A Non-linear Optimization Model to Minimize Flood Risks on Urban Roadways Due to Storm-Drain System Deficiencies

James Ekeh[1] and Manoj K Jha[2]

[1]MKJHA Consulting
[2]Advancement Strategy Consulting

January 20, 2023

Abstract

Urban flooding is caused due to poor drainage design, extreme weather, and excessive rain. Such flooding severely affects the road infrastructure. While there are a number of hydrologic software (e.g., TR-55, HydroCAD, TR-20, HEC-RAS, StreamStats, L-THIA, SWMM, WMOST, MAST, HY-8) available to examine extent of urban flooding, the softwares primarily require walking through a series of manual steps and address each study area individually preventing a collective view of an urban area in an efficient manner for hydrologic analysis. Furthermore, the softwares have no ability to recommend optimal culver pipe sizes to minimize flooding. In this paper, we develop a non-linear optimization formulation to minimize urban flooding using underdrain pipe size as a decision variable. We propose a solution algorithm in an integrated GIS and Python environment. Monte Carlo Simulation is used to simulate rainfall intensity by using empirical data on extreme weather from the National Oceanic and Atmospheric Administration. An example using the storm-drain system for the Baltimore County is performed. The results show that the model is effective in identifying storm-drain deficiencies and correcting them by choosing appropriate storm-drain inlet types to minimize flooding. The proposed method eliminates the need to examine each study area manually using existing hydrologic tools. Future works may include expanding the methodology for large datasets. They may also include a more sophisticated modeling approach for estimating rainfall intensity based on extreme weather patterns.

Hosted file

essoar.10510933.1.docx available at https://authorea.com/users/578230/articles/620293-a-non-linear-optimization-model-to-minimize-flood-risks-on-urban-roadways-due-to-storm-drain-system-deficiencies

A Non-linear Optimization Model to Minimize Flood Risks on Urban Roadways Due to Storm-Drain System Deficiencies

J. Ekeh[1] and M. K. Jha[2]

[1]MKJHA Consulting.

[2]Advancement Strategy Consulting.

Corresponding author: Manoj Jha (`mj@advancementstrategy.com`)

Key Points:

- Relationship of urban flooding to storm-drain system deficiency is studied.
- A non-linear optimization model in an integrated GIS and Python environment is developed.
- A case study using the storm-drain system data from Baltimore County is performed.

Abstract

Urban flooding is caused due to poor drainage design, extreme weather, and excessive rain. Such flooding severely affects the road infrastructure. While there are a number of hydrologic software (e.g., TR-55, HydroCAD, TR-20, HEC-RAS, StreamStats, L-THIA, SWMM, WMOST, MAST, HY-8) available to examine extent of urban flooding, the softwares primarily require walking through a series of manual steps and address each study area individually preventing a collective view of an urban area in an efficient manner for hydrologic analysis. Furthermore, the softwares have no ability to recommend optimal culver pipe sizes to minimize flooding. In this paper, we develop a non-linear optimization formulation to minimize urban flooding using underdrain pipe size as a decision variable. We propose a solution algorithm in an integrated GIS and Python environment. Monte Carlo Simulation is used to simulate rainfall intensity by using empirical data on extreme weather from the National Oceanic and Atmospheric Administration. An example using the storm-drain system for the Baltimore County is performed. The results show that the model is effective in identifying storm-drain deficiencies and correcting them by choosing appropriate storm-drain inlet types to minimize flooding. The proposed method eliminates the need to examine each study area manually using existing hydrologic tools. Future works may include expanding the methodology for large datasets. They may also include a more sophisticated modeling approach for estimating rainfall intensity based on extreme weather patterns.

Plain Language Summary

Urban flooding is attributed to extended periods of rain due to extreme weather and climate change. The situation is exacerbated due to poor drainage system In this paper, we study the relationship of urban flooding to storm-drain system deficiency. A non-linear optimization model in an integrated GIS and Python

environment is developed to optimize storm-drain inlet culverts to minimize urban flooding. Empirical data on rainfall intensity from the National Oceanic and Atmospheric Administration (NOAA) is used in the optimization process. A case study using the storm-drain system data from Baltimore County is performed, which shows that the model is effective in identifying storm-drain deficiencies and correcting them by choosing appropriate storm-drain inlet types to minimize flooding. Future works may include expanding the methodology for large datasets.

1 Introduction

Urban flooding is a serious issue that is primarily caused due to storm-drain system deficiency, extreme weather, and unusual weather patterns, such as excessive rain or hurricanes. Such disasters have surged in recent years. According to the Center for Research on the Epidemiology of Disasters (CRED 2022), over the last 20 years, 7,348 disaster events were recorded. These disasters claimed more than a million lives and led to about US$ 3 trillion in economic losses worldwide.

Many cities in the world have poor storm-drain design which causes flooding due to rain. The situation exacerbates in the event of hurricanes or when the rain is more persistent and intense. A number of methods have been proposed for urban flood management, including numerical simulation and a geographic information system (Eldho et al. 2028), nature-based solution (Bremer et al. 2021), and bioretention (Jones and Jha 2009). However, none of these methods offer an analytical solution for an optimal design of storm-drain culverts based on rainfall intensity, peak discharge, and drainage area. This paper addresses the issue of storm-drain design by offering an analytical optimization solution that can serve as a guide to counties, cities, and municipalities around the world in reducing urban flooding.

2 Literature Review

Cherqui et al (2015) developed an innovative method for assessing urban potential flooding risk and identifying effective risk-reduction measures. The method was based on a spatial analysis and a causal tree. A case study from Bordeaux, France was presented to illustrate the method. Chang and Huang (2015) assessed urban flooding vulnerability with an emergy approach. A systems approach was proposed for assessing vulnerability. Xie et al. (2017) proposed an integrated evaluation framework for urban flooding mitigation. The study incorporated a 2D hydrologic simulation and life-cycle cost analysis into an integrated framework to assess potential flooding risk. Zhou, et al. (2017) performed a study to understand the trends in urban extreme rainfall to urban flooding in China. The study recommended that storm-induced urban flooding should consider both spatial disparities in climate and future changes in extreme rainfall events. Kim et al. (2017) developed a policy-oriented decision-making strategy for urban flooding under climate change. The strategy used a multi-criteria decision approach to show stakeholders' preferences for particular adaptation characteristics in the event of flooding. Flynn and Davidson (2016) discussed the issues associated with poor storm-drain design. They concluded that despite major investments in stormwater infrastructure, urban areas continued to experience urban flooding.

Many studies have proposed green infrastructure and bioretention to curb urban flooding (e.g., Hatt et al. 2004, Villarreal et al. 2004, Tzoulas et al. 2007, Jones and Jha 2009). However, there are practical limitations for actual implementation of green infrastructure and bioretention, including limited resources available to counties, cities, and municipalities.

