Optimal Flood Mitigation over Flood Propagation Approximations

Byron Tasseff[1], Russell Bent[1], and Pascal Van Hentenryck[2]

[1] Los Alamos National Laboratory, Los Alamos NM 87545, USA,
btasseff@lanl.gov, rbent@lanl.gov
[2] University of Michigan, Ann Arbor MI 48109, USA,
pvanhent@umich.edu

Abstract. Globally, flooding is the most frequent among all natural disasters, commonly resulting in damage to infrastructure, economic catastrophe, and loss of life. Since the flow of water is influenced by the shape and height of topography, an effective mechanism for preventing and directing floods is to use structures that increase height, e.g., levees and sandbags. In this paper, we introduce the Optimal Flood Mitigation Problem (OFMP), which optimizes the positioning of barriers to protect critical assets with respect to a flood scenario. In its most accurate form, the OFMP is a challenging optimization problem that combines nonlinear partial differential equations with discrete barrier choices. The OFMP requires solutions that combine approaches from computational simulation and optimization. Herein, we derive linear approximations to the shallow water equations and embed them in the OFMP. Preliminary results demonstrate their effectiveness.

Keywords: flood mitigation, nonlinear programming, mixed integer programming, approximations

1 Introduction

Throughout human history, water-related natural disasters, e.g., the Johnstown Flood of 1889, the Great Mississippi Flood of 1927, and Hurricane Katrina in 2005, have caused immense human suffering and economic consequences. While the causes of such disasters (hurricanes, dam failures, excessive rainfall, etc.) vary, all are characterized by flooding, i.e., the flow of water into undesired areas. As a result, societies and governments have invested considerable resources into controlling and preventing the occurrence of floods. Despite these efforts, risks remain, and floods continue to be a subject of intense scrutiny [10, 9, 5].

One of the most influential factors in flooding is the shape of the ground surface (topography). As an example, under the influence of gravity, water naturally flows downhill and around areas of higher topographic elevation. Topography can be adjusted through construction of permanent structures, such as levees and berms, or temporary structures, such as sandbags. This paper introduces the Optimal Flood Mitigation Problem (OFMP), an optimization problem that

aims to mitigate flooding by adjusting topographic elevation. Its goal is to select the position of barriers, e.g., sandbags or levees, to protect critical assets and/or enable the evacuation of threatened populations.

The OFMP is an inherently difficult optimization problem. Since these barriers divert flow, it is critical to accurately model the flood's propagation, traditionally captured by the two-dimensional (2D) shallow water equations. These nonlinear partial differential equations (PDEs) express flow conversation and momentum along two horizontal dimensions at every point in space and time. In practice, these equations are discretized over space and time, resulting in a set of nonlinear equations of high dimensionality. In addition, the OFMP aims to choose the position of barriers in space, introducing additional sources of nonconvexity and combinatorial explosion. However, unlike many control-related optimization problems, the OFMP optimizes only the initial conditions. Flood propagation is predetermined once initial conditions have been selected; there are limited opportunities to modify the flood behavior once the topography is adjusted. This observation provides the key intuition for our contribution: *the development of a principled approach for approximating the response of a flood to changes in topography that is tractable for current optimization technology.*

The main contributions of this paper can be summarized as follows:

- The formalization of the OFMP problem integrating simulation and optimization in the flood domain;
- The derivation of *linear* lower and upper space-time approximations to the PDEs describing flood propagation;
- The definition of optimization models for the OFMP based on these approximations;
- Preliminary empirical results that highlight the accuracy and tractability of the approximations and demonstrate the potential of optimization technology in this area.

The derivation of linear approximations to flood propagation is a critical step in bringing the OFMP into the realm of optimization technology. Our results show that these approximations can provide reasonable estimates of flood extent and water depth using the historical Taum Sauk dam failure as an example. The empirical results also demonstrate the potential of optimization technology on some preliminary case studies.

It is important to emphasize that the literature associated with optimizing the locations of barriers for flood mitigation is limited. To the best of our knowledge, the closest related work is [6]. They propose an interdiction model for flood mitigation and develop flood surrogates from simulation to serve as proxies for calculating flood response to mitigation efforts. However, these surrogates do not define strict relationships with the original PDEs. There are a number of papers focused on simulation-optimization approaches for flood mitigation, where the PDEs are treated as a black box. These papers are focused on controlling the release of water to prevent floods and do not attempt to exploit the structure of the PDEs themselves. Reference [2] is a recent example of this type of approach and contains an extensive literature review of these methods. The work in [4]

has considered the PDEs, but their focus is on optimizing normal operations of an open-channel system. Hence their model only requires one spatial dimension, whereas the flooding application considered here has an inherent second spatial dimension and is thus significantly more difficult. Finally, the problem of optimizing dike heights with uncertain flood possibilities was considered in [1]. The PDEs for flood propagation are not considered, and probability models for maximum flood depths are used instead.

The rest of this paper is organized as follows: Section 2 discusses the background of flood modeling. Section 3 presents the linear flood relaxations. Section 4 introduces the OFMP and proposes a preliminary optimization model exploiting the linear flood relaxations. Section 5 describes empirical results, and Section 6 concludes the paper.

