed from the second parent solution.

3.4.4.4 Mutation

With the enhanced mutation operator, solutions in this study were guided to implement at a lower cost. As the same as crossover, the probability for each offspring to do mutation is also 0.75. To reduce the cost, the mutation operator increased the flooded areas by choosing new flooded areas from original large flooded areas. For example, the maximum connected conduits were 4 when generating the original flooded areas, while the maximum connected conduits were 2 when choosing new flooded areas by using mutation operator. As a result, one original flooded area can be divided to several small flooded areas to do pipe optimization by changing less pipe diameter and slope. The changing of flooded areas was chosen when the logistic map was $\leqslant 0.5$.

3.4.4.5 Algorithm stopping criteria

There are four conditions outlined by Martí et al. (2016) which MOEA iterations should end if any of the following apply: 1) the present solution is close to optimal; 2) the present solution is acceptable, and no improvement can be made by additional iterations; 3) MOEA cannot converge to any solution; or 4) computation time is adequate. To further justify that the model terminates after 20 generations, the mutual dominance rate indicator (MDR) is used to identify the stopping condition (Martí et al., 20). The equation reads as follows:

$$I_{mdr}({P^*}_t, {P^*}_{t-1}) = \frac{|\Delta({P^*}_{t-1}, {P^*}_t)|}{|{P^*}_{t-1}|} - \frac{|\Delta({P^*}_t, {P^*}_{t-1})|}{|{P^*}_t|} \tag{3.12}$$

where t is the iteration number, ${P^*}_t$ are the solution sets at t_{th} iteration, $|{P^*}_t|$ is the number of ${P^*}_t$ elements, and $\Delta({P^*}_{t-1}, {P^*}_t)$ gives the elements of ${P^*}_{t-1}$ that are dominated by at least one element of ${P^*}_t$. I_{mdr} is the progress (MDR) indicator that provides information regarding the improvement of solutions in each iteration.

4 Results and discussion

4.1 Test area

Before optimization, there were 15 flooded manholes and 7 high-risk manholes in the test area. Using the method of finding flooded area, all these flooded manholes were determined to be within one flooded area. The number of both flooded and high-risk manholes reduced to 0 after optimization. For very limited conduits, the number dropped from 7 to 0 as well. Figures 4.1 and 4.2 show the test area before and after optimization respectively.

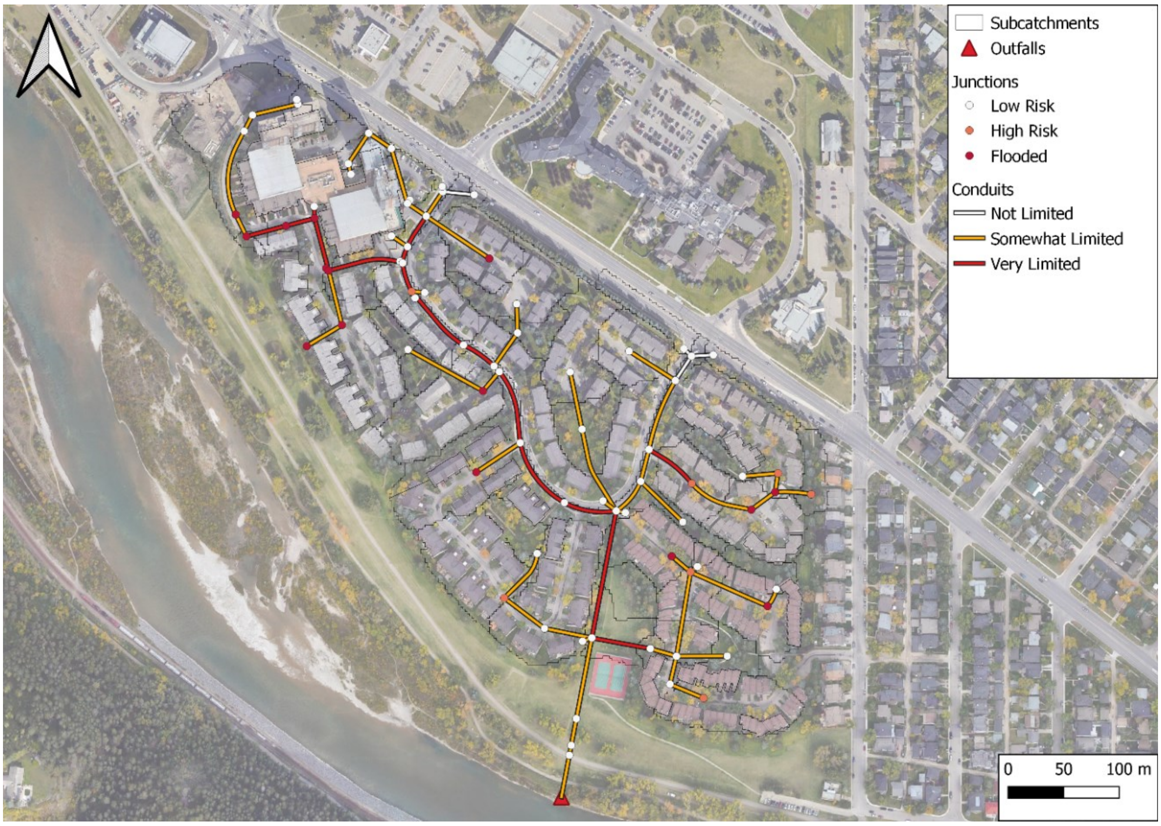


Figure 4.13 Before optimization: existing manhole flooding risks and storm sewer conduit (pipe) conditions in the test area.

Choosing one route (green dash line in Figure 4.3) to check the change between existing and optimal, the profiles of both were generated. Figure 4.3 is the profile of the existing route and Figure 4.4 is the profile of the optimal route. In the existing profile, there were 5 flooded manholes whose HGL was above the ground elevation and these flooded manholes are manholes whose HGL was above the ground elevation and these flooded manholes are concentrated upstream of the pipe routes.

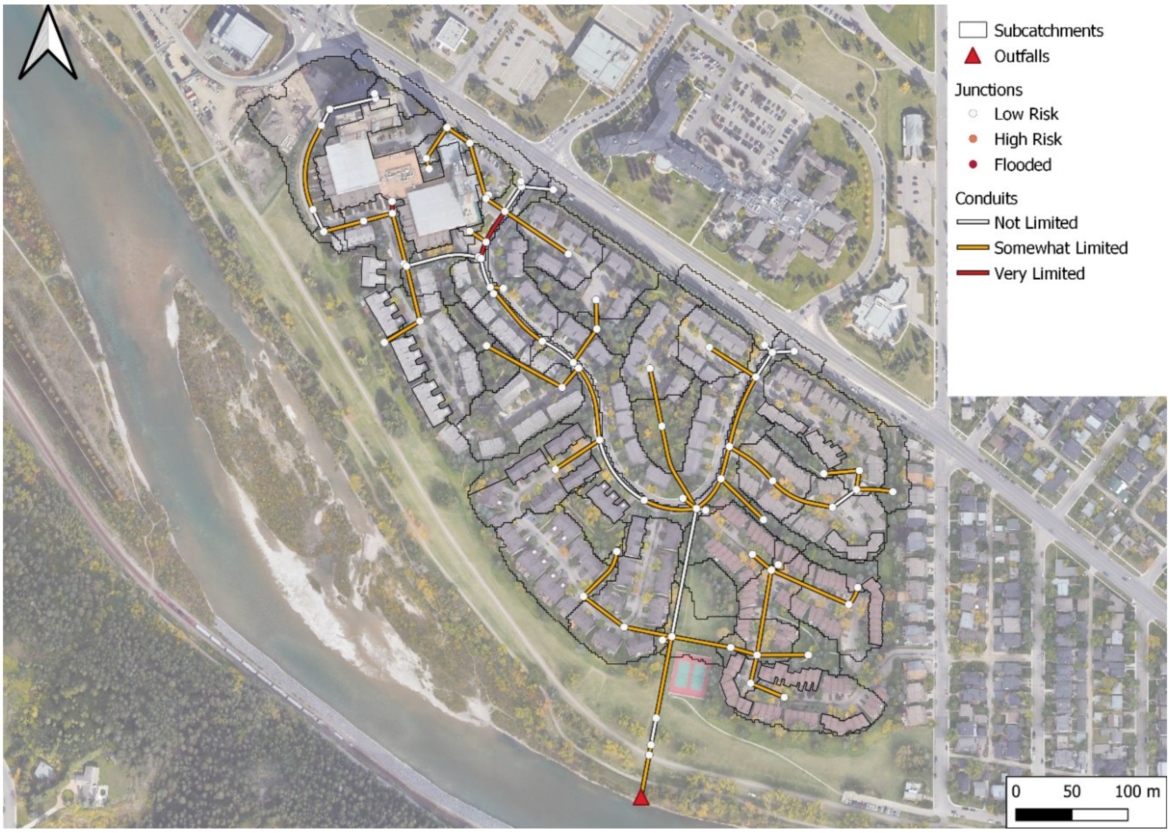
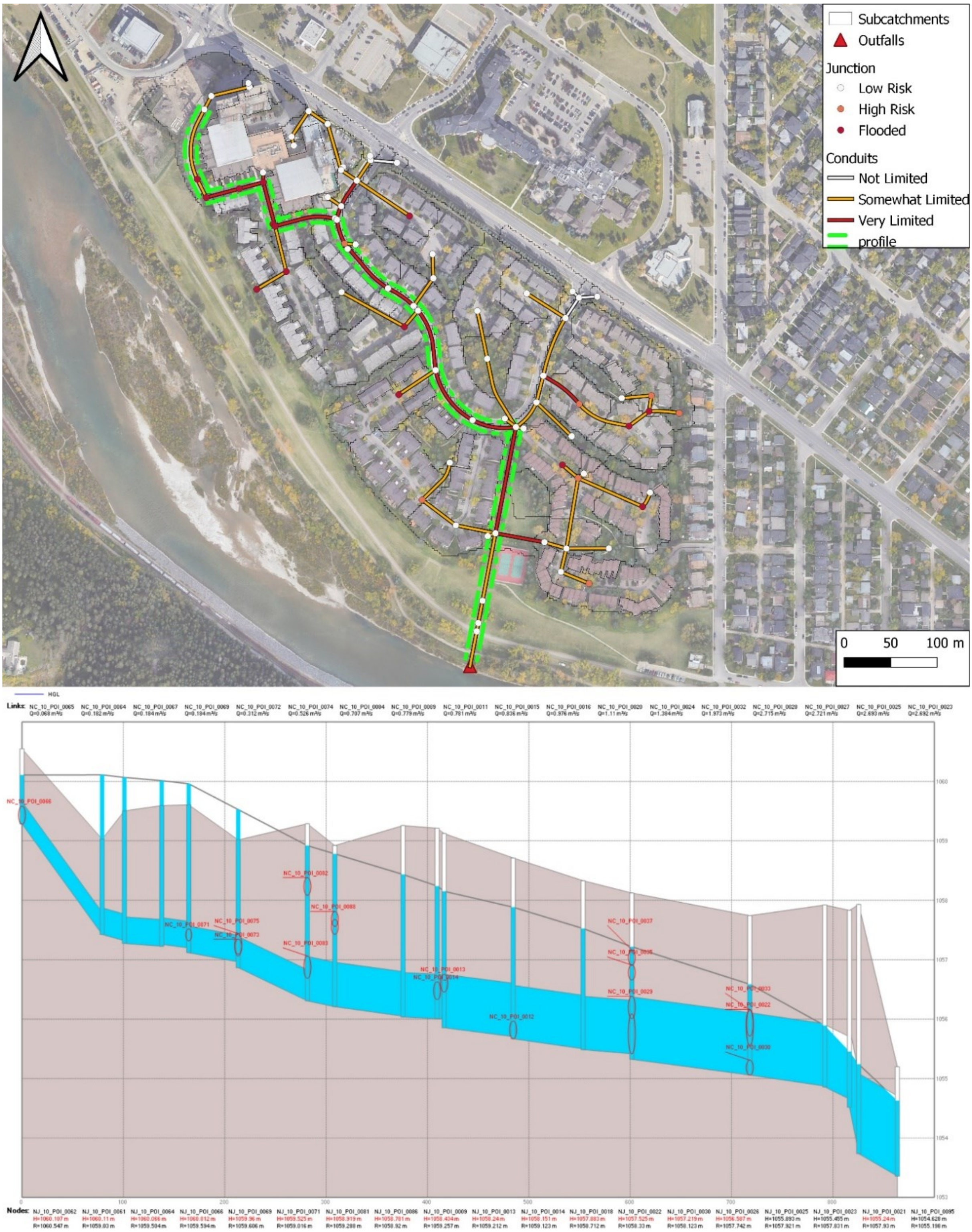


Figure 4.14 After optimization: manhole flooding risks and storm sewer conduit (pipe) conditions in the test area.

As it can be seen from the profiles, most of the pipe size were increased to reduce the effect of flooding. Obviously, the diameter of the most upstream conduit 1 (NC_10_POI_0065) was increased from to 0.375m to 0.525m. Then, from conduit 2 (NC_10_POI_0064) to conduit 3 (NC_10_POI_0072) the diameter was raised from 0.45m to 0.675m. For conduit 4 (NC_10_POI_0074), the diameter was changed from 0.525m to 0.9m. The diameter of conduit 5 (NC_10_POI_0084) to conduit 6 (NC_10_POI_0011) was 0.75 in existing model, and it turned to 0.9m after optimization. For the very limited conduit route conduit 7 (NC_10_POI_0016) to conduit 8 (NC_10_POI_0024), the size increased by 0.15m. The diameter of downstream conduit 9 (NC_10_POI_0032) to conduit 10 (NC_10_POI_0027) grew from 1.05m to 1.5m. And the slope of conduit 11 (NC_10_POI_0027) and conduit 12 (NC_10_POI_0025) had a significant growth, while the elevation of the most upstream and downstream junction kept the same as before. As a result, the number of flooded nodes dropped to 0 after optimization. Due to the significant improvement of pipe optimization, the optimization system was applied to the whole study area to investigate the effect in large scale. Meanwhile, overdesign was observed after optimization especially for conduit 9, 10 and 11 which diameters increased from 1.05m to 1.5m. For the test area, solution with lowest flooded node was determined as optimal solution while cost was not considered and this solution have not been evolved, so overdesign can be found in some solutions. The tables of conduits in selected sewer trunk were presented in table A1 and table A2.

Figure 4.5 presents the objective function values (number of flooded nodes and cost) of each solution. These solutions were not close to the Pareto optimal front as divide and conquer technique and evolving process were not applied in test area due to the only one flooded area. For solutions with 0 flooded node, the cost ranged from 0.6×10^6 CAD$ to 0.7×10^6 CAD$. Also, these solutions have not been evolved.

Yazdi et al. (2016) studied on hydraulic rehabilitation of UDS network by using NSGA-II to change pipe size. The case study area consisted of 101 pipes and nodes, and 102 subcatchments which were similar to the test area in this study. From the results of optimal solutions, the flooded volume decreased from 11m^3 to 7m^3 when the cost was approximately 1.2×10^6 CAD$ (Yazdi et al., 2016). Lin et al. (2020) applied EBDM to do pipe optimization of UDS which case study had a drainage area of 29 ha consisting of 53 subcatchments. The results showed that the cost was about 1.4×10^6 CAD$ while the flooded volume decreased from 15m^3 to 0 m^3 (Lin et al., 2020). However, in this sample test, the flooded volume dropped from 53 m^{3}to 1 m^3 while the cost was around 0.6×10^6 CAD$ which was half as other studies. The comparison indicated a significant improvement in UDS network optimization in this study as the reduction of flooded volume was obvious while the cost was minimal.



Subcatchments
Outfalls
Junction
Low Risk
High Risk
Flooded
Conduits
Not Limited
Somewhat Limited
Very Limited
profile
0
50
100 m
HGL
Links:
NC_10_POI_0065 Q=0.068 m³/s
NC_10_POI_0064 Q=0.182 m³/s
NC_10_POI_0067 Q=0.184 m³/s
NC_10_POI_0069 Q=0.184 m³/s
NC_10_POI_0072 Q=0.312 m³/s
NC_10_POI_0074 Q=0.526 m³/s
NC_10_POI_0084 Q=0.707 m³/s
NC_10_POI_0089 Q=0.779 m³/s
NC_10_POI_0011 Q=0.781 m³/s
NC_10_POI_0015 Q=0.836 m³/s
NC_10_POI_0016 Q=0.976 m³/s
NC_10_POI_0020 Q=1.11 m³/s
NC_10_POI_0024 Q=1.384 m³/s
NC_10_POI_0032 Q=1.973 m³/s
NC_10_POI_0028 Q=2.715 m³/s
NC_10_POI_0027 Q=2.721 m³/s
NC_10_POI_0025 Q=2.693 m³/s
NC_10_POI_0023 Q=2.692 m³/s
NC_10_POI_0066
1060
1059
1058
1057
1056
1055
1054
1053
0
100
200
300
400
500
600
700
800
Nodes:
NJ_10_POI_0062 H=1060.107 m R=1060.547 m
NJ_10_POI_0061 H=1060.11 m R=1059.03 m
NJ_10_POI_0064 H=1060.066 m R=1059.504 m
NJ_10_POI_0066 H=1060.012 m R=1059.594 m
NJ_10_POI_0069 H=1059.96 m R=1059.606 m
NJ_10_POI_0071 H=1059.525 m R=1059.016 m
NJ_10_POI_0081 H=1058.919 m R=1059.288 m
NJ_10_POI_0086 H=1058.781 m R=1058.92 m
NJ_10_POI_0009 H=1058.434 m R=1059.257 m
NJ_10_POI_0013 H=1058.24 m R=1059.212 m
NJ_10_POI_0014 H=1058.151 m R=1059.123 m
NJ_10_POI_0018 H=1057.883 m R=1058.712 m
NJ_10_POI_0022 H=1057.525 m R=1058.33 m
NJ_10_POI_0030 H=1057.219 m R=1058.123 m
NJ_10_POI_0026 H=1056.587 m R=1057.742 m
NJ_10_POI_0025 H=1055.893 m R=1057.921 m
NJ_10_POI_0023 H=1055.455 m R=1057.831 m
NJ_10_POI_0021 H=1055.24 m R=1057.93 m
NJ_10_POI_0095 H=1054.628 m R=1055.198 m

Figure 4.15 Before optimization: the profile of the selected storm sewer trunk in the test area.

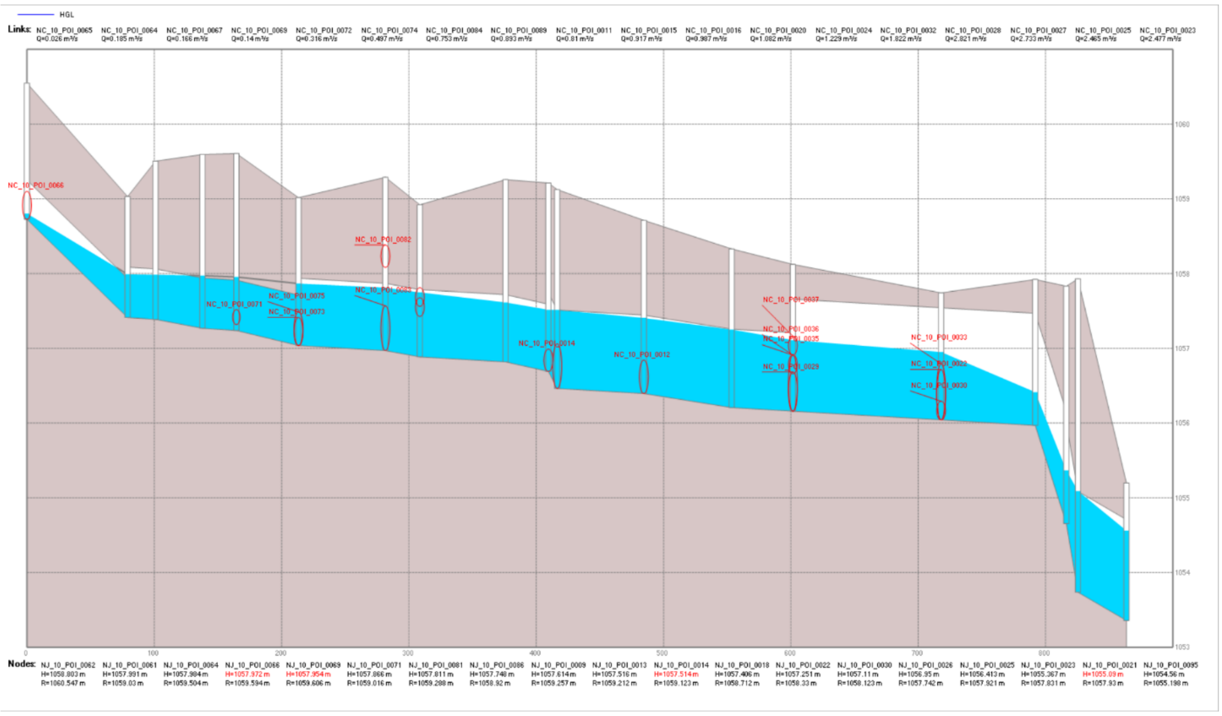

HGL
Links:
NC_10_POI_0065 Q=0.026 m³/s
NC_10_POI_0064 Q=0.185 m³/s
NC_10_POI_0067 Q=0.166 m³/s
NC_10_POI_0069 Q=0.14 m³/s
NC_10_POI_0072 Q=0.316 m³/s
NC_10_POI_0074 Q=0.497 m³/s
NC_10_POI_0084 Q=0.753 m³/s
NC_10_POI_0089 Q=0.893 m³/s
NC_10_POI_0011 Q=0.81 m³/s
NC_10_POI_0015 Q=0.917 m³/s
NC_10_POI_0016 Q=0.987 m³/s
NC_10_POI_0020 Q=1.082 m³/s
NC_10_POI_0024 Q=1.229 m³/s
NC_10_POI_0032 Q=1.822 m³/s
NC_10_POI_0028 Q=2.821 m³/s
NC_10_POI_0027 Q=2.733 m³/s
NC_10_POI_0025 Q=2.465 m³/s
NC_10_POI_0023 Q=2.477 m³/s
NC_10_POI_0066
NC_10_POI_0082
NC_10_POI_0075
NC_10_POI_0071
NC_10_POI_0073
NC_10_POI_0014
NC_10_POI_0012
NC_10_POI_0037
NC_10_POI_0036
NC_10_POI_0035
NC_10_POI_0029
NC_10_POI_0033
NC_10_POI_0030
1060
1059
1058
1057
1056
1055
1054
1053
0
100
200
300
400
500
600
700
800
Nodes:
NJ_10_POI_0062 H=1058.803 m R=1060.547 m
NJ_10_POI_0061 H=1057.991 m R=1059.03 m
NJ_10_POI_0064 H=1057.984 m R=1059.504 m
NJ_10_POI_0066 H=1057.972 m R=1059.594 m
NJ_10_POI_0069 H=1057.954 m R=1059.606 m
NJ_10_POI_0071 H=1057.866 m R=1059.016 m
NJ_10_POI_0081 H=1057.811 m R=1059.288 m
NJ_10_POI_0086 H=1057.748 m R=1058.92 m
NJ_10_POI_0009 H=1057.614 m R=1059.257 m
NJ_10_POI_0013 H=1057.516 m R=1059.212 m
NJ_10_POI_0014 H=1057.514 m R=1059.123 m
NJ_10_POI_0018 H=1057.406 m R=1058.712 m
NJ_10_POI_0022 H=1057.251 m R=1058.33 m
NJ_10_POI_0030 H=1057.11 m R=1058.123 m
NJ_10_POI_0026 H=1056.95 m R=1057.742 m
NJ_10_POI_0025 H=1056.413 m R=1057.921 m
NJ_10_POI_0023 H=1055.367 m R=1057.831 m
NJ_10_POI_0021 H=1055.09 m R=1057.93 m
NJ_10_POI_0095 H=1054.56 m R=1055.198 m

Figure 4.16 After optimization: the profile of the selected storm sewer trunk in the test area.

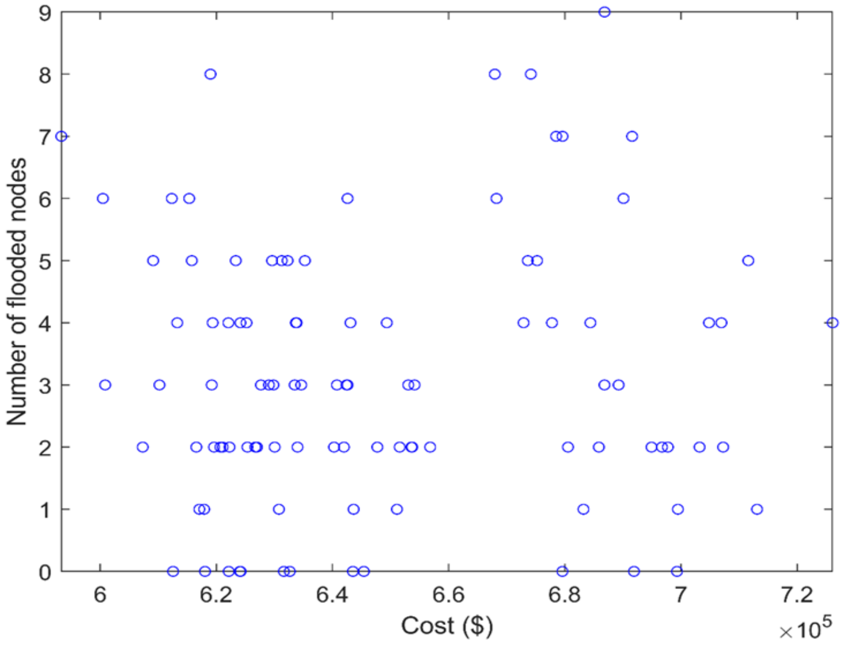


Figure 4.17 The objective function values of each solution – the number of flooded nodes (manholes) vs. cost.

4.2 The entire study area

4.2.1 Results based on 1D model

284 flooded manholes and 47 high-risk manholes were founded in this study area’s 1D model (scenario 1). All these flooded manholes generated 37 flooded areas which are shown in Figure 4.6. One of the optimal solutions with less flooded nodes was selected as an example of the final optimal solution and the results below were based on this final solution. After optimization, the number of flooded nodes decreased to 115 and the number of high-risk nodes dropped to 38 (Figure 4.7).