Some studies have been reported on numerical methods for drainage culvert redesign. For example, Duan, et al. (2016) developed a multiobjective approach for the design of detention tanks in the urban stormwater drainage system. Jun, et al. (2017) developed a storm-drain based bivariate frequency analysis methos to design urban storm drains. Selbig, et al. (2016) investigated the effect of particle size distribution on the design of urban stormwater control measures. Monrabal-martinez, et al. (2019) investigated the seasonal variation in pollutant concentrations and particle size distribution in urban stormwater design. Chen, et al. (2016) developed a tool for urban rainwater management using integrated design workflow.

Majority of the literature presented above are qualitative in nature. While some of them (e.g., Duan, et al., 2016; Eldho et al., 2018) discuss analytical and mathematical approaches for urban flood management, the approaches are still manual since one must go through a series of manual steps to examine each storm-drain outlet individually. This process is very time consuming and cannot ensure a reduction in flooding in each urban segment since direction of flow of water cannot be collectively examined.

Elsevier (ScienceDirect) has put out a list of papers highlighting selected research articles under a web-page titled "Urban Flooding" (Urban Flooding 2022). While these papers do acknowledge the vulnerability of urban cities due to flooding, they fail to deal with the non-linear nature of the optimization problem that may aid in the correction of under-drain system deficiencies.

3 Hydrologic Modeling Software

There are a number of hydrologic modeling software (e.g., TR-55, HydroCAD, TR-20, HEC-RAS, StreamStats, L-THIA, SWMM, WMOST, MAST, HY-8) which are traditionally used to examine flooding risks and make recommendations to address them. The first author undertook a number of flooding analysis using these softwares and reported the results in his dissertation (Ekeh 2020). For brevity, a sample result using HEC-RAS is shown here. The study area is from Baltimore County, MD. The existing culverts and cross sections are shown in Fig. 1. The study area contains 4 cross sections and one culvert.

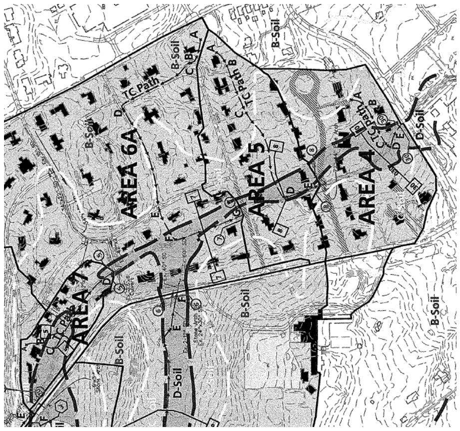

Figure 1. Sample Hydrologic Analysis in HEC-RAS

A cross-section output for 10 years is shown in Table 1. It shows relevant cross-section, culvert, and other values of the watershed, such as peak discharge rate, velocity of the head, slope, maximum channel depth, flow-area, hydrologic depth, and wetted perimeter.

Table 1. Sample HEC-RAS Analysis for a 10-year floodplain

CROSS SECTION OUTPUT Profile #10yr

		Element	Left OB	Channel	Right OB
E.G. Elev (ft)	465.64	Wt. n-Val.	0.070	0.075	0.070
Vel Head (ft)	0.02	Reach Len. (ft)	180.00	180.00	180.00
W.S. Elev (ft)	465.62	Flow Area (sq ft)	458.40	127.17	407.71
Crit W.S. (ft)		Area (sq ft)	458.40	127.17	407.71
E.G. Slope (ft/ft)	0.000552	Flow (cfs)	397.45	189.60	398.95
Q Total (cfs)	986.00	Top Width (ft)	199.88	21.12	148.21
Top Width (ft)	369.21	Avg. Vel. (ft/s)	0.87	1.49	0.98
Vel Total (ft/s)	0.99	Hydr. Depth (ft)	2.29	6.02	2.75
Max Chl Dpth (ft)	7.09	Conv. (cfs)	16919.3	8071.2	16983.3
Conv. Total (cfs)	41973.9	Wetted Per. (ft)	199.94	22.10	148.32
Length Wtd. (ft)	180.00	Shear (lb/sq ft)	0.08	0.20	0.09
Min Ch El (ft)	458.53	Stream Power (lb/ft s)	283.20	0.00	0.00
Alpha	1.13	Cum Volume (acre-ft)	3.49	2.12	4.80
Frctn Loss (ft)	0.22	Cum SA (acres)	1.84	0.27	1.57
C & E Loss (ft)	0.02				

4 Contribution to the State of Hydrologic Science

Most of available literature and hydrologic software offer solution to urban flooding either in a qualitative way or by iterating through a series of manual steps. This makes the process of identifying attributes causing flooding very inefficient. For example, there are thousands of culvert pipes buried under ground and it is not possible to analyze each of them individually unless there is an efficient procedure to analyze their combined effect collectively in understanding the likelihood of future flooding. In other words, while existing studies and hydrologic softwares are useful, they are primarily manual in nature requiring repetitive manual iterations to examine the effect of flooding due to certain rainfall intensity and watershed, cross-section, and culvert characteristics. Under this backdrop, in the paper we develop a non-linear optimization model to efficiently correct the underdrain pipe size which will minimize flooding in a particular jurisdiction, such as a county, city, or municipality.

5 Methodology

We develop a method to collectively examine the peak discharge flow from the storm-drain outlets in a given study region. The hydrologic technique most often used in urban drainage design is the rational method expressed as:

$Q = CIA$ (1)

where:

Q=peak discharge (cfs)

C=runoff coefficient

I=design storm rainfall intensity (in/hr)

A=drainage area (acres)

The quantity, I can be further formulated as a function of extreme weather as follows:

$I = I(ew)$ (2)

Using the TR-55 method, peak discharge, runoff depth, initial abstraction, unit peak discharge, and pond/swamp factor can be computed as follows:

$Q_P = Q_u \mathrm{AQ} F_p$ (3)

$Q = \frac{(P-I_a)^2}{P-I_a+s}$ (4)

$I_a = 0.2s$ (5)

$s = \frac{1000}{\mathrm{CN}} - 10$ (6)

$Q_u = f(T_C, \frac{I_a}{P},\ Rainfall\ Distribution\ Type)$ (7)

$F_p = f(\%\ Ponds\ and\ Swamps)$ (8)

where: A = total watershed area (mile^2); C_N = overall curve number for the watershed; Fp = pond and swamp adjustment factor; I_a = initial abstraction (inch) losses before runoff begins (surface depressions, interception by leaves, evaporation, infiltration); P = precipitation (inch) for 24-hr duration storm of return period for which the study is interested; Q = depth of runoff over entire watershed (inch); Q_p = peak discharge (cfs); Q_u = unit peak discharge (cfs/mile2-inch); s = potential maximum watershed water retention after runoff begins (inch); T_c = time of concentration for the watershed (hr); time for runoff to travel from the furthest distance (by time) in the watershed to the location where to be determined Q_p.