2 Background

The Two-dimensional Shallow Water Equations The 2D shallow water equations are a system of hyperbolic PDEs increasingly used to accurately model flooding phenomena. With recent advances in high-performance computing, numerical solutions to these equations have become tractable for large-scale simulation problems. They are especially useful in the context of urban flooding, where one-dimensional models fail due to increased topographic complexity. With bottom slope, bottom friction, and volumetric source terms, the 2D shallow water equations may be defined as

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial (hu)}{\partial x} + \dfrac{\partial (hv)}{\partial y} = R(x,y,t), \\ \dfrac{\partial (hu)}{\partial t} + \dfrac{\partial}{\partial x}\left(hu^2 + \dfrac{1}{2}gh^2\right) + \dfrac{\partial (huv)}{\partial y} = -gh\dfrac{\partial B}{\partial x} - \dfrac{\tau^x}{\rho}, \\ \dfrac{\partial (hv)}{\partial t} + \dfrac{\partial (huv)}{\partial x} + \dfrac{\partial}{\partial y}\left(hv^2 + \dfrac{1}{2}gh^2\right) = -gh\dfrac{\partial B}{\partial y} - \dfrac{\tau^y}{\rho}, \end{cases} \tag{1}$$

where h is the water depth, u and v are horizontal velocities, B is the bottom topography (or bathymetry), g is the acceleration due to gravity, τ^x and τ^y are horizontal components of the bottom friction, ρ is the water density, and R is a volumetric source term [3]. Although these equations represent the state of the art in flood modeling, even when discretized, they remain nonlinear and nonconvex, making them difficult to optimize over. It is thus beneficial to consider more tractable approximations.

A Hydrostatic Approximation To obtain a more tractable approximation of flood propagation, we instead consider a simplified fluid model similar to that described by Mei, Decaudin, and Hu [8]. In this model, each cell (i, j) exchanges water content with adjacent cells using a set of virtual "pipes." For each time step, the model associates various information with each cell and pipe. In particular, h_{ijt} represents the depth of the water in cell (i, j) at time index t, w_{ijt}

represents the water surface elevation, and B_{ij} represents the topographic elevation. Each cell (i, j) also has four connected pipes, one for each of its four neighboring cells, denoted by W(est), E(ast), N(orth), and S(outh). Each pipe is associated with an outgoing volumetric flux, f^k_{ijt}, which represents the flow of water from cell (i, j) to its neighbor in position k ($k \in W, E, N, S$) at time index t. For instance, f^W_{ijt} represents the outgoing flux from (i, j) to its "western" (left) neighbor (i.e., cell $(i - 1, j)$) at time index t.

In the model, the flux of a pipe is accelerated by the hydrostatic pressure difference between adjacent cells. The water volume V of a cell is integrated using the accumulated flux from all connected pipes. This corresponds to a change in the cell's depth and water surface elevation. These concepts are illustrated visually in Figures 1 and 2.

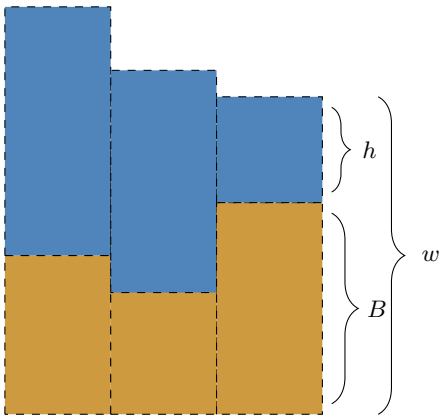


Fig. 1: The pipe flow model is discretized into columnar components, with h denoting the water depth, B the topographic elevation, and w the water surface.

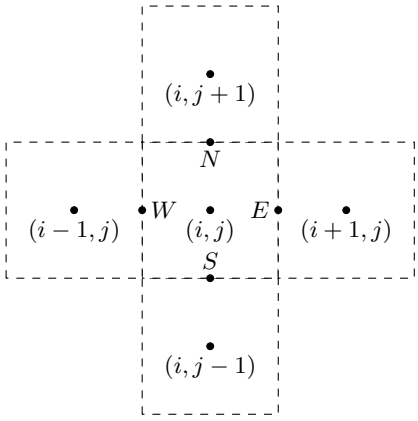


Fig. 2: Two-dimensional representation of the pipe flow discretization containing cell (i, j), adjacent cells, and the four interfaces of (i, j).

For each cell, we first define the *estimated* flux vector $\tilde{\mathbf{f}}_{ijt} = (\tilde{f}^W_{ijt}, \tilde{f}^E_{ijt}, \tilde{f}^N_{ijt}, \tilde{f}^S_{ijt})$ using the hydrostatically derived relation

$$\tilde{f}^k_{ijt} = \max\left(0, f^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}\Delta w^k_{ij,t-1}\right), \tag{2}$$

where A is the cross-sectional area of the pipe, g is the acceleration due to gravity, Δs is the length of the virtual pipe (typically the grid cell spacing, e.g., Δx or Δy), Δt is the simulation time step, and Δw^k_{ijt} is the difference in water surface elevation between cell (i, j) and its k-neighbor at time index t, i.e.,

$$\Delta w^k_{ijt} = (B_{ij} + h_{ijt}) - (B^k_{ij} + h^k_{ijt}). \tag{3}$$

In this approximation, the estimated outgoing flux from a cell may exceed the available water content within that cell. This is obviously not desirable from

a volume conservation standpoint. More importantly, if left uncorrected, this can lead to negative water depths and numerical instabilities. This may be resolved by scaling the outgoing flux with respect to the water content available in the cell. A scaling factor, K_{ijt}, for the outgoing flux vector may be defined as

$$K_{ijt} = \min\left(1, \frac{h_{ij,t-1}\Delta x \Delta y}{\left(\tilde{f}^W_{ijt} + \tilde{f}^E_{ijt} + \tilde{f}^N_{ijt} + \tilde{f}^S_{ijt}\right)\Delta t}\right). \tag{4}$$