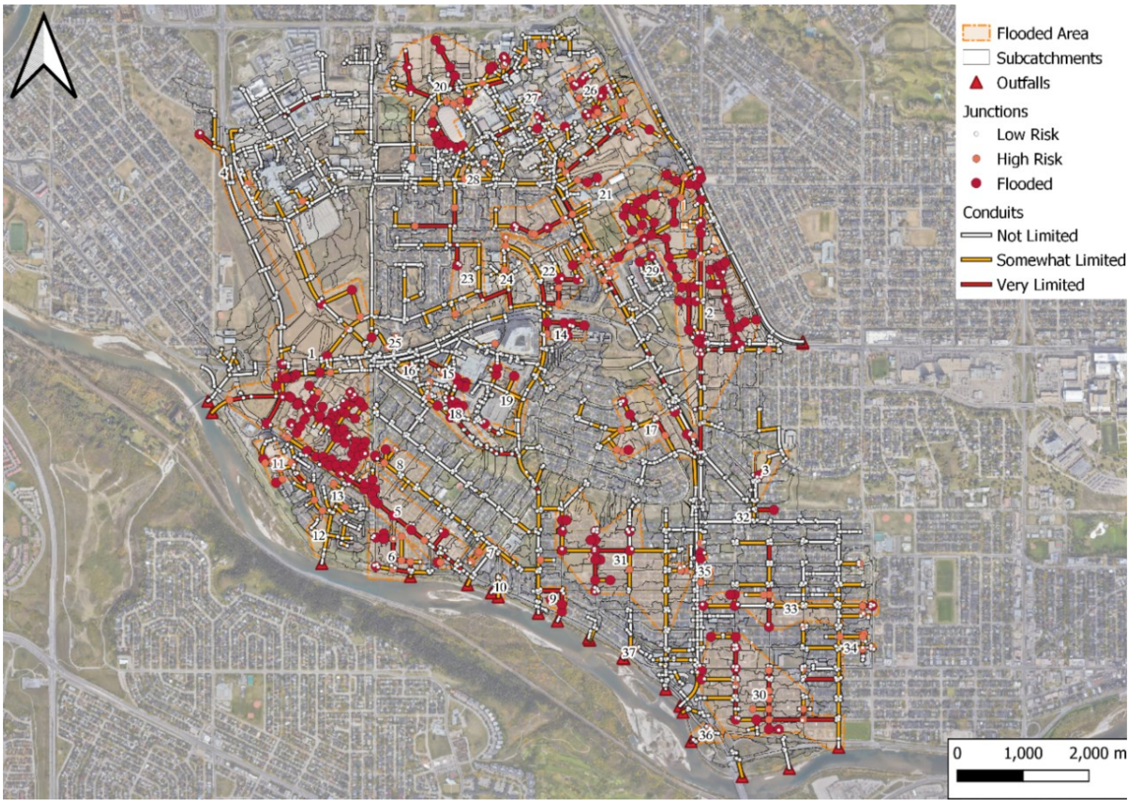


Figure 4.18 Before optimization: flooded areas in the study area based on the 1D model.

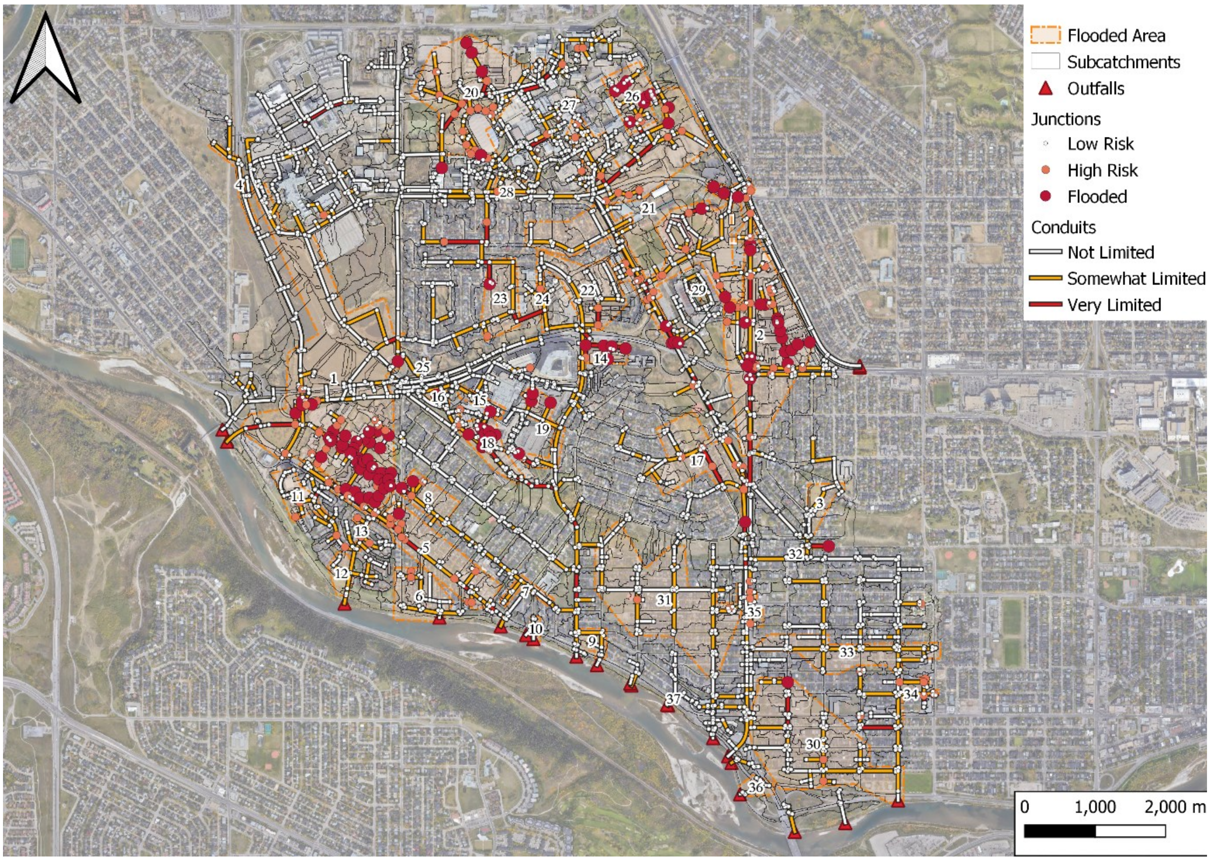


Figure 4.19 After optimization: flooded areas in the study area based on the 1D model.

There was an obvious reduction of flooded nodes in flooded area 1,2 and 21. Figure 4.8 shows the comparison in area 1 where the flooded nodes reduced from 68 to 32 and pipe size variation is also indicated.

The profile, including the most upstream conduits 13 (NC_10_PKD_0248) to the most downstream conduits 14 (NC_10_MON_0039) which reflected the change of pipe slope and size, are presented in Figure 4.9 (existing) and Figure 4.10(optimal). There was a total 15 of flooded nodes before originally and the number dropped to 8 after optimization in this profile. The invert elevation of the most upstream junction 1 (NJ_10_PKD_0258) and most downstream junction 2 (NJ_10_MON_0076) remained the same while the pipe size and slope were changed in this area. All the inlet and outlet elevations of conduits were set to the invert of the junctions. Comparing the profiles, all of the conduits' diameter increased in different degrees. For instance, the diameter of conduit 13 (NC_10_PKD_0248) increased from 0.375m to 0.6m and the diameter of conduit 15 (NC_10_PKD_0246) increased from 0.6m to 1.05m. The CL of conduit 13 and conduit 15 was 0.01 hr and 0.03 hr. Thus, there were 3 increments increasing in conduit 13 and 4 increments increasing in conduit 15. Table A3 and Table A4 present the length, diameter, and slope of conduits in selected sewer trunk.

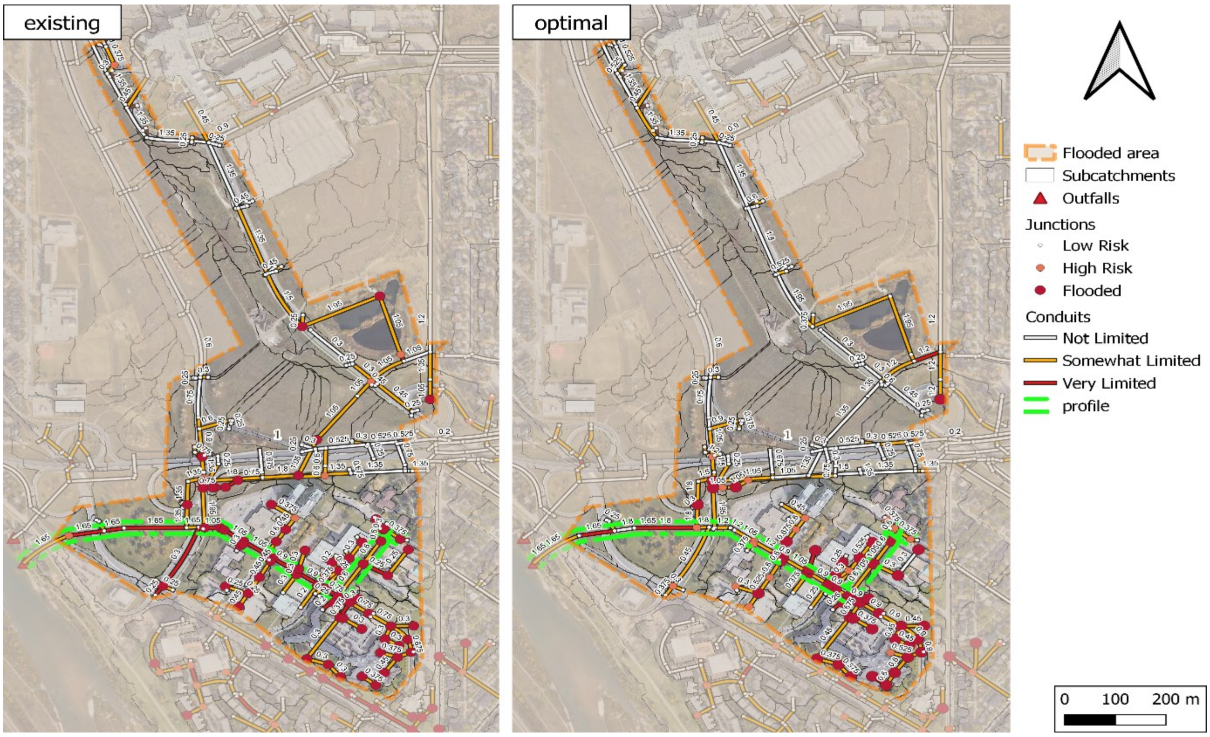


Figure 4.20 The comparison of Flooded Area 1 in the study area before (left) and after (right) optimization, using the 1D model (Scenario 1). The selected route (in green color) is used for generating the profile.

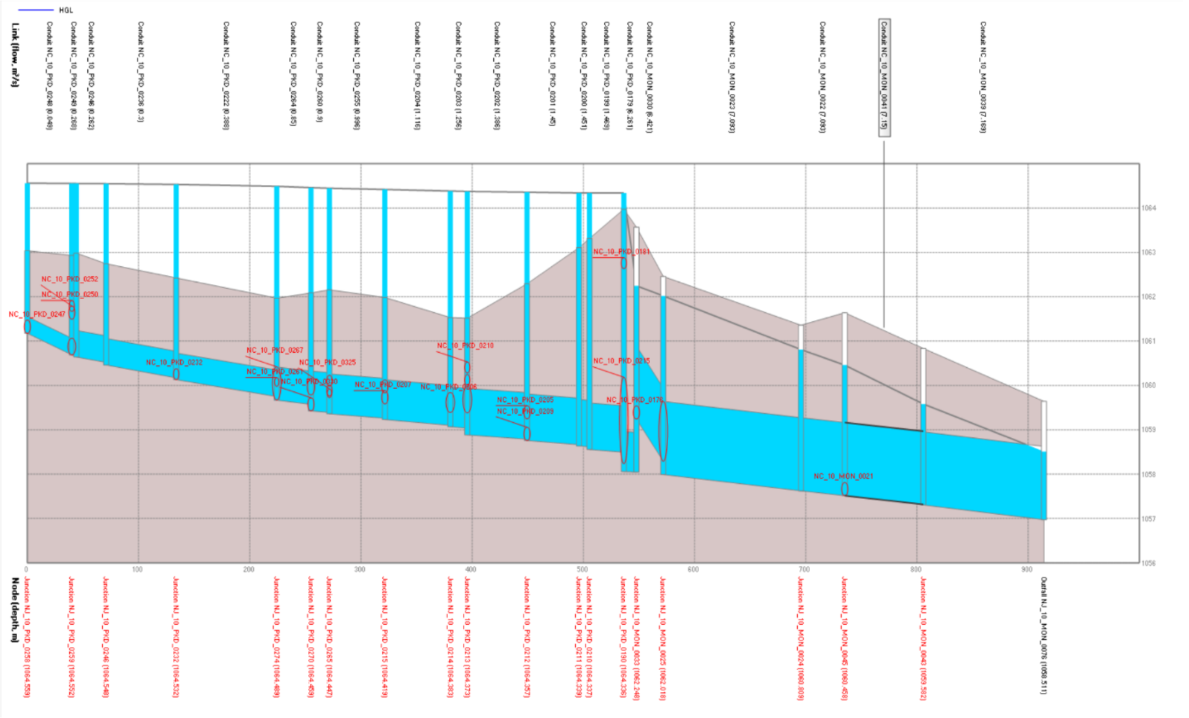


Figure 4.21 Before optimization: the profile of selected sewer trunk in Flooded Area 1of the study area, using the 1D model.

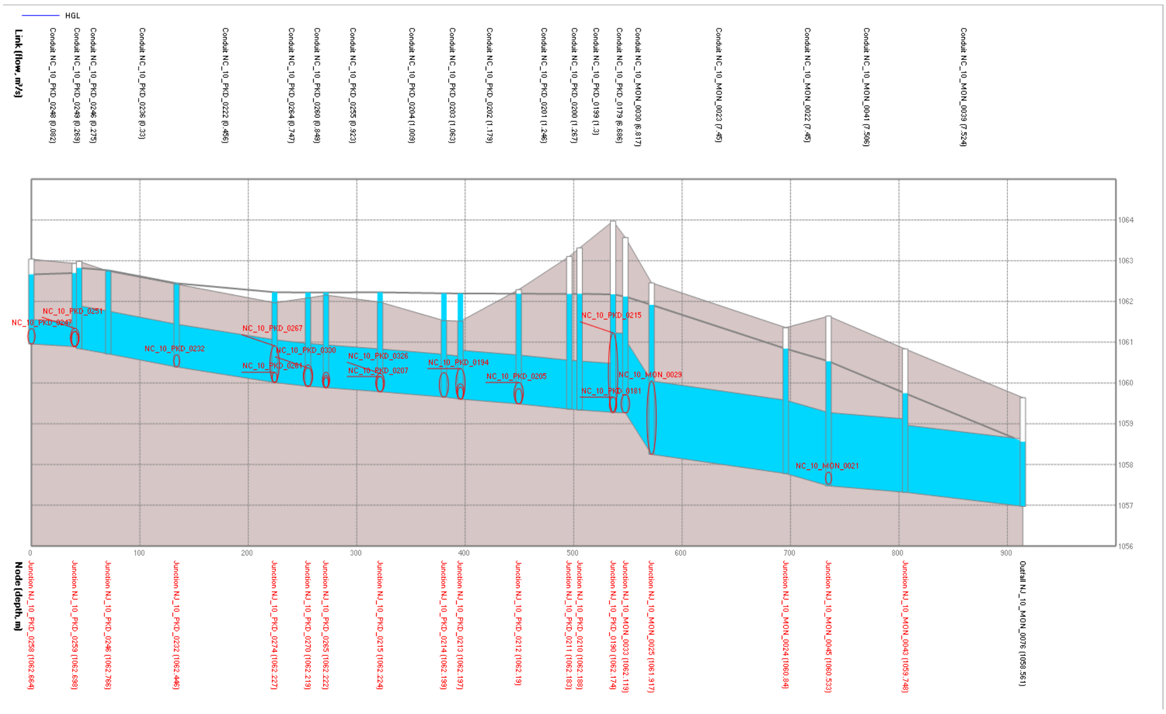


Figure 4.22 After optimization: the profile of selected sewer trunk in Flooded Area 1of the study area, using the 1D model.

By using divide and conquer technique, the study area was divided into 37 flooded areas

(Figure 4.6). For each flooded area, the model simulated 20 divide solutions by changing the pipe slope and diameter while the rest of 36 areas remained the same. Figure 4.11 presents the average reduction number of flooded nodes in the whole study area when each flooded area was optimized. For instance, there were 27 reduced flooded nodes which was the average of 20 divide solutions only when flooded area 1 was optimized and area 2 - area 37 were not changed.

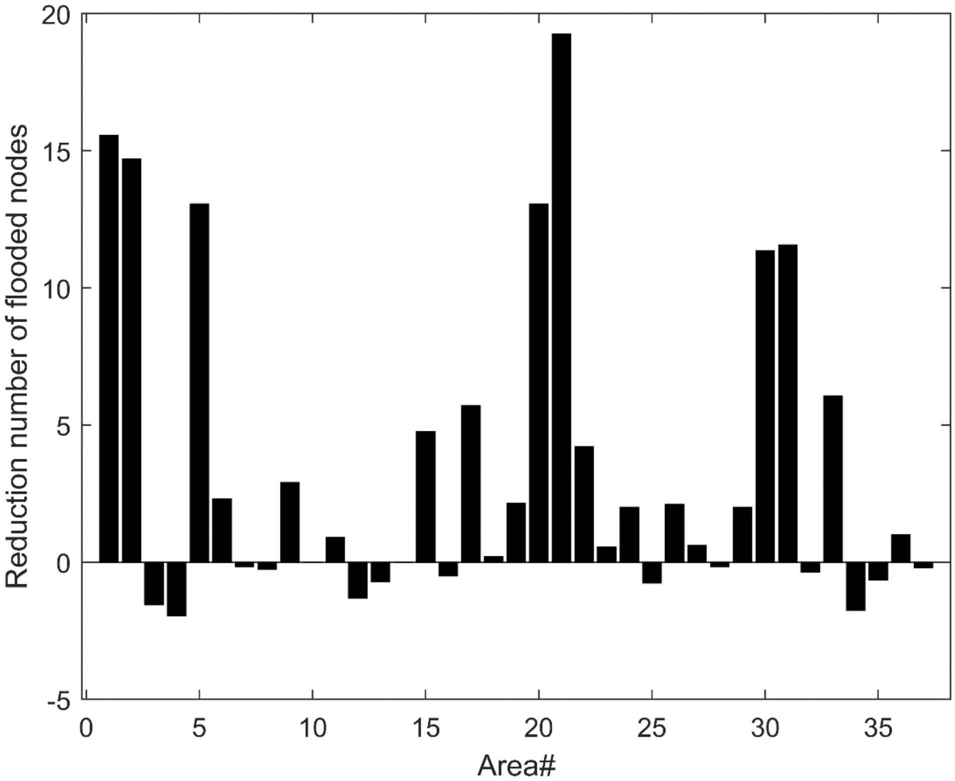


Figure 4.23 The average number of flooded nodes reduction in each area (1D model).

One factor that influenced the reduction number of flooded nodes was the original number of flooded nodes. For example, the optimization to area 1, area 2 and area 21 was much effective in the reduction of total flooded nodes as these areas had large number of flooded nodes before optimization. Area 21 had 28 flooded nodes before optimization and the average reduction was 19 in this area. For areas including few flooded nodes such as area 10,16, 28 and 37 before optimization, the flooded nodes were found not to be reduced on average after optimization.

The other factor was the location of the area. For instance, area 26 was located upstream, and there were several flooded areas located downstream of area 26. Area 17 was also located upstream, but there was no flooded area located downstream. The average reductions of flooded nodes were 2 and 6 in area 26 and area 17 respectively. Besides, the optimization to some areas such as area 3, 4, 7, 8, 12, 13, 16, 25, 28, 32, 34, 35 and area 37 resulted in an increase in the flooded nodes numbers as some changes in pipe size and slope may lead to new flooded junctions in other related flooded areas. For example, area 3 and area 4 were located upstream, and the optimization to these two areas could lead to an increase in flooded nodes in the downstream area.

Looking through the 37 flooded areas, area 4, 7, 13, 14, 19, 23, 24, 28, 32 and 34 were not optimized in the final optimal solution. Among these areas, the average number of flooded nodes was rising due to the optimization of area 4, 7, 13 25, 28 and 34. Also, some of these areas such as area 14, 23, 24 and 32 only contained a small number of conduits and junctions so the optimizations in these areas do not significantly affect the overall flooded number of total study area. Considering the fitness of candidate solutions, these areas were determined not to be optimized in mutation process so as to reduce the number of pipes that need to be changed and control the cost.

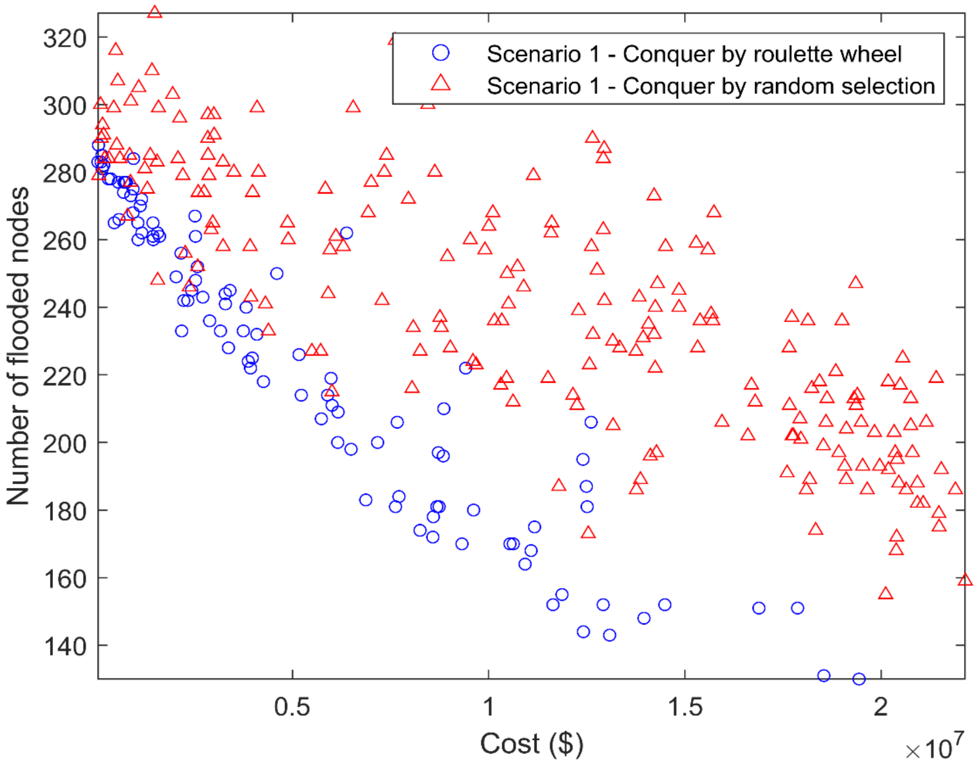


Figure 4.24 The objective function values of solutions (number of flooded nodes vs. cost), obtained by conquer with roulette wheel and by conquer randomly using the 1D model (Scenario 1).

Figure 4.12 describes the difference between conquer technique which using roulette wheel and conquer randomly. From the performance of the total of 200 candidate solutions, it is clear that conquer using roulette wheel was more effective than conquer randomly. For random conquer, there were even more flooded nodes than the initial 284 flooded nodes in some candidate solutions and the costs of most candidate solutions with less flooded nodes (160-200) are above 1.7×10^7 CAD. Meanwhile, if roulette wheel was applied, a high percentage of candidate solutions reduced the number of flooded nodes while the lowest number of flooded nodes was below 120 which was much lower than that in random conquer candidate solutions. Also, in this situation, solutions with 160- 200 flooded nodes only costed 0.5×10^7 – 1.0×10^7 CAD which was more economical than conquer randomly. The reason is, by using roulette wheel, candidate solutions with higher fitness value were more likely to be selected while random conquer do not considered the fitness value and selected each candidate solution with same probability.

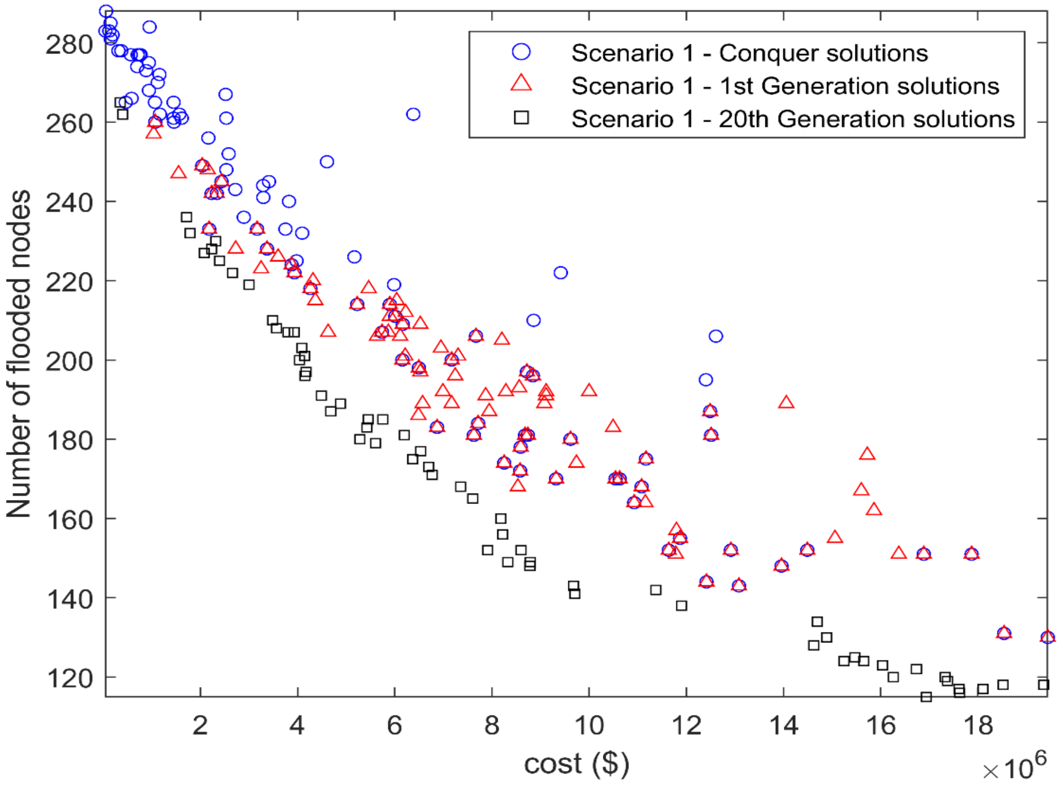


Figure 4.25 The objective function values of solutions obtained after conquer technique, 1st generation evolving and 20th generation evolving in Scenario 1.

The evolving process is shown in Figure 4.13 by plotting 100 solutions after evolving for 0, 1 and 20 generations. These 100 solutions were selected by higher fitness from total of 200 solutions after evolving. Candidate solutions improved significantly from the 1st to 20th generation and were close to the Pareto optimum front. Optimal solutions were more concentrative and more like a curve at the 20th generation.

To determine the generation number, the mutual domination rate (MDR) was calculated with Eq. 3.12 by using 100 candidate solutions. The calculate results are presented in Figure 4.14. With the growth of iterations number, I_{mdr} decreased significantly. At the 20th generation, both I_{mdr} values and linear regression line were all less than 0.1 which proved that solutions in the current generation have limited improvements compared to those in the previous generation (Martí et al., 2016). Therefore, the generation number 20 was determined as an appropriate and efficient number for the iteration stopping criteria.

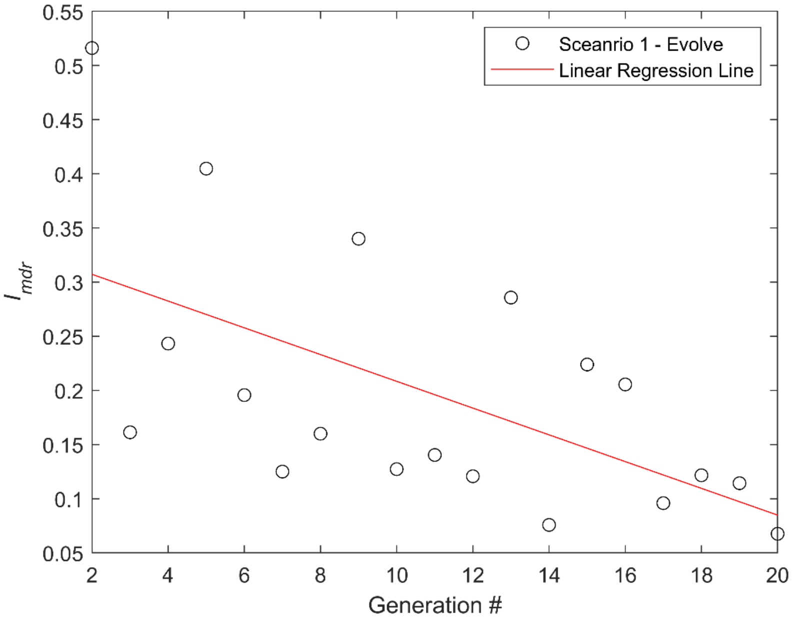


Figure 4.26 Calculation results of I_{mdr} with generations.