I_a can be further defined as a function of extreme weather as:

$I_a = I_a(ew)$ (9)

There are typically three distinct runoff patterns in a watershed: sheet flow, shallow concentrated flow, and channel flow. Each of the flow patterns requires a unique mathematical expression as follows:

$T_c = T_{t(sheet)} + T_{t(\text{shallow concentrated})} + T_{t(channel)}$ (10)

$Sheet\ Flow:\ T_t = \frac{0.007(nL)^{0.8}}{(P_2)^{0.5}S^{0.4}}$ (11)

$Shallow\ Concentrated\ Flow:\ T_t = \frac{L}{3600V}$ (12)

$If\ paved\ surface,\ V = 20.3282S^{0.5}; Unpaved: V = 16.1345S^{0.5}$ (13)

$Channel\ Flow:\ T_t = \frac{L}{3600\ V}; V = \frac{1.49}{n}R^{2/3}S^{0.5}\ (Manning\ Equation)$ (14)

where: L = length of flow pattern (ft) (includes all wiggles in channels); n = Manning's n value; for sheet flow, n represents the ground cover to a depth of about 1.2 inches (3 cm); for channel flow, n represents bank full conditions for an open channel or full conditions for a culvert; P_2 = 2-yr return period, 24-hr duration precipitation for the geographic region where your watershed is located (inch); R = hydraulic radius (ft) of bank full open channel or culvert flowing full (computed automatically if channel cross-section dimensions are input); S = average ground slope of each flow pattern (ft vertical/ft horizontal); T_c = time of concentration for the watershed (hr); time for runoff to travel from the furthest distance (by time) in the watershed to the location where you wish to determine Q_p' T_t = travel time for flow regime of interest (hr) - sheet, shallow concentrated, or channel flow' V = average velocity of water in each flow regime (ft/s).

The flood minimization problem can be formulated as:

$Min\ F = f(Q, T, L, S, V)$ (15)

where: F = Flooding (in cubic ft. per sec.); Q = discharge rate; T = length of time of rain (in hours),; L= Land characteristics (e.g., impervious, grassy, other soil type, etc.); V = volume of the storm-drain; S = slope.

The purpose of Eq. (15) is to illustrate conceptually the independent variables which may influence the flooding. This equation by no means presents an exact relationship between the dependent variable, F and the independent variables. It can be observed from the above equation that lower discharge rate, longer rain hours, smaller volumes of storm-drains, and larger impervious areas will tend to increase the flooding.

A basic concept of optimization problems is that the same problem can be presented either as a minimization problem or a maximization problem. Therefore, while the minimization problem can be formulated as minimization of flooding which can be represented as the difference between inflow and outflow, conversely, the problem can also be represented as a maximization problem to maximize the discharge rate.

The maximization problem can be formulated as:

$Max\ Q(d_1, d_2, \dots\dots,\ d_n)$ (16)

subject to:

$d_i \le x_i \quad \forall\ i$ (17)

where: $Q = peak\ discharge$ (cfs)

$$d_i = design\ variables\ for\ the\ stormwater\ drainge$$

$$x_i = constraints\ placed\ on\ the\ design\ variables$$

The rainfall intensity can be assumed to be as low as 10-year storm event and 25-year storm events; or as high as 50-year storm events and 100-year storm events. Models from NOAA are used to estimate appropriate rainfall intensity considering extreme weather events using a Monte Carlo Simulation.

6 Model Input Parameters

Storm-drain system deficiency means inability of the storm-drain to be effective in storm water runoff conveyance. This relates to poor design of the stormwater inlet systems and waterways, such as channels, conduits, swales and drainage paths. This is the reason why the stormwater design variables are used as input parameters to the optimization model. The model input parameters include rainfall intensity, land characteristics (e.g., gray v. green infrastructure, impervious, grassy, other soil type, etc.), and design variables for the storm-drain (including constrains placed on the design variables). For example, effective radius (R) of various sizes of box culverts can be considered as a design variable.

Using the Rational method, the peak discharge rate can be formulated as:

$Q_1 = CI(ew)A$ (18)

where:

Q_1=peak discharge (cfs)

C=runoff coefficient

$I(ew)$=design storm rainfall intensity expressed as a function of extreme weather (in/hr)

A=drainage area (acres)

The Manning Equation to calculate the outflow to a particular storm-drain inlet is expressed as:

$Q_2 = \frac{\mathrm{KA}R^{2/3}S^{1/2}}{n}$ (19)

where, A = area of the storm-drain; R = hydraulic radius; S = slope of the storm-drain inlet; K = unit conversion factor = 1.49 for English units; and n = Manning Coefficient.

From the inspection of Eqs. (18) and (19), it is clear that flooding will occur if: $Q_1 >$ Q_2; and flooding will not occur if $Q_1 \leq Q_2$. This will result in:

$\mathrm{CI}A_w \leq \frac{KA_s R^{2/3} S^{1/2}}{n}$ (20)

where $A_w = Area\ of\ the\ watershed$; $A_s = Area\ of\ the\ storm{-}drain\ inlet\ pipe$

In this research, it is assumed that storm-drain is an inlet pipe. Therefore, the area of the inlet pipe is $\frac{\pi D^2}{4}$ and hydraulic radius of the pipe is: $\frac{D}{4}$ where D is the pipe diameter in inch. Using these values, Eq. (20) will reduce to:

$\mathrm{CI}A_w \leq \frac{2K\pi D^{\frac{8}{3}} S^{\frac{1}{2}}}{12\times n} = \frac{\mathrm{K}\ D^{\frac{8}{3}} S^{\frac{1}{2}}}{6n}$ (21)

or $I \leq \frac{\mathrm{K}\ D^{8/3} S^{1/2}}{6CA_w n}$ (22)

Because in order to avoid flooding, $Q_1 \leq Q_2$, assuming everything else to be a constant for a given watershed, will lead to the following situation:

$I \leq \frac{\alpha D^{8/3}}{A_w}$ (23)

where,   is a constant, I=Rain Intensity (inch/hr.); D=inlet pipe size (or diameter); A_w=Area of the watershed.

The optimization problem can be presented as:

Min $Q_1 - Q_2$ (24)

subject to:

> $x_0 \geq 2$ (25)
>
> This means the minimum pipe size must be at least 2 inches
>
> $Q_2 \geq Q_1$ (26)
>
> This means outflow must be greater that inflow so as to minimize flooding

x_0 is the pipe diameter (in inches).

7 Rainfall Intensity

The rainfall intensity for 100 years is simulated and plotted in Python as shown in Figure 2.