The estimated outgoing flux vector $\tilde{\mathbf{f}}_{ijt}$ is then scaled by K_{ijt} to produce the actual outgoing flux vector $\mathbf{f}_{ijt}$, i.e.,

$$\mathbf{f}_{ijt} = K_{ijt}\tilde{\mathbf{f}}_{ijt}. \tag{5}$$

The change in water volume may then be computed using the accumulation of incoming flux, $\mathbf{f}^{in}$, and subtraction of outgoing flux, $\mathbf{f}^{out}$. For cell (i, j), the volumetric change in water is thus

$$\begin{aligned}\Delta V_{ijt} &= \left(\sum f^{in}_{ijt} - \sum f^{out}_{ijt}\right)\Delta t \\ &= \left(f^E_{i-1,j,t} + f^W_{i+1,j,t} + f^N_{i,j-1,t} + f^S_{i,j+1,t} - \sum f^k_{ijt}\right)\Delta t.\end{aligned} \tag{6}$$

Finally, the water depth in each cell may be integrated:

$$h_{ijt} = h_{ij,t-1} + \frac{\Delta V_{ijt}}{\Delta x \Delta y}. \tag{7}$$

For completeness, we also suggest the naive reflective boundary conditions

$$h_{ijt} = 0, \mathbf{f}_{ijt} = 0, \tilde{\mathbf{f}}_{ijt} = 0 \tag{8}$$

along the four boundaries of the domain.

3 Linear Approximations of the Pipe Flow Model

The pipe flow model described includes nonlinear terms, even when A, B, g, Δt, Δx, and Δy are treated as constants. Fortunately, these terms are only used for corrective measures, i.e., in Equation (4). We now present two approximations to remove them. For convenience, we call them the lower and upper approximations because they underestimate and overestimate the water being sent from a cell to its neighbors instead of applying the scaling factor K.

Lower Approximation The lower approximation is based on the following idea: if the estimated outgoing flux from a cell exceeds the available water content within that cell, the outgoing flux is approximated as zero, i.e., when

$$h_{ij,t-1}\Delta x \Delta y < \left(\tilde{f}^W_{ijt} + \tilde{f}^E_{ijt} + \tilde{f}^N_{ijt} + \tilde{f}^S_{ijt}\right)\Delta t, \tag{9}$$

$\mathbf{f}_{ijt}$ is approximated as zero. This bypasses the need for Equations (4) and (5). Intuitively, this means that, "when there is not enough water to be transferred," the water is held back within the cell.

Upper Approximation The upper approximation implements another intuitive idea: if the estimated outgoing flux from a cell exceeds the available water content within that cell, the model assumes there is enough water, and no scaling occurs. This again bypasses the need for Equations (4) and (5).

However, it is important to note that, in the case of positive fluxes calculated as a result of differing dry topographies (and thus differing water surface elevations), Equations (4) and (5) provide an additional correction beside scaling. When the available water content within a cell is equal to zero, K_{ijt} is always forced to zero, and the resultant fluxes $\mathbf{f}_{ijt}$ are thus calculated as zero. Although a similar correction is achieved automatically by the lower approximation, it is necessary to impose the constraint $\mathbf{f}_{ijt} = 0$ when $h_{ij,t-1} = 0$ in the upper approximation.

4 The Optimal Flood Mitigation Problem

This section describes two optimization models based on the lower and upper approximations, respectively. Both optimization models aim to protect a set $\mathcal{A}$ of assets by minimizing maximum water depths at asset locations over time. To protect the assets, one or more barriers (e.g., sandbags or levees) can be placed on a cell to increase its elevation; a fixed number of barriers, n, are available for that purpose. The models are similar, differing only in the approximations used. We present them both to give a global view of the lower and upper approximations. Boundary conditions are omitted in the optimization models for simplicity.

Lower Approximation Optimization Model The lower approximation optimization model is presented in Model 1. The objective function (10a) minimizes maximum water depths over the set $\mathcal{A}$ of assets. Constraints (10b) and (10c) limit the number of barriers, n_{ij}, that may be placed in each cell. This number must be no greater than M, the maximum allowable number of barriers per cell, as specified in (10b). The budget of barriers is limited by Constraint (10d). Constraint (10e) defines the water surface elevation as the sum of topographic elevation (i.e., base elevation and barrier additions, each with height ΔB) and water depth. Constraint (10f) defines estimated outgoing flux values, which must always be greater than or equal to zero. Constraints (10g) and (10h) define the outgoing flux values as prescribed by the lower approximation. Constraint (10i) provides a convenient definition for f^{in}, the sum of all incoming flux. Finally, the integration of water depth is defined using an Euler step in Constraint (10j).

Upper Approximation Optimization Model The upper approximation optimization model is presented in Model 2. The model is clearly similar to the lower approximation optimization model. The only differences are in Constraints (11g) and (11h), which ensure that outgoing fluxes are nonzero only when the water depth within a cell is greater than zero, and in Constraint (11j), which ensures nonnegative depths: if the predicted net flux results in the transfer of water greater than what is contained within the cell, this depth is set to zero.

Model 1: Lower approximation optimization model

$$\text{minimize} \quad \sum_{ij\in\mathcal{A}} \max_t h_{ijt} \tag{10a}$$

$$\text{subject to} \quad n_{ij} \in [0, M] \tag{10b}$$

$$n_{ij} = 0;\ \forall (i,j) \in \mathcal{A} \tag{10c}$$

$$\sum_{ij} n_{ij} = n \tag{10d}$$

$$w_{ijt} = (B_{ij} + n_{ij}\Delta B) + h_{ijt} \tag{10e}$$

$$\tilde{f}^k_{ijt} = \max\left(0, f^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}(w_{ij,t-1} - w^k_{ij,t-1})\right) \tag{10f}$$

$$f^k_{ijt} = \tilde{f}^k_{ijt} \quad \text{if} \quad h_{ij,t-1}\Delta x\Delta y \geq \Delta t \sum_k \tilde{f}^k_{ijt} \tag{10g}$$

$$f^k_{ijt} = 0 \quad \text{if} \quad h_{ij,t-1}\Delta x\Delta y < \Delta t \sum_k \tilde{f}^k_{ijt} \tag{10h}$$

$$f^{in}_{ijt} = f^E_{i-1,j,t} + f^W_{i+1,j,t} + f^N_{i,j-1,t} + f^S_{i,j+1,t} \tag{10i}$$

$$h_{ijt} = h_{ij,t-1} + \Delta t \frac{f^{in}_{ijt} - \sum_k f^k_{ijt}}{\Delta x \Delta y} \tag{10j}$$