Table 4.1 presents how the number of non-dominated solutions varies with the number of generations. In 1st generation, the number of non-dominated solutions which were the "absolute" optimal solutions were only 25 while the number of non-dominated solutions was 74 in 20th generation as the number of non-dominated solutions increased with the number of generations to a large extent. To ensure that the same number of solutions were chosen for comparison, 100 solutions were selected from 1st generation even though some dominated solutions were included.

Table 4.4 The number of non-dominated solutions after each generation

Generation	Non-dominated Solutions
1	25
2	31
3	31
4	37
5	42
6	46
7	48
8	50
9	50
10	55
11	57
12	58
13	63
14	66
15	67
16	73
17	73
18	74
19	70
20	74

Looking at the 20th generation optimal solutions, there were 74 non-dominated solutions. Among these solutions, the selected times of each area divide solution were different. For example, divide solutions of area 1 were picked 14 times and divide solutions of area 2 were picked 67 times. Furthermore, divide solutions of area 5, 20, 21 were selected 69, 36, 26 times respectively. From Figure 4.12, it proved that the optimization to area 1, 2, 5, 20, 21 was more effective in the reduction of flooded nodes. Thus, divide solutions of these area with higher fitness were selected as optimal solutions with higher possibilities, which accelerated the convergence of the offspring solutions. On the contrary, divide solutions of area 7, 11, 12, 13, 14, 15, 23 and 24 were not picked among 74 non-dominated solutions because the optimization effects of these area were minimal or even lead to an increase of the flooded nodes.

From the optimal solution, which flooded areas are suitable for this optimization system of changing pipe size and slope can be obviously discovered. The optimal solution provides engineers with a tentative idea of expected effect by pipe optimization and corresponding cost.

However, this method may be limited for some flooded area, such as area 12 and area 25 To figure out this problem, other method, for instance, BMPs and LID, are supposed to be applied to improve flood situation.

4.2.2 Results based on 2D model

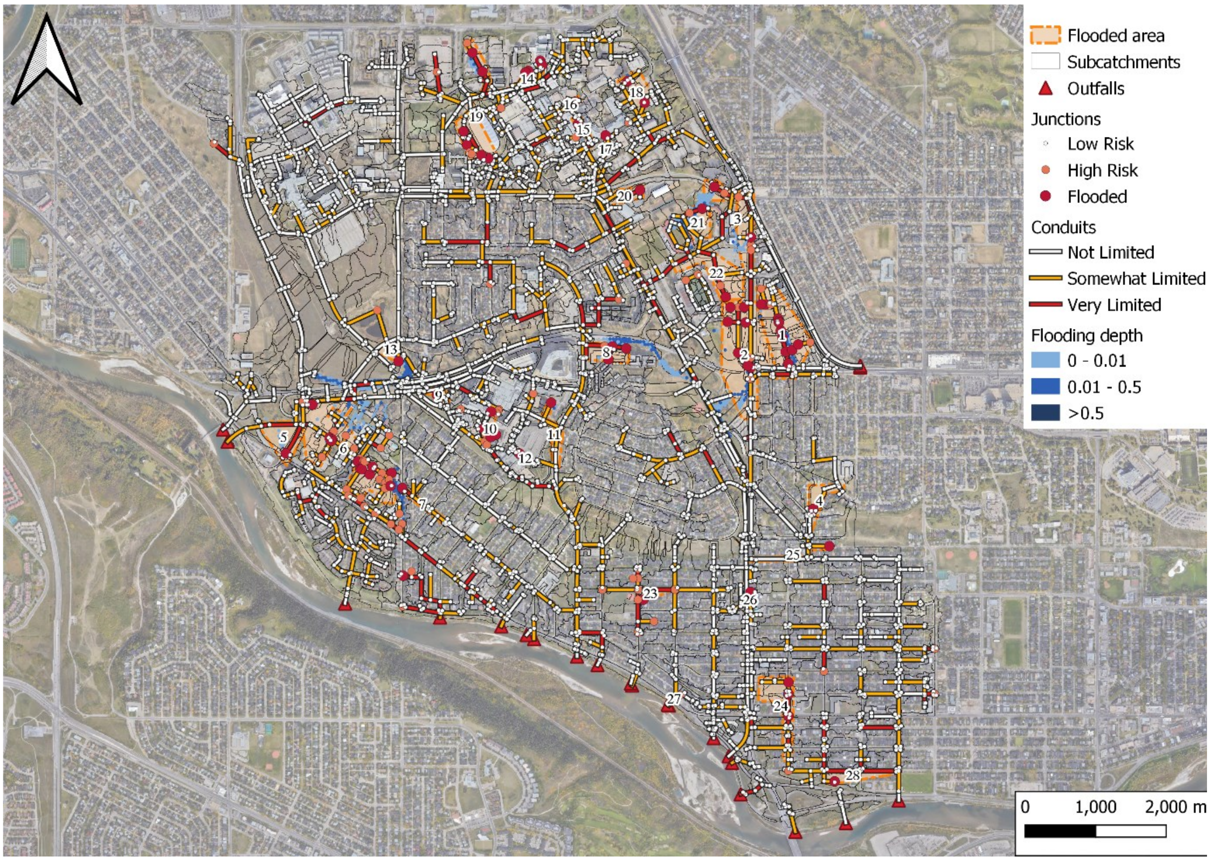


Figure 4.27 Before optimization: flooded area in the study area by using the 2D model.

The result of 2D simulation of existing study area including 28 flooded areas is shown in Figure 4.15. It was created by using interconnected nodes, cells and links which represent the physical features of the area being modeled. 73 flooded manholes and 79 high-risk manholes were found in the study area scenario 2. In this study, both optimization results based on scenario 1 and scenario 2 were simulated by 2D model to compare with the original 2D model. Under scenario 1, there were 284 flooded manholes and 342 very limited conduits while the numbers reduced to 73 and 294 respectively under scenario 2. And the flooded areas dropped from 37 to 28 when the 1D model converted to 2D model.

Scenario 1 was using the PCSWMM 1D modelling results of the study area for optimization and scenario 2 was using the PCSWMM 2D modelling results for optimization. Thus, scenario 1 only considered the flood at junctions while in scenario 2 water would spread to the ground surface when the junction was flooded. As a result, the HGL in scenario 1 was much higher than the HGL in scenario 2, and there were more flooded nodes in scenario 1.

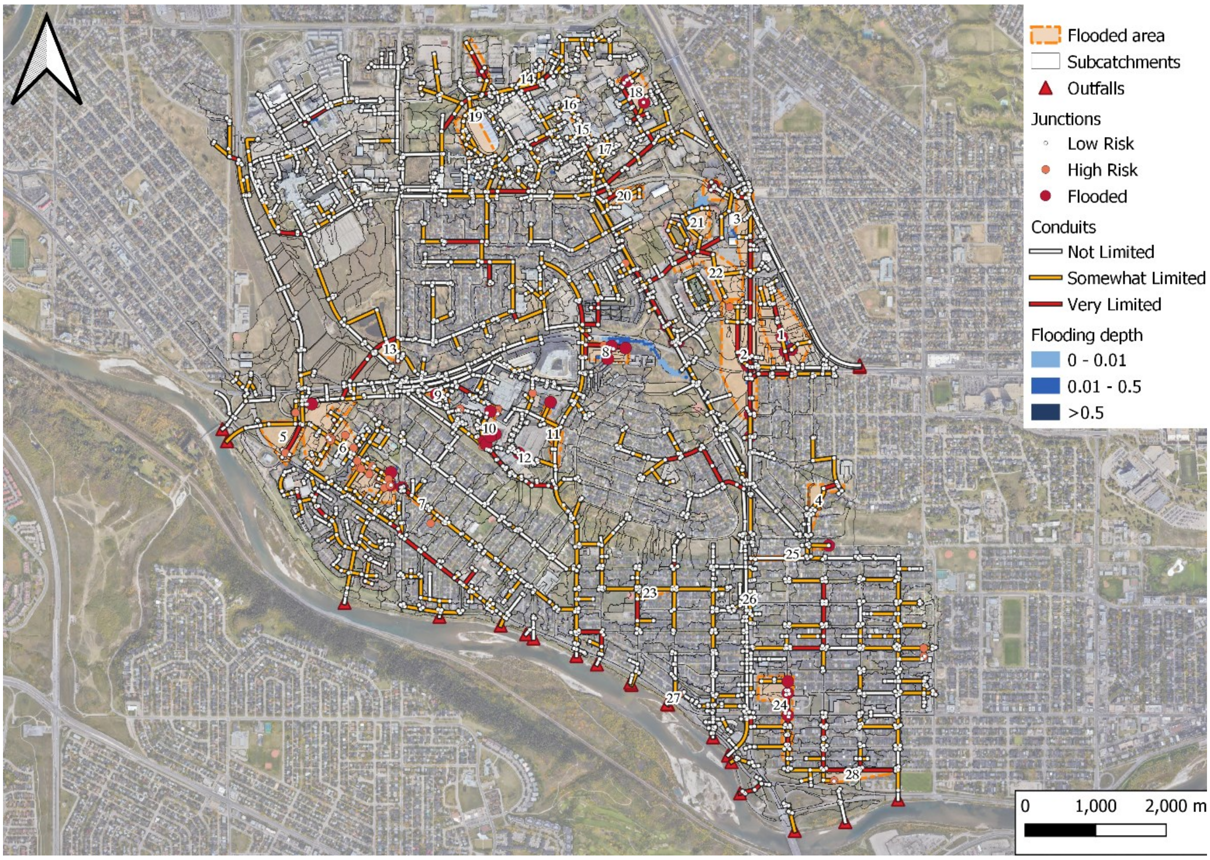


Figure 4.28 After optimization: flooded area in the study area of Scenario 1 by using the 2D model.

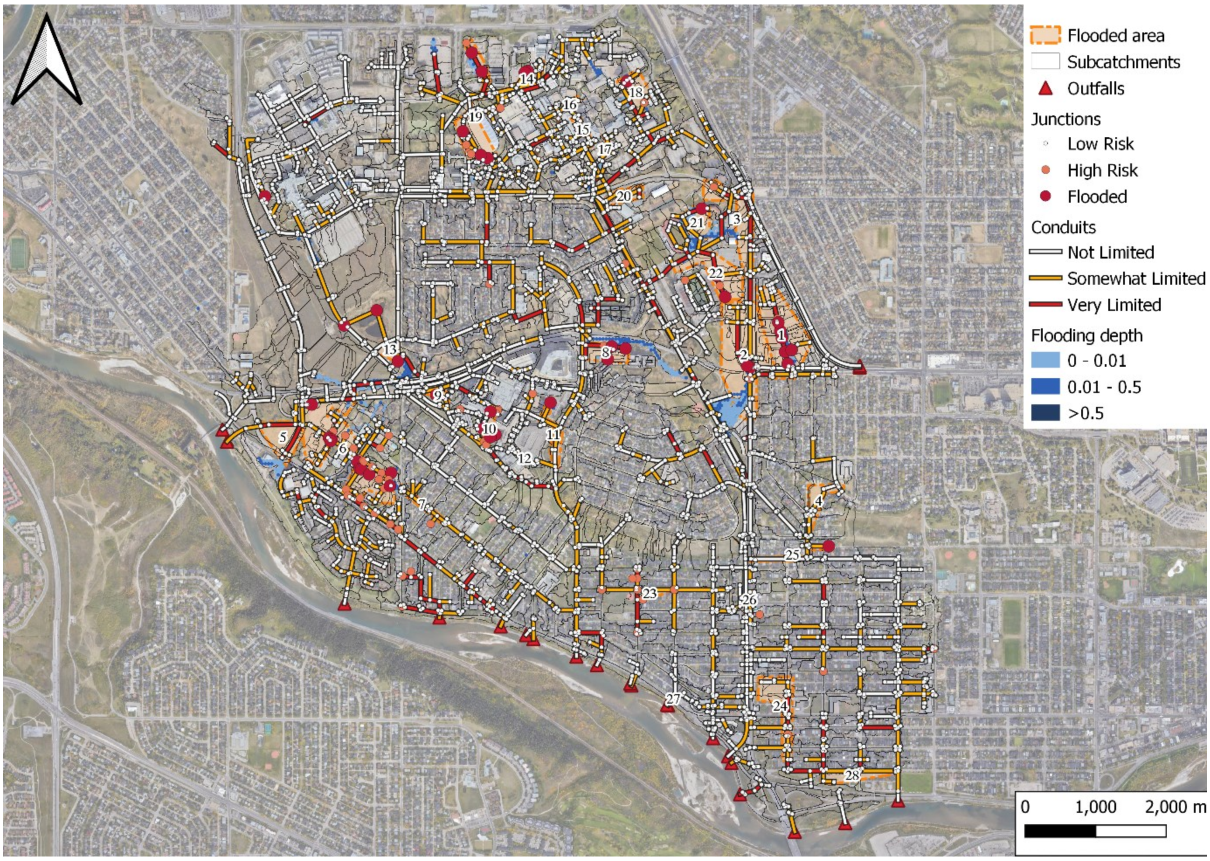


Figure 4.29 After optimization: flooded area in the study area of Scenario 2 by using the 2D model.

After optimization, there were 30 flooded nodes and 30 high-risk nodes in 2D model of scenario 1 (Figure 4.16). And 44 flooded nodes and 59 high-risk nodes were found in 2D model of scenario 2 (Figure 4.17). The total area of surface flooding in existing condition (Figure 4.15) was 13.17ha and the total area of surface flooding after optimization (Figure 4.16 and Figure 4.17) were 3.67ha and 11.03ha, separately. Compared to Figure 4.17, the improvement of surface flooding was more significant in Figure 5.16 as less area of surface flooding was observed.

Figure 4.18 compares flooded area 6 in existing and optimal study area 2D model of scenario 1 and Figure 4.19 compares flooded area 6 in existing and optimal study area 2D model of scenario 2. In existing area, large-scale of surface flooding were founded surrounding area 6. As shown in Figure 4.18, it is obvious that the flooded nodes in area 6 decreased from 11 to 3 when most of conduits were upsized. At the same time, the large-scale surface ponding in existing model almost disappeared when the flooding depth decreased below 0 after optimization. However, the improvement of optimization results in scenario 2 (Figure 4.19) was not significant as scenario 1 as there was only one reduction number of flooded nodes even if most of pipes were upsized. In addition, the flooding depth decreased slightly in this area whereas large-scale of surface flooding still existed after optimization.

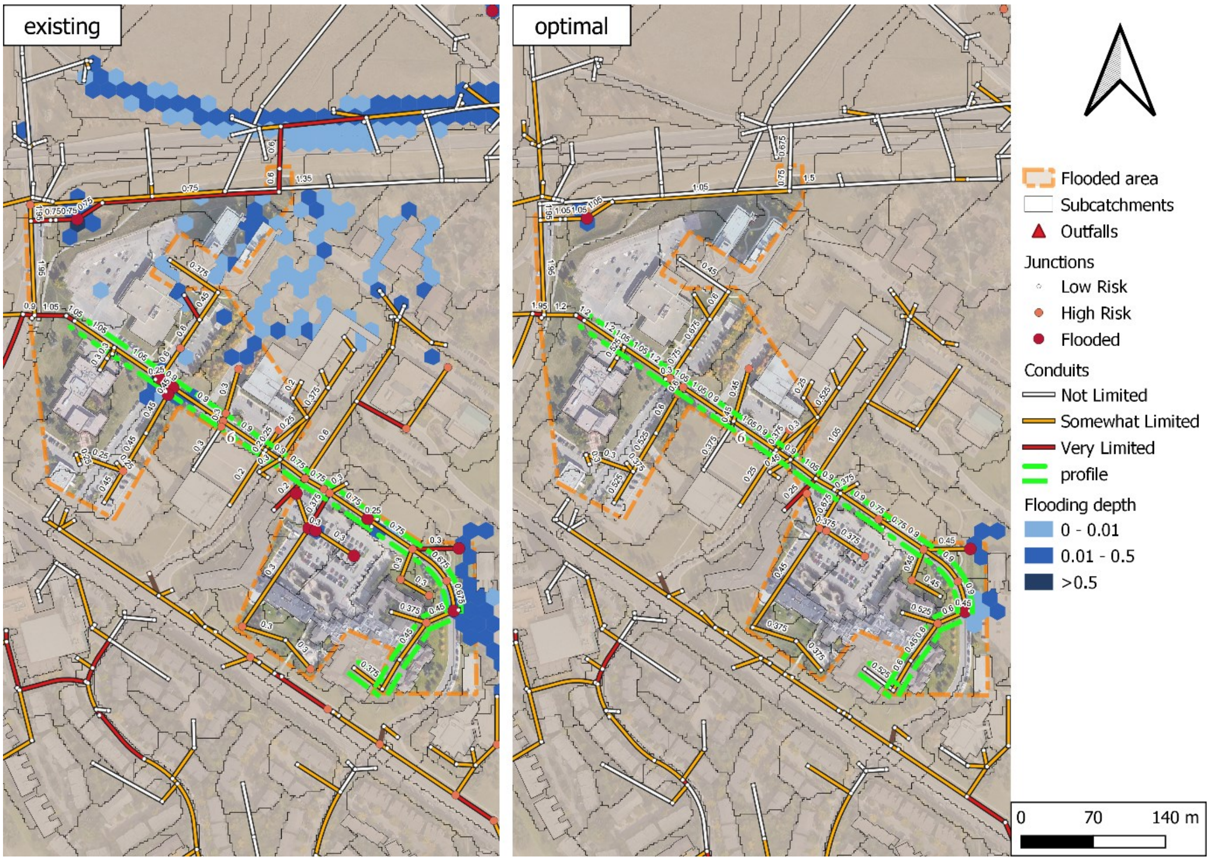


Figure 4.30 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 2D model. The selected sewer trunk route (green color) is used for generating the profile.

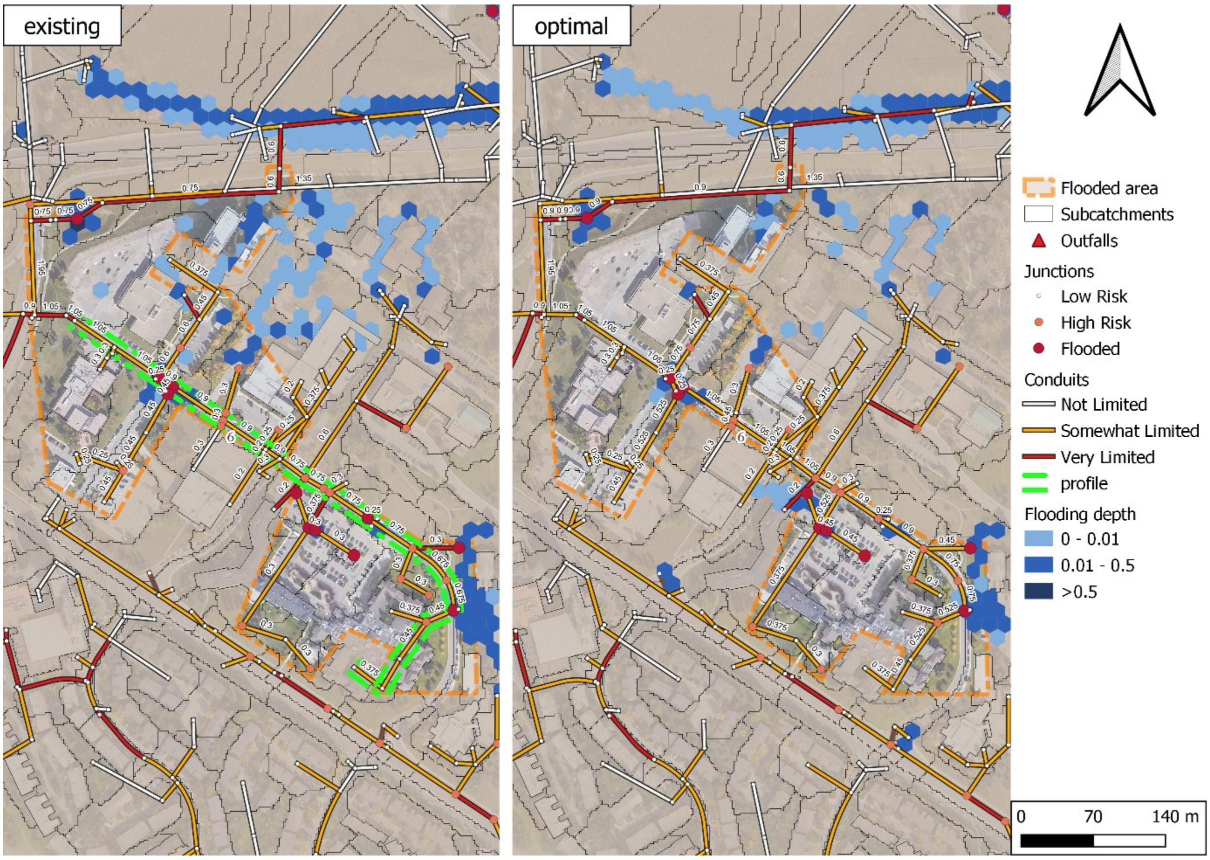


Figure 4.31 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 2 by using the 2D model.

The 2D modeling profiles of a typical route in flooded area 6 before and after optimization in profile are compared in Figure 4.20 and Figure 4.21. A total of 4 flooded nodes were observed in existing profile and there was only 1 flooded node after optimization. The slope of this pipe route was decreased slightly while all of conduits were upsized. For instance, the diameter of conduit 16 (NC_10_PKD_0285) and conduit 17 (NC_10_PKD_0255) increased from 0.45m to 0.6m and 0.9m to 1.05m respectively. The length, diameter and slope of conduits in selected sewer trunk were shown in Table A5 and Table A6.

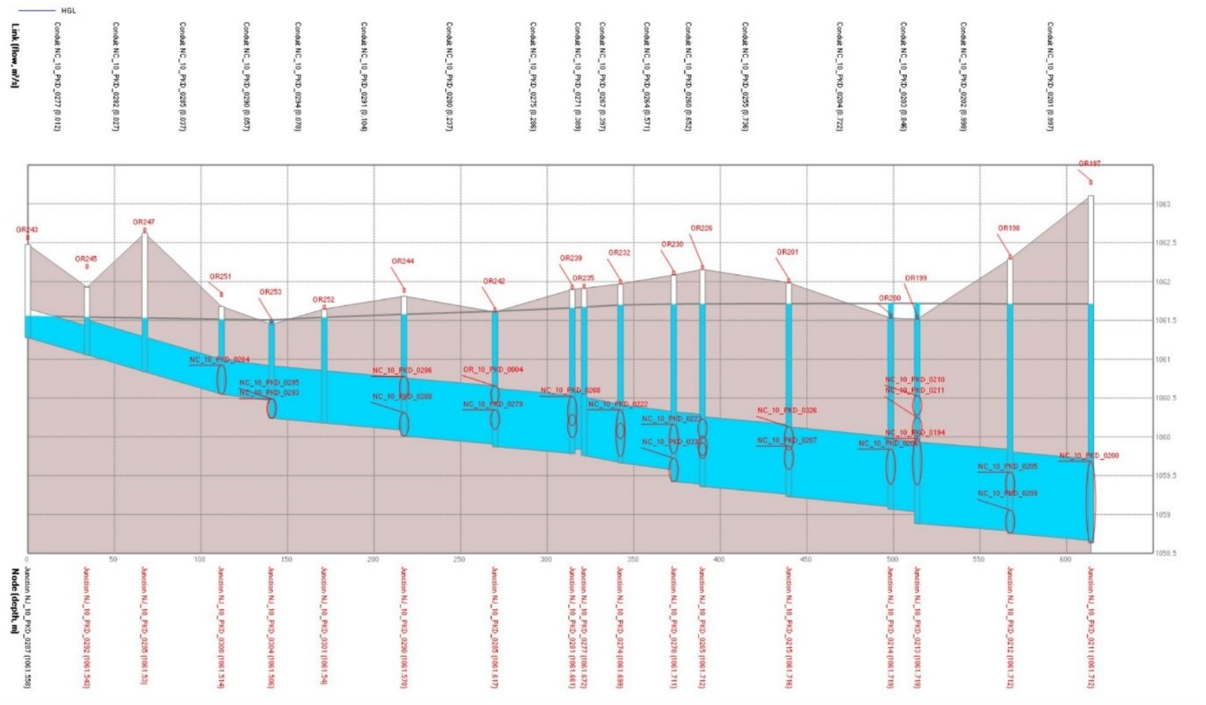


Figure 4.32 Before optimization: the profile of selected storm sewer trunk route in Flooded Area 6 of the study area, using the 2D model.

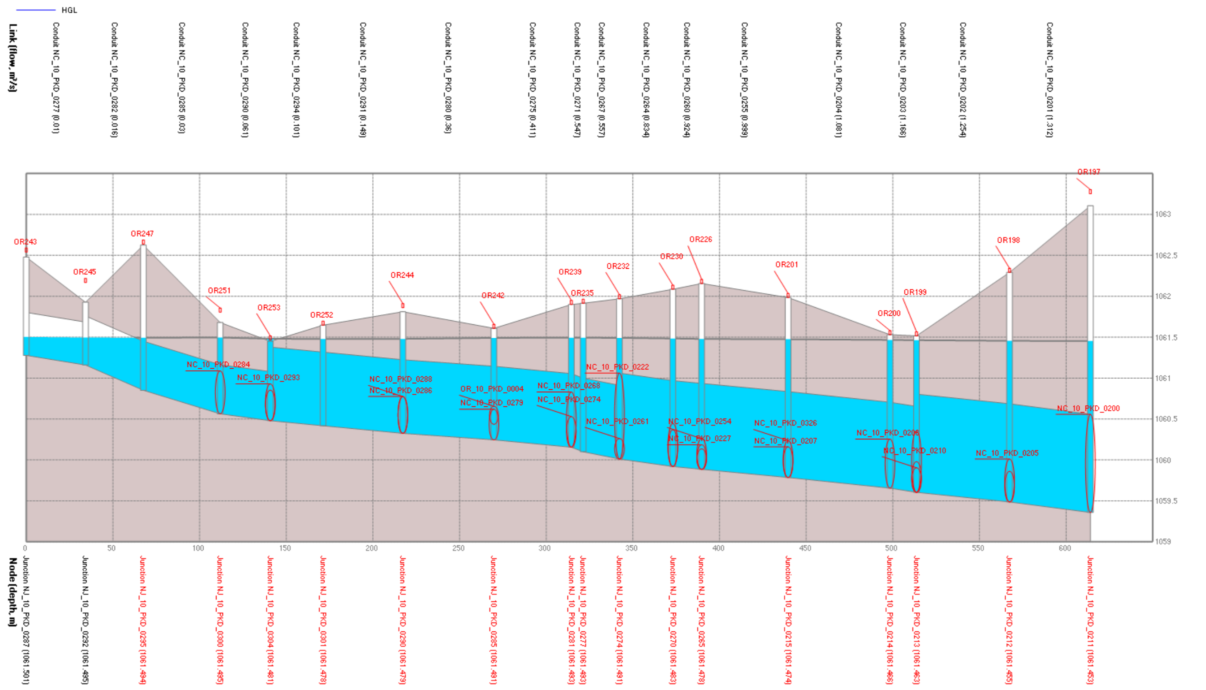


Figure 4.33 After optimization: the profile of selected storm sewer trunk route in Flooded Area 6 of the study area, under Scenario 1 by using the 2D model.