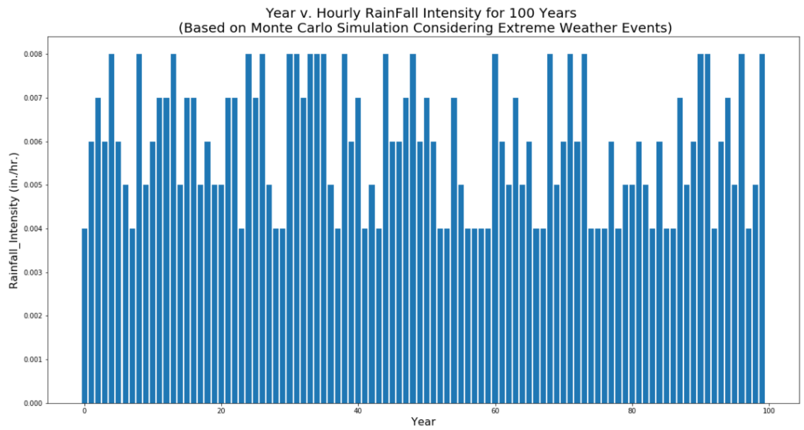

Figure 2. Plot of 100-Year Hourly Rainfall Intensity Based on Extreme Weather Events

Alternatively, another empirical formula or chart can be used to calculate the rainfall intensity which will be an input to the optimization model. For example, EPAs national stormwater calculator can be used to calculate the rainfall intensity (EPA 2020) as shown in Figs. (3)-(5) below.

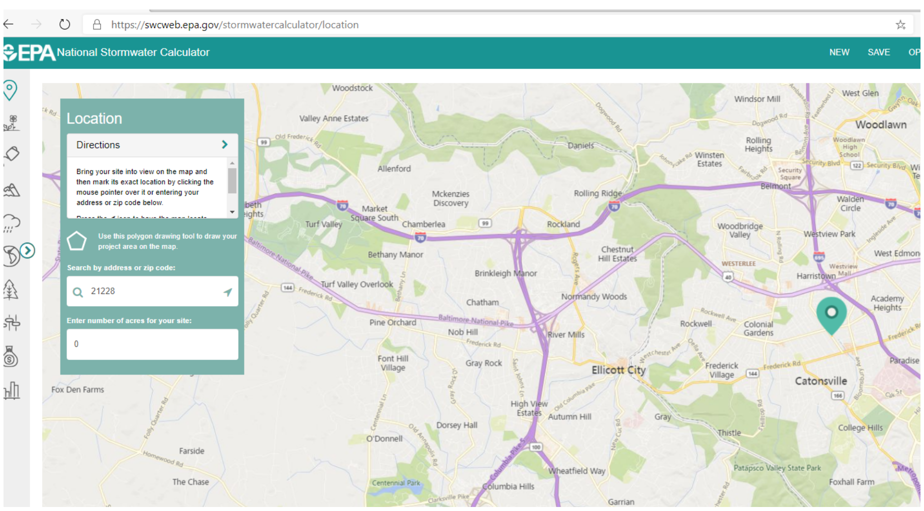

Figure 3. EPA method to calculate the rainfall intensity-screenshot 1.

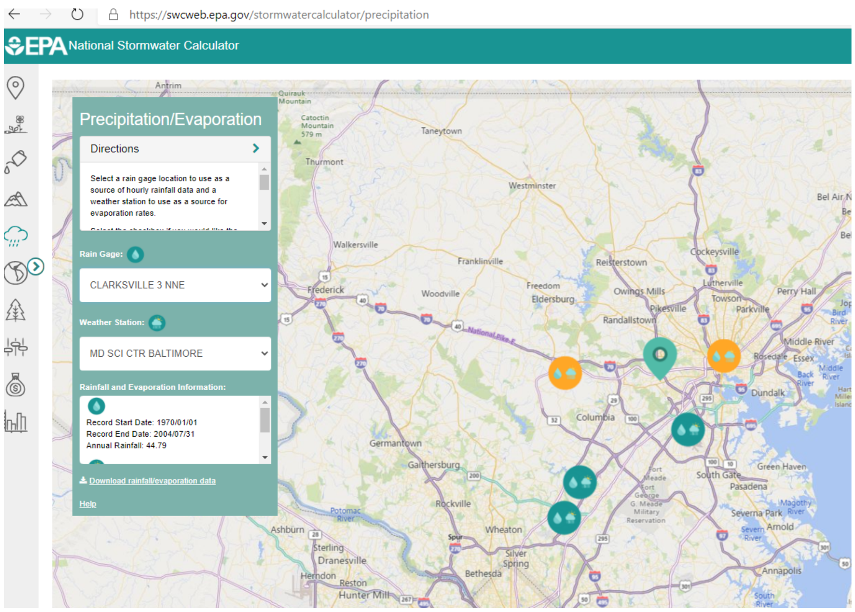

Figure 4. EPA method to calculate the rainfall intensity-screenshot 2.

Figure 5. EPA method to calculate the rainfall intensity-screenshot 3

Based on the bottom graph of Fig. 5, the maximum 24-hour rainfall for Baltimore County is 6.65 which results into an hourly rainfall intensity of 0.277.

8 Non-Linear Optimization in Python

An inspection of Eq. (23) reveals that because the rainfall intensity varies non-linearly with the pipe diameter, it is a non-linear optimization problem. In Python, the minimize function provides a common interface to unconstrained and constrained minimization algorithms for multivariate scalar functions in *scipy.optimize*. To demonstrate the minimization function, the following non-linear optimization problem is considered as an illustration:

Min $x_1 x_4 \left(x_1 + x_2 + x_3\right) + x_3$ (27)

subject to

$x_1 x_2 x_3 x_4 \geq 25$ (28)

$x_{21} + x_{22} + x_{23} + x_{24} = 40$ (29)

$1 \leq x_1, x_2, x_3, x_4 \leq 5$ (30)

$x_0 = (1, 5, 5, 1)$ (31)

This problem has a nonlinear objective that the optimizer attempts to minimize. The nonlinear nature of the objective function is obvious by the left-hand side of Eq. (27) where degree of the decision-variables is 4. The variable values at the optimal solution are subject to both equality (=40) and inequality (>25) constraints. The product of the four variables must be greater than 25 while the sum of squares of the variables must also equal 40. In addition, all variables must be between 1 and 5 and the initial guess is $x_1 = 1$, $x_2 = 5$, $x_3 = 5$, and $x_4 = 1$.

