Optimization of Critical Infrastructure with Fluids

by

Byron Alexander Tasseff

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Marina A. Epelman, Co-Chair
Professor Pascal Van Hentenryck, Georgia Tech, Co-Chair
Dr. Russell Bent, Los Alamos National Laboratory
Professor Seth Guikema
Professor Jerome Lynch

Byron Alexander Tasseff
byron@tasseff.com
ORCID iD: 0000-0002-5043-8305

To Sarah,
the infinite support of my irrelaxable nonlinearity.

Acknowledgments

I owe a tremendous amount of gratitude to several people for their enduring support over the course of my degree. It all began with Russell Bent at Los Alamos National Laboratory, who first introduced me to the fascinating discipline of optimization. At the time, I was working primarily as a computational physicist. I couldn't understand why Russell was so obsessed with *approximating* and *relaxing* physical equations. It seemed like heresy. Nonetheless, I gradually came around to this new way of thinking, and I credit Russell for the largest step thus far taken during my intellectual development.

However, Russell has provided much more than a vast amount of technical inspiration and guidance. Without him and other supportive colleagues at Los Alamos National Laboratory, including Scott Backhaus, David Cremer, Kim Edlund, Beth Hornbein, Lisa Inkret, and Matt Kirkland, my oscillation between full-time work and graduate school would have been logistically impossible. I thank them for their willingness to support such an endeavour (don't graduate students *become* technical staff and not the other way around?), despite the bureaucratic difficulties it entailed. I will never forget it.

Russell was also the first to introduce me to Pascal Van Hentenryck. "I met a famous mathematician today!" I remember proudly telling my mom after Pascal's visit to Los Alamos in 2013. Three years later, I would begin to know Pascal more closely as my Ph.D. adviser, which has been an honor. I am consistently surprised by Pascal's ability to find new ways of looking at a problem. I'm also appreciative of his emphasis on the application of his work to the real world. It is rare to observe genuine altruism in academia, but Pascal has proven to be an exception. It would be easy for him to take a hiatus, but he instead chooses to work at a breakneck pace in pursuit of solutions to large societal problems. Aside from his playful brilliance and intense work ethic, Pascal has been a source of great reassurance during the most difficult portions of my Ph.D.

After I had achieved candidacy, Pascal moved to Georgia Tech, which required me to name another "official" adviser at the University of Michigan. After Pascal's coaxing

and with some hesitation, Marina Epelman agreed to serve in this role, and it was my sincere hope that I would not become a burden. Prior to this, I had taken a convex optimization course taught by Marina, and I was convinced that she was the most technically capable professor in the department. Naturally, I was intimidated at the outset. As I continued to update her on my work, however, she became more interested and involved. Ultimately, she ended up playing a fundamental role in the later stages of this dissertation. I have been routinely impressed by Marina's patience and willingness to ask questions, but I am most grateful for her attention to detail. Collaborating with her on our water network design work (Chapter 5) will remain a cherished memory. I have never worked with someone who shares my desire for meticulously crafted, powerful composition. I will forever regard Marina as the Leo Tolstoy of nonlinear programming.

Since I returned to Los Alamos in 2018 after two years in Ann Arbor, two colleagues in particular have been incredibly supportive in providing technical mentoring and funding. The first is Donatella Pasqualini, who gave me a great amount of freedom during our work on water network design. This ultimately led to Chapter 5 of this dissertation and the ability to secure further funding in the area of water network optimization, which in turn led to Chapter 6. The second is Carleton Coffrin, who has served as a wonderful role model and whose technical guidance helped tremendously in Chapters 6, 7, and 8. I am constantly amazed by Carleton's planning skills, technical breadth, patience, and ability to quickly generate good new ideas. I am grateful for his support and friendship.

There are many others whose technical foundations and assistance played a large role in my work. The flood-related studies in Chapters 3 and 4 were largely borne from my mentorship under David Judi, now at Pacific Northwest National Laboratory. Chapters 5–8 build upon the InfrastructureModels software ecosystem. David Fobes and Kaarthik Sundar of Los Alamos National Laboratory, as well as many others, have played a critical role in the development of the Julia packages that support this ecosystem. Clayton Barrows, Devon Sigler, Jonathan Stickel, and Ahmed Zamzam of the National Renewable Energy Laboratory, as well as Yang Liu of Stanford University, have acted as collaborators on our ongoing work related to Chapter 6. Anatoly Zlotnik of Los Alamos National Laboratory served as a collaborator on Chapter 7. Finally, Seth Guikema and Jerome Lynch provided feedback during the drafting of this dissertation.

I'd similarly like to express my gratitude to the friends and connections made during my time in Ann Arbor. First, I'd like to thank my course professors, especially Brian

Denton, Marina Epelman, Jon Lee, and Neda Masoud, whose quality of instruction was extraordinary. I'd also like to thank the members of Pascal's lab whose bonds will not be forgotten. His postdocs, especially Ferdinando Fioretto, Antoine Legrain, and Terrence Mak, have been wonderful examples and good friends. Finally, I'd like to thank the students in Pascal's lab who made the challenge of graduate school much more bearable. Four students stand out in particular: Geunyeong Byeon, Palaniappan Chellappan, Gabriel Hackebeil, and Connor Riley. I admire each of you in unique ways.

My family has also been supportive. My dad, Tom, instilled in me that "school is prison" growing up, which is hilariously ironic in retrospect. I nonetheless credit him for my unrelenting anti-authoritarianism, a distinguishing trait of good scientists and their broader superset, the clinically insane. I thank my mom, Colleen, for her encouragement and compassion. She has loved me no matter where my life has led and is always the first person I call. I finally thank my older siblings, Lindsay and George. I cannot imagine how different my life would look if it were not for their serving as role models.

I finally thank my wife Sarah, without whom this undertaking would have been impossible. Sarah temporarily left her job at Los Alamos National Laboratory to accompany me in Ann Arbor, sacrificing innumerable personal and professional prospects. She selflessly provided every facet of support possible and great comfort during very difficult times. Despite the large number of enormously challenging life changes we've experienced over the past five years, she has remained the most thoughtful and kind person that I've ever met, and every day, I feel grateful that Sarah loves a schlub like me. Sarah also designed the beautiful frontispiece of this dissertation on her own generous accord.

Finally, this material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256260 and under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. 89233218CNA000001.

Table of Contents

List of Figures

List of Tables

List of Appendices

List of Acronyms

2D two-dimensional.

3D three-dimensional.

AC alternating current.

BB branch and bound.

BCE before Common Era.

FRM flood risk management.

LP linear programming.

LP/NLP-BB linear programming/nonlinear programming branch and bound.

MICP mixed-integer convex programming.

MICQP mixed-integer convex quadratic programming.

MILP mixed-integer linear programming.

MINCP mixed-integer nonconvex nonlinear programming.

MINLP mixed-integer nonlinear programming.

MINQP mixed-integer nonconvex quadratic programming.

MIQP mixed-integer quadratic programming.

MLD Maximal Load Delivery.

NLP nonlinear programming.

OBBT optimization-based bound tightening.

OBCG optimization-based cut generation.

ODE ordinary differential equation.

OFMP Optimal Flood Mitigation Problem.

OWF Optimal Water Flow.

PDE partial differential equation.

PGA peak ground acceleration.

PGV peak ground velocity.

SOCP second-order cone programming.

List of Symbols

Surface water modeling

Horizontal positions	x, y	Bathymetric elevation	B
Water surface elevation	w	Water depth	$h := w - B$
Horizontal velocities	u, v	Horizontal discharges	hu, hv
Time	t	Water density	ρ
Gravitational acceleration	g	Bottom friction	τ
Roughness coefficient	n	Volumetric source	R

Water network design

Set of nodes	$\mathcal{N}$	Set of pipe arcs	$\mathcal{A}$
Set of reservoirs	$\mathcal{S}$	Set of demand junctions	$\mathcal{J}$
Set of resistances at $a \in \mathcal{A}$	$\mathcal{R}_a$	Total hydraulic head	h
Volumetric flow rate	q	Length of a pipe	L
Diameter of a pipe	D	Resistance per unit length	r, p
Demand at a junction	d	Cost per unit length	c
Head loss exponent	α	Design decision	z

Water network operation

Set of nodes	$\mathcal{N}$	Set of links	$\mathcal{L}$
Set of pipes	$\mathcal{A} \subset \mathcal{L}$	Set of short pipes	$\mathcal{S} \subset \mathcal{L}$
Set of valves	$\mathcal{V} \subset \mathcal{L}$	Set of regulators	$\mathcal{W} \subset \mathcal{L}$
Set of pumps	$\mathcal{P} \subset \mathcal{L}$	Set of demands	$\mathcal{D} \subset \mathcal{N}$
Set of reservoirs	$\mathcal{R} \subset \mathcal{N}$	Set of tanks	$\mathcal{T} \subset \mathcal{N}$
Set of time points	$\mathcal{K}$	Set of time intervals	$\mathcal{K}'$
Number of time points	K	Number of time intervals	K'
Total hydraulic head	h	Volumetric flow rate	q
Length of a pipe	L	Resistance per unit length	r
Valve or pump status	z	Pump head gain	G
Tank volume	V	Time step length	Δt
Power consumption	P	Energy consumption	E
Efficiency	η	Energy cost	c

(*continued on next page…*)

(*...continued from previous page*)

Natural gas networks

Set of junctions	$\mathcal{J}$	Set of receipts	$\mathcal{R}$
Set of deliveries	$\mathcal{D}$	Set of arc components	$\mathcal{A}$
Set of pipes	$\mathcal{P} \subset \mathcal{A}$	Set of short pipes	$\mathcal{S} \subset \mathcal{A}$
Set of resistors	$\mathcal{T} \subset \mathcal{A}$	Set of loss resistors	$\mathcal{U} \subset \mathcal{A}$
Set of valves	$\mathcal{V} \subset \mathcal{A}$	Set of regulators	$\mathcal{W} \subset \mathcal{A}$
Set of compressors	$\mathcal{C} \subset \mathcal{A}$	Supply flow	s
Demand flow	d	Resistance of a pipe	w
Resistance of a resistor	τ	Pressure	p
Mass flow	f	Pressure ratio	α
Component status	z	Flow direction	y

Power transmission networks

Set of buses	$\mathcal{N}$	Set of forward lines	$\mathcal{E}$
Set of reverse lines	$\mathcal{E}^R$	Set of generators	$\mathcal{G}$
Set of loads	$\mathcal{L}$	Set of shunts	$\mathcal{H}$
Line admittance	Y	Line charging	Y^c
Shunt admittance	Y^s	Transformer properties	T
Power	S	Phase angle difference	θ
Voltage	V	Component status/scalar	z

Abstract

Many of the world's most critical infrastructure systems control the motion of fluids. Despite their importance, the design, operation, and restoration of these infrastructures are sometimes carried out suboptimally. One reason for this is the intractability of optimization problems involving fluids, which are often constrained by partial differential equations or nonconvex physics. To address these challenges, this dissertation focuses on developing new mathematical programming and algorithmic techniques for optimization problems involving difficult nonlinear constraints that model a fluid's behavior. These new contributions bring many important problems within the realm of tractability.