Model 2: Upper approximation optimization model

$$\text{minimize} \quad \sum_{ij\in\mathcal{A}} \max_t h_{ijt} \tag{11a}$$

$$\text{subject to} \quad n_{ij} \in [0, M] \tag{11b}$$

$$n_{ij} = 0;\ \forall (i,j) \in \mathcal{A} \tag{11c}$$

$$\sum_{ij} n_{ij} = n \tag{11d}$$

$$w_{ijt} = (B_{ij} + n_{ij}\Delta B) + h_{ijt} \tag{11e}$$

$$\tilde{f}^k_{ijt} = \max\left(0, f^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}(w_{ij,t-1} - w^k_{ij,t-1})\right) \tag{11f}$$

$$f^k_{ijt} = \tilde{f}^k_{ijt} \quad \text{if} \quad h_{ij,t-1} > 0 \tag{11g}$$

$$f^k_{ijt} = 0 \quad \text{if} \quad h_{ij,t-1} \leq 0 \tag{11h}$$

$$f^{in}_{ijt} = f^E_{i-1,j,t} + f^W_{i+1,j,t} + f^N_{i,j-1,t} + f^S_{i,j+1,t} \tag{11i}$$

$$h_{ijt} = \max\left(0, h_{ij,t-1} + \Delta t \frac{f^{in}_{ijt} - \sum_k f^k_{ijt}}{\Delta x \Delta y}\right) \tag{11j}$$

5 Empirical Results

This section reports some preliminary results regarding the proposed approximations and the associated optimization models.

5.1 Evaluation of the Flood Model Relaxations

This section compares differences among the discussed simulation models, i.e., the 2D shallow water equations, pipe flow, lower approximation, and upper approximation models. The comparison uses the historical Taum Sauk dam failure as an example scenario, with a thirty meter spatial resolution and a grid containing approximately 38,000 cells. In the models, a gravitational acceleration constant of 9.80665 m/s^2 was used, and the dam failure was modeled as a time-dependent volumetric point source, using a hydrograph similar to a United States Geological Survey estimate [11]. In the shallow water equations model, a Manning's roughness coefficient of 0.035 was used, and time steps varied based on a Courant condition. In the remaining models, various constant cross-sectional pipe areas and time steps were used. Note that, in a simulation context, Mei, Decaudin, and Hu did not suggest using constant cross-sectional pipe areas or time steps; however, our intent was to simplify these models as much as possible.

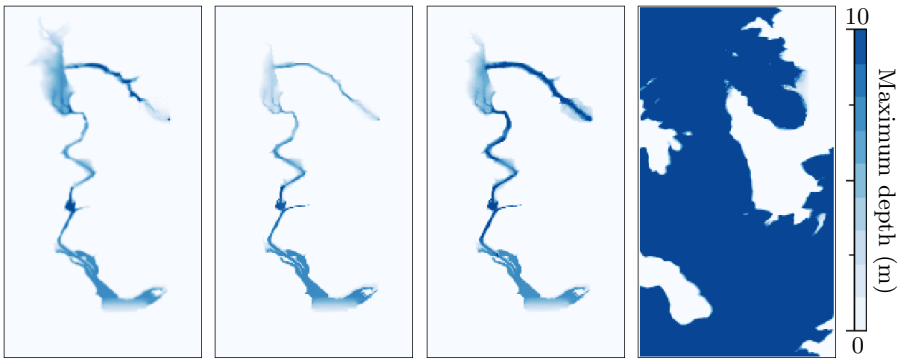
For flood mitigation, we are primarily concerned with the accuracy of maximum depth estimates over a simulation's time extent.[3] Figure 3 compares images of maximum depth results from a 2D shallow water equations model (**SWE**) similar to [3], as well as pipe flow (**P**), lower approximation (**L**), and upper approximation (**U**) models which used various pipe areas and time steps.

The top row of Figure 3 compares **SWE** with **P**, **L**, and **U** using a parameterization calibrated to minimize the root-mean-square error between **P** and **SWE**. **P** and **L** provided very reasonable estimates of **SWE**, but **U** greatly overestimated maximum depths. This is because, in **U**, the large pipe area of 500 m^2 resulted in unrestricted large fluxes and thus poor volume conservation. In the second row, the pipe area was substantially decreased, and the pipe flow and lower approximation models overestimated **SWE**, although **U** behaved more reasonably. Finally, in the third row, as Δt was decreased, **U** began to converge upon **P** and **L**. Most model parameterizations provided reasonable simulated flood extents, similar to those found in the literature [11, 7].