The comparison of solutions' objective function values obtained by conquer with roulette wheel in scenario 1 and scenario 2 is shown in Figure 4.22. The objective function values of conquer solutions in scenario 1 were more evenly dispersed, whereas the values in scenario 2 concentrated around 280 of flooded nodes which was closed to existing flooded nodes. It indicated that the result based on scenario 1 is more effective than the result based on scenario 2. Because the number of changed pipe was larger in scenario 1 as there were more flooded nodes and very limited conduits in scenario 1.

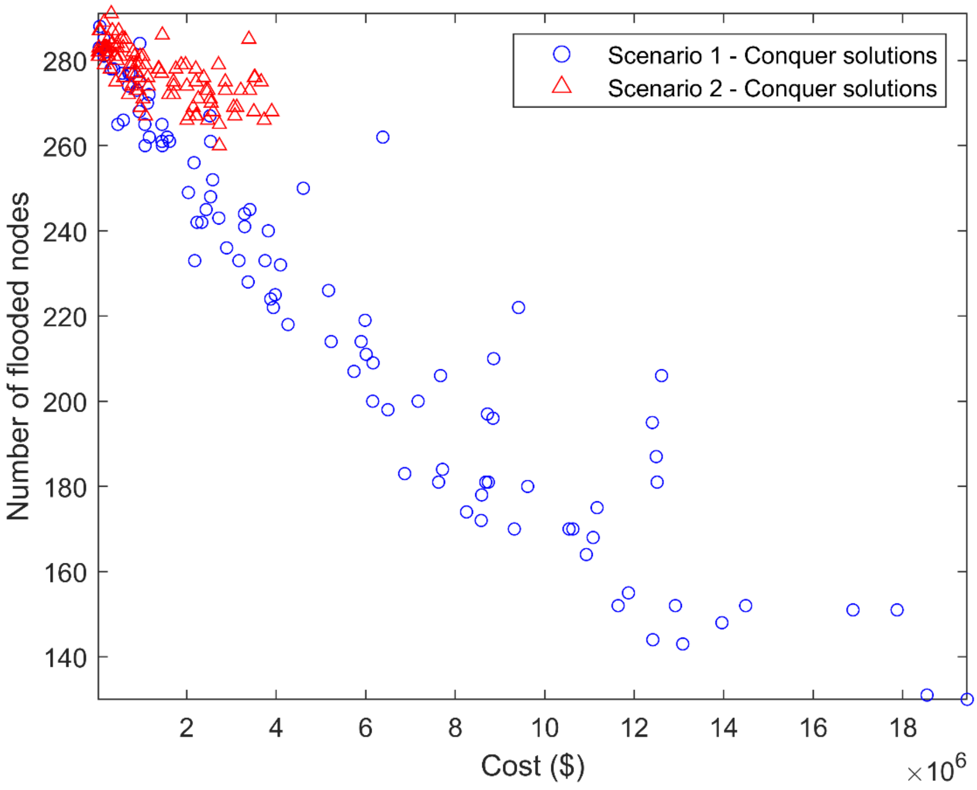


Figure 4.34 The comparison between Scenario 1 and Scenario 2 for conquer solutions' objective function values (number of flooded nodes vs. cost).

Figure 4.23 presents the objective function values of solutions obtained after conquer technique, 1st generation of evolving and 20th generation of evolving in scenario 2. Compared to conquer solutions and 1st generation solutions, 20th generation solutions were more like a curve. The number of flooded nodes reduced to about 250 in some of 20th generation solutions while the number of 1st generation solutions was around 270 which were close to conquer solutions.

However, the improvement from 1st generation solutions to 20th generation solutions in scenario 2 was not as significant as it in scenario 1 when compared with Figure 4.12.

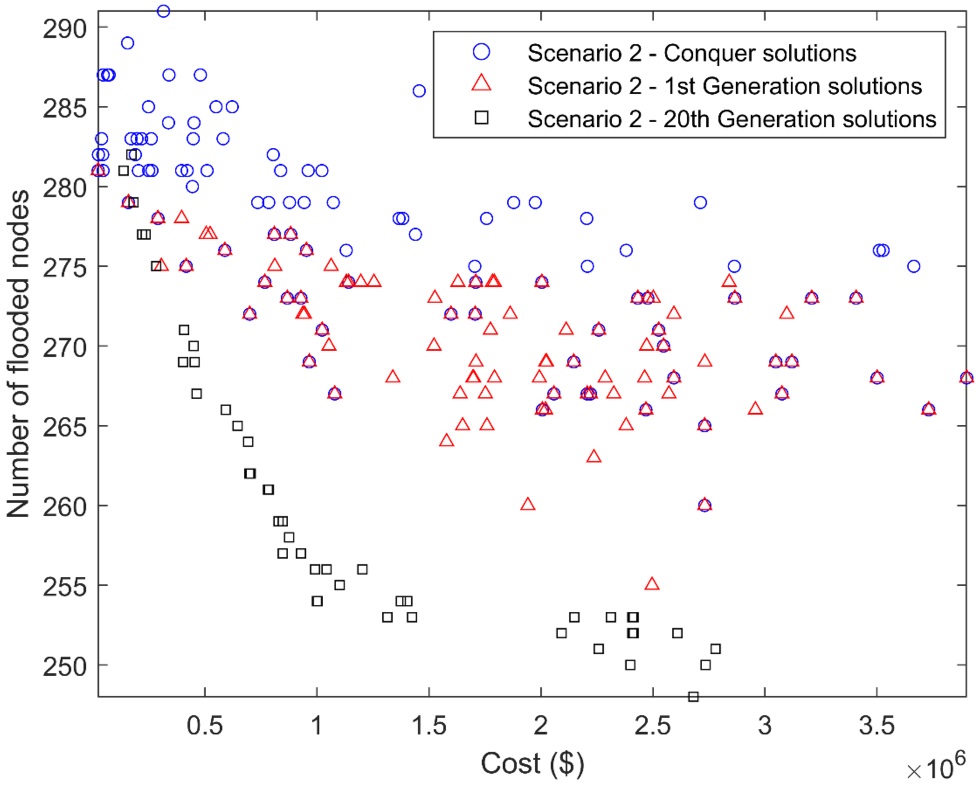


Figure 4.35 The objective function values of solutions obtained after conquer technique, 1st generation of evolving and 20th generation of evolving in scenario 2.

Although these results were compared in 2D model, all the optimal processes in MATLAB were through 1D modeling. The main reason is that PCSWMM is not an open-source software, the open-source code MATLAB using is from EPASWMM. However, there is no 2D model in EPASWMM. Even if the source code of PCSWMM is opened, another limitation is that doing 2D modeling is time-consuming as it costs about 3 hours to run a 2D model for one time. During the process of optimization, the model needs to be run for 2660 times in total. If 2D modeling is applied in optimal process, the running time of the model in MATLAB will exceed 330 days. Although parallel computing is applied, the total running time would be shorten to approximately 30 days as parallel computing cannot ensure the running speed of each core is same and running speed is limited by computer memory. Overall, doing 2D modeling in MATLAB optimal process is still unpractical.

Compared 2D model and 1D model, the pipe variation in 1D model is more effective before and after optimization as the flooded manholes reduced from 284 to 115 in 1D model. In addition, 1D model focused on the stormwater pipe system, while 2D model took surface flooding into consideration. In this study, the optimization aimed at changing of stormwater pipe size and slop but not surface condition. Therefore, 1D modeling, which is more intuitive and efficient, should be used to verify the effect of the pipe optimization system.

4.2.3 Limitations

Although the optimization method in this study gives users flexibility in using it in different study areas, in different scales and with customized modifications, some limitations still exist in this study. First, the invert elevation at the junction were assumed the same as the inlet and outlet elevations of connected sewer pipes but in most engineering cases there are differences between the invert elevation at the junction and the inlet and outlet elevations of connected sewer pipes. Also, pipes were connected based on the bottom of pipe in this study however different pipe connection methods are applied in practical engineering. Second, this study assumed that the downstream pipe size cannot be less than the upstream pipe size. But in practical engineering, the downstream pipe can be found to be smaller than the upstream pipe size in some cases, where the downstream pipe slope is larger and thus the downstream pipe capacity is larger or equivalent, compared to the upstream pipe.

Third, there is a limit to the number of increments to increase the pipe size in the optimization, in which the maximum increments for raising pipe size were limited to four. However, the magnitude of pipe size and slope variation depends on actual engineering conditions and topographical conditions. The pipe size was only considered to be upsized or remain the same when determining the initial candidate solutions, i.e., decrease of pipe size was not considered. Therefore, overdesign can be observed in some areas especially for the test area as solutions were not evolved in the test area. And this method does not consider drop manholes which are utilized in steep areas or when the inlet pipe has an invert elevation significantly higher than the invert of the outlet pipe.

Moreover, when varying the pipe size, all pipes were assumed to be circular pipes and only the diameters are changed. But in the original situation, there are a small number of pipes in rectangular shape. For this type of pipes, only the heights (maximum depth of cross section) were changed while the widths remained the same. And velocity in sewer pipes was not considered in this study.

At last, the method in this study is only suitable for SWMM models including PCSWMM, EPASWMM, XPSWMM, etc., and cannot be used for other stormwater management software (e.g., MIKE) because they are not open-source software. And all the optimal process in MATLAB were through the 1D modeling as the 2D modeling is too time-consuming and it cannot be applied to MATLAB optimal process.

5 Conclusion and future research

5.1 Conclusion

In this study, a multi-objective optimization algorithm based on PICEA-g for a large-scale storm sewer network optimization has been proposed, which can be also applied to sanitary sewer network. Sewer pipe size and slope were determined as decision variables, and the number of flooded nodes and cost were taken as objective functions. The new optimization system can be used as a tool to assist drainage network engineers, managers and other decision makers in developing optimization strategies and projects for sewer system rehabilitation or upgrade to mitigate urban pluvial flooding in the context of climate change and increased anthropogenic activities.

Differing from previous optimization algorithms in the literature, significant improvements were made to the optimization algorithm, including new approaches for initializing candidate solutions, applying divide and conquer technique, enhanced goal vector boundaries and fitness calculation, and enhanced genetic operators by adopting the logistic map and roulette wheel. These improvements were demonstrated to significantly enhance the algorithm's performance.

Specific and detailed improvements are as follows: (1) The initial candidate solutions were created diversely to cover the entire possible value range of objective functions and to avert the aggregation of produced optimal solutions so that the non-dominated solutions can be closer to the Pareto optimal front. (2) At the same time, the creation of candidate solutions was limited by decision variable constraints to ensure the pipe connection and follow the UDS design criteria. (3) In the process of divide and conquer, the optimization process was first decomposed for each individual flooded area, and then these divide solutions were combined by conquer technique into new sets of conquer solutions for the entire study area. (4) With the aid of roulette wheel, areas and divide solutions with a higher ratio of flooded nodes had a higher chance of being selected by conquer technique. (5) Cutting plane was utilized in this study to sort candidate solutions and increase their diversity by keeping each goal vector useful with modified genetic operators. (6) In addition, enhanced fitness calculation helped sort the dominance relationships between candidate solutions. (7) In evolving process, the logistic map generated a chaotic sequence that increased the randomness and diversity of offspring solutions and roulette wheel helped accelerate the convergence of the offspring solutions.

Compared with previous studies that only optimized sewer pipe size in a small area (with less than 100 pipes), this study considered the optimizations of both sewer pipe size and slope in a real-life, complex storm sewer network in a large area (with 2930 pipes) to improve its performance in urban pluvial flooding. The new optimization system was first demonstrated in a small test area, and then in the entire study area. For the test area, the flooded manholes and high-risk manholes were reduced from 15 and 7 to 0 after optimization at a cost of 0.6×10^6 CAD$. The comparison with previous case studies indicated a significant improvement in sewer network optimization in this study because the reduction of flooded volume was substantial while the cost was minimal.

For the entire study area, the flooded manholes decreased from 284 to 115 after the optimization and high-risk manholes decreased from 47 to 38 using the 1D model at a cost of 17×10^6 CAD$. Using the 2D model, the flooded nodes dropped from 73 to 30 and 44 under Scenarios 1 and 2, respectively, where Scenario 1 was the condition that using the PCSWMM 1D modelling results of the study area for optimization and scenario 2 was the condition that using the PCSWMM 2D modelling results of the study area for optimization. Also, the high-risk manholes dropped from 79 to 30 and 59 under Scenarios 1 and 2, respectively; and the cost was around 2.5×10^6 CAD$ when using scenario 2. In this study, results from the 1D modeling were more intuitive and efficient to verify the effect of storm sewer pipe system optimization as they aimed at sewer pipe size and slope variations, without consideration of ground surface condition. In practical engineering, the optimal solution can be chosen and modified in accordance with other constraints, such as engineering design criteria, budget availability and risk assessment.

5.2 Future research

For future research, further improvements to the proposed optimization system can be made in the following aspects. As the current optimization only focuses on changing sewer pipe size and slope, the improvements to urban flooding are still limited. In order to better solve this problem, other measures such as dry ponds, storage tanks, LID and BMPs can be combined with varying sewer pipe size and slope so that better flood mitigation effect of urban drainage system can be achieved. In this study, the study area was divided into different flooded areas, and flooded areas where pipe upgrades had a marginal effect have been found. Thus, these flooded areas could be prioritized for adding dry ponds and storage tanks or applying LID and BMPs in future research. For implementing LID-BMPs, incorporating GIS information and conducting geodata analysis automatically can help to identify the feasible location.

Currently, the way to determine the selection of decision variables is through junction HGL, and the optimization is only focused on pipes in selected flooded areas generated by flooded nodes. Also, the way to adjust pipe size and slope is based on current pipe capacity limitation. Other criteria and parameters, such as hours flooded, total flood volume, maximum flood rate, etc., can be used to choose the decision variables and figure out the sewer pipe optimization range in study area to achieve more effective solutions.

Besides, the optimization system in this study aims at storm sewer network based on the PCSWMM model. To expand the scope of application, this optimization system needs to be adjusted to sanitary system or combined system and incorporate other stormwater management software such as MIKE in future studies. In this way, the proposed optimization system can better provide reliable optimization solutions to decision makers when facing different engineering optimization problems. Also, other algorithms, such as NSGA-II and ε-MOEA, can be used in same study area to compare with the solutions generated from PICEA-g.

At last, the engineering practicability of this method is supposed to be improved. For instance, the upgrades of pipe size were limited to 4 increments in this study, but the magnitudes of pipe size and slope variations depend on actual engineering conditions and topographical conditions. And a drop manhole can be utilized in steep ground areas or where the elevation of the inlet pipe is much higher than the elevation of the outlet pipe. Thus, GIS can be incorporated for decision making and Python can be used instead of MATLAB as it has geodata analysis packages such as Geopanda and GDAL. Incorporating GIS can help us not just change the current pipe system and adjust pipe slope with topographical conditions but also add drop manholes at feasible locations.

Reference

Adams, B. J., Fraser, H. G., Howard, C. D., & Sami Hanafy, M. (1986). Meteorological data analysis for drainage system design. *Journal of environmental Engineering*, 112(5), 827-848.

Afshar, A., Shojaei, N., & Sagharjooghifarahani, M. (2013). Multi-objective calibration of reservoir water quality modeling using multi-objective particle swarm optimization (MOPSO). *Water resources management*, 27(7), 1931-1947.

Ahiablame, L. M., Engel, B. A., & Chaubey, I. (2012). Effectiveness of low impact development practices: literature review and suggestions for future research. *Water, Air, & Soil Pollution*, *223*(7), 4253-4273.

Ahmadi, A., Zolfagharipoor, M. A., & Nafisi, M. J. J. (2018). Development of a hybrid algorithm for the optimal design of sewer networks. *Journal of Water Resources Planning and Management*, 144(8).

Antonio, L. M., & Coello, C. A. C. (2013, June). Use of cooperative coevolution for solving large scale multi-objective optimization problems. *In 2013 IEEE Congress on Evolutionary Computation* (pp. 2758-2765). IEEE.

Azadnia, A., & Zahraie, B. (2010). Optimization of nonlinear Muskingum method with variable parameters using multi-objective particle swarm optimization. *In World Environmental and Water Resources Congress 2010: Challenges of Change* (pp. 2278-2284).

Baltar, A. M., & Fontane, D. G. (2008). Use of multi-objective particle swarm optimization in water resources management. *Journal of water resources planning and management*, 134(3), 257-265.

Barraud, S., Gibert, J., Winiarski, T., & Bertrand Krajewski, J. L. (2002). Implementation of a monitoring system to measure impact of stormwater runoff infiltration. *Water Science and Technology*, *45*(3), 203-210.

Barreto, W., Vojinovic, Z., Price, R., & Solomatine, D. (2010). Multi-objective evolutionary approach to rehabilitation of urban drainage systems. *Journal of Water Resources Planning and Management -Asce*, 136(5), 547–554.

Bayas-Jiménez, L., Iglesias-Rey, P. L., & Martínez-Solano, F. J. (2019). Multi-Objective Optimization of Drainage Networks for Flood Control in Urban Area Due to Climate Change. *Multidisciplinary Digital Publishing Institute Proceedings*, 48(1), 27.

Burch, N., Holte, R., Müller, M., O'Connell, D., & Schaeffer, J. (2010). Automating layouts of sewers in subdivisions. *In ECAI 2010* (pp. 655-660). IOS Press.

Choi, Y. H., Jung, D., Lee, H. M., Yoo, D. G., & Kim, J. H. (2017). Improving the quality of pareto optimal solutions in water distribution network design. *Journal of Water Resources Planning and Management*, 143(8), 04017036.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3), 256-279.

Cooper, I. M., John, M. P., Lewis, R., Mumford, C. L., & Olden, A. (2014, July). Optimising large scale public transport network design problems using mixed-mode parallel multi-objective evolutionary algorithms. *In 2014 IEEE Congress on Evolutionary Computation* (CEC) (pp. 2841-2848). IEEE.

Deb, K., & Agrawal, R. B. (1994). Simulated binary crossover for continuous search space (Technical Report No. IITK/ME/SMD-94027). Indian Institute of Technology, Kanpur.

Deb, K., Mohan, M., & Mishra, S. (2005). Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary computation*, 13(4), 501-525.

Deb, K., Pratap, Amrit, Agarwal, Sameer, Meyarivan, TAMT. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions* on, 6(2), 182-197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002, May). Scalable multi-objective optimization test problems. *In Proceedings of the 2002 Congress on Evolutionary Computation.* CEC'02 (Cat. No. 02TH8600) (Vol. 1, pp. 825-830). IEEE.

Deb, K.,Mohan,M., & Mishra, S. (2003). Towards a quick computation of well-spread pareto-optimal solutions. *In Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference* (LNCS 2632), pages 222–236.

Delelegn, S. W., Pathirana, A., Gersonius, B., Adeogun, A. G., & Vairavamoorthy, K. (2011). Multi-objective optimisation of cost–benefit of urban flood management using a 1D2D coupled model. *Water Science and Technology*, 63(5), 1053-1059.

Friese, R. D. (2016, May). Efficient genetic algorithm encoding for large-scale multi-objective resource allocation. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (pp. 1360-1369). IEEE.

Fu, G., Khu, S. T., & Butler, D. (2009). Use of surrogate modelling for multi-objective optimisation of urban wastewater systems. *Water Science and Technology*, 60(6), 1641–1647. https://doi.org/10.2166/wst.2009.508

Gaur, A., Talukder, A. K., Deb, K., Tiwari, S., Xu, S., & Jones, D. (2017, November). Finding near-optimum and diverse solutions for a large-scale engineering design problem. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). IEEE.

Hadka, D., & Reed, P. (2013). Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2), 231–259.

Hammond, M. J., Chen, A. S., Djordjević, S., Butler, D., & Mark, O. (2015). Urban flood impact assessment: A state-of-the-art review. *Urban Water Journal*, 12(1), 14-29.

Hassan, W. H., Jassem, M. H., & Mohammed, S. S. (2018). A GA-HP model for the optimal design of sewer networks. *Water Resources Management*, 32(3), 865–879.

Heydari Mofrad, H., & Yazdi, J. (2022). An enhanced multi-objective evolutionary algorithm for the rehabilitation of urban drainage systems. *Engineering Optimization*, 54(2), 349-367.

Hong, W. J., Yang, P., & Tang, K. (2021). Evolutionary computation for large-scale multi-objective optimization: A decade of progresses. *International Journal of Automation and Computing*, 18(2), 155-169.

Huber, W. C., Rossman, L. A., & Dickinson, R. E. (2005). EPA storm water management model, SWMM5. *Watershed models*, *338*, 359

Huber, W., & Roesner, L. (2012). The History and Evolution of the EPA SWMM.

IEEE Congress on Evolutionary Computation. IEEE, pp. 1263–1270.

Jato-Espino, D., Sillanpää, N., Charlesworth, S. M., & Andrés-Doménech, I. (2016). Coupling GIS with stormwater modelling for the location prioritization and hydrological simulation of permeable pavements in urban catchments. *Water*, 8(10), 451.

Kebede, S. A. (2014). Optimal design of urban stormwater drainage system under uncertainity. A thesis submitted to the UNESCO-IHE.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. *In Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary computation*, 10(3), 263-282.

Lin, R., Zheng, F., Savic, D., Zhang, Q., & Fang, X. (2020). Improving the effectiveness of multi-objective optimization design of urban drainage systems. *Water Resources Research*, 56(7), e2019WR026656.

Liong, S.-Y., Chan, W. T., & ShreeRam, J. (1995). Peak-flow forecasting with genetic algorithm and SWMM. *Journal of Hydraulic Engineering*, 121(8), 613-617.

Liu, H., Zhang, C., Jiang, Y., Meng, F., & Fu, G. (2018). Measuring surplus capacity for multi-objective optimal design of foul sewer systems. *Urban Water Journal*, 15(8), 723–731.

Liu, W., Liu, L., & Dong, Z. (2013, July). On the use of multi-objective particle swarm optimization for allocation of water resources. *In 2013 Ninth International Conference on Natural Computation* (ICNC) (pp. 612-617). IEEE.

Lohn, J. D., Kraus, W. F., & Haith, G. L. (2002, May). Comparing a coevolutionary genetic algorithm for multi-objective optimization. *In Proceedings of the 2002 Congress on Evolutionary Computation.* CEC'02 (Cat. No. 02TH8600) (Vol. 2, pp. 1157-1162). IEEE.

Lowe, S. A. (2010). Sanitary sewer design using EPA storm water management model (SWMM). *Computer Applications in Engineering Education*, 18(2), 203-212.

Lü, R., Guan, X., Li, X., & Hwang, I. (2016). A large-scale flight multi-objective assignment approach based on multi-island parallel evolution algorithm with cooperative coevolutionary. *Science China Information Sciences*, 59(7), 1-17.

Maheepala, U. K., Takyi, A. K., & Perera, B. J. C. (2001). Hydrological data monitoring for urban stormwater drainage systems. *Journal of Hydrology*, *245*(1-4), 32-47.

Maier, H. R., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L. S., Cunha, M. C., et al. (2014). Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environmental Modelling & Software*, 62, 271–299.

Martí, L., García, J., Berlanga, A., Molina, J.M., 2016. A stopping criterion for multi-objective optimization evolutionary algorithms. Inf. Sci. 367, 700–718.

May, R. M. (2004). Simple mathematical models with very complicated dynamics. *The Theory of Chaotic Attractors*, 85-93.

Men, H., Lu, H., Jiang, W., Xu, D., 2020. Mathematical Optimization Method of Low-Impact

Development Layout in the Sponge City. *Mathematical Problems in Engineering*. 2020.

Mohammadiun, S., Yazdi, J., Neyshabouri, S. A. A. S., & Sadiq, R. (2018). Development of a stochastic framework to design/rehabilitate urban stormwater drainage systems based on a resilient approach. *Urban Water Journal*, 15(2), 167–176.

Murphy, L. J., Dandy, G. C., & Simpson, A. (1993). Pipe network optimization using an improved genetic algorithm: Department of Civil and Environmental Engineering, University of Adelaide.

Ngamalieu-Nengoue, U. A., Iglesias-Rey, P. L., Martínez-Solano, F. J., Mora-Meliá, D., &

Saldarriaga Valderrama, J. G. (2019). Urban drainage network rehabilitation considering storm tank installation and pipe substitution. *Water*, 11(3), 515.

Ngamalieu-Nengoue, U. A., Martínez-Solano, F. J., Iglesias-Rey, P. L., & Mora-Meliá, D. (2019). Multi-objective optimization for urban drainage or sewer networks rehabilitation through pipes substitution and storage tanks installation. *Water*, 11(5), 935.

Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., ... & ASCE Task Committee on Evolutionary Computation in Environmental and Water Resources Engineering. (2010). State of the art for genetic algorithms and beyond in water resources planning and management. *Journal of Water Resources Planning and Management*, 136(4), 412-432.

Noory, H., Liaghat, A. M., Parsinejad, M., Haddad, O. B., & Vazifedoust, M. (2010). Environmental and economic multi-objective model for managing irrigation and drain water. *In 9th International Drainage Symposium held jointly with CIGR and CSBE/SCGAB Proceedings*, 13-16 June 2010, Québec City Convention Centre, Quebec City, Canada (p. 1). American Society of Agricultural and Biological Engineers.

Paknejad, P., Khorsand, R., & Ramezanpour, M. (2021). Chaotic improved PICEA-g-based multi-objective optimization for workflow scheduling in cloud environment. *Future Generation Computer Systems,* 117, 12-28.

Park M, Chung G, Yoo C, Kim JH (2012) Optimal Design of Stormwater detention basin using the genetic algorithm. *KSCE J Civ Eng* 16(4):660–666

Parsopoulos, K. E., & Vrahatis, M. N. (2002, March). Particle swarm optimization method in multi-objective problems. *In Proceedings of the 2002 ACM symposium on Applied computing* (pp. 603-607).

Purshouse, R. C., & Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE transactions on evolutionary computation*, 11(6), 770-784.

Qian, H., & Yu, Y. (2017, February). Solving high-dimensional multi-objective optimization problems with low effective dimensions. *In Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

Rathnayake, U., & Tanyimboh, T. (2015). Multi-objective optimization of combined sewer systems using SWMM 5.0. J. *Civil Eng. Architect. Res*, 2(10), 985-993.

Ray, T., & Liew, K. M. (2002). A swarm metaphor for multi-objective design optimization. *Engineering optimization*, 34(2), 141-153.

Rossman, L.A. (2015). Storm Water Management Model User's Manual Version 5.1.

Rungo, M., & Olesen, K. W. (2003, October). Combined 1-and 2-dimensional flood modelling. *In Proceeding 4th Iranian Hydraulic Conference* (pp. 21-23).

Salerno, F., Gaetano, V., & Gianni, T. (2018). Urbanization and climate change impacts on surface water quality: Enhancing the resilience by reducing impervious surfaces. *Water research*, 144, 491-502.

Sanders, B. F. (2007). Evaluation of on-line DEMs for flood inundation modeling. *Advances in water resources*, 30(8), 1831-1843.

Savic, D. (2002). Single-objective vs. multi-objective optimisation for integrated decision support. Paper presented at the Integrated Assessment and Decision Support*, Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society.*

Schmit, T. G., Thomas, M., & Ettrich, N. (2004). Analysis and modeling of flooding in urban drainage systems. *Journal of Hydrology*, 299(3-4), 300-311.

Senior, M., Scheckenberger, R., & Bishop, B. (2018). Modeling catchbasins and inlets in SWMM. *Journal of Water Management Modeling*.

Seyedashraf, O., Bottacin-Busolin, A., & Harou, J. J. (2021). A Disaggregation-Emulation Approach for Optimization of Large Urban Drainage Systems. *Water Resources Research,* 57(8), e2020WR029098.

Shojaeefard, M. H., Akbari, M., & Asadi, P. (2014). Multi objective optimization of friction stir welding parameters using FEM and neural network. *International journal of precision engineering and manufacturing,* 15(11), 2351-2356.

Srinivas, M., & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4), 656-667.