For the above illustrative nonlinear optimization problem represented, an optimization procedure is developed in Python using the nonlinear optimization solver called Sequential Least Squares Programming (SLSQP). This solver which is based on the principles of sequential least squares is a package within Python's model called *Scipy.optimize*. The screenshot below shows the optimization model set up and the solution obtained from Python.

```
[17]: import numpy as np
      from scipy.optimize import minimize
```

```
[3]: def objective(x):
         return x[0]*x[3]*(x[0]+x[1]+x[2])+x[2] # this is Eq. (4.23)

     def constraint1(x):
         return x[0]*x[1]*x[2]*x[3]-25.0 # this is Eq. (4.24)

     def constraint2(x):
         sum_eq = 40.0 # this is Eq. (4.25)
         for i in range(4):
             sum_eq = sum_eq - x[i]**2
         return sum_eq
```

```
[4]: # initial guesses
     n = 4 # This means there are four decicion variables
     x0 = np.zeros(n)
     x0[0] = 1.0
     x0[1] = 5.0
     x0[2] = 5.0
     x0[3] = 1.0
```

```
[5]: # show initial objective
     print('Initial SSE Objective: ' + str(objective(x0)))

     Initial SSE Objective: 16.0
```

Figure 6. Python Non-Linear Optimization Screenshot 1

```
[6]: # optimize
     b = (1.0,5.0) # this is Eq. (4.24)
     bnds = (b, b, b, b)
     con1 = {'type': 'ineq', 'fun': constraint1}
     con2 = {'type': 'eq', 'fun': constraint2}
     cons = ([con1,con2])
     solution = minimize(objective,x0,method='SLSQP',\
                         bounds=bnds,constraints=cons)

[7]: x = solution.x

[40]: # show final objective
      print('Final SSE Objective: ' + str(objective(x)))

      Final SSE Objective: 39.16601048855094

[41]: # print solution
      print('Solution')
      print('x1 = ' + str(x[0]))
      print('x2 = ' + str(x[1]))
      print('x3 = ' + str(x[2]))
      print('x4 = ' + str(x[3]))

      Solution
      x1 = 2.0000000000000586
      x2 = 5.291502622137112
      x3 = 2.0
      x4 = 2.000000000000063
```

Figure 7. Python Non-Linear Optimization Screenshot 2

9 Example, Results, and Discussion

We apply the proposed methodology to an underdrain pipe culvert network for Baltimore County, Maryland. A GIS map of the Baltimore County storm-drain network used for the study is shown in Fig. 8.

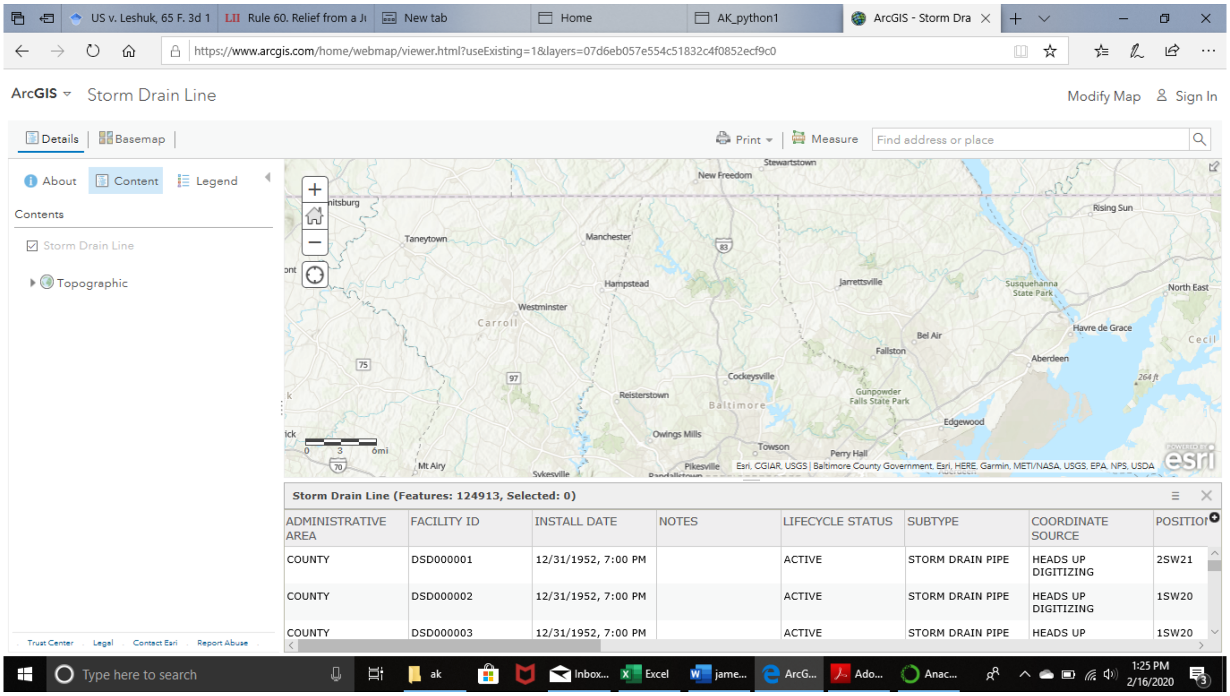

Figure 8. Storm-drain Map of Baltimore County

The map has 124,913 culvert pipes and contains information, such as watershed, pipe roughness, flowrate, velocity, geometric slope, and other relevant information to conduct the optimization. Several scatter plots are shown in Figs. 9-11 to understand the input characteristics of the inlet pipes and flow. For example, Fig. 9 shows a scatter plot of velocity v. flowrate. It shows that majority of the pipes have a velocity ranging from 0 to about 180 ft. per sec. and flowrate of about 0-1,300 cubic ft. per sec.

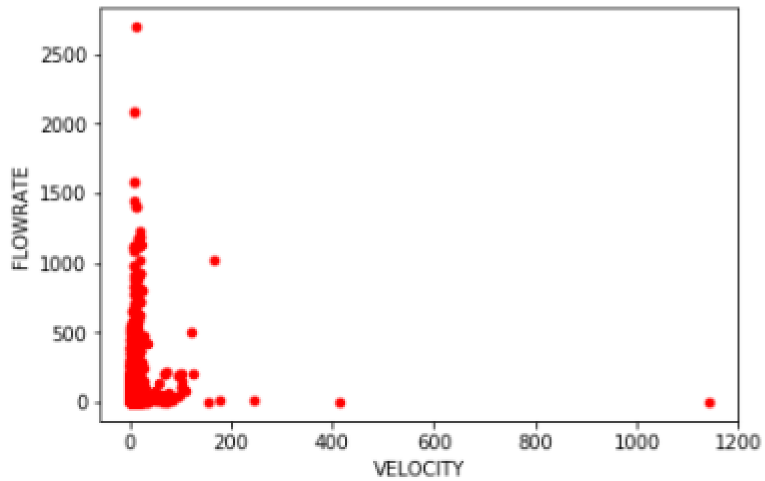

Figure 9. Scatter Plot of Velocity (ft. per Sec.) v. Flowrate (cubic ft. per sec.)