The first focus of this dissertation is on surface water systems. Specifically, we introduce the Optimal Flood Mitigation Problem, which optimizes the positioning of structural measures to protect critical assets with respect to a predefined flood scenario. Two solution approaches are then developed. The first leverages mathematical programming but does not tractably scale to realistic scenarios. The second uses a physics-inspired metaheuristic, which is found to compute good quality solutions for realistic scenarios.

The second focus is on potable water distribution systems. Two foundational problems are considered. The first is the optimal water network design problem, for which we derive a novel convex reformulation, then develop an algorithm found to be more effective than the current state of the art on select instances. The second is the optimal pump scheduling (or Optimal Water Flow) problem, for which we develop a mathematical programming relaxation and various algorithmic techniques to improve convergence.

The final focus is on natural gas pipeline systems. Two novel problems are considered. The first is the Maximal Load Delivery (MLD) problem for gas pipelines, which aims at finding a feasible steady-state operating point that maximizes load delivery for a severely damaged gas network. The second is the joint gas-power MLD problem, which couples damaged gas and power networks at gas-fired generators. In both problems, convex relaxations of nonconvex dynamical constraints are developed to increase tractability.

Chapter 1

Introduction

Critical infrastructure, which includes flood control systems, potable water distribution systems, and bulk energy transport systems, are vital to the physical and economic security of modern societies. Despite their importance, the design, operation, and restoration of these infrastructures are sometimes carried out suboptimally. This suboptimality may exist for a variety of reasons, including a lack of data to drive the decision-making process, inaccurate forecasts of infrastructure supply and demand, and the neglect of tail risks. Another fundamental difficulty, however, resides in the *modeling* of infrastructure systems, even when their behavior can be accurately approximated using well-established physical relationships. This difficulty is especially apparent for critical infrastructures that control the motion of fluids, whose dynamics are often captured by partial differential equations (PDEs) and nonconvex physics. These properties sometimes render optimization problems in which the physical models are embedded intractable. To address this challenge, this dissertation develops techniques for increasing the tractability of infrastructure optimization problems for systems that control fluids.

The topic is certainly not new, and the control of large-scale fluid systems has challenged humankind for millenia. These systems were also the progenitors of civilization. Even in prehistorical times, humans have required nearby sources of water to survive and thrive. Around 10,000 years ago saw the beginning of the Neolithic Revolution, when human culture transitioned from a hunter-gatherer lifestyle to one of agriculture and settlement [144]. As settled villages began to develop, the control of fluid naturally followed, with early wastewater drainage systems appearing around 6,500 before Common Era (BCE) and irrigation systems around 6,000 BCE [48], [127]. These and other

innovations led to the formation of nascent civilizations in the millenia that followed.

Increasing populations in fertile regions over the following years motivated early large-scale infrastructure development as humans attempted to tame the forces of nature. Complex urban water supply and sanitation systems began to form in the Indus Valley around 2,400 BCE, which included underground sewage systems as well as public and private baths [80]. Early flood myths have been dated around the same time, e.g., the Great Flood of Gun-Yu, which has physical evidence of having occurred around 1,900 BCE [146]. Narratives of the flood detail the efforts of the early Chinese Xia dynasty to mitigate the disaster through the contruction of dikes, dams, and canals [32]. These early examples of water management foreshadowed infrastructure developments to come.

However, infrastructure systems of the present day that control water and other fluids are not so different from their ancient predecessors. Fundamentally, they channelize or restrict fluid flow for either delivery or diversion while leveraging the physical forces of gravity and friction. The advent of interconnected electric power grids in the twentieth century was perhaps one of the largest paradigm shifts for these systems in millennia. Readily available energy in the form of electricity allowed for the widespread installation of water lifting devices (pumps) and the scheduled storage and release of water in elevated reservoirs. It also motivated the investment in massive infrastructure projects, including hydroelectric dams and natural gas transmission networks. Finally, to protect these systems and the growing populations that relied upon them, greater investments were made to mitigate the negative effects of frequent natural disasters, including floods.

Although humans have made great strides toward curbing the often disorderly nature of fluids, they have not yet mastered it. Natural disasters like floods, which have plagued civilization since its inception, continue to pose threats to even the most technologically advanced societies. Continually increasing and migrating populations require new and expanded infrastructure systems that are more resilient to change and disruption. The interconnection and growing interdependencies of these and other systems further render their optimal management less intuitive and often intractable. These observations suggest there is much to be learned before full control of these systems can be achieved.

This dissertation suggests that the following millenium will be one in which humanity becomes capable of fully predicting and controlling the motion of fluids. It serves as an early effort within an increasingly larger body of literature to interface the ancient study of fluid dynamics with the more recent but burgeoning fields of mathematical op-

timization and computer science. More specifically, this dissertation develops a variety of mathematical and computational approaches intended to optimize large-scale infrastructure decisions while recognizing the fluid dynamics that constrain them. It studies three application domains to highlight the importance of developing tailored techniques: (i) flood mitigation, (ii) potable water distribution, and (iii) natural gas transmission.

At present, optimization tasks involving these domains are challenging for three primary reasons: (i) the nonconvexity of fluid dynamical constraints, (ii) the combinatorial nature of decisions and controllable infrastructure elements, and (iii) computational limitations. This dissertation broadly applies two methods to address these challenges. The first is convex or linear reformulation of the fluid dynamical constraints, which render the problems more amenable to current optimization technologies. The second is the restriction of combinatorial and continuous search spaces, which aids in the faster convergence of solution techniques. Both of these approaches will likely remain key for tractably solving similar fluid-constrained infrastructure problems in the years to come.

The rest of this chapter provides technical foundations for the remainder of this dissertation. To begin, Section 1.1 provides a short introduction to the infrastructure optimization problems considered herein and the physical models that constrain them. Specifically, Section 1.1.1 describes models for flood propagation and mitigation; Section 1.1.2 describes models for water distribution systems; and Section 1.1.3 describes models for natural gas transmission systems. Then, Section 1.2 reviews the principles of mathematical optimization as they pertain to the following chapters. Specifically, it reviews linear optimization in Section 1.2.1; nonlinear optimization in Section 1.2.2; mixed-integer linear optimization in Section 1.2.3; and mixed-integer nonlinear optimization in Section 1.2.4. Finally, Section 1.3 briefly outlines the remaining chapters.

1.1. Infrastructure Problems and Fluid Models

This section summarizes the infrastructure optimization problems considered in this dissertation and the physical models that constrain them. The computational challenges that arise during the solution of these problems are discussed in Section 1.2. Finally, although this section provides very brief descriptions of the fluid models we consider, a more thorough review of the models and their derivations is provided in Chapter 2.

1.1.1. Flood Mitigation Optimization

Chapters 3 and 4 consider optimization problems that aim at minimizing flooding in predefined critical locations through the placement of structural mitigation measures. In their most accurate forms, these problems are constrained by the two-dimensional (2D) shallow water equations, a set of PDEs that model the flow of surface water, i.e.,

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = R(x, y, t), \tag{1.1a}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial (huv)}{\partial y} = -gh\frac{\partial B}{\partial x} - g\frac{n^2}{h^{1/3}}|u|u, \tag{1.1b}$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial B}{\partial y} - g\frac{n^2}{h^{1/3}}|v|v, \tag{1.1c}$$

where h is the water depth, u and v are horizontal velocities, R is a volumetric source term, g is the acceleration due to gravity, B is the bottom topography (or bathymetric elevation), and n is the Manning's roughness coefficient [30]. Even in a simulation setting, these PDEs are difficult to solve. The optimization problems are of the form

$$\underset{(x,h,u,v)\in\mathcal{X}}{\text{minimize}} \qquad \eta(x, h, u, v) \tag{1.2a}$$

$$\text{subject to} \qquad \text{Constraints (1.1)}, \tag{1.2b}$$

where x denotes decision variables not present in Equations (1.1) but implicitly affecting their solution, and $\mathcal{X}$ denotes the set of feasible decisions. Here, $\mathcal{X}$ may include practical constraints, e.g., where mitigation measures may be applied or how measures are linked to the multiple variables of Constraints (1.1). Nonetheless, Constraints (1.1) present the greatest computational challenge, as decisions x affect the PDEs' solution, (h, u, v).

1.1.2. Water Distribution System Optimization

Chapters 5 and 6 consider optimization problems that aim at minimizing the cost of design or operation of a potable water distribution system. The transport of water in these systems is physically limited by the "head loss" equations for pipes, which model

energy losses from friction in the steady-state regime. These equations are of the form

$$\Delta h = Lrq\,|q|^{\alpha-1}\,, \tag{1.3}$$

where Δh is the head loss, interpretable as a loss in energy along the pipe, L is the pipe length, r is the pipe resistance, q is the rate of volumetric flow, and α is an exponent based on engineering approximations. As an example, assuming the water system is modeled as a directed graph $\mathcal{G} := (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes (or junctions) and $\mathcal{A}$ is the set of arcs (or pipes), a simplified water system problem is of the form

$$\begin{aligned}
\underset{(x,q,h)\in\mathcal{X}}{\text{minimize}} \quad & \eta(x,q,h) && \text{(1.4a)}\\
\text{subject to} \quad & h_i - h_j = L_{ij} r_{ij} q_{ij} \left|q_{ij}\right|^{\alpha-1},\ \forall (i,j)\in\mathcal{A} && \text{(1.4b)}\\
& \sum_{(j,i)\in\delta_i^-} q_{ji} - \sum_{(i,j)\in\delta_i^+} q_{ij} = d_i,\ \forall i\in\mathcal{N}, && \text{(1.4c)}
\end{aligned}$$

where $\mathcal{X}$ denotes the set of feasible decisions, neglecting important physical relationships; $\eta(x,q,h)$ denotes an arbitrary objective function; h_i, $i\in\mathcal{N}$ denotes the total hydraulic head at a node; and d_i denotes the (often fixed) flow rate demand at node $i\in\mathcal{N}$. Additionally, the set of arcs incident to node $i\in\mathcal{N}$ where i is the tail (respectively, head) of the arc is denoted by $\delta_i^+ := \{(i,j)\in\mathcal{A}\}$ (respectively, $\delta_i^- := \{(j,i)\in\mathcal{A}\}$).

In Problem (1.4), Constraints (1.4b) model head loss relationships along pipes and Constraints (1.4c) model flow conservation at nodes. Although Constraints (1.4c) are linear, Constraints (1.4b) model nonconvex nonlinear relationships among heads and flows. Because of this nonconvexity property, as later elaborated upon in Section 1.2, the head loss relationships represent one of the greatest computational challenges from an optimization perspective. Another difficulty resides in the discreteness of the feasible set $\mathcal{X}$. Addressing these two properties is thus a large focus of Chapters 5 and 6.