Sun SA, Djordjević S, Khu ST (2011) A general framework for flood risk based storm sewer network design. *Urban Water* J 8(1):13–27

Sutherland, R., & Jelen, S. L. (2003). Stormwater quality modeling improvements needed for SWMM. *Journal of Water Management Modeling*.

The City of A Water Resources. (2011). *Stormwater Management & Design Manual*.

Vairavamoorthy, K., & Ali, M. (2000). Optimal design of water distribution systems using genetic algorithms. *Computer-Aided Civil and Infrastructure Engineering, 15*(5), 374-382.

Vélez, C. A., Lobbrecht, A., Price, R., Mynett, A. E., Popescu, I., Galvis, A., & Restrepo, I. (2007). Optimization of urban wastewater systems using model based design and control. Case study of call, Colombia. *Saneamiento básico y ambiental en america latina*, 1, 358-370.

Vojinovic, Z., Sahlu, S., Torres, A. S., Seyoum, S. D., Anvarifar, F., Matungulu, H., & Kapelan, Z. (2014). Multi-objective rehabilitation of urban drainage systems under uncertainties. *Journal of Hydroinformatics*, 16(5), 1044-1061.

Walski, T. M. (2001). The wrong paradigm—Why water distribution optimization doesn't work. *Journal of Water Resources Planning and Management*, 127(4), 203–205.

Walski, T. M., Chase, D. V., Savic, D. A., Grayman, W. M., Beckwith, S., & Koelle, E. (2003). Advanced water distribution modeling and management: Haestad press.

Wang, Q., Guidolin, M., Savic, D., & Kapelan, Z. (2015). Two-objective design of benchmark problems of a water distribution system via MOEAs: Towards the best-known approximation of the true Pareto front. *American Society of Civil Engineers.*

Wang, Q., Zhou, Q., Lei, X., & Savić, D. A. (2018, August). Comparison of multi-objective optimization methods applied to urban drainage adaptation problems. *American Society of Civil Engineers.*

Wang, R., Purshouse, R. C., & Fleming, P. J. (2013). Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(4), 474-494.

Watanabe, S., Ito, M., & Sakakibara, K. (2015, May). A proposal on a decomposition-based evolutionary multi-objective optimization for large scale vehicle routing problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2581-2588). IEEE.

Yazdanfar, Z., & Sharma, A. (2015). Urban drainage system planning and design–challenges with climate change and urbanization: a review. *Water Science and Technology*, 72(2), 165-179.

Yazdi, J. (2018). Rehabilitation of urban drainage systems using a resilience-based approach. *Water resources management*, 32(2), 721-734.

Yazdi, J., Sadollah, A., Lee, E. H., Yoo, D. G., & Kim, J. H. (2017). Application of multi-objective evolutionary algorithms for the rehabilitation of storm sewer pipe networks. *Journal of Flood Risk Management*, 10(3), 326-338.

Yazdi, J., Yoo, D. G., & Kim, J. H. (2017). Comparative study of multi-objective evolutionary algorithms for hydraulic rehabilitation of urban drainage networks. *Urban Water Journal*, 14(5), 483-492.

Yu, Y., Zhou, Y., Guo, Z., van Duin, B., & Zhang, W. (2022). A new LID spatial allocation optimization system at neighborhood scale: Integrated SWMM with PICEA-g using MATLAB as the platform. *Science of The Total Environment*, 831, 154843.

Zare, S. O., Saghafian, B., Shamsai, A., & Nazif, S. (2012). Multi-objective optimization using evolutionary algorithms for qualitative and quantitative control of urban runoff. *Hydrology & Earth System Sciences Discussions*, 9(1).

Appendix

Example A1. An example of Eq. 3.5-3.7

An example of an optimization problem with 2 minimizing objectives, 4 candidate solutions, and 4 goal vectors are used in Paknejad et al. (2021) to explain this fitness method as shown in Figure A1.

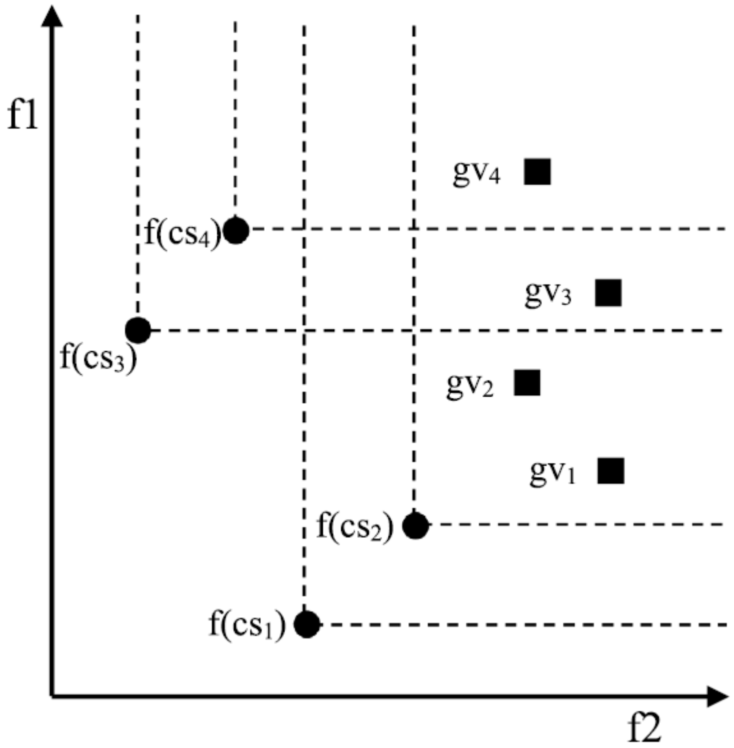


Figure A1 An example of how to calculate fitness value in a two-objective problem (Paknejad et al., 2021)

Candidate solutions CS_2 and CS_4 are parent solutions while CS_1 and CS_3 are offspring solutions, so $N = 2$. In this example, the fitness values of the four candidate solutions are calculated as:

$$Fit_{cs1} = \frac{1}{n_{gv1}} + \frac{1}{n_{gv2}} + \frac{1}{n_{gv3}} + \frac{1}{n_{gv4}} = \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{19}{12}$$

$$Fit_{cs2} = \frac{1}{n_{gv1}} + \frac{1}{n_{gv2}} + \frac{1}{n_{gv3}} + \frac{1}{n_{gv4}} = \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{19}{12}$$

$$Fit_{cs3} = \frac{1}{n_{gv3}} + \frac{1}{n_{gv4}} = \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$$

$$Fit_{cs4} = \frac{1}{n_{gv4}} = \frac{1}{4}$$

In this case, the fitness values of *CS₁* and *CS₂* are the same while *CS₃* is dominated by *CS₄*. Using the calculation of fitness value of *CS₃* as an example, *gv₃* and *gv₄* are the two goal vectors that are dominated by *CS₃*. Therefore, the fitness value of *CS₃* is equal to the sum of $\frac{1}{n_{gv3}}$ and $\frac{1}{n_{gv4}}$.

The fitness values of the four goal vectors are calculated as:

$$Fit_{gv1} = \frac{1}{1 + \left(\frac{n_{gv1} - 1}{2N - 1}\right)} = \frac{1}{1 + \left(\frac{2-1}{4-1}\right)} = \frac{3}{4}$$

$$Fit_{gv2} = \frac{1}{1 + \left(\frac{n_{gv2} - 1}{2N - 1}\right)} = \frac{1}{1 + \left(\frac{2-1}{4-1}\right)} = \frac{3}{4}$$

$$Fit_{gv3} = \frac{1}{1 + \left(\frac{n_{gv3} - 1}{2N - 1}\right)} = \frac{1}{1 + \left(\frac{3-1}{4-1}\right)} = \frac{3}{5}$$

$$Fit_{gv4} = \frac{1}{1 + \left(\frac{n_{gv4} - 1}{2N - 1}\right)} = \frac{1}{1 + \left(\frac{4-1}{4-1}\right)} = \frac{1}{2}$$

Example A2. An example of Eq. 3.8-3.9

New fitness values of the four candidate solutions are calculated as (Yu et al, 2022):

$$rank_{cs1} = 1 + 0 = 1$$

$$rank_{cs2} = 1 + 1 = 2$$

$$rank_{cs3} = 1 + 0 = 1$$

$$rank_{cs4} = 1 + 1 = 2$$

$$Fit_{cs1} = \frac{1}{n_{gv1}} \times Fit_{gv1} + \frac{1}{n_{gv2}} \times Fit_{gv2} + \frac{1}{n_{gv3}} \times Fit_{gv3} + \frac{1}{n_{gv4}} \times Fit_{gv4} + \frac{1}{rank_{cs1}}$$
$$= \frac{1}{2} \times \frac{3}{4} + \frac{1}{2} \times \frac{3}{4} + \frac{1}{3} \times \frac{3}{5} + \frac{1}{4} \times \frac{1}{2} + \frac{1}{1} = \frac{22}{10}$$

$$Fit_{cs2} = \frac{1}{n_{gv1}} \times Fit_{gv1} + \frac{1}{n_{gv2}} \times Fit_{gv2} + \frac{1}{n_{gv3}} \times Fit_{gv3} + \frac{1}{n_{gv4}} \times Fit_{gv4} + \frac{1}{rank_{cs2}}$$
$$= \frac{1}{2} \times \frac{3}{4} + \frac{1}{2} \times \frac{3}{4} + \frac{1}{3} \times \frac{3}{5} + \frac{1}{4} \times \frac{1}{2} + \frac{1}{2} = \frac{17}{10}$$

$$Fit_{cs3} = \frac{1}{n_{gv3}} \times Fit_{gv3} + \frac{1}{n_{gv4}} \times Fit_{gv4} + \frac{1}{rank_{cs3}} = \frac{1}{3} \times \frac{3}{5} + \frac{1}{4} \times \frac{1}{2} + \frac{1}{1} = \frac{53}{40}$$

$$Fit_{cs4} = \frac{1}{n_{gv4}} \times Fit_{gv4} + \frac{1}{rank_{cs4}} = \frac{1}{4} \times \frac{1}{2} + \frac{1}{2} = \frac{5}{8}$$

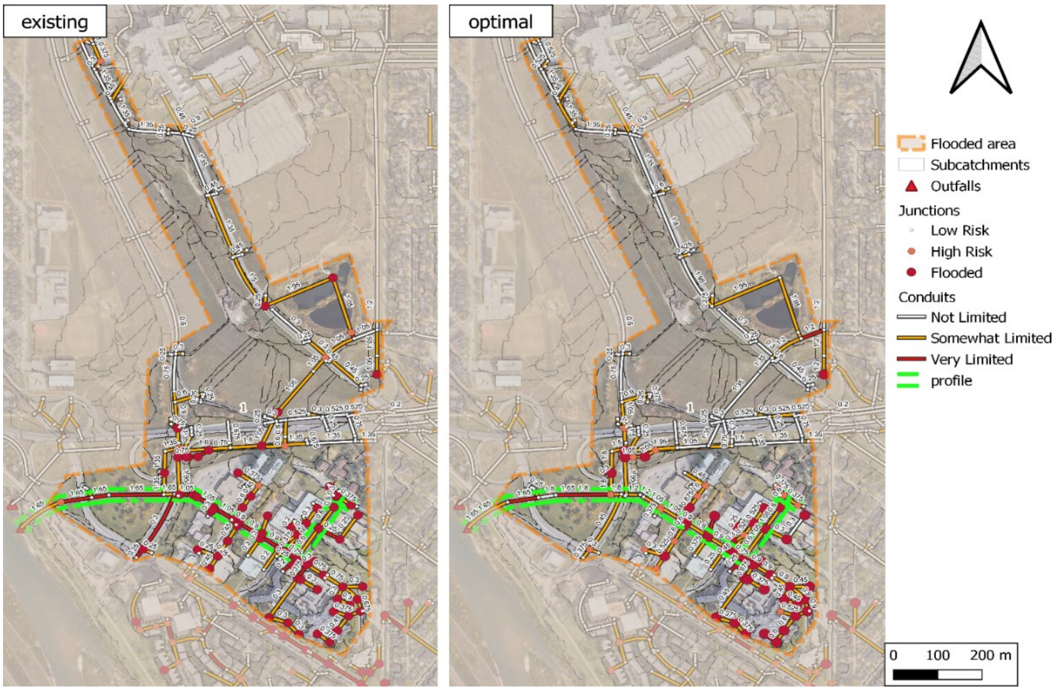


Figure A2 The comparison of Flooded Area 1 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

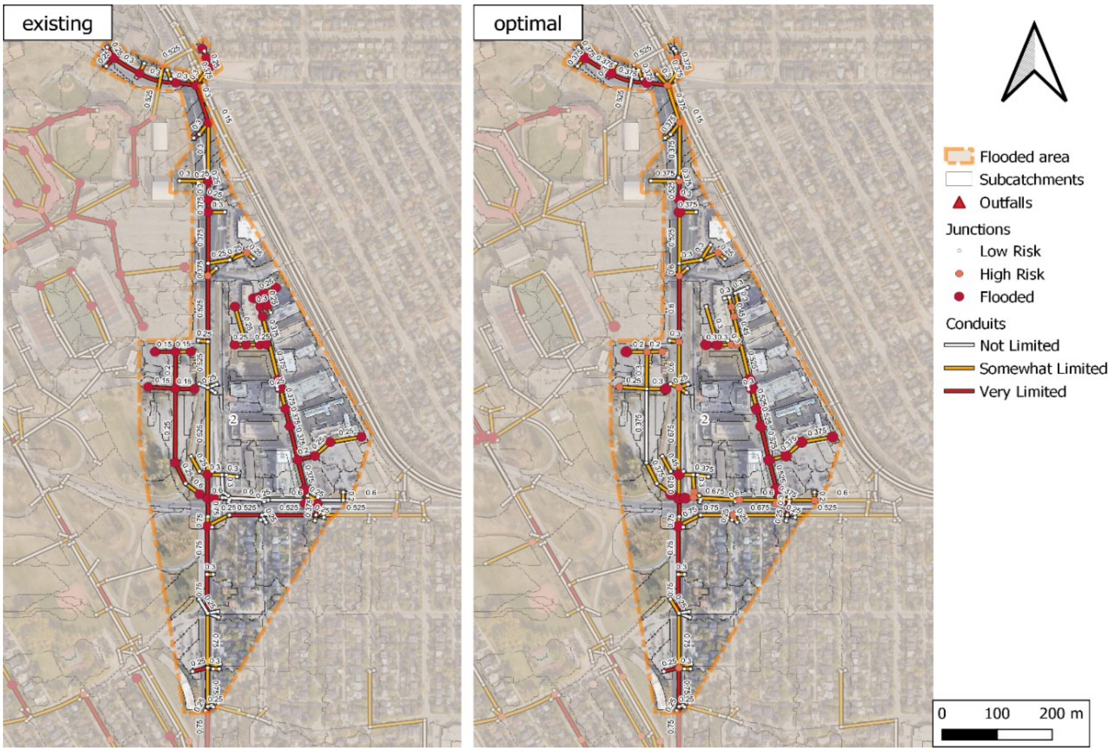


Figure A3 The comparison of Flooded Area 2 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

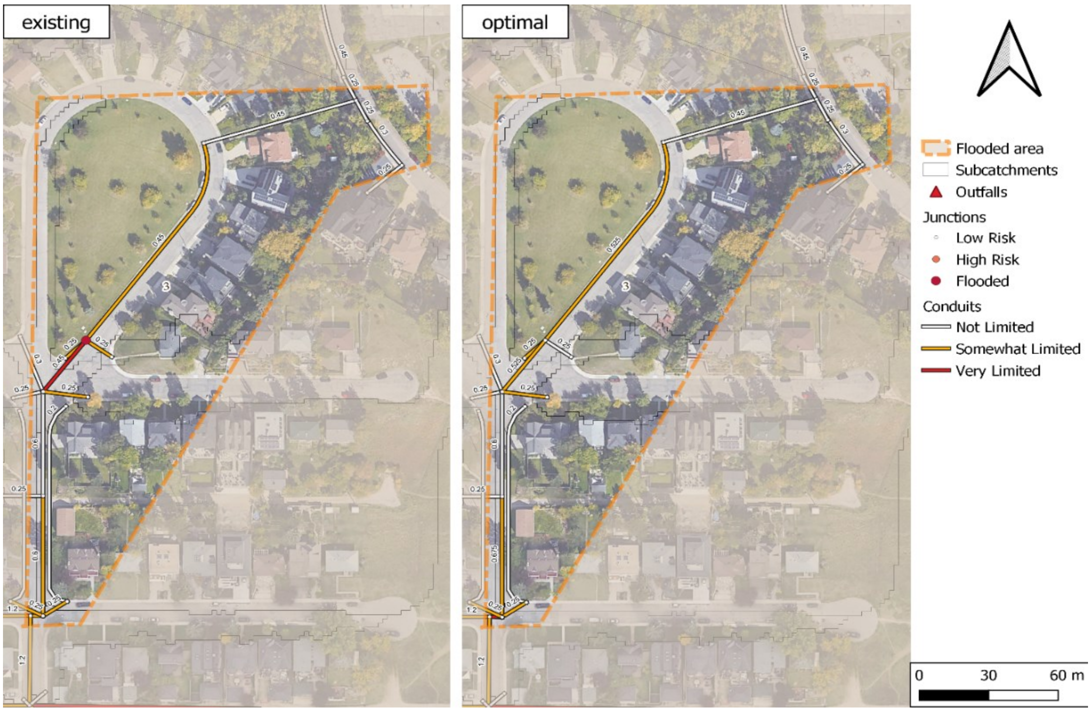


Figure A4 The comparison of Flooded Area 3 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

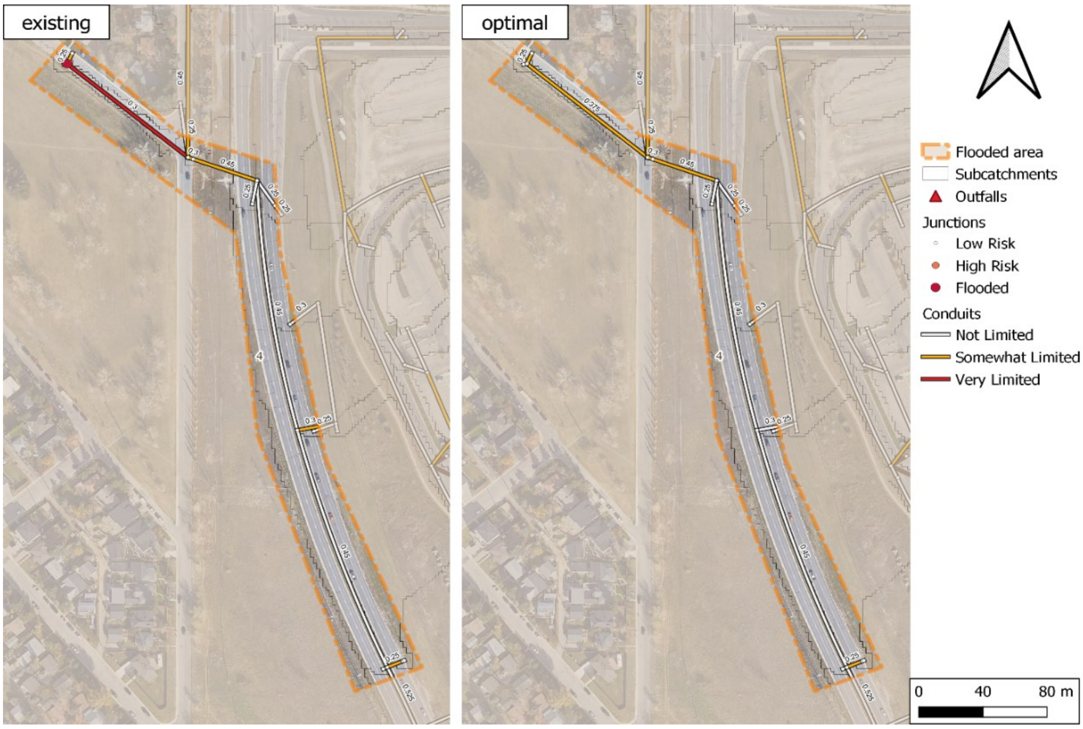


Figure A5 The comparison of Flooded Area 4 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

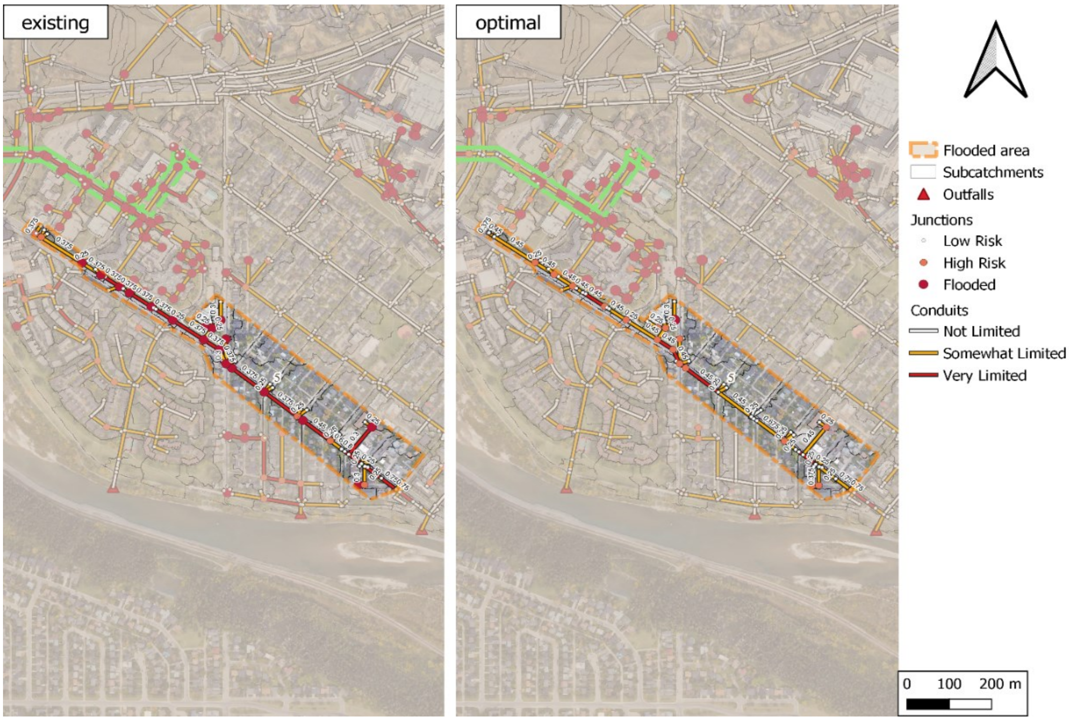


Figure A6 The comparison of Flooded Area 5 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

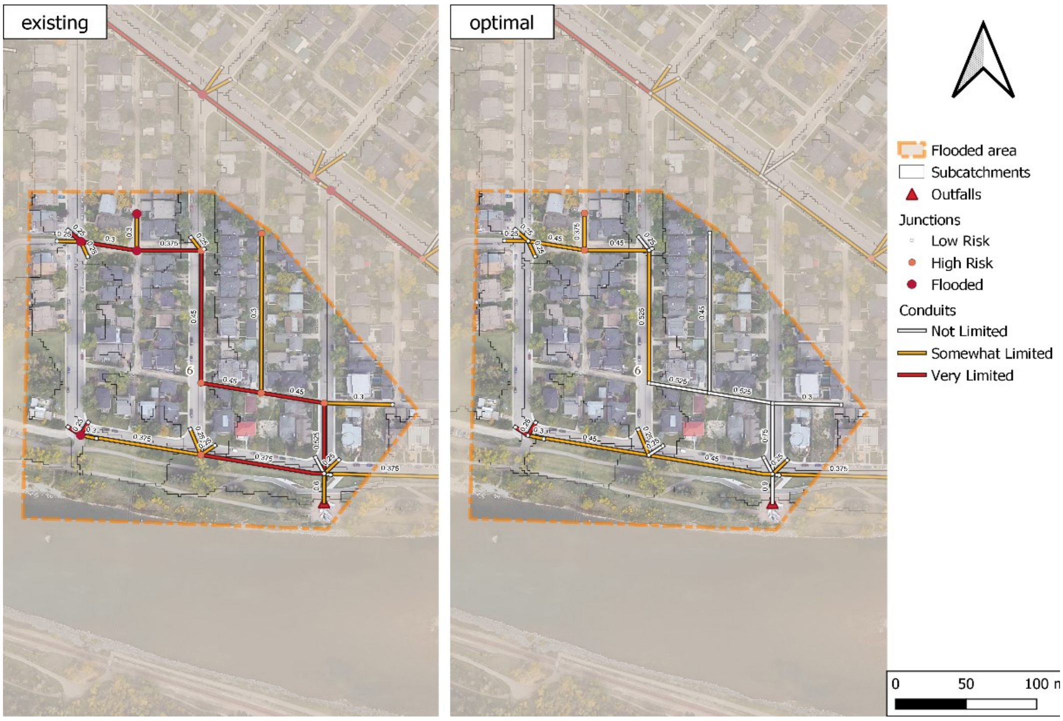


Figure A7 The comparison of Flooded Area 6 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

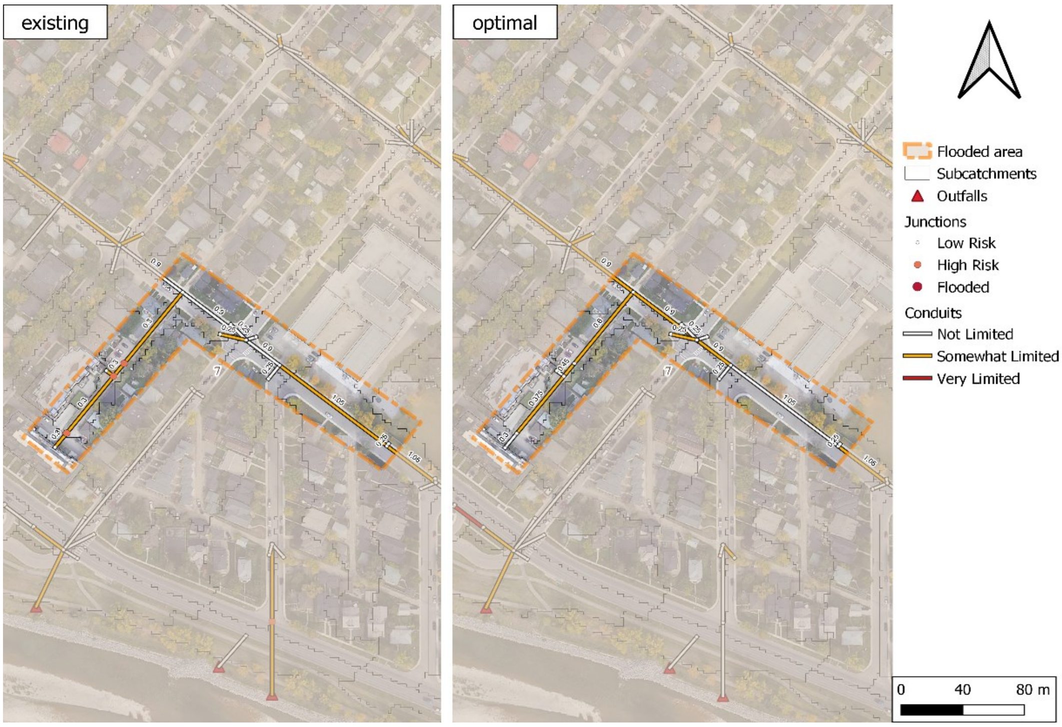


Figure A8 The comparison of Flooded Area 7 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