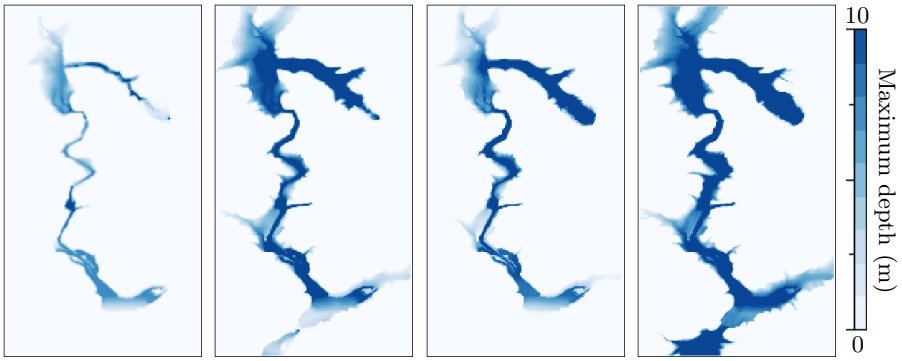
Finally, Figure 4 reports volume conservation error for selected upper approximation parameterizations. As anticipated, the pipe flow and lower approximation models conserved volume well, with error on the order of machine epsilon. The upper approximation accumulated error more rapidly, although it displayed good convergence as the time step was decreased.

It is important to note a unique difference between pipe flow simulations and traditional two dimensional hydrodynamic simulations based on the shallow water equations. When using a full two-dimensional shallow water model, the

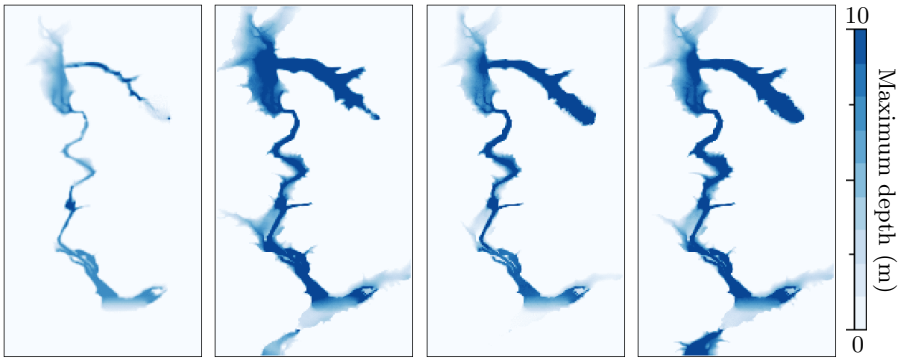
[3] This is different from evacuation settings, in which the flood arrival time at various locations is critical information.



(a) **SWE** (b) $\mathbf{P}_{\Delta t=1.0,A=500}$ (c) $\mathbf{L}_{\Delta t=1.0,A=500}$ (d) $\mathbf{U}_{\Delta t=1.0,A=500}$



(e) **SWE** (f) $\mathbf{P}_{\Delta t=1.0,A=5}$ (g) $\mathbf{L}_{\Delta t=1.0,A=5}$ (h) $\mathbf{U}_{\Delta t=1.0,A=5}$



(i) **SWE** (j) $\mathbf{P}_{\Delta t=0.1,A=5}$ (k) $\mathbf{L}_{\Delta t=0.1,A=5}$ (l) $\mathbf{U}_{\Delta t=0.1,A=5}$

Fig. 3: Maximum flood depths for ten-hour simulations of the historical Taum Sauk dam failure using shallow water equations (**SWE**), pipe flow (**P**), lower approximation (**L**), and upper approximation (**U**) models. Pipe flow, lower approximation, and upper approximation models are compared using constant time steps (Δt) of 1.0 and 0.1 s and cross-sectional pipe areas (A) of 500 and 5 m^2.

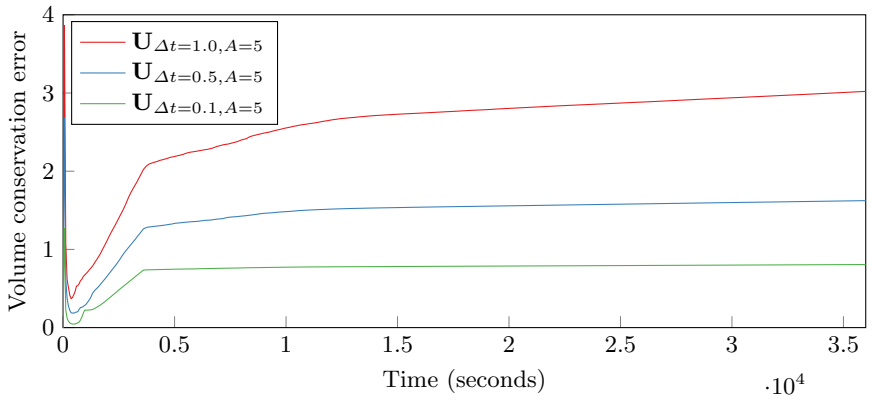


Fig. 4: Volume conservation error in upper approximation models ($\mathbf{U}$) for ten-hour simulations of the Taum Sauk dam break using various time steps. Volume conservation error was computed as $(V_{computed} - V_{added})/V_{added}$.

Taum Sauk scenario can be almost fully simulated using a simulation time extent of three hours. In contrast, the pipe flow and approximated models allow for faster or slower propagation, depending on the model parameterization. As an example, the large pipe area used to produce simulation results in the top row of Figure 3 resulted in fast propagation; the flood was fully propagated in less than an hour. The smaller pipe areas used in the second and third rows resulted in slower propagation; a time extent of roughly three hours was required. In general, as the cross-sectional pipe area decreased, a longer time extent was required for full propagation. Nonetheless, since flood mitigation is primarily concerned with protecting assets, and thus maximum water depth, we found differences in flood propagation speed acceptable for our current application.

5.2 The Potential of Optimization

This section describes some small case studies to highlight the potential and challenges of optimization for flood mitigation and, more generally, the integration of simulation and optimization.

Experimental Setting The lower and upper optimization models were implemented using the C++ CPLEX interface and run on twenty Intel Xeon E5-2660 v3 cores at 2.60 GHz, with 128 GB of memory. Conditional expressions and min/max functions were eliminated using big-M transformations. No attempt was made to optimize the model or exploit problem structure.