1.1.3. Natural Gas System Optimization

Chapters 7 and 8 consider optimization problems that aim at maximizing the delivery of load in a damaged natural gas transmission system. The modeling of such a system is highly similar to the modeling of a water distribution system, as in Section 1.1.2. Specifically, the transport of gas in these systems is limited by the "pressure-flow" equations

for gas pipelines, which also model energy losses from friction in the steady-state regime, similar to the water system engineering Equation (1.3). These are generally of the form

$$p_1^2 - p_2^2 = wf|f|, \tag{1.5}$$

where p_1 and p_2 are pressures at the inlet and outlet of the pipe, respectively, w is the mass flow resistance of the pipe, and f is the mass flow rate through the pipe. Assuming the gas system is a directed graph $\mathcal{G} := (\mathcal{J}, \mathcal{P})$, where $\mathcal{J}$ is the set of nodes (or junctions) and $\mathcal{P}$ is the set of arcs (or pipes), a simplified gas system problem is of the form

$$\begin{aligned}
&\underset{(x,f,p)\in\mathcal{X}}{\text{minimize}} && \eta(x,f,p) && \text{(1.6a)}\\
&\text{subject to} && p_i^2 - p_j^2 = w_{ij} f_{ij} \left|f_{ij}\right|, \ \forall (i,j) \in \mathcal{P} && \text{(1.6b)}\\
& && \sum_{(j,i)\in\delta_i^-} f_{ji} - \sum_{(i,j)\in\delta_i^+} f_{ij} = d_i, \ \forall i \in \mathcal{J}, && \text{(1.6c)}
\end{aligned}$$

where $\mathcal{X}$ denotes the (again, discrete) set of feasible decisions, neglecting important physical relationships; $\eta(x, f, p)$ denotes an arbitrary objective function; p_i, $i \in \mathcal{J}$ denotes the pressure at a junction; and d_i denotes the mass flow demand or supply at $i \in \mathcal{J}$. Similar to Chapters 5 and 6, Chapters 7 and 8 address the nonconvexity of Constraints (1.6b) and the discreteness of the set $\mathcal{X}$ to yield more tractable problems.

1.2. Principles of Mathematical Optimization

Section 1.1 presented simplified representations of the optimization problems considered in this dissertation and some relevant physical models that constrain them. There are two interesting properties that unite these optimization problems. First, the fluid dynamical constraints are nonlinear and nonconvex. Second, there are often a number of discrete decision variables that are used to model each problem. In this section, we thus provide a general background on optimization theory and algorithmic techniques that can be used to solve combinatorial, nonlinear nonconvex optimization problems. A hierarchy of these techniques and where they're applied is illustrated in Figure 1.1.

In general, the subject of optimization is concerned with determining the best solution to a problem, with respect to predefined criteria, and subject to some set of constraints.

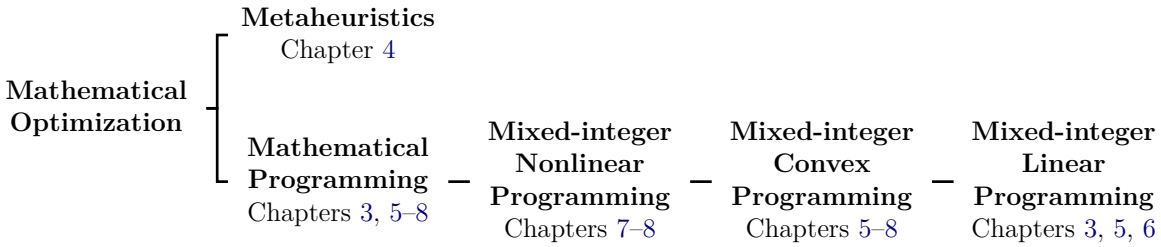


Figure 1.1: Hierarchy of optimization methods used in this dissertation.

Very broadly, there are two types of solution techniques for solving these problems. The first are "mathematical programming" methods, which rely on mathematically rigorous algorithms whose convergence guarantees are typically well-defined for particular problem classes. Examples are encountered in dynamic programming, linear programming, and convex programming. We remark that, in mathematical programming, the solution technique ultimately used depends much on *how the optimization problem is formulated*. Since there are often many possible formulations for a single optimization task, deriving a *suitable formulation* of the problem, which can then be solved via compatible deterministic methods, is an important step for developing tractable solution techniques.

The second type of solution techniques are "metaheuristic" methods. These algorithms typically provide few or no guarantees on solution quality or, in the case of many constrained optimization problems, solution feasibility. Examples include evolutionary algorithms, simulated annealing, and other so-called derivative-free techniques that rely on random perturbations of the decision variables. Although metaheuristics are capable of providing quality solutions on difficult problems containing tens to hundreds of decision variables, they often suffer on problems of higher dimensionality. As such, metaheuristics can work well on problems in which (i) a large number of variable bounds and constraints are satisfied via a "black box" method (e.g., a simulation), (ii) the remaining number of variables is relatively small, and (iii) remaining constraints are amenable to being embedded in the objective via functions that penalize infeasibilities.

Because of the lack of solution guarantees associated with metaheuristics, as well as the many computational advances in mathematical programming that have occurred over the past few decades, this dissertation focuses primarily on the development of mathematical programming techniques for the problems that are considered. The exception is the metaheuristic-inspired algorithm for optimal flood mitigation developed

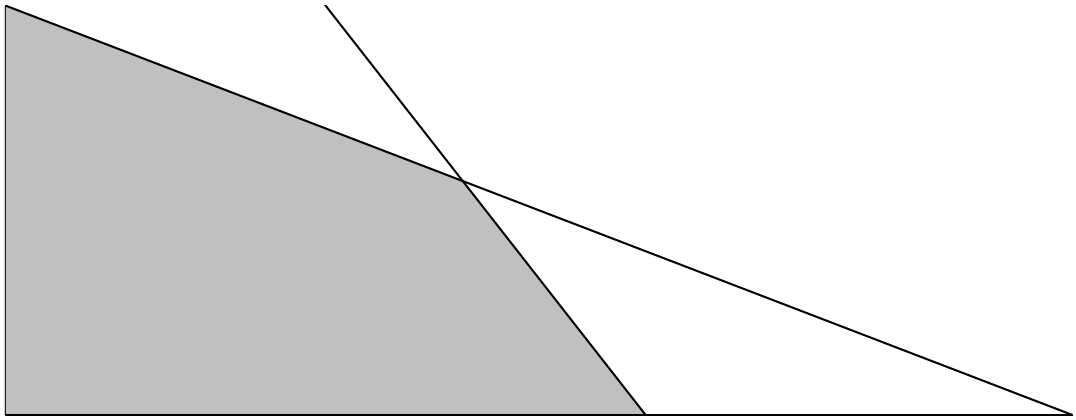


Figure 1.2: Illustration of a linear program's feasible region.

in Chapter 4, which is motivated by limitations of the mathematical programming algorithm developed in Chapter 3. Next, we provide a bottom-up overview for various classes of mathematical programming problems and summarize common solution algorithms.

1.2.1. Linear Optimization

Any linear programming (LP) problem can be written in its canonical form as

$$\begin{aligned} &\text{minimize} && c^\top x && \text{(1.7a)} \\ &\text{subject to} && Ax \leq b && \text{(1.7b)} \\ & && x \geq 0, && \text{(1.7c)} \end{aligned}$$

where c is a vector of fixed cost coefficients, x is a vector of decision variables, A is a matrix of linear constraint coefficients, and b is a vector of fixed constants. Geometrically, the set of feasible solutions to an LP is represented by a convex polyhedron. An illustration of such a polyhedron, which is the LP's feasible set, is depicted in Figure 1.2, where the variable space is assumed to be of two dimensions. The polyhedron defines all solutions that satisfy the linear Constraints (1.7b) and bound Constraints (1.7c).

In mathematical programming, LPs are regarded as "easy" to solve, and modern algorithms are routinely capable of solving problems containing millions of variables and constraints. Development of solution techniques for LPs is usually assumed to have begun with the simplex method of Dantzig in 1947 [43], which traverses vertices of

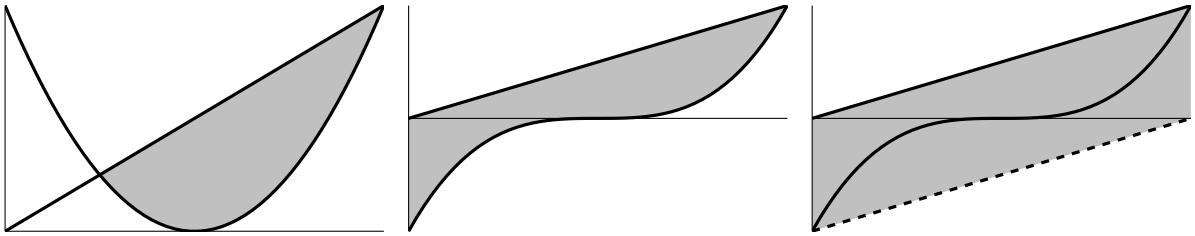


Figure 1.3: Illustrations of nonlinear program feasible regions.

the LP's polyhedron. The ellipsoid method, a polynomial-time algorithm that instead iteratively reduces the size of the solution set, was developed by Khachiyan in 1979 [11]. Finally, interior point methods, which traverse the interior of the polyhedron, began to increase in efficiency and popularity shortly after the ellipsoid method's development. Today, interior point methods are competitive with others when applied to problems of small size (less than one million variables and constraints) and are without competition for problems of larger size [63]. Due to the efficiency of LP solvers, LPs often serve as foundations for formulating or solving more difficult classes of optimization problems.

1.2.2. Nonlinear Optimization

Any nonlinear programming (NLP) problem can be written in its canonical form as

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & f(x) && \text{(1.8a)} \\ \text{subject to} \quad & g_i(x) \leq 0, \ \forall i \in \{1, 2, \dots, m\} && \text{(1.8b)} \\ & h_j(x) = 0, \ \forall j \in \{1, 2, \dots, \ell\}, && \text{(1.8c)} \end{aligned}$$

where $\mathcal{X}$ is an open set and $g_i(x)$, $h_j(x)$ are affine or nonlinear functions. For the purpose of this dissertation, there are two relevant problem subclasses: convex nonlinear and nonconvex nonlinear. A convex problem is a variant in which $\mathcal{X}$ is a convex set; $f(\cdot)$ is a convex function; all $g_i(\cdot)$ are convex functions; and all $h_j(\cdot)$ are affine functions. A nonconvex problem is one in which any of these functions are nonconvex, or if any $h_j(\cdot)$ is a nonlinear function. Examples of convex and nonconvex feasible sets are illustrated in Figure 1.3. The left example depicts a convex set and the middle, a nonconvex set.