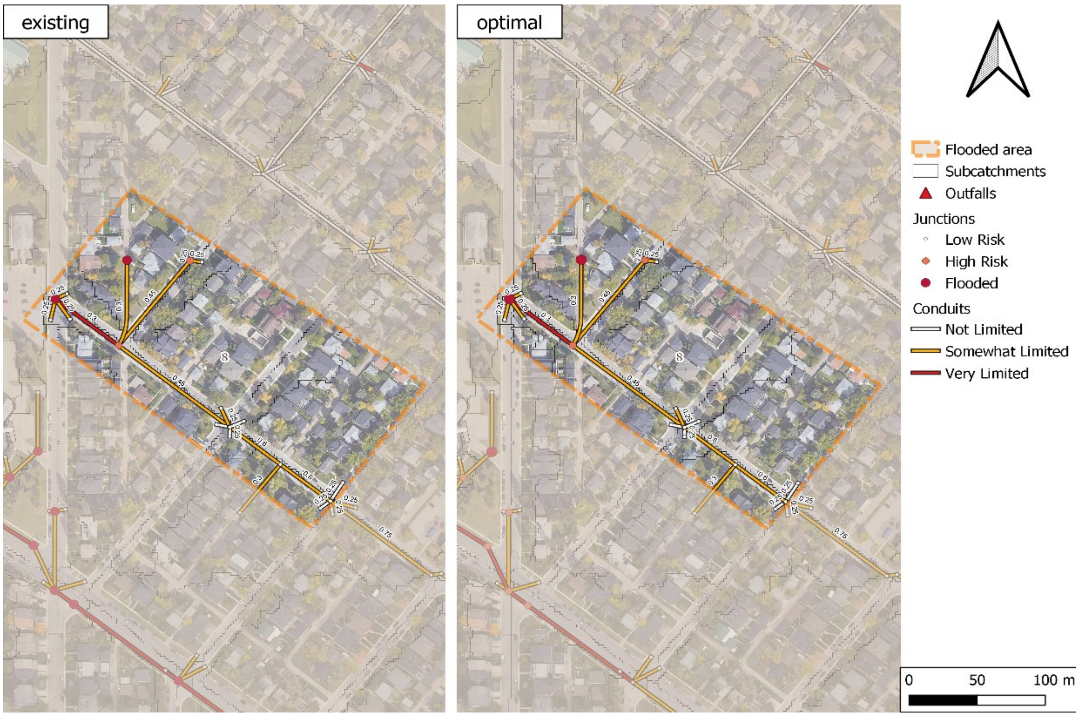


Figure A9 The comparison of Flooded Area 8 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

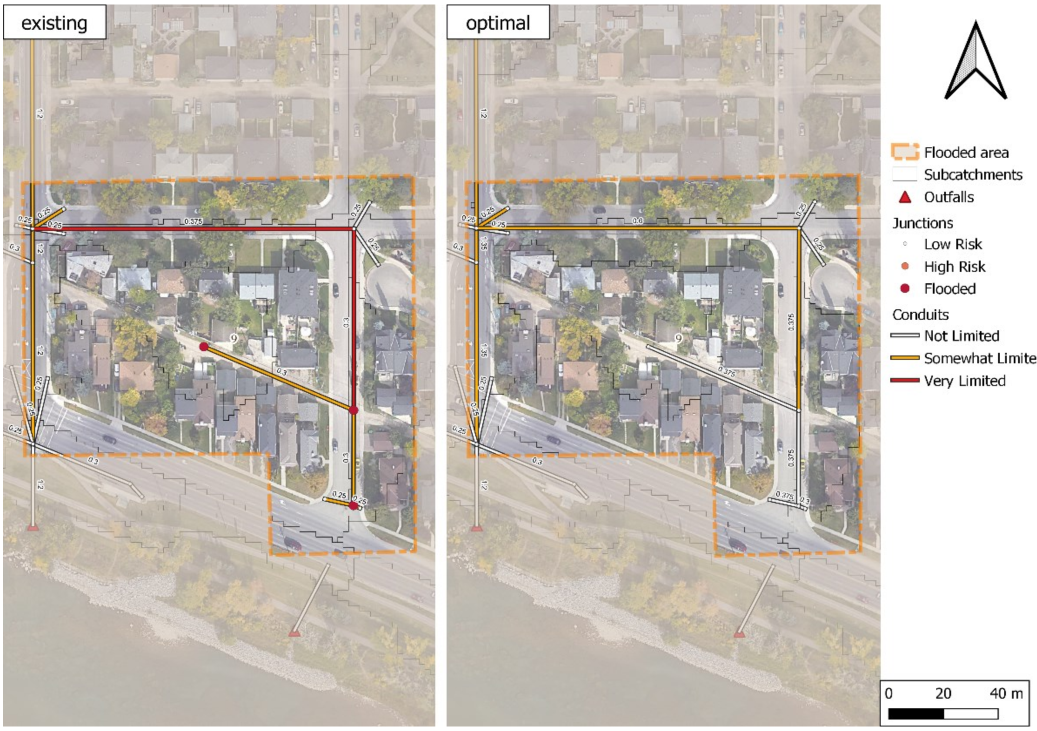


Figure A10 The comparison of Flooded Area 9 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

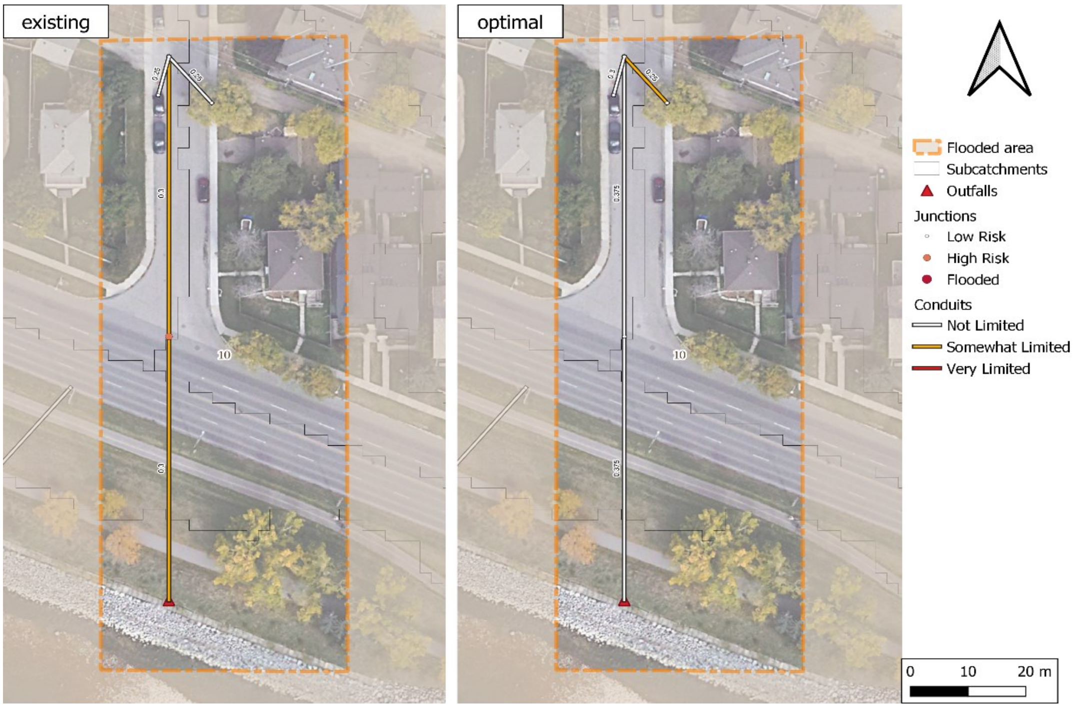


Figure A11 The comparison of Flooded Area 10 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

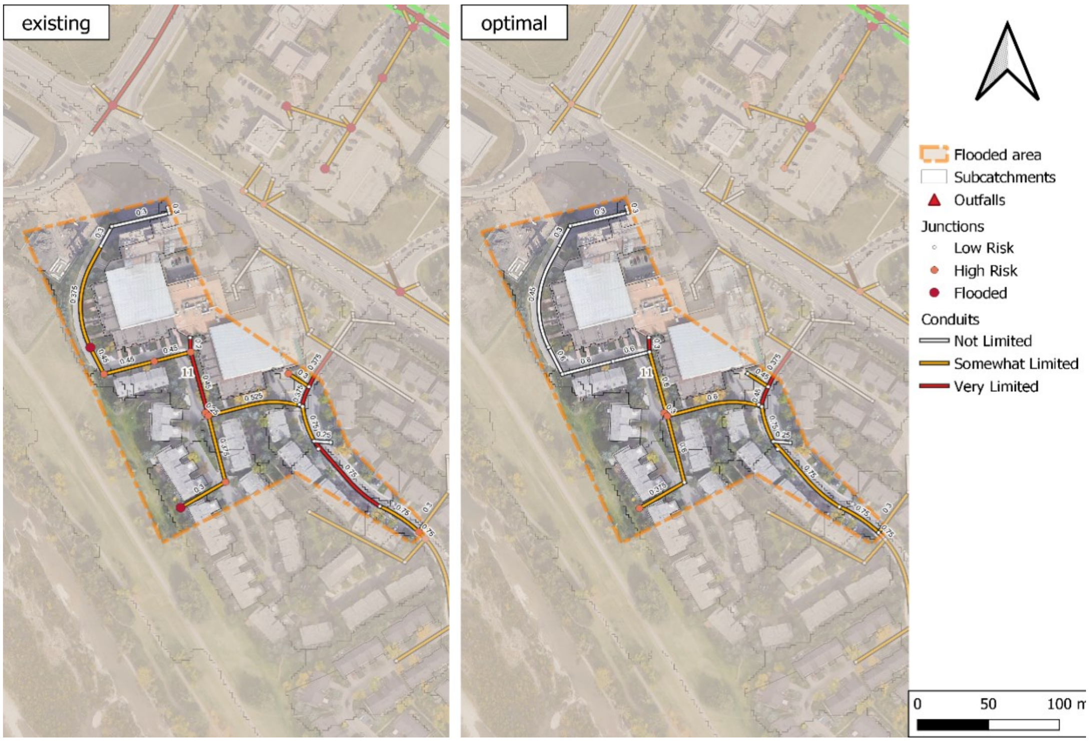


Figure A12 The comparison of Flooded Area 11 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

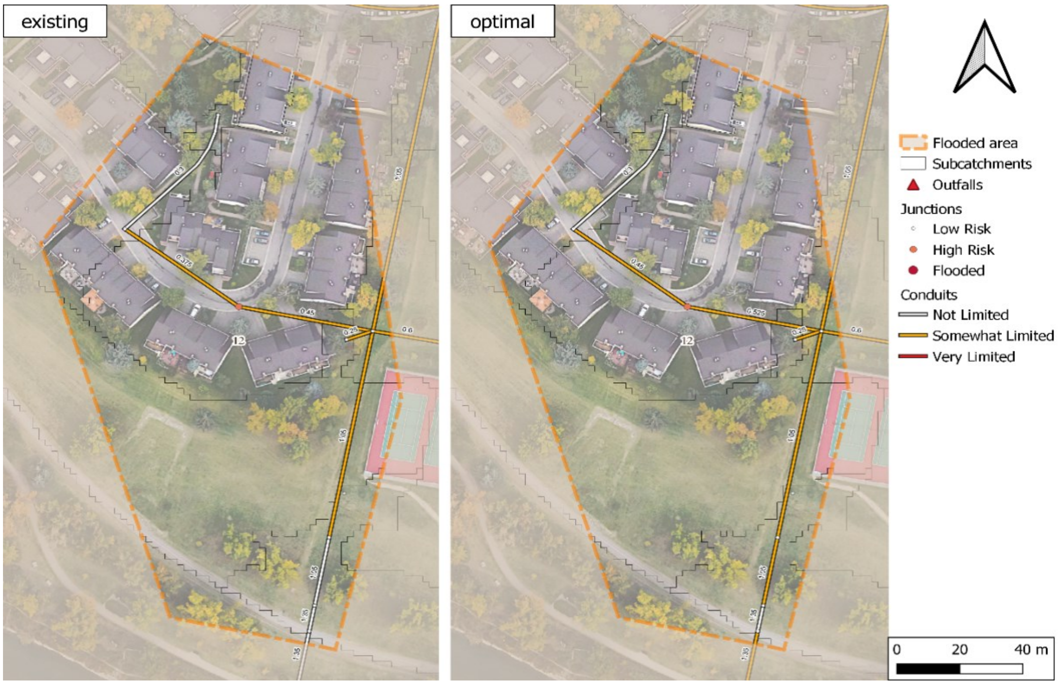


Figure A13 The comparison of Flooded Area 12 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

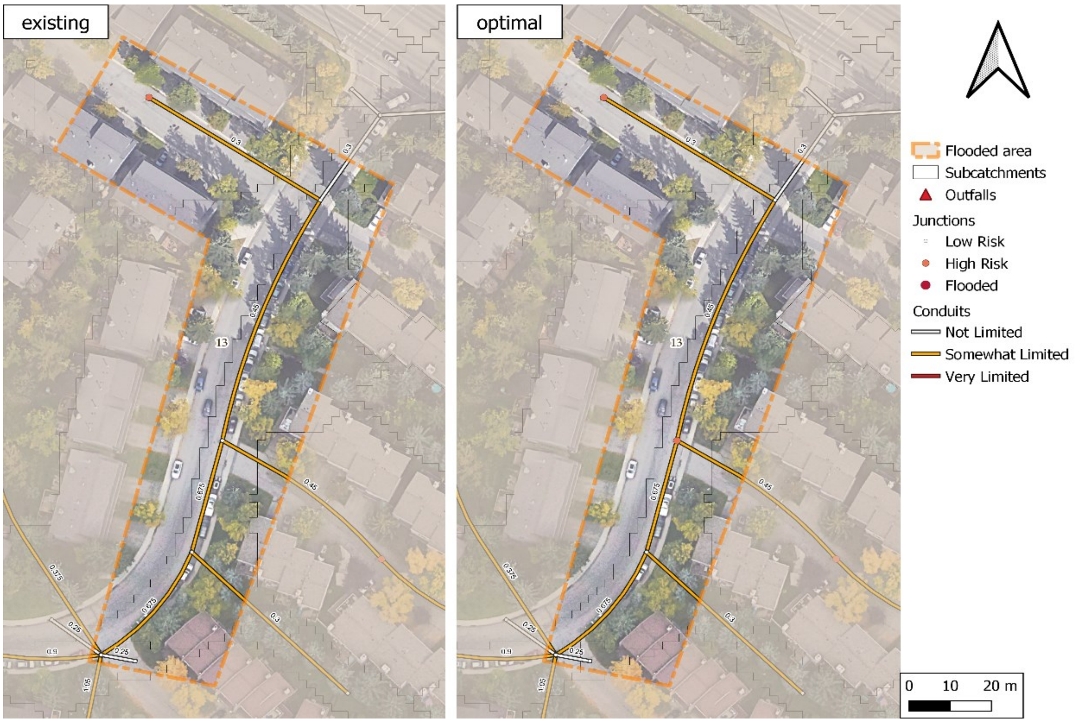


Figure A14 The comparison of Flooded Area 13 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

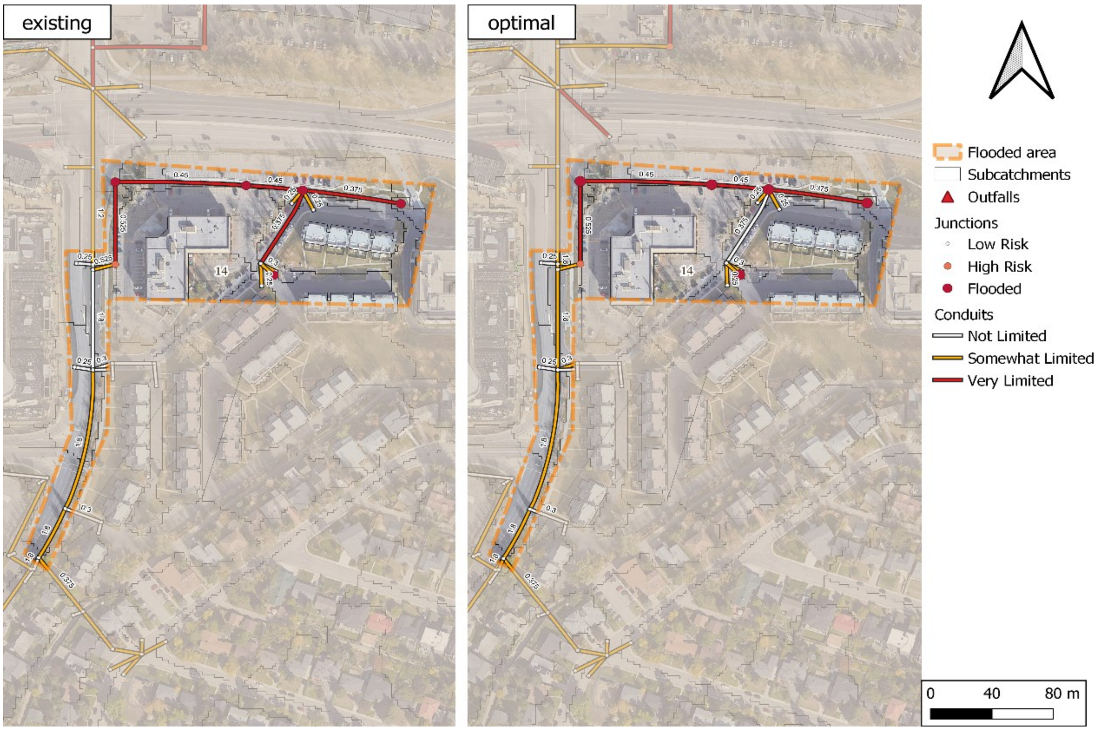


Figure A15 The comparison of Flooded Area 14 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

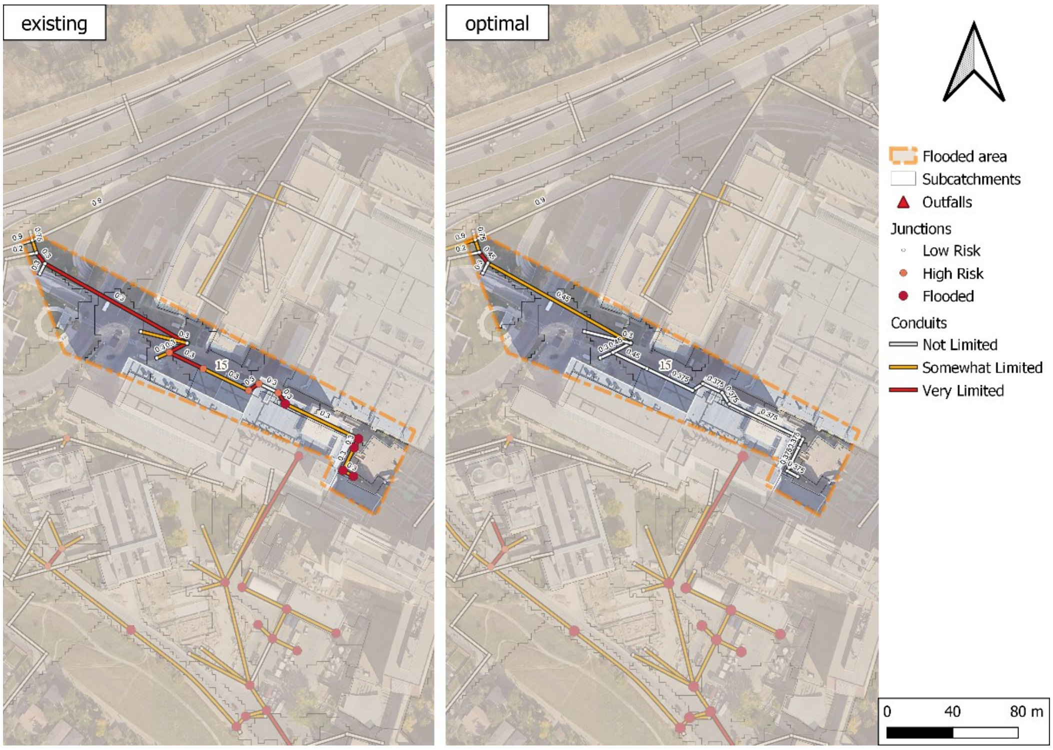


Figure A16 The comparison of Flooded Area 15 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

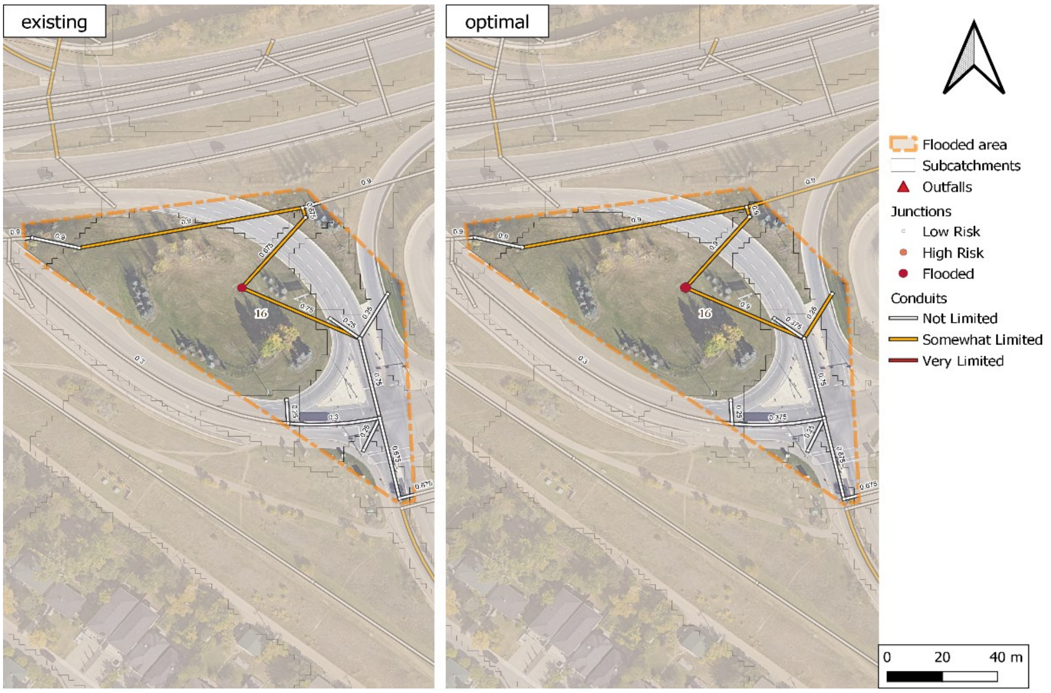


Figure A17 The comparison of Flooded Area 16 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

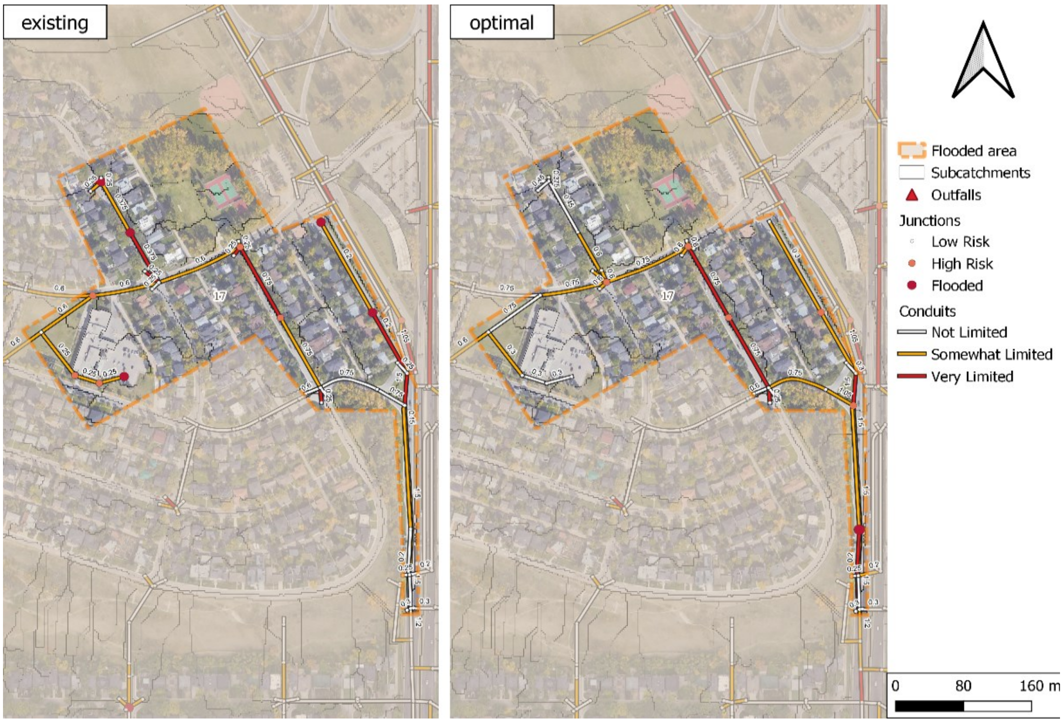


Figure A18 The comparison of Flooded Area 17 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

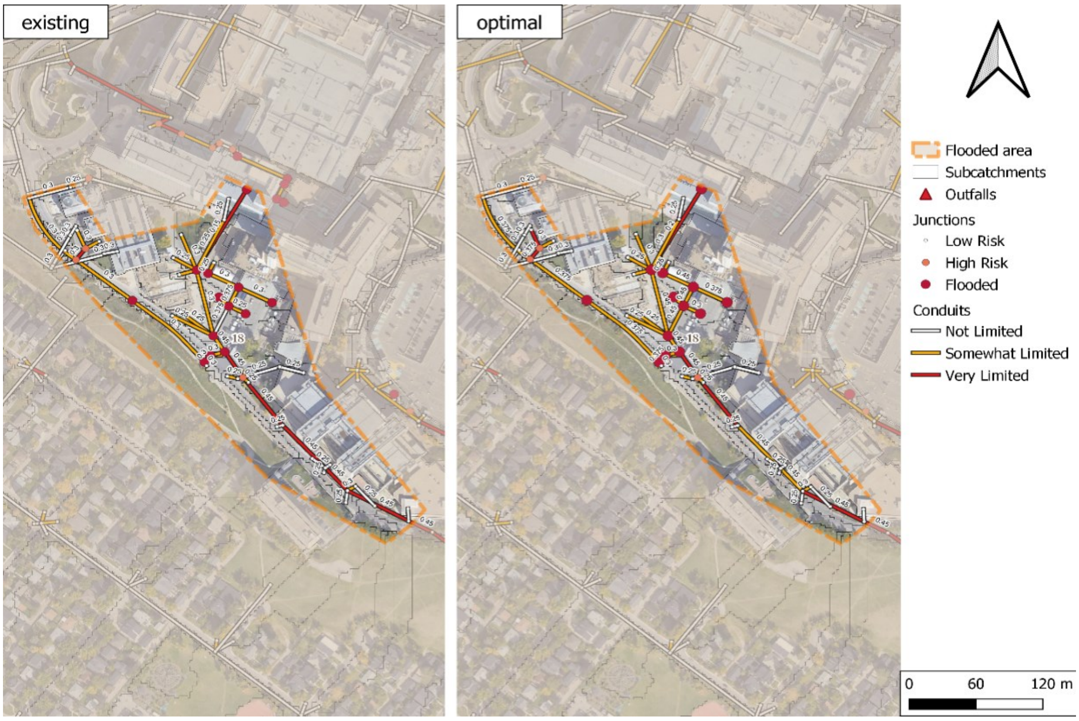


Figure A19 The comparison of Flooded Area 18 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