A Simple Case Study To validate the optimization model, an 8x8 scenario was constructed, with Δx and Δy equal to one meter. In this scenario, a topographic gradient was introduced, from the top to the bottom of the domain, with elevations linearly decreasing from 0.7 to zero meters in steps of 0.1 meters. Four cells near the top of the domain were initialized to contain one meter of water depth.

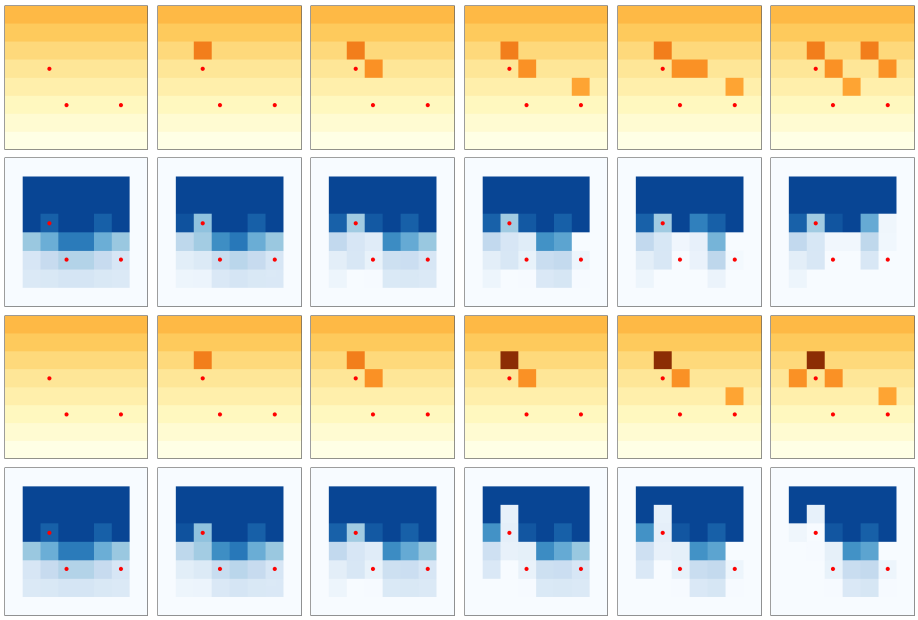


Fig. 5: Optimal elevation fields and maximum depths using the lower approximation, where the allowable number of barriers per cell is one (first two rows) or two (next two rows), and the total number of barriers ranges from zero to five. Darker oranges and blues correspond to larger topographic elevations and depths, respectively, and red circles correspond to assets that were protected.

Under the influence of gravity and in the presence of the topographic gradient, the water was pushed down the domain over time. Three assets to protect were arbitrarily placed throughout the domain, and individual barrier heights (ΔB) of 0.5 meters were employed. A constant time step of 0.1 seconds was used, and eight time steps were simulated. The optimization problem was varied to understand how solutions changed using various rules for resource allocation. In particular, the experiments studied limits on the total number of barriers and limits on the number of allowable barriers per cell.

Optimal Asset Protection Figure 5 displays optimization results from the lower optimization models. Observe that, when only one barrier was allowed per cell, the optimization model tried to mitigate flooding in the asset regions almost one at a time, before placing more barriers in interesting places throughout the domain. When two barriers were allowed per cell, it clearly became preferential to protect the topmost asset, which received a large amount of water over the duration of the simulation. Figure 6 displays optimization results from the upper optimization models. These show similarly interesting outcomes. In the one barrier per cell case, the optimization decided to protect the topmost asset less in favor of protecting the bottom assets. When two barriers were allowed per cell and enough barriers were available, it was clearly beneficial to protect the

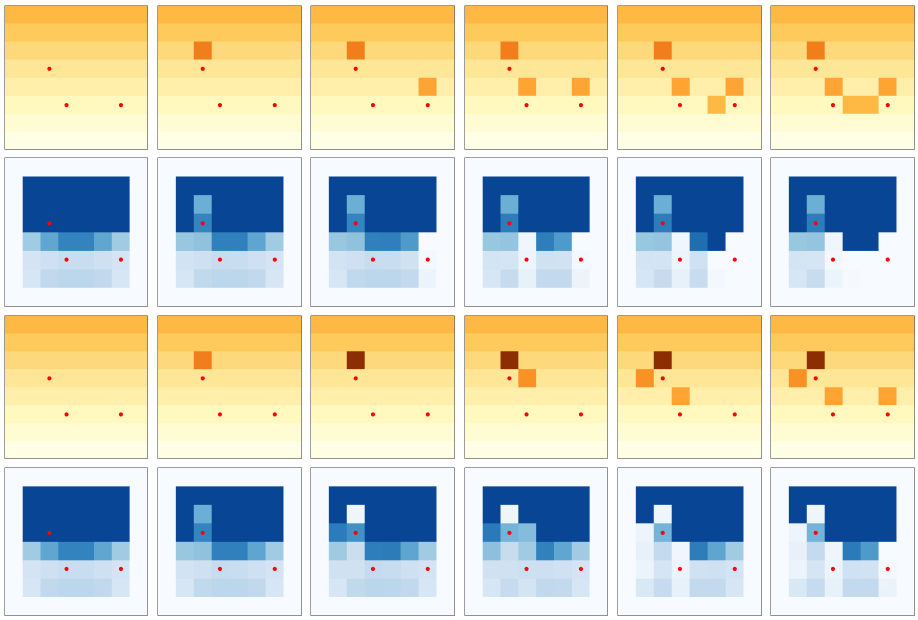


Fig. 6: Optimal elevation fields and maximum depths using the upper approximation (same setting as the lower approximation in Fig. 5).

topmost asset as much as possible from the water above it, which greatly reduced the objective value. It is also important to observe the non-monotonic behavior of the optimization results. Allowing more barriers often changed their optimal positions. This was the case when moving from four to five barriers in the top row and when moving from three to four in the bottom row. Since the barrier placements differed in both models, it was important to study how they behaved using the other model. These results are shown in the last two columns of Table 1. Column o_{opt} gives the optimal solutions, and the last column describes the objective value obtained when the optimal solution of the upper model was used in the lower model and vice-versa. They are particularly interesting, as they sometimes show significant differences in objective values. This indicates the need to apply robust optimization techniques. In practice, of course, solutions could be evaluated using full hydrodynamic simulations for various scenarios.