For many classes of convex optimization problems, several polynomial-time global solution methods exist, e.g., interior point methods, as discussed in Section 1.2.1. For

nonconvex optimization problems, fewer methods are available and often do not guarantee global optimality. Some algorithms, e.g., interior point methods, are capable of providing locally optimal solutions to nonconvex problems but provide few convergence guarantees. Other algorithms, e.g., spatial branch and bound (BB), solve a number of partitioned subproblems and provide solutions within a predefined optimality tolerance. However, in many practical applications, using *accurate convex approximations* or *tight convex relaxations* of nonconvex constraints is often a reliable and efficient approach. An example of such a relaxation is shown in the last illustration of Figure 1.3. In this dissertation, a common theme is using convex approximations and relaxations of nonconvex physical constraints to render the associated optimization problems more tractable.

1.2.3. Mixed-integer Linear Optimization

Mixed-integer linear programming (MILP) problems can be written as

$$\begin{aligned} &\text{minimize} && c^\top x && \text{(1.9a)}\\ &\text{subject to} && Ax \leq b && \text{(1.9b)}\\ & && x_j \geq 0,\ \forall j \in \mathcal{C} && \text{(1.9c)}\\ & && x_j \in \{0,1\},\ \forall j \in \mathcal{B}, && \text{(1.9d)} \end{aligned}$$

where $\mathcal{C}$ is the index set defining variables that can take on continuous values and $\mathcal{B}$ is the index set defining variables that must take on discrete (in this case, binary) values. An example of a MILP's feasible region is illustrated in Figure 1.4. Here, the gray region depicts the set defined by the linear and bound Constraints (1.9b) and (1.9c). The points within this gray region depict integer-feasible solutions to the problem.

Generally, MILPs cannot be solved in polynomial time, although several solution techniques exist. Broadly, there are three types of methods commonly used to solve MILPs. The first are BB algorithms, which rely on systematically exploring a MILP's search space via fixing select binary variables (i.e., branching) and solving continuously-relaxed subproblems (LPs) that *bound* the problem's global solution (i.e., bounding). The second are cutting plane algorithms, which solve continuous relaxations (LPs) to computationally derive valid linear inequalities. After a sufficient number of inequalities have been appended to the problem, solving the resultant continuously-relaxed MILP

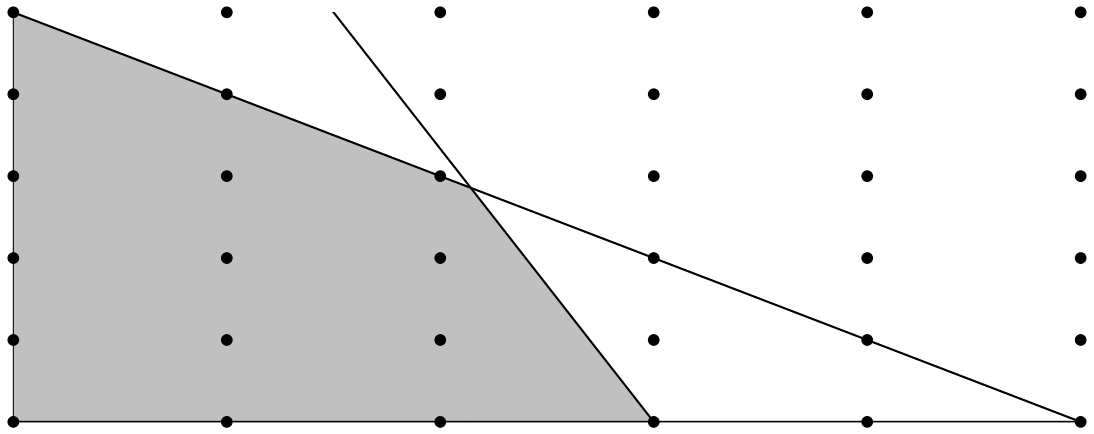


Figure 1.4: Illustration of a mixed-integer linear program's feasible region.

yields its integer-feasible global optimum. The final type of methods are heuristic algorithms, which aim at determining low-cost but not necessarily globally optimal solutions to Problem (1.9). In practice, modern commercial MILP solvers, which this dissertation relies heavily upon, combine all of these techniques to solve challenging problems.

1.2.4. Mixed-integer Nonlinear Optimization

Mixed-integer nonlinear programming (MINLP) problems combine the nonlinear aspects of Section 1.2.2 and the discrete aspects of Section 1.2.3. A MINLP can be written as

$$\begin{aligned}
&\underset{x\in\mathcal{X}}{\text{minimize}} && f(x) && \text{(1.10a)}\\
&\text{subject to} && g_i(x) \leq 0,\ \forall i \in \{1,2,\dots,m\} && \text{(1.10b)}\\
& && h_j(x) = 0,\ \forall j \in \{1,2,\dots,\ell\} && \text{(1.10c)}\\
& && x_k \in \{0,1\},\ \forall k \in \mathcal{B}. && \text{(1.10d)}
\end{aligned}$$

Similar to the classifications of NLPs in Section 1.2.2, MINLPs can be either *mixed-integer convex programming (MICP) problems* or *mixed-integer nonconvex nonlinear programming (MINCP) problems* depending on the nature of the set $\mathcal{X}$, the convexity of $g_i(\cdot)$, and the linearity of $h_j(\cdot)$. Figure 1.5 illustrates examples of three relevant feasible sets: (i) the feasible set of an MICP, (ii) the feasible set of a MINCP, and (iii) the feasible set of one possible MILP *relaxation* of (ii). Similar to Figure 1.4, gray represents each set bounded by functional constraints, and circles depict integer-feasible points.

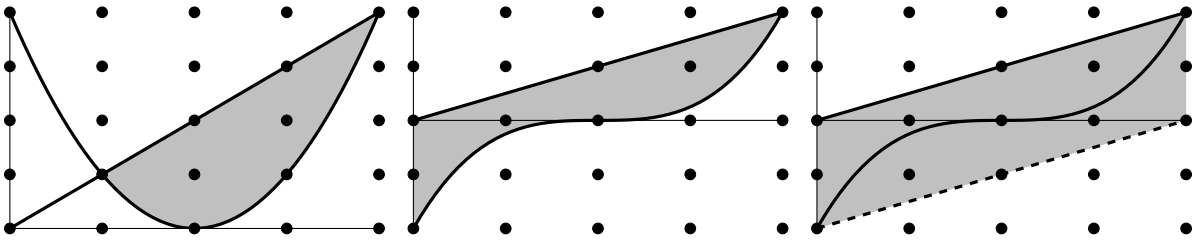


Figure 1.5: Illustrations of mixed-integer nonlinear program feasible regions.

Methods that solve MINLPs are often descendents of techniques discussed in Sections 1.2.2 and 1.2.3. For example, to address nonconvex nonlinearities in a MINCP, convex relaxations can be employed to formulate a more tractable MICP. If the problem is an MICP, nonlinear BB, in which MICP subproblems are continuously-relaxed convex programs, can then be used to determine a globally optimal solution. Similarly, valid inequalities can be derived from solutions to NLP subproblems to develop cutting plane algorithms. In practice, modern MINLP and MICP solvers often lack the reliability and efficiency of modern MILP solvers, unless the MINLP is of a specific subclass (e.g., mixed-integer quadratic). As such, many techniques rely on successive linear approximations or relaxations of the MINLP to better leverage more efficient MILP techniques.

All of the problems considered in this dissertation, described at high levels in Section 1.1, are MINCPs and thus of generally high difficulty when compared to other applied problems. First, the primary sources of nonlinear nonconvexity arise from modeling the physical dynamics of these infrastructure systems. There are also two sources of discreteness in these problems that render them *mixed-integer*. The first are system components that must be controlled via discrete decisions and affect system dynamics (e.g., the opening or closing of a valve). The second are discrete decisions that pertain to the optimization objective (e.g., the placement of a flood barrier or the addition of a network design component). Addressing the nonconvex and discrete aspects of these problems in order to make them tractable is a large technical focus of this dissertation.

1.3. Dissertation Overview

The remainder of this dissertation details the several optimization applications depicted in Figure 1.6. However, prior to describing the novel contributions of Chapters 3–8,

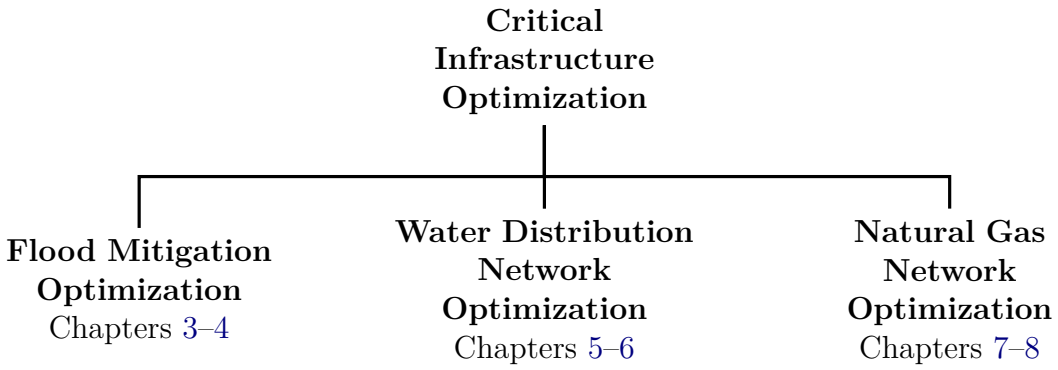


Figure 1.6: Hierarchy of critical infrastructure applications in this dissertation.

we begin with a thorough review of the fluid dynamical models that constrain these applied problems for the interested reader. That is, in Chapter 2, we describe and derive all relevant fluid dynamical approximations used in the rest of this dissertation. These derivations expand upon the very brief model descriptions provided in Section 1.1. Chapter 2 is thus intended to provide *physical justification* for the infrastructure models we consider thereafter. For the reader who is only interested in the optimization aspects of this dissertation, Chapter 2 can be skipped without overall loss of context.

The rest of this dissertation is broadly divided into the three parts depicted in Figure 1.6. The first part comprises Chapters 3 and 4, which describe techniques for solving the Optimal Flood Mitigation Problem (OFMP), previously outlined in Section 1.1.1. Chapter 3 introduces and formulates a first version of the OFMP. To solve the problem using modern MILP technologies, a linear approximation of the flooding dynamics is developed, and MILP models of the resultant OFMP, with the linearized flood propagation dynamics embedded, are then empirically compared. Although the linearized flood models are shown to be good approximators of the 2D shallow water equations, the optimization methodology is found to be incapable of scaling to realistic scenarios.