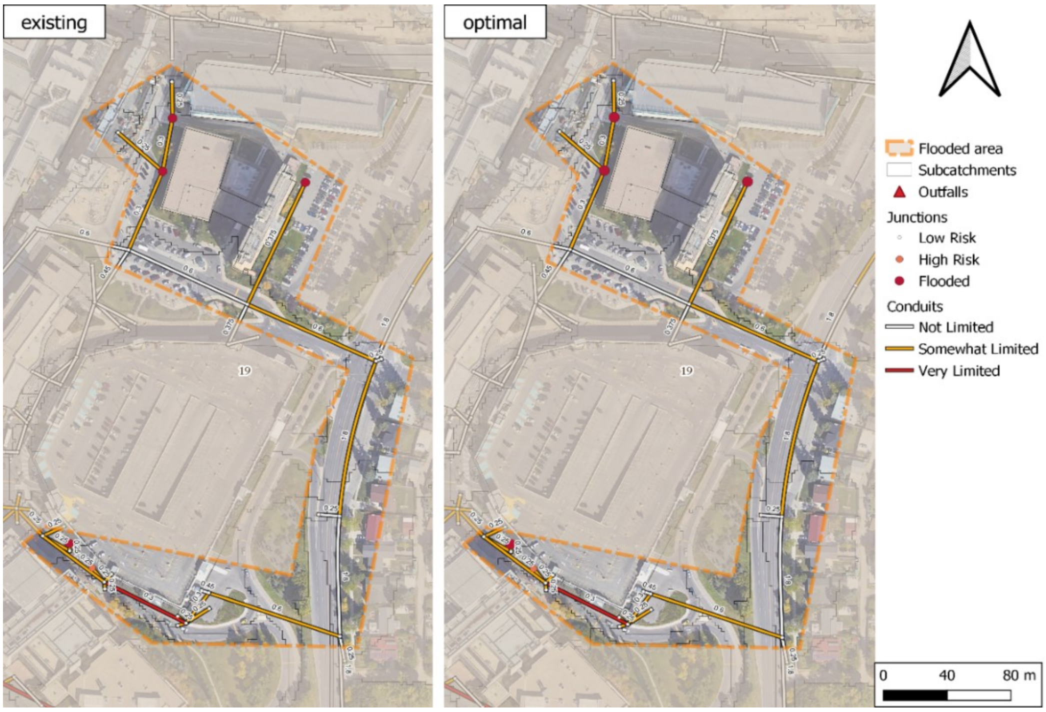


Figure A20 The comparison of Flooded Area 19 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

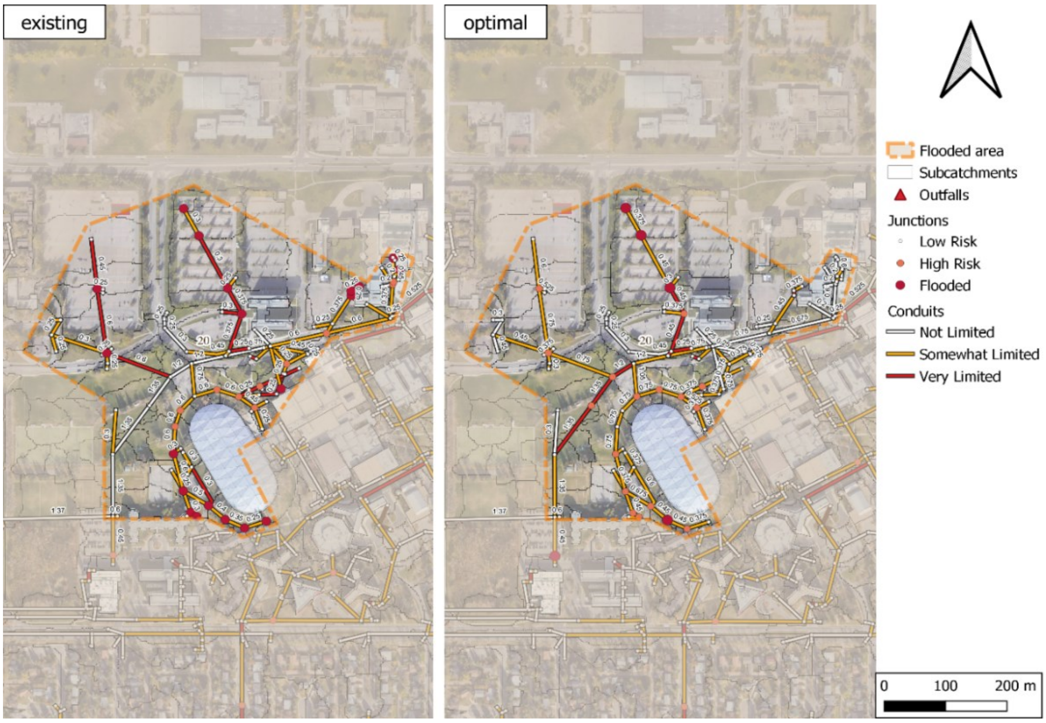


Figure A21 The comparison of Flooded Area 20 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

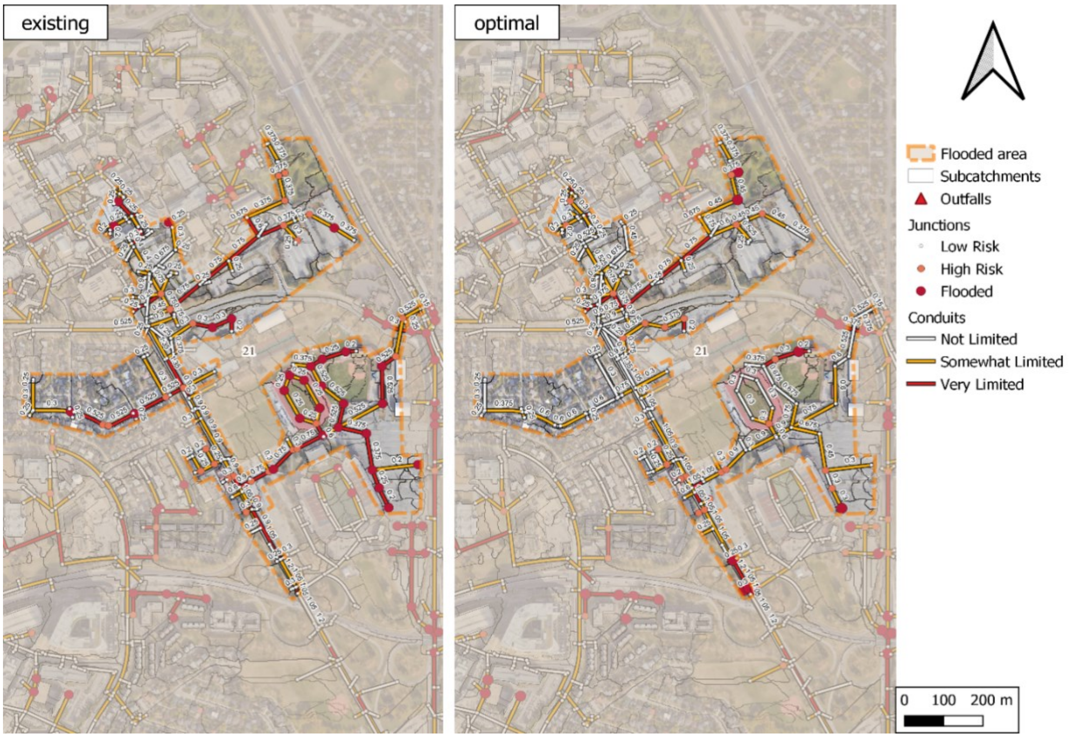


Figure A22 The comparison of Flooded Area 21 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

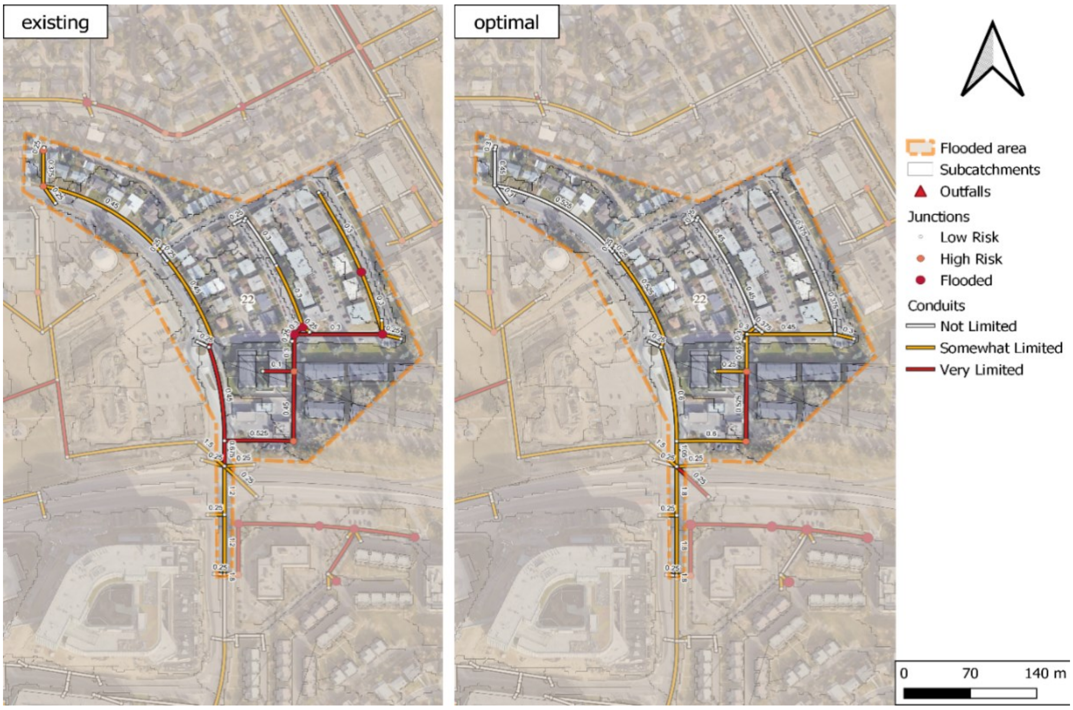


Figure A23 The comparison of Flooded Area 22 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

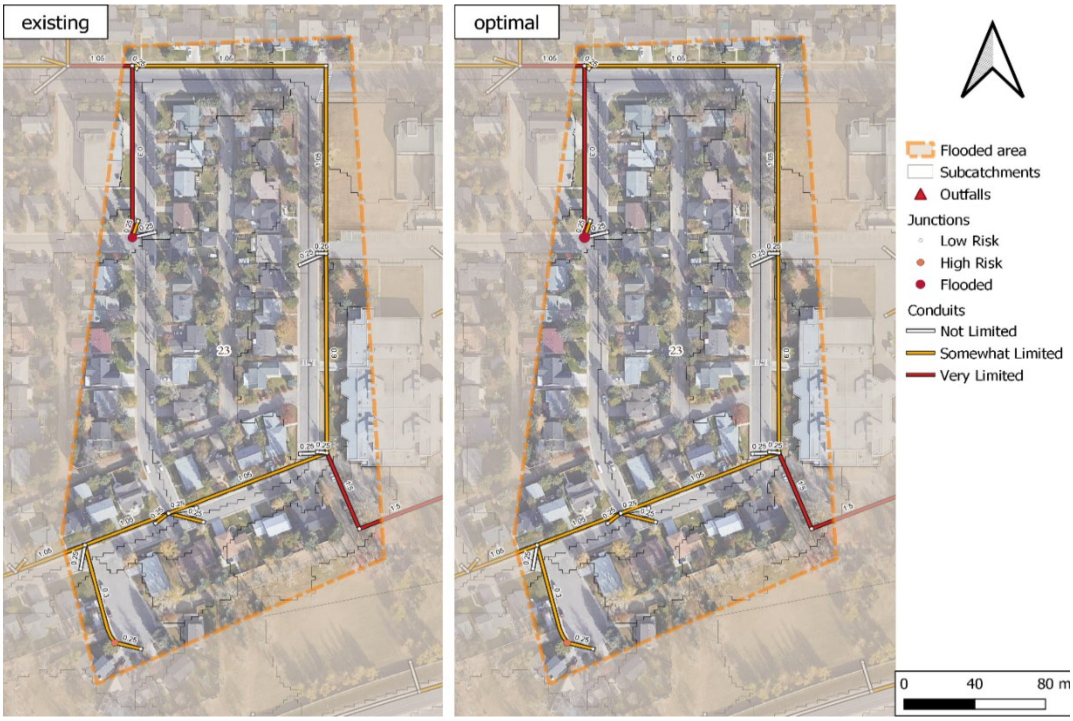


Figure A24 The comparison of Flooded Area 23 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

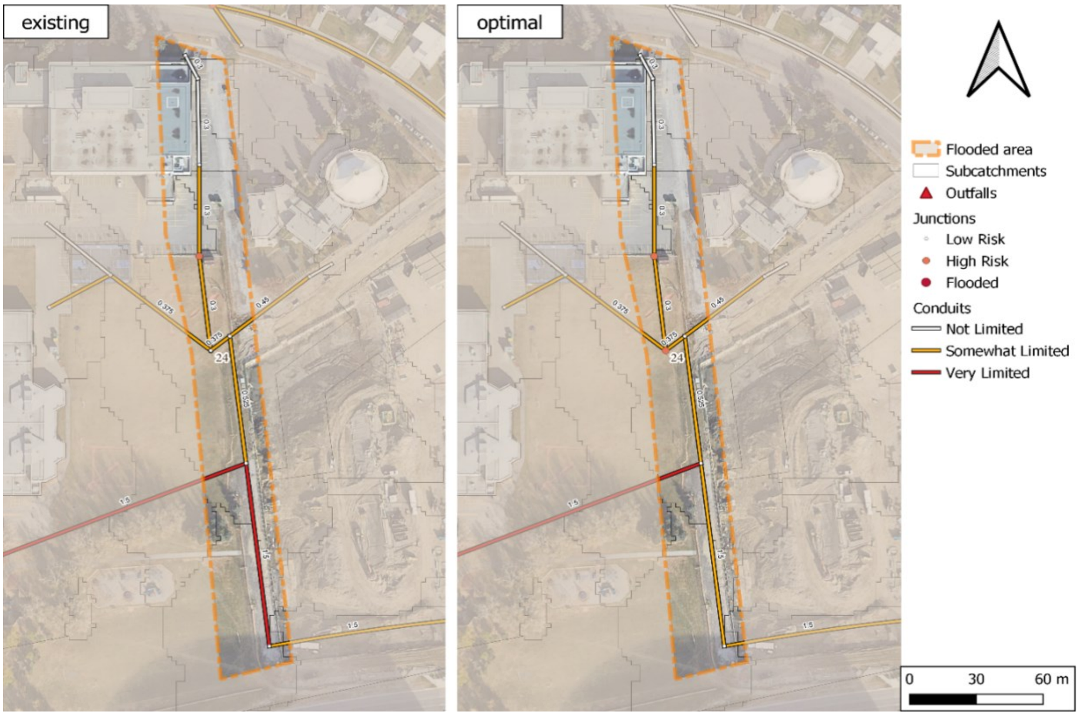


Figure A25 The comparison of Flooded Area 24 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

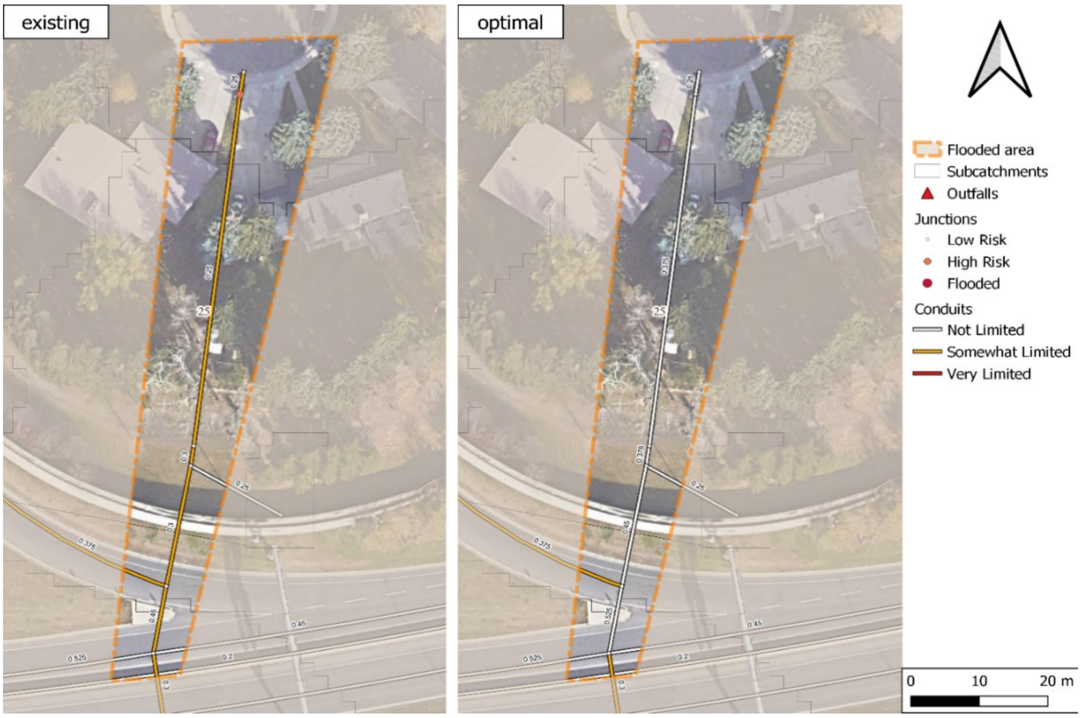


Figure A26 The comparison of Flooded Area 25 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

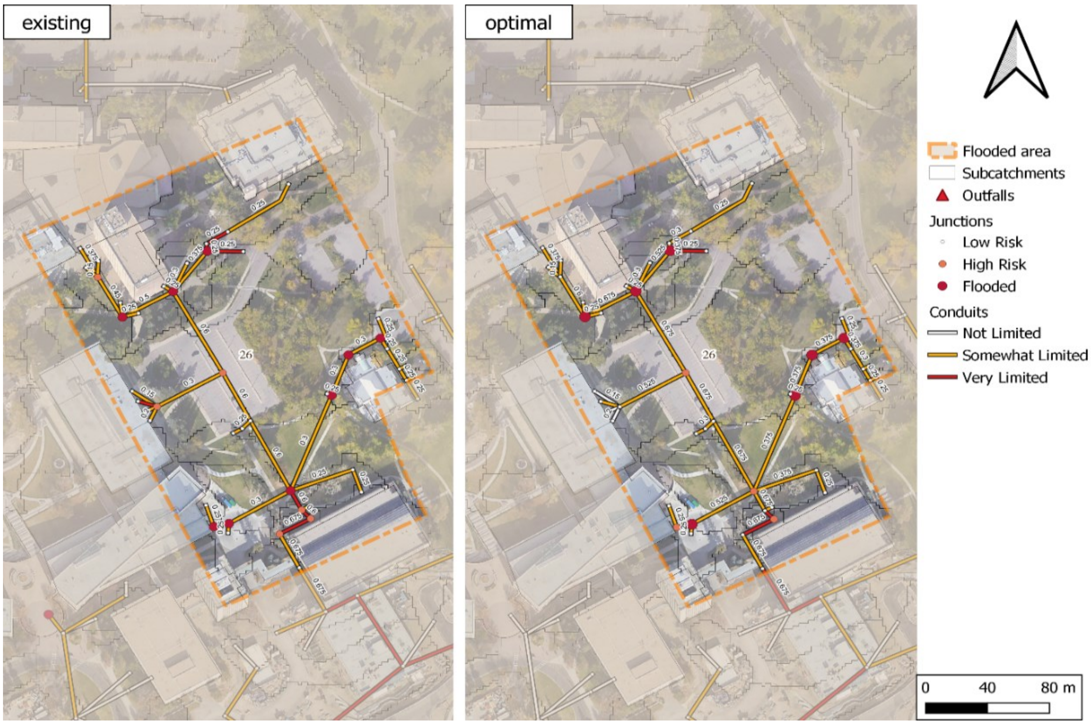


Figure A27 The comparison of Flooded Area 26 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

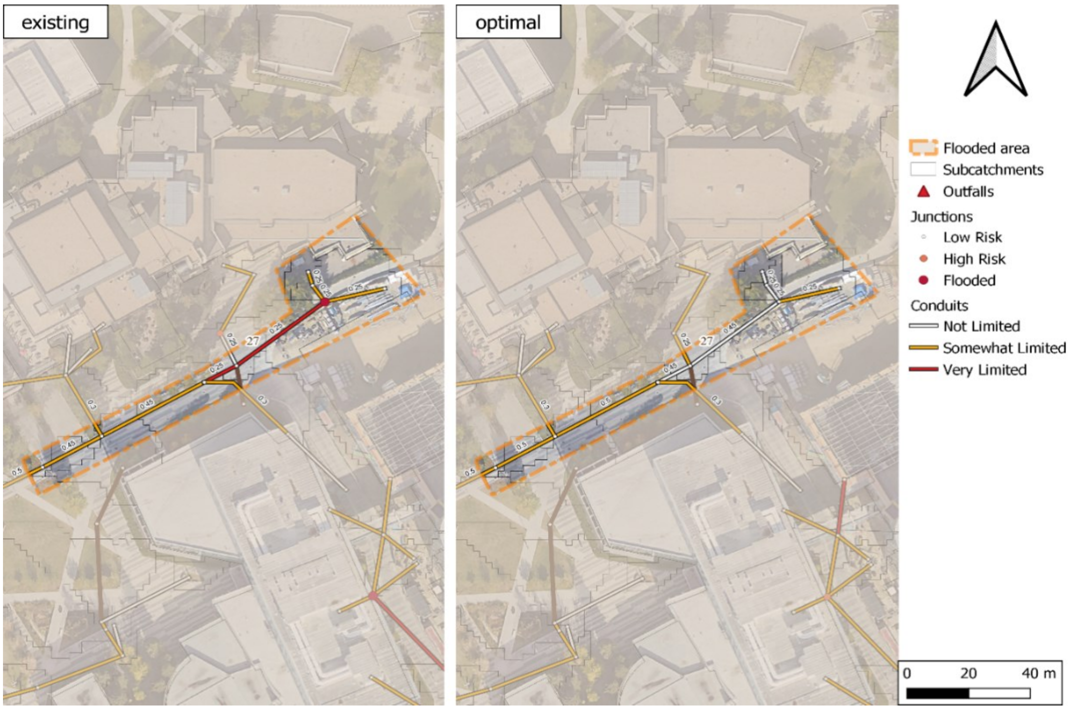


Figure A28 The comparison of Flooded Area 27 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

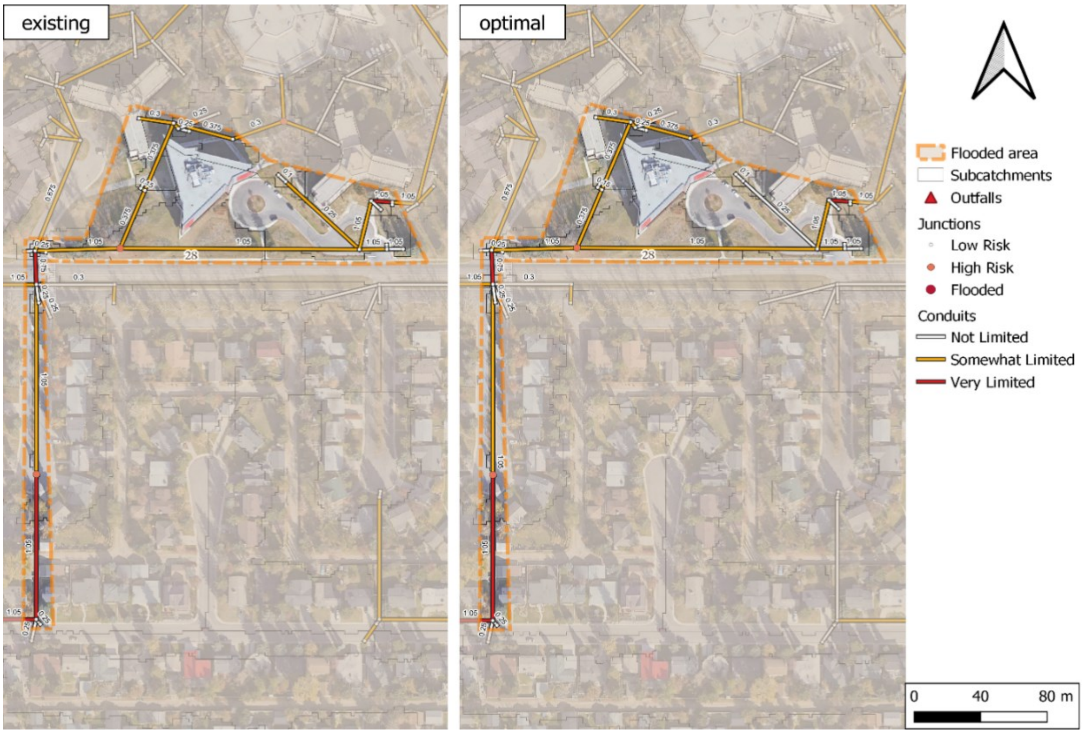


Figure A29 The comparison of Flooded Area 28 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

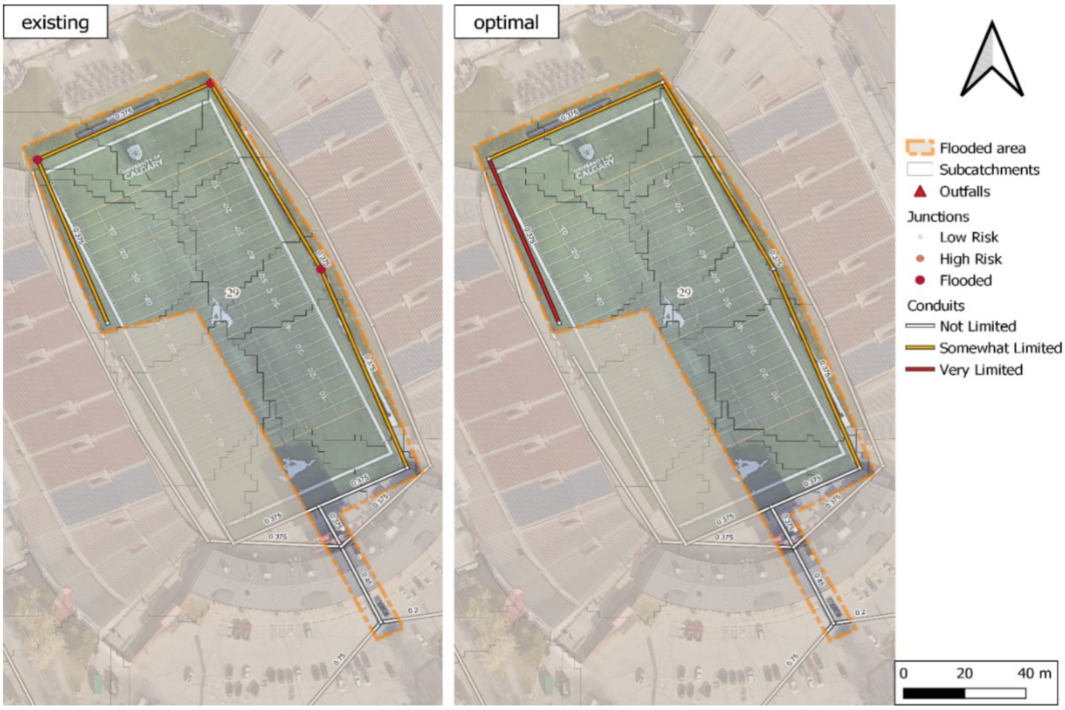


Figure A30 The comparison of Flooded Area 29 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