A plot of design length and flowrate is shown in Fig. 10. It shows that flowrates are higher for shorter length pipes.

```
[123]: df.plot(kind='scatter',x='DESIGNLENGTH',y='FLOWRATE',color='blue')
       plt.show()
```

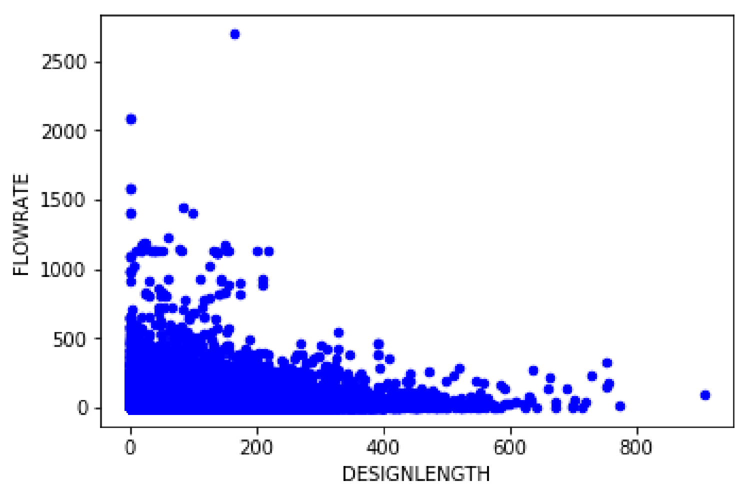

Figure 10. Scatter Plot of Design length v. Flowrate

Fig. 11 shows a scatter plot of geometric slope v. velocity. It shows that velocity is generally higher for smaller slopes.

```
[124]: df.plot(kind='scatter',x='GEOMETRICSLOPE',y='VELOCITY',color='green')
       plt.show()
```

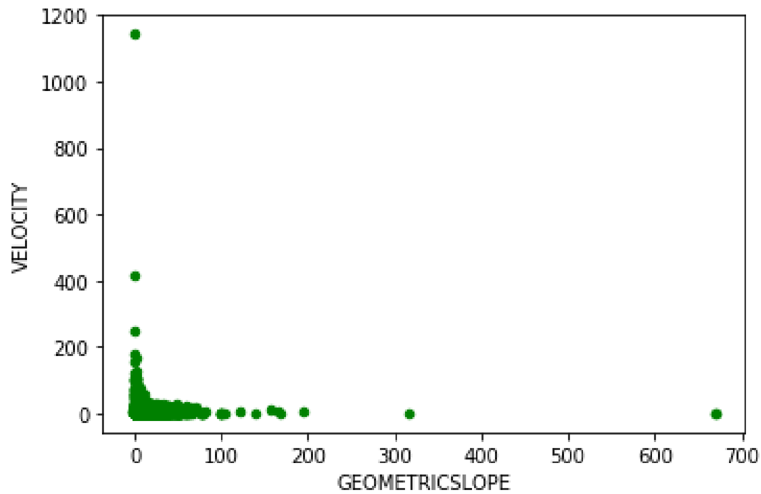

Figure 11. Scatter Plot of Geometric Slope v. Velocity

The optimization is performed in Python for a rainfall intensity of 0.277 inch per hour and additional assumed values of input parameters. Because the optimization procedure is nonlinear, an initialization step is necessary. In this step certain dummy initial variables are used based on which initial flooding rate (in cfs) is calculated. The following input values are considered for the first example: Runoff coefficient=0.84; Rainfall intensity =0.277 inches per hour; Geometric slope=2%; Roughness=0.012; Initial value of the pipe diameter=2 inches; Initial value of the watershed area=10,000 sq. ft.; allowable bounds of pipe size=[2,100]; and allowable bounds of the watershed=[10000, 100000].

Using the above values, initial flooding is obtained to be 2,326.49 cfs; optimal flooding is obtained to be 1.36e-12 cfs; and optimal pipe size is obtained to be 56.85 inches. This means, the optimization procedure obtains an optimal pipe size which will result in a zero (or less) flooding. The initialization process and results are shown in Figs. 12-13 below:

[399]:

```
# minimization problem Eqs. (4.28)-(4.30)
# FIRST SCENERIO, i_r=0.277
# SECOND SCENERIO, i_r=0.05
#x[0]=pipe diameter in inches, this is the first decision variable
#x[1]=Watershed area in sq. ft.
c=0.84 # runoff coefficient, can be changed as desired
i_r=0.277 # First Scenerio
#i_r=0.05 # Second scenerio
s_0=2 # slope
n_r=0.012 # roughness, can be changed as desired
def objective(x):
    a_2=((x[0]**2)*0.25*(22/7))/144 # This is the area of the pipe in ft.^2
    r_h=x[0]/48 # Hydraulic radius, in ft.
    q_2=(a_2*(r_h**(2/3))*(s_0**0.5))/n_r # this is Manning's Equation, flow rate is in cfs
    q_1=c*i_r*x[1] # This is the rational method, flow rate is in cfs
    return (q_1-q_2)

def constraint1(x):
    return x[0]-2.0

def constraint2(x):
    a_2=((x[0]**2)*0.25*(22/7))/144 # This is the area of the pipe in ft.^2
    r_h=x[0]/48 # Hydraulic radius, in ft.
    q_2=(a_2*(r_h**(2/3))*(s_0**0.5))/n_r # this is Manning's Equation, flow rate is in cfs
    q_1=c*i_r*x[1] # This is the rational method, flow rate is in cfs
    return (q_2-q_1)-0
```

[400]:

```
# initial guesses
n = 2 # this means there are two decision variables
x0 = np.zeros(n)
x0[0] = 2 # This is the initial value of the pipe diameter in inches
x0[1]=10000 # This is the initial value of the watershed area in sq. ft.
```

[401]:

```
# show initial objective
print('Initial Flooding rate in cfs: ' + str(objective(x0)))
```

```
Initial Flooding rate in cfs: 2326.490860092466
```

Figure 12. First screenshot of the python optimization code under the first scenario

```
[402]: # optimize
b1 = (2,100) # this is the allowable range of pipe sizes which can be changed
b2=(10000,100000) # this is the allowable range of watershed area in sq. ft. which can be changed.
bnds = (b1,b2)
con1 = {'type': 'ineq', 'fun': constraint1}
con2 = {'type': 'eq', 'fun': constraint2}
cons = ([con1,con2])
solution = minimize(objective,x0,method='SLSQP',\
                    bounds=bnds,constraints=cons)

[403]: x = solution.x

[404]: # show final objective
print('Minimum Flooding: '  + str(objective(x)))

Minimum Flooding: 1.3642420526593924e-12

[405]: # print solution
print('Solution')
print('Optimial Pipe Size = ' + str(x[0]))
print('Optimal Watershed Area = ' + str(x[1]))

Solution
Optimial Pipe Size = 56.85340635515565
Optimal Watershed Area = 10000.000083007115
```

Figure 13. Second screenshot of the python optimization code under the first scenario

Another example is performed using the following input values: Runoff coefficient = 0.84; Rainfall intensity = 0.05 inches per hour; Geometric slope=2%; Roughness=0.012; Initial value of the pipe diameter = 2 inches; Initial value of the watershed area=10,000 sq. ft.; allowable bounds of pipe size= [2,100]; and allowable bounds of the watershed=10000, 100000].