To address OFMP scalability, a metaheuristic-based optimization approach is developed in Chapter 4. Three variants of this method are then compared: (i) a direct approach using only derivative-free optimization, (ii) an augmented approach using properties of the fluid's path to restrict the search space, and (iii) a sequential approach. The latter two techniques are found to be successful, allowing the algorithm to efficiently compute good solutions to realistic, large-scale OFMPs. Moreover, unlike

Chapter 3, the algorithm of Chapter 4 (i) solves the 2D shallow water equations, (ii) uses a continuous representation of the decision space instead of a discrete one, and (iii) makes soft structural mitigation decisions, including the positions of revegetation sites.

The next part of the dissertation comprises Chapters 5 and 6, which both concern the optimization of potable water distribution systems. Chapter 5 considers the water network design problem, which contains nonconvex head loss functions and discrete resistance choices with varying costs. Traditionally, to resolve the nonconvexities of this problem, relaxations of the head loss constraints have been applied to yield a more tractable MICP. However, design solutions to these relaxed problems may not be feasible with respect to the full nonconvex physics. In this chapter, it is shown that, in fact, the original MINCP can be reformulated *exactly* as an MICP. Using this MICP as a foundation, a global optimization algorithm is developed, leveraging heuristics, outer approximations, and feasibility cutting planes for infeasible designs. Finally, the algorithm is compared against the previous state of the art in network design over a number of standard benchmarks, showing large improvements on moderately-sized instances.

Chapter 6 then addresses a separate but similar problem related to the optimal *operation* of potable water distribution systems. Unlike Chapter 5, the problem of Chapter 6 does not concern design decisions but instead *scheduling* decisions for controllable components in the network (e.g., the activation and deactivation of pumps). The discrete nature of these controls and the temporal evolution of the network complicate the scheduling problem considerably in several unique ways compared to the design problem. To address these difficulties, the chapter aims at developing tractable, relaxation-based solution techniques for the problem. Compared to the literature, it develops new preprocessing and algorithmic methods to improve upon limitations of the current state of the art. These new formulations and techniques are then empirically compared across three small- and moderately-sized water network instances with different structures.

The next part of the dissertation comprises Chapters 7 and 8, which consider natural gas transmission systems. To address the operational challenges arising from system disruptions, Chapter 7 considers the task of determining a feasible steady-state operating point for a damaged gas pipeline network while ensuring the maximal delivery of load. We formulate the mixed-integer nonconvex MLD problem, which proves difficult to solve on large-scale networks. To address this challenge, we present an MICP relaxation of the MLD problem and use it to determine bounds on the transport capacity of

a gas pipeline system. To demonstrate the effectiveness of the relaxation, the exact and relaxed formulations are compared across a large number of randomized damage scenarios on nine natural gas pipeline network models ranging in size from tens to thousands of junctions. A proof-of-concept application, which assumes network damage from a set of synthetic earthquakes, is also presented to demonstrate the utility of the proposed optimization-based capacity evaluation in the context of risk assessment for natural disasters. For all but the largest network, the relaxation-based method is found to be suitable for use in evaluating the impacts of multi-contingency network disruptions.

Chapter 8 then provides an important extension of Chapter 7, wherein the interdependencies between gas and power grids are explicitly considered. Although typically operated independently, coordination of these systems during severe disruptions can allow for targeted delivery to lifeline services, including gas delivery for residential heating and power delivery for critical facilities. To address the challenge of estimating maximum joint network capacities under such disruptions, we consider the nonconvex *joint* gas-power MLD problem. Similar to Chapter 7, to increase its tractability, we present an MICP relaxation of the joint problem. Then, to demonstrate the relaxation's effectiveness in determining bounds on network capacities, exact and relaxed MLD formulations are compared across various multi-contingency scenarios on nine joint networks ranging in size from tens to over a thousand nodes. The relaxation-based method is observed to accurately and efficiently estimate the impacts of severe joint network disruptions

Finally, Chapter 9 concludes the dissertation. Here, we reflect on the commonalities of Chapters 3–8 and the theoretical and computational challenges encountered therein. Specifically, we describe the benefits and limitations of approximation- and relaxation-based optimization techniques for critical infrastructure problems that include fluid dynamical constraints. We also reflect on the lack of general optimization techniques that can efficiently handle these constraints. Finally, we describe the possible futures of both computational fluid dynamics and mathematical programming and how advances in each could potentially improve upon the work considered in this dissertation.

Chapter 2

Fluid Dynamics Background

The applications of fluid dynamics are numerous, ranging from the simulation of microfluidic devices to global-scale climate phenomena. An overview of the fluid models used in this dissertation is illustrated in Figure 2.1. Fundamentally, almost any application involving liquid or gas flow could be modeled from first principles using the conservation laws of mass and momentum described in Section 2.1. However, for practical or computational reasons, approximations of these equations must often be used in their place. For most "simple" (or Newtonian) fluids, the Navier-Stokes equations serve as an excellent approximation. These equations are derived from first principles in Section 2.2. In the context of large-scale flooding, even these equations are computationally prohibitive. As such, the 2D shallow water equations, which are used in the later flood modeling efforts throughout Chapters 3 and 4, are derived in Section 2.3.

Unlike the 2D shallow water equations, there are several important fluid dynamical models, which have emerged for either practical or historical reasons, that are *not* fully derivable from first principles. One important subset comprises models of steady-state *incompressible* flow through a circular pipe. These models are ultimately empirical approximations of the Navier-Stokes equations under very strict assumptions. To appreciate their lines of derivation, the Euler equations are derived as an approximation of the Navier-Stokes equations in Section 2.4. The Bernoulli equation is then derived from the Euler equations in Section 2.5. This equation gives rise to the multiple pipe flow engineering equations that are subsequently derived. Pipe flow models for water distribution systems are first discussed in Section 2.6 and provide the physical foundations for Chapters 5 and 6. The pipe flow models for natural gas pipeline systems are

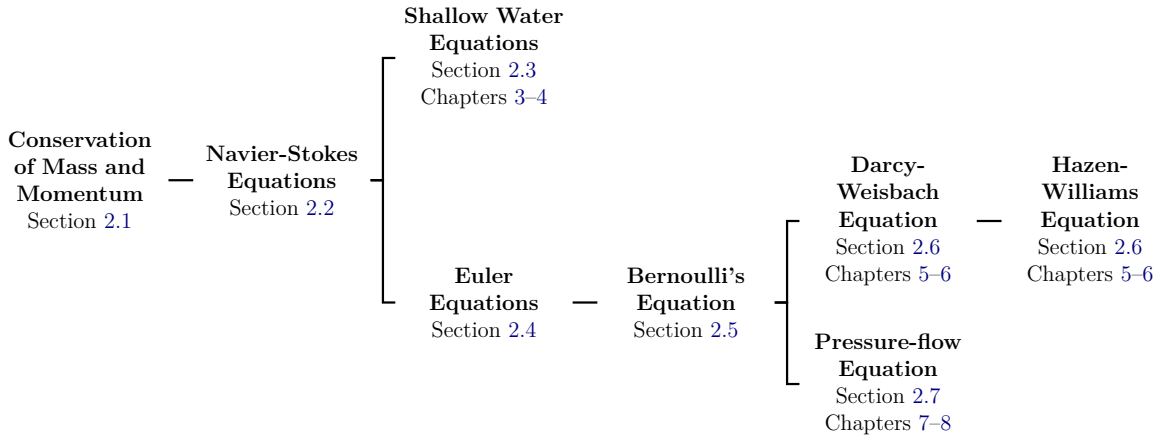


Figure 2.1: Hierarchy of fluid dynamical models considered in this dissertation.

described in Section 2.7 and provide the physical foundations for Chapters 7 and 8.

The following derivations of conservation laws, the Navier-Stokes equations, and the shallow water equations are adapted from [44]. Euler and Bernoulli equation derivations are adapted from [109]. The discussion of pipe flow equations for water is based on [21]. Finally, derivations of pipe flow equations for gas are adapted from [102, Chapter 4].

2.1. Conservation Laws

Consider an arbitrary control volume Ω. Mass conservation within Ω is defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\,\mathrm{d}V = -\int_{\partial\Omega}(\rho\mathbf{v})\cdot\mathbf{n}\,\mathrm{d}A, \tag{2.1}$$

where ρ is the fluid's density, $\mathbf{v} = (u, v, w)$ is the fluid's velocity, and $\mathbf{n}$ is the unit normal vector of the surface $\partial\Omega$. Using this relation, applying Gauss's theorem gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\,\mathrm{d}V = -\int_{\Omega}\nabla\cdot(\rho\mathbf{v})\,\mathrm{d}V. \tag{2.2}$$

Then, applying the Leibniz integral rule (and the fact that Ω is arbitrary) gives

$$\int_{\Omega}\left[\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v})\right]\mathrm{d}V = 0 \implies \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0. \tag{2.3}$$

This is the canonical mass conservation equation for a compressible fluid [44].

Again letting Ω denote a control volume, conservation of linear momentum requires

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\mathbf{v}\,\mathrm{d}V = -\int_{\partial\Omega}(\rho\mathbf{v})\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}A + \int_{\Omega}\rho\mathbf{b}\,\mathrm{d}V + \int_{\partial\Omega}\mathbf{T}\mathbf{n}\,\mathrm{d}A, \tag{2.4}$$

where $\mathbf{b}$ is the force acting throughout the volume of the fluid and $\mathbf{T}$ is the stress tensor of external contact forces. That is, the second term on the right-hand side denotes body forces acting on Ω, and the last term denotes external forces acting on the surface $\partial\Omega$. Applying Gauss's theorem to the previous linear momentum Equation (2.4) then gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\mathbf{v}\,\mathrm{d}V + \int_{\Omega}\nabla\cdot(\rho\mathbf{v}\mathbf{v})\,d\mathrm{V} - \int_{\Omega}\rho\mathbf{b}\,\mathrm{d}V - \int_{\Omega}\nabla\cdot\mathbf{T}\,\mathrm{d}V = 0. \tag{2.5}$$

Applying the Leibniz integral rule gives

$$\int_{\Omega}\left[\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) - \rho\mathbf{b} - \nabla\cdot\mathbf{T}\right]\mathrm{d}V = 0. \tag{2.6}$$

Again, similar to the mass continuity equation, since Ω is arbitrary, this simplifies to

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) - \rho\mathbf{b} - \nabla\cdot\mathbf{T} = 0, \tag{2.7}$$

which is the canonical momentum conservation equation for a compressible fluid [44].