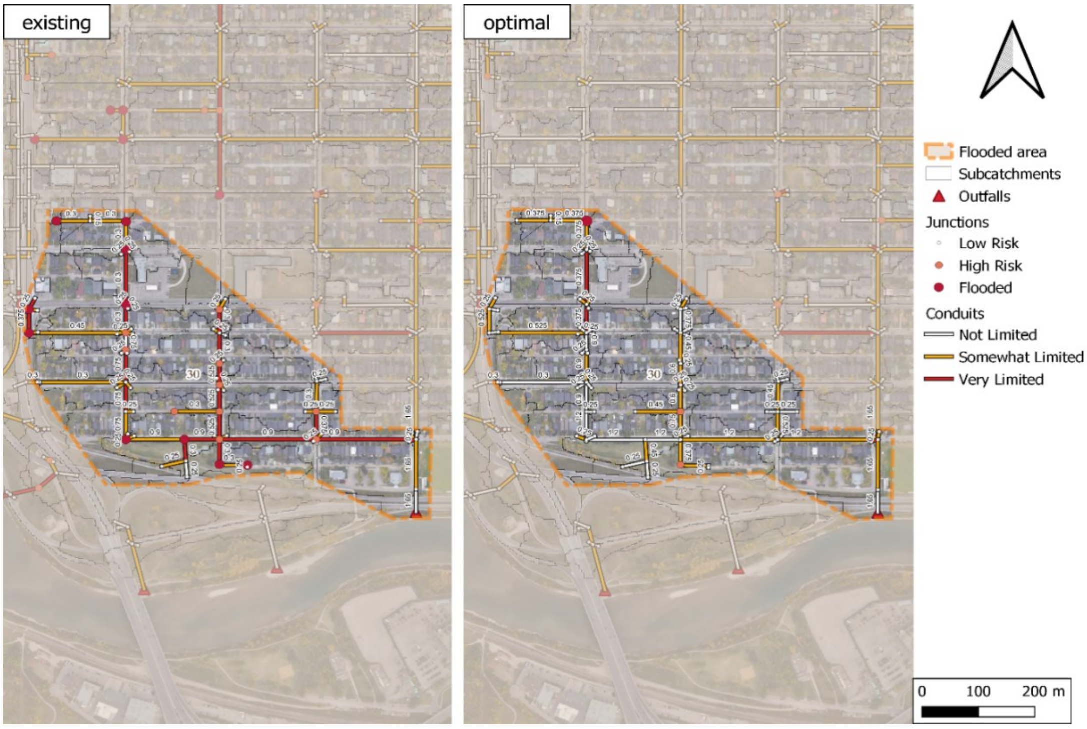


Figure A31 The comparison of Flooded Area 30 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

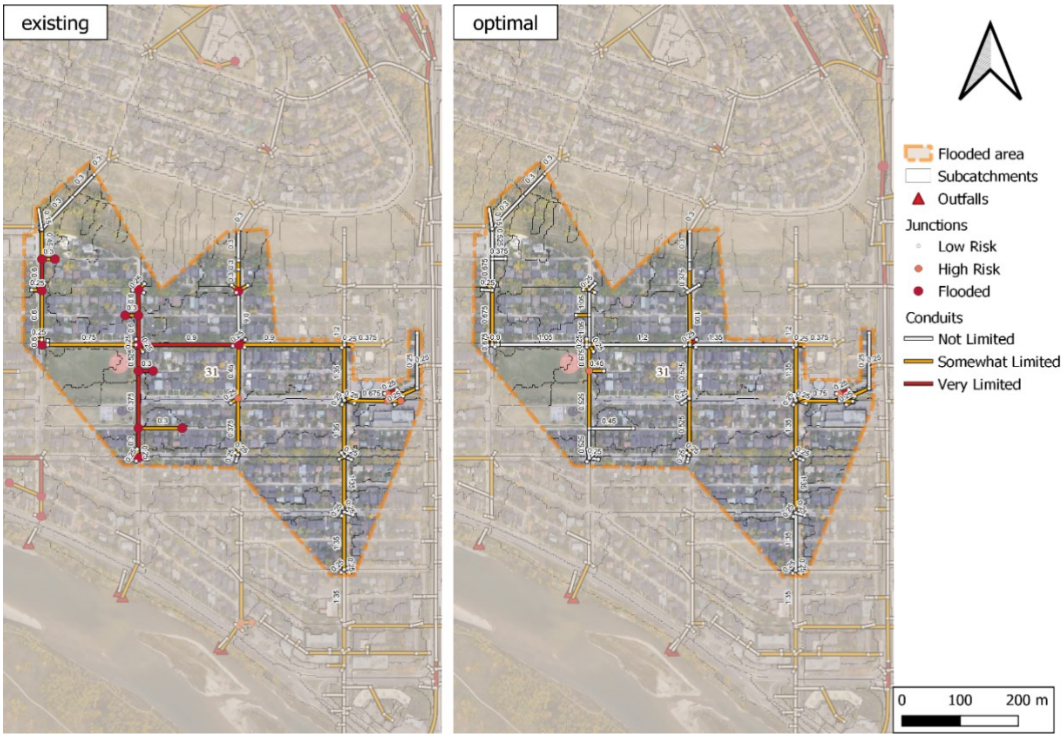


Figure A32 The comparison of Flooded Area 31 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

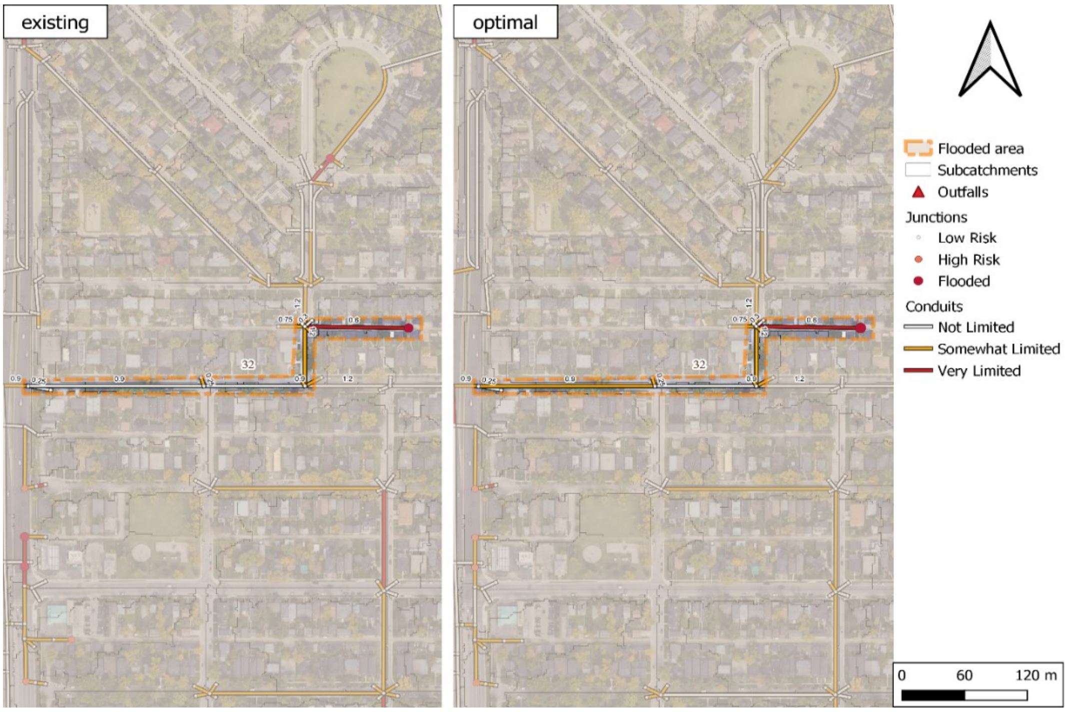


Figure A33 The comparison of Flooded Area 32 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

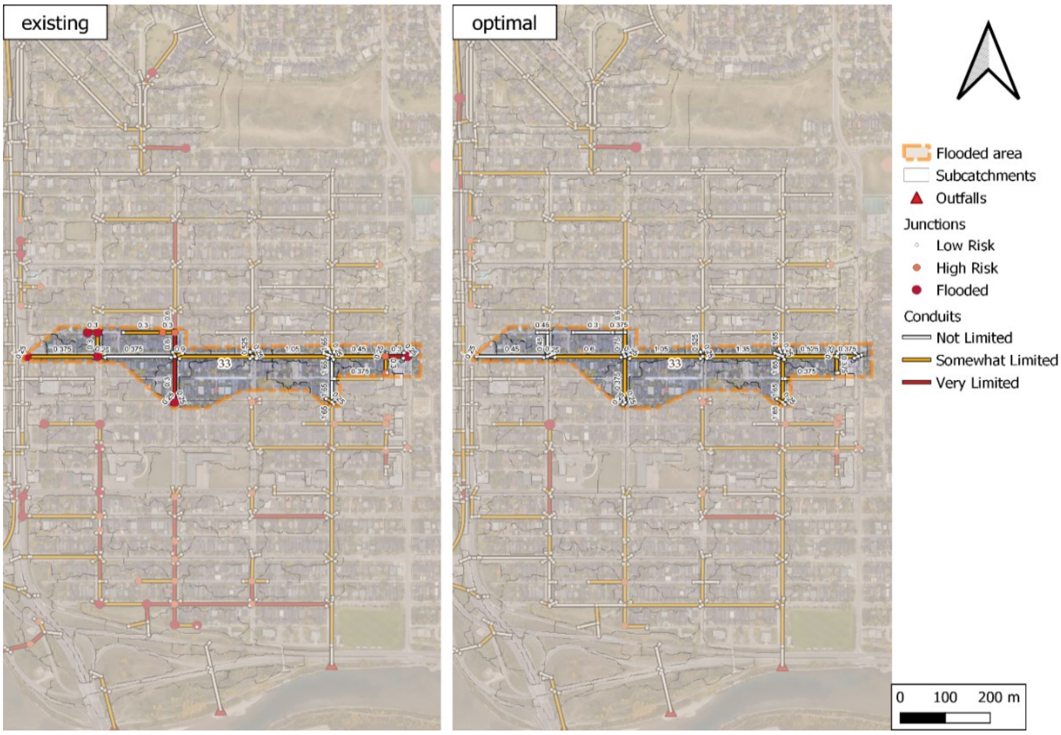


Figure A34 The comparison of Flooded Area 33 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

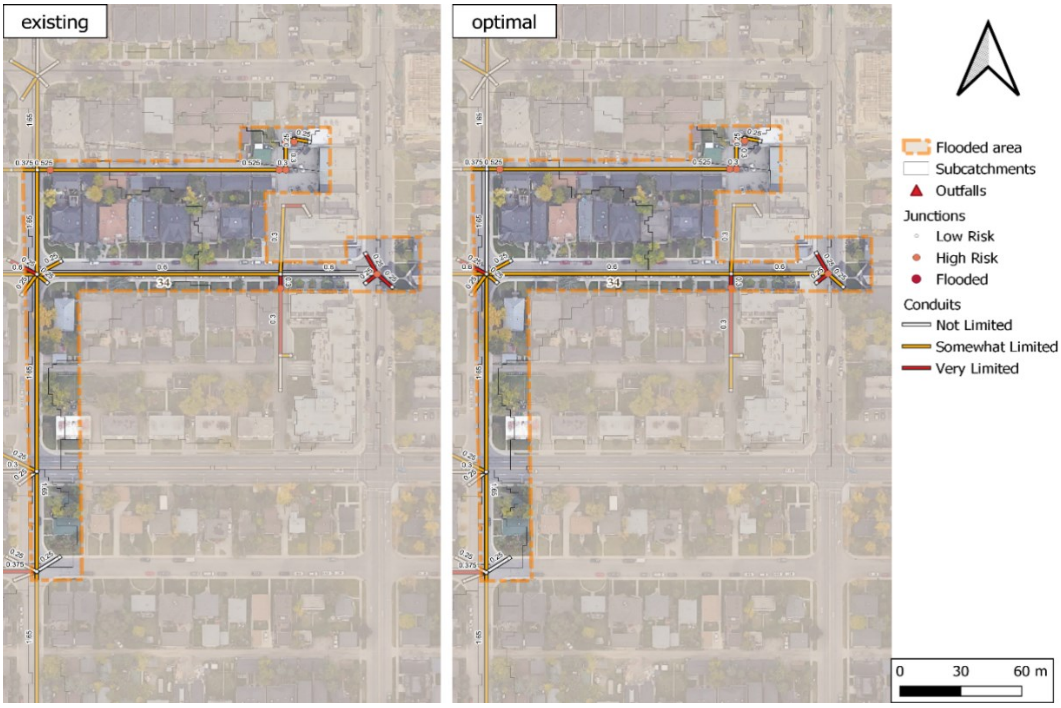


Figure A35 The comparison of Flooded Area 34 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

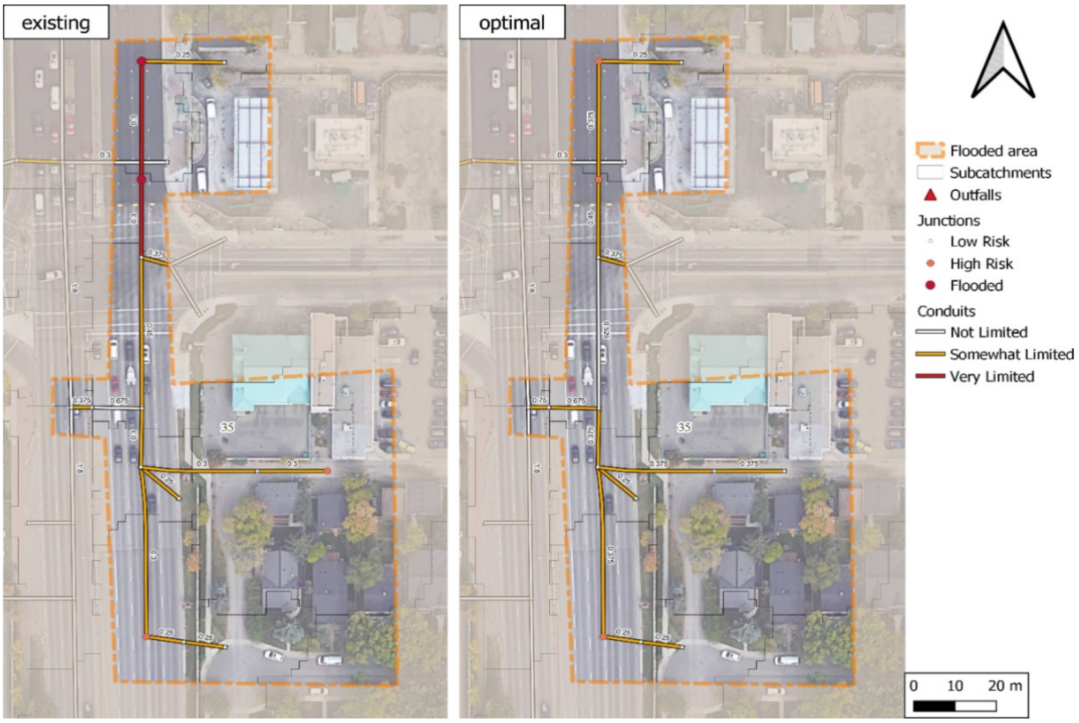


Figure A36 The comparison of Flooded Area 35 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

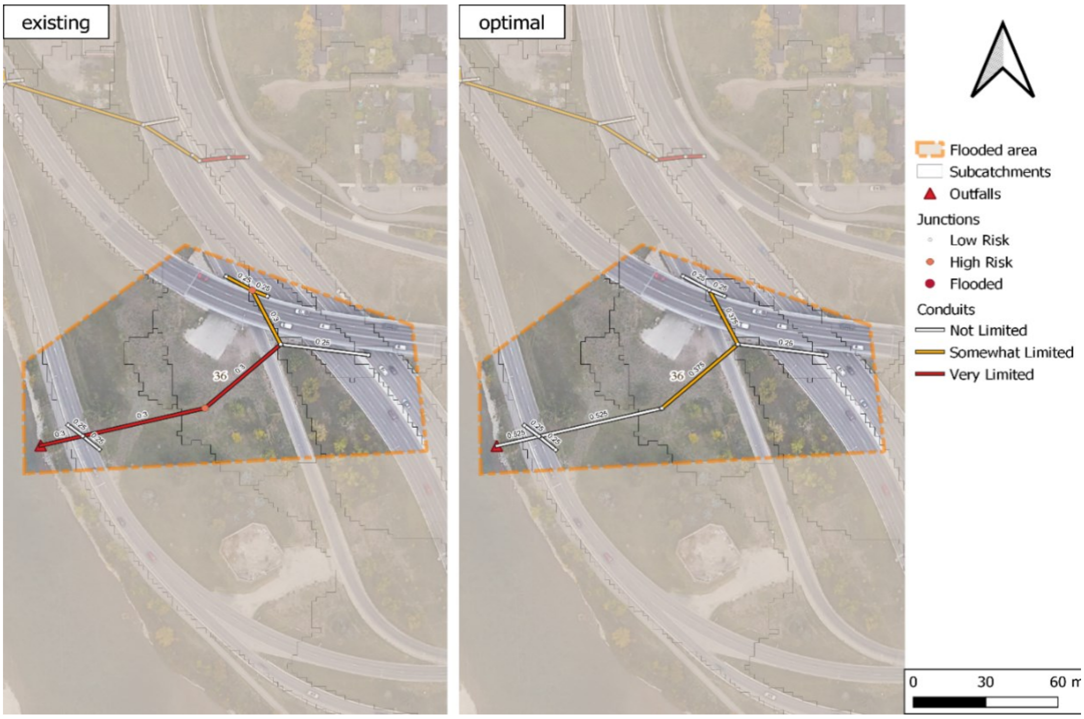


Figure A37 The comparison of Flooded Area 36 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

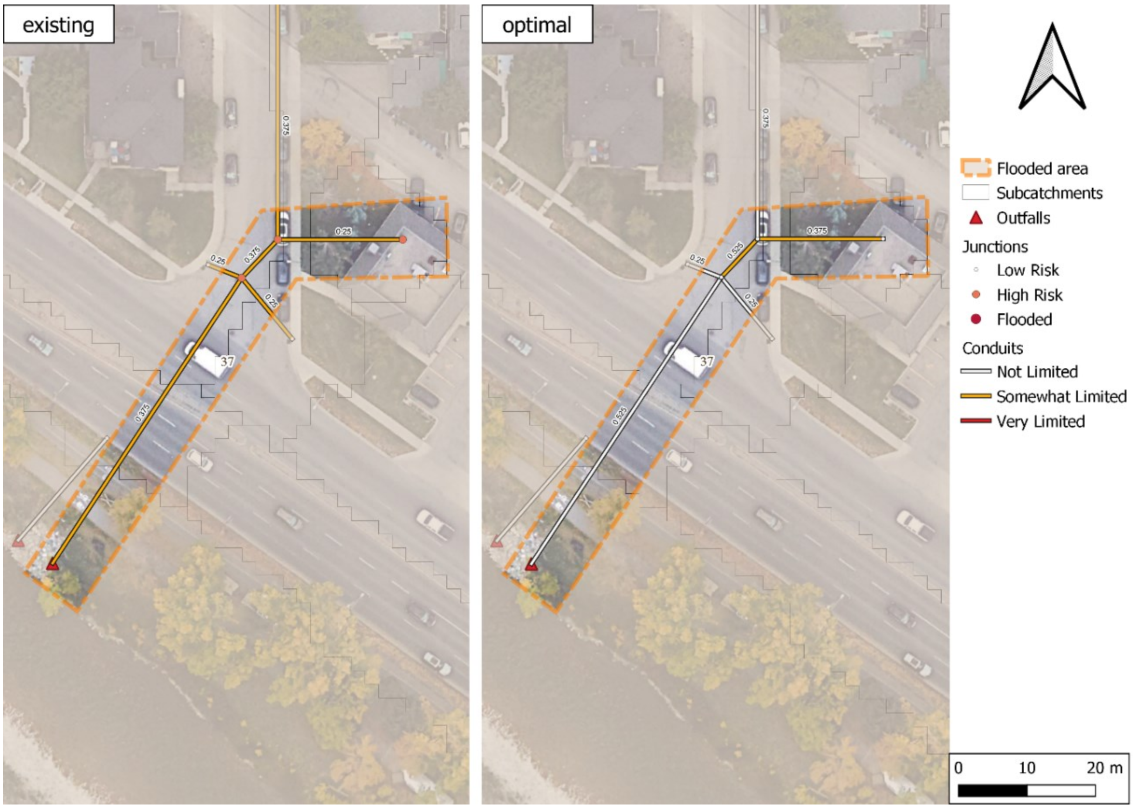


Figure A38 The comparison of Flooded Area 37 in the study area before (left) and after (right) optimization, under Scenario 1 by using the 1D model.

Table A1 Before optimization: the selected conduits of the storm sewer trunk in the test area.

Name	Length (m)	Geom1 (m)	Slope (m/m)
NC_10_POI_0011	33.6	0.75	0.00080
NC_10_POI_0015	6.96	0.75	0.00086
NC_10_POI_0016	68.04	0.9	0.00220
NC_10_POI_0020	68.95	0.9	0.00244
NC_10_POI_0023	38.1	1.35	0.00992
NC_10_POI_0024	48.08	0.9	0.00146
NC_10_POI_0025	9.14	1.35	0.08576
NC_10_POI_0027	24.38	1.05	0.00837
NC_10_POI_0028	73.96	1.05	0.00251
NC_10_POI_0032	116.42	1.05	0.00228
NC_10_POI_0064	21.93	0.45	0.00483
NC_10_POI_0065	79.02	0.375	0.02353
NC_10_POI_0067	36.91	0.45	0.00125
NC_10_POI_0069	26.6	0.45	0.00229
NC_10_POI_0072	48.8	0.45	0.00316
NC_10_POI_0074	68.26	0.525	0.00819
NC_10_POI_0084	27.05	0.75	0.00362
NC_10_POI_0089	67.34	0.75	0.00235

Table A2 After optimization: the selected conduits of the storm sewer trunk in the test area.

Name	Length (m)	Geom1 (m)	Slope (m/m)
NC_10_POI_0011	33.6	0.9	0.00380
NC_10_POI_0015	6.96	1.05	0.03272
NC_10_POI_0016	68.04	1.05	0.00100
NC_10_POI_0020	68.95	1.05	0.00272
NC_10_POI_0023	38.1	1.35	0.00992
NC_10_POI_0024	48.08	1.05	0.00100
NC_10_POI_0025	9.14	1.35	0.08576
NC_10_POI_0027	24.38	1.5	0.05398
NC_10_POI_0028	73.96	1.5	0.00100
NC_10_POI_0032	116.42	1.5	0.00100
NC_10_POI_0064	21.93	0.675	0.00110
NC_10_POI_0065	79.02	0.525	0.01664
NC_10_POI_0067	36.91	0.675	0.00324
NC_10_POI_0069	26.6	0.675	0.00110
NC_10_POI_0072	48.8	0.675	0.00411
NC_10_POI_0074	68.26	0.9	0.00100
NC_10_POI_0084	27.05	0.9	0.00310
NC_10_POI_0089	67.34	0.9	0.00100

Table A3 Before optimization: the selected conduits of the storm sewer trunk in the study area, using the 1D model.

Name	Length (m)	Diameter (m)	Slope (m/m)
NC_10_MON_0022	39.43	1.65	0.00276
NC_10_MON_0023	123.75	1.65	0.00297
NC_10_MON_0030	24.18	1.65	0.03936
NC_10_MON_0039	108.59	1.65	0.00307
NC_10_MON_0041	70.59	1.65	0.00278
NC_10_PKD_0179	11.32	0.9	0.00097
NC_10_PKD_0199	30.87	1.05	0.00198
NC_10_PKD_0200	9.5	1.05	0.00211
NC_10_PKD_0201	46.66	1.05	0.00199
NC_10_PKD_0202	53.7	1.05	0.00181
NC_10_PKD_0203	15.3	0.9	0.00229
NC_10_PKD_0204	58.81	0.9	0.00221
NC_10_PKD_0222	90.35	0.6	0.00423
NC_10_PKD_0236	63	0.6	0.00460
NC_10_PKD_0246	26.89	0.6	0.00446
NC_10_PKD_0248	40	0.375	0.01215
NC_10_PKD_0249	4	0.6	0.01225
NC_10_PKD_0255	49.86	0.9	0.00201
NC_10_PKD_0260	16.63	0.9	0.00204
NC_10_PKD_0264	30.79	0.75	0.00299

Table A4 After optimization: the selected conduits of the storm sewer trunk in the study area, using the 1D model.

Name	Length (m)	Diameter (m)	Slope (m/m)
NC_10_MON_0022	39.43	1.8	0.00768
NC_10_MON_0023	123.75	1.8	0.00381
NC_10_MON_0030	24.18	1.8	0.04219
NC_10_MON_0039	108.59	1.65	0.00307
NC_10_MON_0041	70.59	1.8	0.00222
NC_10_PKD_0179	11.32	1.95	0.00100
NC_10_PKD_0199	30.87	1.2	0.00193
NC_10_PKD_0200	9.5	1.2	0.00198
NC_10_PKD_0201	46.66	1.2	0.00276
NC_10_PKD_0202	53.7	1.2	0.00220
NC_10_PKD_0203	15.3	1.05	0.00333
NC_10_PKD_0204	58.81	1.05	0.00221
NC_10_PKD_0222	90.35	1.05	0.00423
NC_10_PKD_0236	63	1.05	0.00508
NC_10_PKD_0246	26.89	1.05	0.00484
NC_10_PKD_0248	40	0.6	0.00168
NC_10_PKD_0249	4	0.675	0.01178
NC_10_PKD_0255	49.86	1.05	0.00201
NC_10_PKD_0260	16.63	1.05	0.00204
NC_10_PKD_0264	30.79	1.05	0.00299

Table A 5 Before optimization: the selected conduits of the storm sewer trunk in the study area, using the 2D model.

Name	Length (m)	Diameter (m)	Slope (m/m)
NC_10_PKD_0201	46.66	1.05	0.00199
NC_10_PKD_0202	53.7	1.05	0.00181
NC_10_PKD_0203	15.3	0.9	0.00229
NC_10_PKD_0204	58.81	0.9	0.00221
NC_10_PKD_0255	49.86	0.9	0.00201
NC_10_PKD_0260	16.63	0.9	0.00204
NC_10_PKD_0264	30.79	0.75	0.00299
NC_10_PKD_0267	20.95	0.75	0.00420
NC_10_PKD_0271	6.85	0.75	0.00292
NC_10_PKD_0275	44.64	0.75	0.00202
NC_10_PKD_0277	34.16	0.375	0.00650
NC_10_PKD_0280	52.56	0.75	0.00200
NC_10_PKD_0282	33.38	0.45	0.00653
NC_10_PKD_0285	44.31	0.45	0.00650
NC_10_PKD_0290	28.95	0.45	0.00290
NC_10_PKD_0291	46.03	0.675	0.00202
NC_10_PKD_0294	30.44	0.675	0.00204

Table A6 After optimization: the selected conduits of the storm sewer trunk in the study area under scenario 1, using the 2D model.

Name	Length (m)	Diameter (m)	Slope (m/m)
NC_10_PKD_0201	46.66	1.2	0.00274
NC_10_PKD_0202	53.7	1.2	0.00220
NC_10_PKD_0203	15.3	1.05	0.00333
NC_10_PKD_0204	58.81	1.05	0.00221
NC_10_PKD_0255	49.86	1.05	0.00201
NC_10_PKD_0260	16.63	1.05	0.00204
NC_10_PKD_0264	30.79	1.05	0.00299
NC_10_PKD_0267	20.95	0.9	0.00420
NC_10_PKD_0271	6.85	0.9	0.00818
NC_10_PKD_0275	44.64	0.9	0.00202
NC_10_PKD_0277	34.16	0.525	0.00345
NC_10_PKD_0280	52.56	0.9	0.00150
NC_10_PKD_0282	33.38	0.6	0.00923
NC_10_PKD_0285	44.31	0.6	0.00650
NC_10_PKD_0290	28.95	0.6	0.00290
NC_10_PKD_0291	46.03	0.9	0.00202
NC_10_PKD_0294	30.44	0.9	0.00204