Using the above values, initial flooding is obtained to be 419.69 cfs; optimal flooding is obtained to be -1e-08 cfs; and optimal pipe size is obtained to be 29.92 inches. The initialization process and results are shown in Fig. 14-15. A comparison of both set of results shows that an underestimated value of rainfall intensity may result in a reduced optimal pipe size. Therefore, it makes sense to use a realistic value of the rainfall intensity using extreme weather.

[290]:

```python
# minimization problem Eqs. (4.28)-(4.30)
# THIRD SCENERIO, Initial Flooding Calculation for a range of slope from the Baltimore County Database
#x[0]=pipe diameter in inches, this is the first decision variable
#x[1]=Watershed area in sq. ft.
c=0.84 # runoff coefficient, can be changed as desired
i_r=0.277 # from the EPA Chart (6.65/24=0.277)
s_0=pd.DataFrame(df['GEOMETRICSLOPE'])
n_r=0.012 # roughness, can be changed as desired
def objective(x):
    a_2=((x[0]**2)*0.25*(22/7))/144 # This is the area of the pipe in ft.^2
    r_h=x[0]/48 # Hydraulic radius, in ft.
    q_2=(a_2*(r_h**(2/3))*(s_0**0.5))/n_r # this is Manning's Equation, flow rate is in cfs
    q_1=c*i_r*x[1] # This is the rational method, flow rate is in cfs
    return (q_1-q_2)

def constraint1(x):
    return x[0]-2.0

def constraint2(x):
    a_2=((x[0]**2)*0.25*(22/7))/144 # This is the area of the pipe in ft.^2
    r_h=x[0]/48 # Hydraulic radius, in ft.
    q_2=(a_2*(r_h**(2/3))*(s_0**0.5))/n_r # this is Manning's Equation, flow rate is in cfs
    q_1=c*i_r*x[1] # This is the rational method, flow rate is in cfs
    return (q_2-q_1)-0
```

[291]:

```python
# initial guesses
n = 2 # this means there is one decicion variable
x0 = np.zeros(n)
x0[0] = 2 # This is the initial value of the pipe diameter in inches
x0[1]=10000 # This is the initial value of the watershed area
```

Figure 14. Screenshot of the initialization of the optimization process

```
[415]: # show initial objective
       print('Initial Flooding rate in cfs: ' + str(objective(x0)))
```

```
Initial Flooding rate in cfs:         GEOMETRICSLOPE
0         2326.490860
1         2326.589195
2         2326.408966
3         2326.391046
4         2326.385245
5         2326.221652
6         2326.269035
7         2326.800000
8         2326.560541
9         2326.560541
10        2326.311207
11        2326.569696
12        2326.552688
13        2326.586940
14        2326.490860
15        2326.399309
16        2326.108742
17        2326.391046
18        2326.590331
19        2326.590331
20        2326.335775
21        2326.115690
22        2326.800000
23        2326.800000
24        2326.604483
25        2326.490860
26        2326.630677
27        2326.464188
28        2326.800000
29        2326.461354
...               ...
124883    2325.991790
124884    2326.800000
124885    2326.033358
124886    2326.541355
124887    2326.800000
124888    2326.581405
```

Figure 15. Initial flooding for the Baltimore county database for different slopes

A plot of geometric slope v. initial flooding is shown in the Figure below.

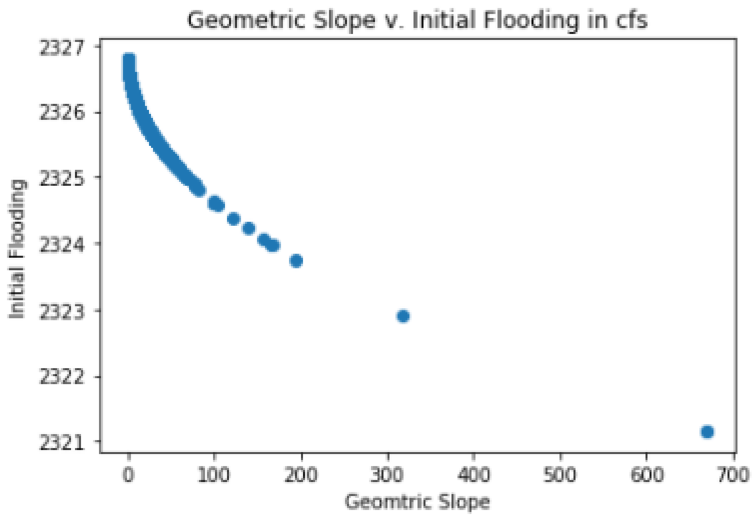

Figure 16. Plot of geometric slope v. initial flooding

It can be seen that as the slope increases initial flooding decreases. It means that it may not be necessary to optimize inlet pipe size for locations for higher slopes since the risk of flooding is less.

Using the optimization algorithm, a revised pipe size (rounded off to the nearest integer) is obtained. As an example, number of flooding issues in Jones Falls watershed is reviewed based on the optimization procedure developed in the paper, and an adjusted pipe size is recommended. The result is shown in the Table 2.

Table 2. Recommended Pipe Size for Jones Falls Watershed

Pipe Number	**Street Name**	**Number of Flooding**	**Discharge rate (cfs)**	**Underground cu**
1	Ridge Terrace	9	49	-
2	Crossland Road	18	64	18” CMP
3	Midfield Road	15	245	42” X 27” Elliptica
4	Southvale Road	12	45	None
5	Seven Mile Lane	28	158	36” RCCP
6	Fairway Road	25	27	42” X 27” Elliptic
7	Overbrook Road	20	92	15”D Conc. Circul
8	Barton Oaks Road	19	50	-
9	Traymore Road	3	24	-
10	Lorry Lane	10	28	21”D Conc. Circul
11	Lee court	6	80	I8” D Conc. Circu
12	Greenvale Road	12	63	15”D Conc. Circul
13	Slade Avenue	27	251	24”D Conc. Circul

Pipe Number	Street Name	Number of Flooding	Discharge rate (cfs)	Underground cu
14	Marnat Road	17	196	36”D Conc. Circul
15	Carla Road	9	94	18”D Conc. Circul

It can be seen that the recommended pipe size to minimize flooding is higher than the existing pipe size.

10 Conclusions and Future Works

The paper formulated a non-linear mathematical optimization problem to address the flooding on urban roadways due to deficient culvert pipe geometry. The study was carried out to investigate the optimal culvert capacity associated with the hydraulic analysis that could justify the sizing of the pipe culvert to minimize flooding. The GIS and Python-based non-linear optimization was performed using storm-drain data from Baltimore County, Maryland. Based on the sample results, it is shown that the GIS and Python-based non-linear optimization model within a Python environment is an effective tool in identifying storm-drain deficiencies and correct them by choosing appropriate storm-drain inlet types to minimize flooding. Moreover, the developed procedure eliminates the need to perform manual analysis using existing hydrologic software, although those softwares can be used in subsequent stages for micro-level analysis.