2.2. The Navier-Stokes Equations

The Navier-Stokes equations are a set of PDEs that accurately describe the motion of viscous fluids. They are used to model a variety of important phenomena in science and engineering, including water flow through a pipe and air flow around an obstacle [100], [106]. To obtain these PDEs from Equations (2.3) and (2.7), various assumptions are made concerning the fluid, the fluid's density ρ, the body forces $\mathbf{b}$, and the stress tensor $\mathbf{T}$. To begin, we first assume that gravity is the only relevant body force $\mathbf{b}$, i.e.,

$$\rho\mathbf{b} = \rho\mathbf{g}, \tag{2.8}$$

where $\mathbf{g}$ is the gravitational force that acts upon the fluid. We also assume that the fluid is Newtonian (as is typical of water and air, for example), which implies the relationship

$$\mathbf{T} := -p\mathbf{I} + \bar{\mathbf{T}}, \tag{2.9}$$

where p is termed the "static-fluid pressure" and $\bar{\mathbf{T}} \in \mathbb{R}^{3\times 3}$ is the viscous stress tensor. Finally, we assume that the flow is incompressible. This implies ρ is constant within a small element volume $\mathrm{d}V$ that moves with the flow velocity. With the given assumptions, the three-dimensional (3D) Navier-Stokes equations are then traditionally written as

$$\nabla \cdot \mathbf{v} = 0, \tag{2.10a}$$

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) = -\nabla p + \rho\mathbf{g} + \nabla \cdot \bar{\mathbf{T}}. \tag{2.10b}$$

Expanding the previous notation over all four spatiotemporal dimensions gives

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{2.11a}$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} = \frac{\partial(\tau_{xx} - p)}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z}, \tag{2.11b}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} = \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial(\tau_{yy} - p)}{\partial y} + \frac{\partial \tau_{yz}}{\partial z}, \tag{2.11c}$$

$$\frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho uw)}{\partial x} + \frac{\partial(\rho vw)}{\partial y} + \frac{\partial(\rho w^2)}{\partial z} = -\rho g + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial(\tau_{zz} - p)}{\partial z}. \tag{2.11d}$$

These PDEs constitute the general Cartesian form of the Navier-Stokes equations.

2.3. The Shallow Water Equations

The 2D shallow water equations are a set of PDEs derived from the Navier-Stokes equations under the assumption that horizontal length scales are much larger than the vertical scale. This is a reasonable assumption for modeling large-scale floods, where water depths are much smaller than typical flood wavelengths [71]. Prior to the PDEs' derivation, we introduce $h(x, y, t)$ to denote the total depth of the water column,

$B(x, y)$ to denote the bathymetric height below the water column, and $\eta(x, y, t) := B(x, y) + h(x, y, t)$ to denote the free surface elevation of the water column. We begin with four boundary conditions applied at the "bottom" surface, $z = B$, i.e.,

$$u(x, y, B, t) = v(x, y, B, t) = 0, \tag{2.12a}$$

$$u\frac{\partial B}{\partial x} + v\frac{\partial B}{\partial y} + w = 0, \tag{2.12b}$$

$$\tau_B^x - \tau_{xx}\frac{\partial B}{\partial x} - \tau_{xy}\frac{\partial B}{\partial y} - \tau_{xz} = 0, \tag{2.12c}$$

$$\tau_B^y - \tau_{yx}\frac{\partial B}{\partial x} - \tau_{yy}\frac{\partial B}{\partial y} - \tau_{yz} = 0. \tag{2.12d}$$

Here, Equation (2.12a) is the "no-slip" boundary condition, where adhesive forces between the fluid and the boundary are assumed to be greater than the cohesive forces of the fluid *at* the boundary. Equation (2.12b) is the "no normal flow" condition, which prohibits fluid from traveling *through* the boundary. Finally, Equations (2.12c) and (2.12d) are the "bottom shear stress" conditions, where τ_B^x and τ_B^y are predetermined.

As with the bottom, at the "free surface" ($z = \eta$), we apply three boundary conditions:

$$p(x, y, \eta, t) = p_a, \tag{2.13a}$$

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} - w = 0, \tag{2.13b}$$

$$\tau_\eta^x + \tau_{xx}\frac{\partial \eta}{\partial x} + \tau_{xy}\frac{\partial \eta}{\partial y} - \tau_{xz} = 0, \tag{2.13c}$$

$$\tau_\eta^y + \tau_{yx}\frac{\partial \eta}{\partial x} + \tau_{yy}\frac{\partial \eta}{\partial y} - \tau_{yz} = 0. \tag{2.13d}$$

In this set of conditions, Equation (2.13a) enforces constant atmospheric pressure, p_a, at the fluid's surface, Equation (2.13b) prohibits relative flow that is normal to the surface, and Equations (2.13c) and (2.13d) model the horizontal shear stresses at the surface.

Equation (2.11d) represents the Navier-Stokes z-momentum equation. Using the assumptions that the vertical velocity is small and that only the pressure derivative and

gravitational terms are substantial, the z-momentum equation then reduces to

$$\frac{\partial p}{\partial z} = \rho g. \tag{2.14}$$

This implies that

$$p = g \int_z^\eta \rho \, \mathrm{d}z = \rho g(\eta - z) + p_a. \tag{2.15}$$

This is the distribution of hydrostatic pressure for a column of water. It follows that

$$\frac{\partial p}{\partial x} = \rho g \frac{\partial \eta}{\partial x}, \tag{2.16a}$$

$$\frac{\partial p}{\partial y} = \rho g \frac{\partial \eta}{\partial y}. \tag{2.16b}$$

We have assumed that vertical velocity is small and that the exchange of vertical momentum is negligible. Furthermore, we know the pressure increase is linear with depth (making the flows along a column parallel). Thus, we can now "depth-average" the system of equations to remove a dimension, i.e., integrate everything from $z = B$ to η. Integrating the Navier-Stokes continuity Equation (2.11a) from $z = B$ to $z = \eta$ gives

$$\int_B^\eta \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \mathrm{d}z + w\big|_{z=\eta} - w\big|_{z=B} = 0. \tag{2.17}$$

Applying the Leibniz integral rule to Equation (2.17) gives

$$\begin{aligned} \frac{\partial}{\partial x} \int_B^\eta u \, \mathrm{d}z + \frac{\partial}{\partial y} \int_B^\eta v \, \mathrm{d}z - \left(u\big|_{z=\eta} \frac{\partial \eta}{\partial x} - u\big|_{z=B} \frac{\partial B}{\partial x} \right) - \\ \left(v\big|_{z=\eta} \frac{\partial \eta}{\partial y} - v\big|_{z=B} \frac{\partial B}{\partial y} \right) + w\big|_{z=\eta} - w\big|_{z=B} = 0. \end{aligned} \tag{2.18}$$

Additionally, the *depth-averaged* horizontal velocities, $\bar{u}$ and $\bar{v}$ are defined as

$$\bar{u} = \frac{1}{h} \int_B^\eta u \, \mathrm{d}z \implies h\bar{u} = \int_B^\eta u \, \mathrm{d}z, \tag{2.19}$$

$$\bar{v} = \frac{1}{h} \int_B^\eta v \, \mathrm{d}z \implies h\bar{v} = \int_B^\eta v \, \mathrm{d}z. \tag{2.20}$$

Finally, applying boundary conditions to the continuity Equation (2.18) gives

$$\frac{\partial h}{\partial t} + \frac{\partial (h\bar{u})}{\partial x} + \frac{\partial (h\bar{v})}{\partial y} = 0. \tag{2.21}$$

Next, we simplify the x-momentum Equation (2.11b) for the Navier-Stokes equations. Assuming that density ρ is constant and integrating the left-hand side over depth gives

$$\int_B^\eta \left[\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} \right] \mathrm{d}z = \frac{\partial (h\bar{u})}{\partial t} + \frac{\partial (h\bar{u}^2)}{\partial x} + \frac{\partial (h\bar{u}\bar{v})}{\partial y} + \cdots, \tag{2.22}$$

where the latter $(\cdots)$ are differential advective terms, often neglected in geophysical modeling [55]. Integrating the right-hand side over depth and recalling $\frac{\partial p}{\partial x} = \rho g \frac{\partial \eta}{\partial x}$ gives

$$-\rho g h \frac{\partial \eta}{\partial x} + \tau_\eta^x - \tau_B^x + \frac{\partial}{\partial x} \int_B^\eta \tau_{xx} + \frac{\partial}{\partial y} \int_B^\eta \tau_{xy}. \tag{2.23}$$

Note that a similar procedure may also be applied to the y-momentum equation.

Finally, combining the continuity, x-momentum, and y-momentum equations gives

$$\frac{\partial h}{\partial t} + \frac{\partial (h\bar{u})}{\partial x} + \frac{\partial (h\bar{v})}{\partial y} = 0, \tag{2.24a}$$

$$\frac{\partial (h\bar{u})}{\partial t} + \frac{\partial (h\bar{u}^2)}{\partial x} + \frac{\partial (h\bar{u}\bar{v})}{\partial y} = -gh\frac{\partial \eta}{\partial x} + \frac{1}{\rho}\left[\tau_\eta^x - \tau_B^x + F_x\right], \tag{2.24b}$$

$$\frac{\partial (h\bar{v})}{\partial t} + \frac{\partial (h\bar{u}\bar{v})}{\partial x} + \frac{\partial (h\bar{v}^2)}{\partial y} = -gh\frac{\partial \eta}{\partial y} + \frac{1}{\rho}\left[\tau_\eta^y - \tau_B^y + F_y\right]. \tag{2.24c}$$

In the case of 2D flood modeling, which is considered in Chapters 3 and 4, F_x and F_y are typically neglected. Note that there are several practical benefits of modeling floods using the 2D shallow water equations over the full 3D Navier-Stokes equations. Most importantly, the model requires only two spatial dimensions, providing a substantial computational advantage. Furthermore, the equations are easily parameterized using initial conditions that can be derived from readily available data, e.g., the bathymetric profile $B(x, y)$, surface stress models $\tau_\eta^{x,y}(x, y)$, and local bed friction $\tau_B^{x,y}(x, y)$.

2.4. The Euler Equations

The Euler equations are an approximation of the Navier-Stokes equations under the assumption that the fluid viscosity is negligible (or inviscid). This implies that all elements of the viscous stress tensor, $\bar{\mathbf{T}}$, in Equation (2.9) are equal to zero, yielding

$$\nabla \cdot \mathbf{v} = 0, \tag{2.25a}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla p + \rho \mathbf{g}. \tag{2.25b}$$

Then, assuming the fluid's density ρ is constant and spatially uniform yields

$$\nabla \cdot \mathbf{v} = 0, \tag{2.26a}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{v}) = -\frac{1}{\rho}\nabla p + \mathbf{g}. \tag{2.26b}$$

These are the incompressible Euler equations with constant and uniform density. Finally, assuming that velocity is constant over time, factoring the momentum equation using the result of the continuity equation, and expanding vector notation then yields

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{2.27a}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{2.27b}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial v}{\partial y}, \tag{2.27c}$$

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g, \tag{2.27d}$$

which are the steady-state incompressible Euler equations in three spatial dimensions.