Evolution of the Objective Value Figure 7 depicts the value of the objective function as the number of available barriers increased, for cases where the models allowed one or two barriers per cell. The critical information is the importance of using multiple barriers at a specific location, since it brings significant benefits as the number of barriers increases. We anticipate similar behavior when the number of allowable barriers per cell is increased to three or four. Note also that the lower and upper approximations behaved comparably as the number of maximum barriers was increased and, as expected, the upper objective value was always greater than the lower objective value.

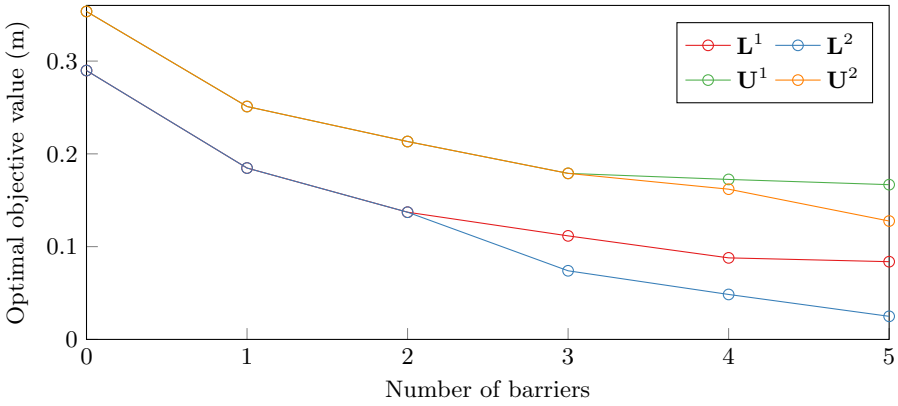


Fig. 7: Optimal objective function values using lower and upper approximations ($\mathbf{L}$ and $\mathbf{U}$) and increasing allowable numbers of barriers. Also compared are the differences among objective values when one barrier is allowed per cell ($\mathbf{L}^1$ and $\mathbf{U}^1$) versus two barriers per cell ($\mathbf{L}^2$ and $\mathbf{U}^2$).

Model	t_{CPU} (s)	n_{nodes}	n_{var}	n_{con}	n_{bin}	o_{opt} (m)	o_{com} (m)
$\mathbf{L}_3^1$	49.65	37808	2280	4410	840	0.111607	0.134935
$\mathbf{L}_4^1$	83.66	71528	2280	4410	840	0.0878997	0.123499
$\mathbf{L}_5^1$	91.53	66485	2280	4410	840	0.0837807	0.112588
$\mathbf{L}_3^2$	82.74	86964	2646	4998	975	0.0739754	0.134935
$\mathbf{L}_4^2$	134.58	121477	2651	5004	978	0.0484597	0.155655
$\mathbf{L}_5^2$	80.61	56586	2651	5004	978	0.0248889	0.13016
$\mathbf{U}_3^1$	55.56	41487	2027	3840	840	0.178971	0.239858
$\mathbf{U}_4^1$	46.17	25037	2027	3840	840	0.17244	0.234095
$\mathbf{U}_5^1$	123.50	83850	2027	3840	840	0.166794	0.225759
$\mathbf{U}_3^2$	77.29	55398	2379	4292	1018	0.178971	0.197646
$\mathbf{U}_4^2$	263.72	203168	2382	4297	1019	0.161937	0.163313
$\mathbf{U}_5^2$	234.85	108909	2382	4297	1019	0.127602	0.157531

Table 1: General statistics and analysis of selected optimization models.

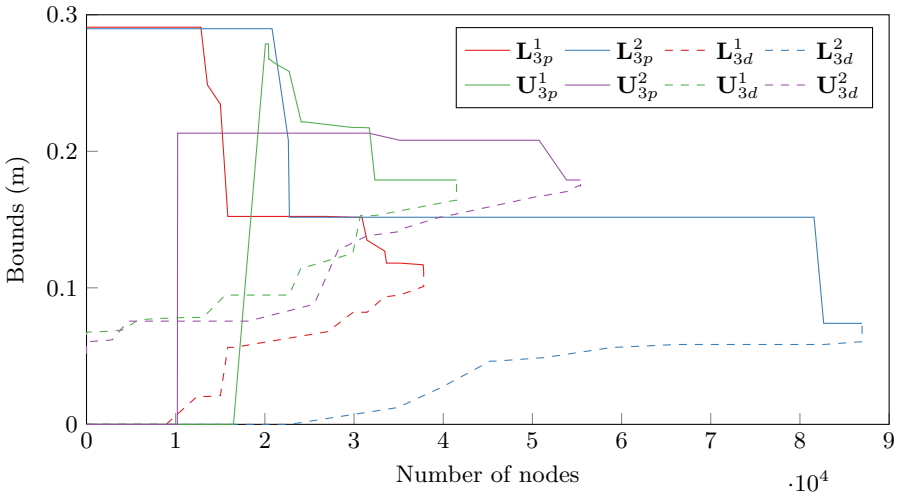


Fig. 8: Performance of CPLEX on problems where $n = 3$; p and d indicate the primal and dual, respectively.