The results showed that deficient and poor design of culvert pipes are key contributors to urban flooding. For the Baltimore County case study, flooding was attributed to the difference between in-flow and out-flow. Whenever there was excess in-flow, a flooding scenario was observed. This scenario was removed by performing optimization and selecting optimal pipe sizes.

Future works may include expanding the non-linear optimization methodology on larger datasets with complex hydrological features as well as modeling the spatial disparity when considering the effects of extreme weather/climate change. Additional sensitivity analysis for a range of input values, such as roughness, geometric slope, and watershed area can also be undertaken in future works.

Acknowledgments

This paper is based on the first author’s doctoral work and not part of any external funding. The authors confirm contribution to the paper as follows: J. Ekeh and M. Jha: Study conception, design, and development of the mathematical formulation; data collection: J. Ekeh; development of the computer code: J. Ekeh and M. Jha; draft manuscript preparation: M. Jha and J. Ekeh. All authors reviewed the results and approved final version of the manuscript.

Open Research

The data used for the research is downloaded as an ArcGIS map from ArcGIS online: https://www.arcgis.com/index.html

References

Bremer, L.L., B. Keeler, P. Pascua, R. Walker, and E. Sterling. 2021. Nature-based solutions, sustainable development, and equity, In Nature-based Solutions and water Security: An Action Agenda for the 21st Century, J. Cassin, J. H. Matthews, and E. L. Gunn (eds.), 81-105, Elsevier, ISBN 9780128198711, https://doi.org/10.1016/B978-0-12-819871-1.00016-6.

Chang, L.-F. and S.-L. Huang. 2015. "Assessing Urban Flooding Yulnerability with an Emergy Approach," *Landscape and Urban Planning*, 143(Nov. 2015), 11-24.

Chen, Y., H.W. Samuelson, and Z. Tong. 2016. Integrated Design Workflow and a New Tool for Urban Rainwater Management, Journal of Environmental Management, 180(15), 45-51.

Cherqui, F., A. Belmeziti, D. Granger, A. Sourdril., and P. L. Gauffre. 2015. "Assessing Urban Potential Flooding Risk and Identifying Effective Risk-Reduction Measures", *Science of the Total Environment*, 514(1), 418-425.

CRED. 2022. Human cost of disasters: An overview of the last 20 years, 2000-2019, Centre of Research on the Epidemiology of Disasters, UN Office of Disaster Risk Reduction.

Duan, H.-F., F. Li, and H. Yan. 2016. Multi-objective Optimal Design of Detention Tanks in the Urban Stormwater Drainage System: Uncertainty and Sensitivity Analysis, Water Resources Management. 30 (Oct. 2016), 13, 4635-4648.

Ekeh, J. 2020. *A GIS-based optimization model to minimize urban flood risk due to storm drain system deficiencies*, Doctoral Dissertation, Morgan State University, Baltimore, MD.

Eldho, T.I., P.E. Zope, and A.T. Kulkarni. 2018. Urban Flood Management in Coastal Regions Using Numerical Simulation and Geographic Information System, In Integrated Disaster Science and Management: Global Case Studies in Mitigation and Recovery, P. Samui, D. Kim, and C. Ghosh (eds.) 205-219, Elsevier.

EPA. 2020. https://swcweb.epa.gov/stormwatercalculator (last accessed 10/28/2020).

Flynn, C.D. and C.I. Davidson. 2016. "Adapting the social-ecological system framework for urban stormwater management: the case of green infrastructure adoption," *Ecology and Society*, 21(4):19. http://dx.doi.org/10.5751/ES-08756-210419.

Hatt, B. E., T. D. Fletcher, C. J. Walsh, and S. L. Taylor. 2004. "The influence of urban density and drainage infrastructure on the concentrations and loads of pollutants in small streams." *Environmental Management*, 34:112–124. http://dx.doi.org/10.1007/s00267-004-0221-8

Jones, D. and M.K. Jha. 2009. Green Infrastructure: Assessing the Benefits of Bioretention over Traditional Stormwater Management, in *Environmental*

Sciences and Sustainability, M. Jha, C. Long, N. Mastorakis, C. Bulucea (eds.), WSEAS Press, www.wseas.org, ISBN: 978-960-474-136-6, pp. 134-141.

Jun, C., X. Qin, T.-Y. Gan, Y.-K. Tung, and C. D. Michele. 2017. Bivariate Frequency Analysis of Rainfall Intentisity and Duration for Urban Stormwater Infrastructure Design, Journal of Hydrology, 553 (Oct. 2017), 374-383.

Kim, Y., D. A. Eisenberg, E.N. Bondank, M. V. Chester, G. Mascaro, and B. S. Underwood. 2017. "Fail-Safe and Safe-to-fail Adaptation: Decision-Making for Urban Flooding under Climate Change," *Climate Change*, 145 (3-4), 397-412.

Monrabal-martinez, C., T. M. Muthanna, and T. Meyn. 2019. Seasonal variation in pollutant concentrations and particle size distribution in urban stormwater-design implications for BMPs. http://documents.irevues.inist.fr/bitstream/handle/2042/60419/1C 119MON.pdf (last accessed on 2/20/2019).

Selbig, W.R., M.N. Fienen, M.N., J.A., Horwatich, and R.T. Bannerman. 2016. The Effect of Particle Size Distribution on the Design of Urban Stormwater Control Measures, water, 8(1), 17; https://doi.org/10.3390/w8010017.

Tzoulas, K., K. Korpela, S. Venn, V. Yli-Pelkonen, A. Kaźmierczak, J. Niemela, and P. James. 2007. "Promoting ecosystem and human health in urban areas using green infrastructure: a literature review." *Landscape and Urban Planning*, 81(3):167–178. http://dx.doi.org/10.1016/j.landurbplan.2007.02.001.

Urban Flooding. 2022. Urban Flooding, ScienceDirect (Elsevier). https://www.sciencedirect.com/topics/earth-and-planetary-sciences/urban-flooding (accessed: 3/17/2022).

Villarreal, E. L., A. Semadeni-Davies, and L. Bengtsson. 2004. "Inner city stormwater control using a combination of best management practices." *Ecological Engineering*, 22:279–298. http://dx.doi.org/10.1016/j.ecoleng.2004.06.007.

Xie, J., H. Chen, Z. Liao, X. Gu, D. Zhu, and J. Zhang. 2017. "An Integrated Assessment of Urban flooding Mitigation Strategies for Robust Decision Making," *Environmental Modelling & Software*, 95 (Sept. 2017), 143-155.

Zhou, X., Z. Bai, and Y. Yang. 2017. "Linking Trends in Urban Extreme Rainfall to Urban Flooding in China," *Internal Journal of Climatology*, 37(13), 4586-4593.