2.5. Bernoulli's Equation

Bernoulli's principle in fluid dynamics states that an increase in fluid speed occurs simultaneously with a decrease in static pressure or potential energy. To obtain Bernoulli's equation, we are interested in integrating Equations (2.27b)-(2.27d) over a *streamline*,

i.e., the path that a fluid element will travel within a fixed velocity field. Applying a number of differential transformations and integrating over the streamline then yields

$$p_1 + \frac{1}{2}\rho|\mathbf{v}_1|^2 + \rho g z_1 = p_2 + \frac{1}{2}\rho|\mathbf{v}_2|^2 + \rho g z_2, \tag{2.28}$$

where the subscripts 1 and 2 correspond to streamline positions $\mathbf{x}_1$ and $\mathbf{x}_2$. Equation (2.28) is the traditional form of Bernoulli's equation without energy losses. In practice, the friction between the material containing the fluid and the fluid itself contributes to losses in energy. This gives rise to an empirical extension of the Bernoulli principle,

$$\Delta h = \left(\frac{|\mathbf{v}_1|^2}{2g} + \frac{p_1}{\rho g} + z_1\right) - \left(\frac{|\mathbf{v}_2|^2}{2g} + \frac{p_2}{\rho g} + z_2\right) \approx \left(\frac{p_1}{\rho g} + z_1\right) - \left(\frac{p_2}{\rho g} + z_2\right), \tag{2.29}$$

which is appropriately termed the "energy equation." Here, Δh is the loss in total hydraulic head (or "head") from friction. When analysis is limited to flow through a pipe with constant cross-sectional area, the velocities are approximately equal, and the right-hand side is used instead. However, Equation (2.29) is not predictive unless all variables on the right-hand side are known a priori. This would require detailed knowledge of the relationship between the pressures in a pipe at a specific flow rate [21].

2.6. Incompressible Flow of Water Through a Pipe

Engineering design requires a relationship that predicts Δh as a function of the fluid, velocity, pipe material, and pipe diameter. Engineering efforts to determine such an empirical relationship appeared to have begun in earnest around 1770 with Antoine Chézy and culminated in 1845 with Julius Weisbach, who proposed the relationship

$$\Delta h = \frac{fL}{D}\frac{v^2}{2g}, \tag{2.30}$$

where f is the "friction factor," L is the pipe length, D is the pipe diameter, and v is the fluid velocity. In 1857, Henry Darcy proposed an accurate method for determining f as a function of pipe roughness and pipe diameter, which was then continually improved as a function of the fluid's Reynolds number over the twentieth century [21]. The combination of Equation (2.30) and Darcy's findings on f gave the equation its moniker, the Darcy-

Weisbach equation. The form of the equation used throughout this dissertation is

$$\Delta h = \frac{8Lfq|q|}{\pi^2 g D^5}, \tag{2.31}$$

where q is the volumetric flow rate (or flow) and Δh is directed based on the sign of q.

For all of its empirical advantages, the Darcy-Weisbach equation did not garner widespread approval among engineering practitioners until after publication of the Moody diagram in 1944. This diagrammatic approach allows for the easier selection of f depending on properties of the fluid and pipe, which aided engineers prior to the Computer Age. Before this time, and even up until the present day, engineers have used an empirical approximation to the Darcy-Weisbach equation, termed the Hazen-Williams equation:

$$\Delta h = \frac{10.7Lq|q|^{0.852}}{\kappa^{1.852} D^{4.8704}}, \tag{2.32}$$

where 10.7 is a constant in standard units and κ is the pipe's roughness, which depends on material properties. Note that unlike f in Equation (2.31), κ in Equation (2.32) is instead treated as a fixed constant, which certainly aided in its adoption throughout the earlier twentieth century. However, this constant approximation also implies greater limitations: the equation is typically only applicable to water in a turbulent flow regime, and past studies on its applicability have suggested the equation is limited to cases in which the pipe diameter is larger than 5 cm and the velocity is less than 3 m/s [101].

In this dissertation, the incompressible head loss Equations (2.31) and (2.32) are both considered. Since the application in which they're used typically satisfies the turbulence, pipe size, and fluid velocity assumptions, i.e., steady-state potable water distribution system optimization, the Hazen-Williams approximation is considered valid in this context. However, because the assumption of a friction factor f that depends on the fluid's Reynolds number entails greater nonlinearity in the Darcy-Weisbach equation, f is often assumed to be fixed in optimization applications [61], [140]. Using the assumption of a constant f, when all terms but Δh and q are fixed, both head loss forms reduce to

$$\Delta h = Lrq\,|q|^{\alpha-1}\,. \tag{2.33}$$

Here, α is the exponent required by Equation (2.31) or (2.32) (i.e., 2 or 1.852, respec-

tively), and r is the resistance per unit length. The resistance per unit length comprises all non-length constants in Equations (2.31) and (2.32) and is in units of $(\mathrm{m}^3/\mathrm{s})^{-\alpha}$. We consider this standard, general form of pipe head loss throughout this dissertation.

2.7. Incompressible Flow of Gas Through a Pipe

Consider a 2D variant of Equation (2.29) where the pressure p at a distance x from the inlet of a pipe changes to a pressure $p + \mathrm{d}p$ at a distance $x + \mathrm{d}x$ from the inlet of the pipe. Similarly, let z and the velocity u also change across the differential element $\mathrm{d}x$. Assuming the fluid density is constant, Bernoulli's energy equation is then written as

$$\mathrm{d}h = \left(\frac{u^2}{2g} + \frac{p}{\rho g} + z\right) - \left(\frac{(u+\mathrm{d}u)^2}{2g} + \frac{p+\mathrm{d}p}{\rho g} + z + \mathrm{d}z\right). \tag{2.34}$$

In a gas pipeline, the head loss due to friction across the element $\mathrm{d}x$ is accurately modeled using a differential form Weisbach's relationship in Equation (2.30), namely

$$\mathrm{d}h = \frac{f}{D}\frac{u^2}{2g}\mathrm{d}x. \tag{2.35}$$

Assuming any change in velocity is small, Equation (2.34) can then be rewritten as

$$-\mathrm{d}p = \frac{f\rho u^2}{2D}\mathrm{d}x + \rho g \mathrm{d}z. \tag{2.36}$$

Assuming mass flow continuity along the pipe, i.e., $\rho u = \rho_1 u_1$, it can be deduced that $u = \frac{p_1}{p}u_1$ and $\rho = \frac{p}{p_1}\rho_1$. Substituting these relationships into Equation (2.36) gives

$$-p\mathrm{d}p = \frac{f}{2D}\rho_1 p_1 u_1^2 \mathrm{d}x + \frac{p^2}{p_1}\rho_1 g \mathrm{d}z. \tag{2.37}$$

The thermodynamical equation of state for a gas that relates pressure and density is

$$p_1 = ZRT\rho_1, \tag{2.38}$$

where Z is the gas compressibility factor, R is the individual gas constant, and T is the

average temperature. This relationship allows for the rewriting of Equation (2.37) as

$$-p\mathrm{d}p = \frac{f}{2D}\rho_1^2 u_1^2 ZRT\mathrm{d}x + \frac{p^2}{ZRT}g\mathrm{d}z. \tag{2.39}$$

Letting n denote quantities at standard pressure and temperature, continuity guarantees

$$\rho_1 u_1 = \rho_n u_n = \frac{\rho_n^2 q_n^2}{(\pi D^2/4)^2}, \tag{2.40}$$

where q_n is the volumetric flow rate at standard conditions. Equation (2.39) reduces to

$$-p\mathrm{d}p = \frac{8ZRTf\rho_n^2}{\pi^2 D^5}q_n^2\mathrm{d}x + \frac{p_{\mathrm{av}}^2}{ZRT}g\mathrm{d}z, \tag{2.41}$$

where the pressure in the elevation term is assumed to be the average pressure, p_{av}. Next, the gas constant R is related to the gas constant for air, R_{air}, via $R = R_{\mathrm{air}}/S$, where S is termed the *specific gravity* of the gas, and where $S = \rho_n/\rho_{\mathrm{air},n}$. This implies $\rho_n = \frac{Sp_n}{R_{\mathrm{air}}T_n}$. Substitution of this relationship for ρ_n within Equation (2.41) then yields

$$-p\mathrm{d}p = \frac{8ZSTf}{\pi^2 R_{\mathrm{air}}D^5}q_n^2\left(\frac{p_n}{T_n}\right)^2\mathrm{d}x + \frac{p_{\mathrm{av}}^2 S}{ZR_{\mathrm{air}}T}g\mathrm{d}z. \tag{2.42}$$

Finally, integrating this equation from $x = 0$, $p = p_1$ to $x = L$, $p = p_2$ then yields

$$p_1^2 - p_2^2 = \frac{16LZSTf}{\pi^2 R_{\mathrm{air}}D^5}q_n^2\left(\frac{p_n}{T_n}\right)^2 + \frac{2p_{\mathrm{av}}^2 S}{ZR_{\mathrm{air}}T}gh, \tag{2.43}$$

which is a rewriting of the canonical *general flow equation* for steady-state gas flow [102].

Assuming a horizontal pipe, the elevation term is zero, and Equation (2.43) becomes

$$p_1^2 - p_2^2 = \frac{16LZSTf}{\pi^2 R_{\mathrm{air}}D^5}q_n^2\left(\frac{p_n}{T_n}\right)^2. \tag{2.44}$$

Letting μ_n denote the mass flow rate (i.e., $\rho_n q_n$), Equation (2.44) can be rewritten as

$$p_1^2 - p_2^2 = \frac{16LZR_{\mathrm{air}}Tf}{\pi^2 SD^5}\mu_n^2 \implies p_1^2 - p_2^2 = \frac{16LZRTf}{\pi^2 D^5}\mu_n^2. \tag{2.45}$$

In practice, however, the gas industry uses several approximations of this equation, primarily based on assumptions of the friction factor f. One of the most common approximations is the Weymouth equation, which is typically only valid in high-pressure gas transmission networks that operate in the fully-turbulent flow regime. It assumes

$$\sqrt{\frac{1}{f}} = CD^{1/6}E \implies f = (CD^{1/6}E)^{-2}, \tag{2.46}$$

where C is a constant and E is the pipe's *efficiency factor*, which typically varies between 0.8 and one, depending on pipe's roughness. Substitution in Equation (2.45) yields [102]

$$p_1^2 - p_2^2 = \frac{16LZRT}{\pi^2 C^2 D^{16/3} E^2}\mu_n^2. \tag{2.47}$$

Other approximations are based on the Colebrook-White equation, which is an empirical relation that expresses f as a function of the fluid's Reynolds number and the pipe's relative roughness. In the turbulent regime, f can be computed via an *explicit* approximation of the Colebrook-White equation for turbulent flow, presented as [148]

$$f = \left[2\log\left(\frac{3.7D}{\epsilon}\right)\right]^{-2}, \tag{2.48}$$

where ϵ is used to denote the absolute roughness of the pipe (in units of length).