Computational Results Finally, Table 1 presents preliminary computational results. The first column describes the instance in terms of lower ($\mathbf{L}$) or upper ($\mathbf{U}$) approximations. The superscript represents the maximum number of barriers per cell, and the subscript represents the total number of barriers. The second column denotes (wall clock) execution time, in seconds. The third column shows the number of nodes in the search tree. The fourth, fifth, and sixth columns describe the number of variables, constraints, and binary variables after presolve. Column o_{opt} describes the optimal objective value in meters of flood depth. The last column describes the objective value obtained when the optimal solution of the upper model was used for the lower model and vice-versa.

As mentioned earlier, no attempt was made to optimize the model or to exploit the problem structure. The instances have about 2,000 (mostly binary) variables and 4,000 constraints, and they can typically be solved in a few minutes. In general, CPLEX was not able to find high-quality solutions quickly, which substantially increased computation times. Integrating good primal heuristics should improve performance significantly. This is illustrated in Figure 8, where CPLEX spent significant time improving the primal bound.

6 Conclusion

Each year, flood-related disasters cause billions of dollars in damage, loss of life, and significant human suffering. Resources such as levees and berms are constructed and utilized to mitigate the consequences of such events. The deployment of these mitigation efforts is often ad hoc and relies on subject matter expertise, as computational methods are immature due to the complexity of

embedding flood models in modern optimization technologies. The goal of this paper was to establish the foundations for a more principled approach to flood mitigation. It introduced the Optimal Flood Mitigation Problem (OFMP), which aims at integrating simulation and optimization tightly by including flood simulation equations as part of the optimization model. To ensure the tractability of the approach, the main contribution of the paper is the development of linear, physics-based approximations of flood propagation models. Experimental results on the Taum Sauk dam failure show the potential of the models for predicting flood extent and maximum water depths. The integration of these approximations in optimization models was tested on a small case study, demonstrating the potential of optimization in this context.

Our current work is focused on addressing the computational challenges raised by the OFMP. Surprisingly, state of the art MIP solvers are not capable of exploiting the structure of this application. In particular, they do not seem to recognize that, once the barriers are placed, the problem is predetermined. That is, given a fixed topographic elevation field, only the deterministic simulation step remains. A combination of constraint programming (for fast propagation of the water depths) and linear programming (for computing a strong lower bound) has much potential in addressing this challenge. In addition, it would be interesting to consider whether strong dominance relationships hold between candidate solutions. More generally, exploiting the natural separation between mitigation decisions and flood propagation variables will be key when scaling to realistic problems.

Acknowledgments We gratefully acknowledge our early discussions on flood modeling and mitigation with Feng Pan and David Judi at Pacific Northwest National Laboratory [6]. The work at LANL was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396.

References

1. Brekelmans, R., den Hertog, D., Roos, K., Eijgenraam, C.: Safe Dike Heights at Minimal Costs: The Nonhomogeneous Case. Operations Research 60(6), 1342–1355 (Dec 2012), http://pubsonline.informs.org/doi/abs/10.1287/opre.1110.1028
2. Che, D., Mays, L.W.: Development of an Optimization/Simulation Model for Real-Time Flood-Control Operation of River-Reservoirs Systems. Water Resources Management 29(11), 3987–4005 (Jun 2015), http://link.springer.com/10.1007/s11269-015-1041-8
3. Chertock, A., Cui, S., Kurganov, A., Wu, T.: Well-balanced positivity preserving central-upwind scheme for the shallow water system with friction terms. International Journal for Numerical Methods in Fluids 78(6), 355–383 (2015), http://dx.doi.org/10.1002/fld.4023, fld.4023
4. Colombo, R.M., Guerra, G., Herty, M., Schleper, V.: Optimal Control in Networks of Pipes and Canals. SIAM Journal on Control and Optimization 48(3), 2032–2050 (Jan 2009), http://epubs.siam.org/doi/abs/10.1137/080716372
5. Downton, M.W., Miller, J.Z.B., Pielke, R.A.: Reanalysis of U.S. National Weather Service Flood Loss Database. Natural Hazards Review 6(1), 13–22 (Feb 2005), http://ascelibrary.org/doi/10.1061/(ASCE)1527-6988(2005)6%3A1(13)
6. Judi, D., Tasseff, B., Bent, R., Pan, F.: LA-UR 14-21247: Topography-based flood planning and optimization capability development report. Tech. rep., Los Alamos National Laboratory (2014)
7. Kalyanapu, A.J., Shankar, S., Pardyjak, E.R., Judi, D.R., Burian, S.J.: Assessment of GPU computational enhancement to a 2D flood model. Environmental Modelling and Software 26(8), 1009–1016 (2011), http://dx.doi.org/10.1016/j.envsoft.2011.02.014
8. Mei, X., Decaudin, P., Hu, B.G.: Fast hydraulic erosion simulation and visualization on GPU. In: 15th Pacific Conference on Computer Graphics and Applications. pp. 47–56. Maui, HI (2007)
9. Pielke, R.A., Downton, M.W.: Precipitation and Damaging Floods: Trends in the United States, 193297. Journal of Climate 13(20), 3625–3637 (Oct 2000), http://journals.ametsoc.org/doi/abs/10.1175/1520-0442(2000)013%3C3625%3APADFTI%3E2.0.CO%3B2
10. Proverbs, D., Mambretti, S., Brebbia, C., Wrachien, D.: Flood Recovery Innovation and Response III. WIT Press, Southampton, UK (2012)
11. Rydlund, P.H.: Peak discharge, flood profile, flood inundation, and debris movement accompanying the failure of the upper reservoir at the Taum Sauk Pump Storage Facility near Lesterville, Missouri. Tech. rep. (2006)