No matter the approximation used for f, the general form of the pressure-flow relationship for modeling losses in high-pressure gas pipelines is conveniently written as

$$p_1^2 - p_2^2 = Lr\mu_n^2, \tag{2.49}$$

where r is the resistance per unit length of the pipe. Similar to the head loss Equation (2.33) for water flow through a pipe, the resistance per unit length comprises all non-length constant terms in Equation (2.45) and is defined a priori in units of $\mathrm{m}^{-3}\mathrm{s}^{-2}$. We consider this the standard form of gas pipeline pressure loss throughout this dissertation.

Chapter 3

Optimization of Structural Flood Mitigation Strategies Using Mixed-integer Linear Approximations

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 resulted in tremendous human suffering and economic consequences. While the causes of these disasters vary (e.g., hurricanes, dam failures, or excessive rainfall), all are characterized by the phenomenon of flooding, i.e., the undesired flow of water into areas that are usually not submerged. As a result of the frequent and negative outcomes associated with floods, societies have historically invested considerable resources into controlling and preventing their occurrence. Despite these efforts, mitigation decisions can often be viewed as suboptimal in hindsight. As such, the optimal management of flood risk continues to be a perennial and ever-evolving topic of great importance [49], [103], [107].

Modern flood risk management (FRM) is a continuous process of identifying issues, defining objectives, assessing risks, appraising options, implementation, monitoring, and review. Within this framework, risk assessment is regarded as a cyclic process that includes the design and evaluation of alternative management strategies. Such strategies commonly include both "hard" and "soft" structural mitigation measures, e.g., the construction of dams (hard) and wetland storage (soft) [122]. Measures can also be

temporary (e.g., sandbags) or permanent (e.g., levees). However, for complex scenarios, the number of feasible strategies is extremely large and practically difficult to explore. As such, the manual design and assessment of these strategies, whether conducted in a real-world or simulation-based setting, can be time-consuming and expensive. This limitation may result in vastly suboptimal FRM strategies. To aid in the FRM process, an optimization-based decision support approach for proposing structural mitigation designs can serve as a useful tool within the overall risk assessment phase of FRM.

To this end, this chapter, which is based on [132], introduces an optimization-based technique for proposing the placement of hard structural mitigation measures. The formulation of the problem begins from an intuitive understanding of flood propagation. Specifically, one of the most influential factors in flooding is the shape of the ground surface (or topography). As an example, under the influence of gravity, water naturally flows downhill and around areas of higher topographic elevation. Topography can, of course, be adjusted through the construction of permanent or temporary structural measures. This chapter thus proposes the OFMP, a difficult optimization problem that aims at mitigating a flood by adjusting topographic elevation. Its goal is to select the positioning of hard structural measures, e.g., sandbags or levees, to protect predefined locations of critical assets and/or to enable the evacuation of threatened populations.

The OFMP is an inherently difficult optimization problem. Since structural barriers divert flow, it is critical to accurately model the flood's propagation, which is well-captured by hydrodynamic simulations that solve the 2D shallow water equations. As described in Section 2.3, these PDEs express mass and momentum conservation along two horizontal dimensions at every point in space and time. In practice, these PDEs are *discretized* over space and time, resulting in a set of nonlinear equations of high dimensionality. In addition, the OFMP aims at choosing the position of barriers in space, introducing additional sources of nonconvexity and combinatorial challenges. However, unlike many control-related optimization problems, the OFMP optimizes only the initial conditions. Flood propagation is predetermined once initial conditions have been selected, i.e., there are limited opportunities to modify a flood's 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 when using current optimization technology.*

The primary novel contributions of this chapter are summarized as follows:

- The formalization of the OFMP problem, integrating simulation and optimization;
- The derivation of *linear* lower and upper approximations to 2D flood PDEs;
- The definition of optimization models for the OFMP using these approximations;
- Empirical results that highlight the accuracy and tractability of the approximations and demonstrate the potential of applying optimization technology in this area.

The derivation of linear approximations to flood propagation is a critical step in bringing the OFMP within the realm of optimization tractability. 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 small, contrived 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, aside from the work of this chapter and of Chapter 4, the closest related work is [76]. There, the authors propose an interdiction model for flood mitigation and develop flood surrogates from simulation data to serve as proxies for calculating flood responses to mitigation efforts. However, these surrogates do not define strict relationships with the original PDEs. In contrast, this chapter develops *dynamic approximations* of the flood PDEs.

A number of studies consider simulation-optimization approaches for reservoir operation, where the PDEs associated with the flood dynamics are treated as a black box. An extensive literature review of these studies can be found in [27]. The work described by [38] considers the full PDEs, but their focus is on optimizing normal operations of an open-channel system. Finally, the problem of optimizing dike heights with uncertainty in flooding estimates is considered by [19]. However, in their study, the PDEs for flood propagation are not considered, and probability models for maximum flood depths are used in place of deterministic physical models. Our work considers deterministic models.

Chapter Overview The rest of this chapter is organized as follows: Section 3.1 discusses the background of flood modeling; Section 3.2 presents the linear flood relaxations; Section 3.3 introduces the OFMP and proposes an optimization model exploiting these relaxations; Section 3.4 gives empirical results; and Section 3.5 concludes the chapter.

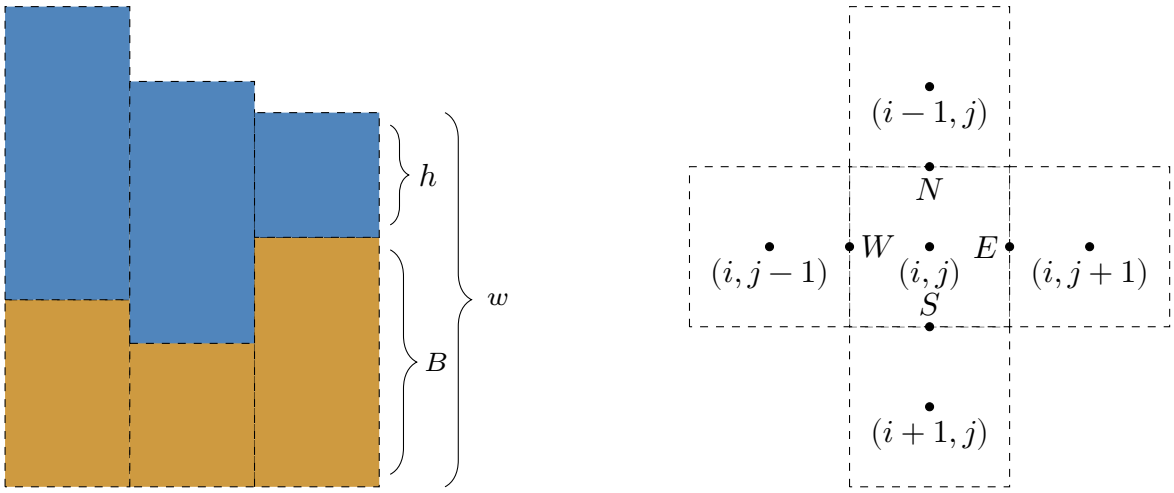
3.1. Background

The Two-dimensional Shallow Water Equations As described in Section 2.3, the 2D shallow water equations are a system of hyperbolic PDEs increasingly used to accurately model flooding phenomena. With advances in high-performance computing over the past two decades, numerical solutions to these equations have recently become tractable for large-scale simulation problems. They are especially useful in the context of urban flooding, where one-dimensional models often fail due to increased topographic complexity. Neglecting select terms of Equations (2.24) while including bottom slope, bottom friction, and volumetric sources, the 2D shallow water equations are written as

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} &= R(x,y,t), \\ \frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial (huv)}{\partial y} &= -gh\frac{\partial B}{\partial x} - \frac{\tau^x}{\rho}, \\ \frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) &= -gh\frac{\partial B}{\partial y} - \frac{\tau^y}{\rho}, \end{aligned} \tag{3.1}$$

where h is the water depth, u and v are horizontal velocities, B is the bottom topography (or bathymetric elevation), 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 [30]. 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 reasonable to consider more tractable model 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 [91]. 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} denotes the depth of the water in cell (i, j) at time index t, w_{ijt} denotes the water surface elevation, and B_{ij} denotes 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 flow rate (hereafter termed "flow"), q^k_{ijt}, which models the transport of water from cell (i, j) to its neighbor in position $k \in \{W, E, N, S\}$



(a) Vertical discretization. **(b) Horizontal discretization.**

Figure 3.1: Discretization of the pipe flow shallow water equations approximation. The pipe flow model is discretized using (a) columnar vertical components, with h denoting the water depth, B the topographic elevation, and w the net water surface elevation; and (b) two-dimensional horizontal components, where the behavior of quantities at the center cell (i, j) is dependent on adjacent cells and the four interfaces of (i, j).

at time index t. For example, q^W_{ijt} is the flow from (i, j) to $(i, j-1)$ at time index t.

In the model, the flow of a pipe is accelerated by the hydrostatic pressure difference between adjacent cells. The water volume V_{ij} of a cell is integrated using the accumulated flow from all connected pipes. This corresponds to a change in the cell's depth and water surface elevation. These concepts are illustrated pictorially in Figure 3.1.

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

$$\tilde{q}^k_{ijt} = \max\left(0, q^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}\Delta w^k_{ij,t-1}\right), \tag{3.2}$$

where A is the cross-sectional area of the pipe (assumed to be equal for all pipes), 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 modeling 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.,

$$\Delta w^k_{ijt} = (B_{ij} + h_{ijt}) - (B^k_{ij} + h^k_{ijt}). \tag{3.3}$$

In this approximation, the estimated outgoing flow from a cell may *exceed* the available water content within that cell. This is not desirable from the perspective of mass conservation. More importantly, if left uncorrected, this can lead to negative water depths and numerical instabilities. This can be resolved, however, by scaling the outgoing flow with respect to the water content available within a cell (i, j) at time t. To accomplish this, we introduce a scaling factor K_{ijt} for the outgoing flow, defined as

$$K_{ijt} = \min\left(1, \frac{h_{ij,t-1}\Delta x \Delta y}{\left(\tilde{q}^W_{ijt} + \tilde{q}^E_{ijt} + \tilde{q}^N_{ijt} + \tilde{q}^S_{ijt}\right)\Delta t}\right). \tag{3.4}$$

The estimated outgoing flow vector $\tilde{\mathbf{q}}_{ijt}$ is then scaled by K_{ijt} to produce the actual (i.e., volume-corrected) outgoing flow vector $\mathbf{q}_{ijt}$. This implies the scaling relationship

$$\mathbf{q}_{ijt} = K_{ijt}\tilde{\mathbf{q}}_{ijt}. \tag{3.5}$$

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

$$\begin{aligned}\Delta V_{ijt} &= \left(\sum q^{\text{in}}_{ijt} - \sum q^{\text{out}}_{ijt}\right)\Delta t \\ &= \left(q^E_{i,j-1,t} + q^W_{i,j+1,t} + q^N_{i+1,j,t} + q^S_{i-1,j,t} - \sum_{k\in\{W,E,N,S\}} q^k_{ijt}\right)\Delta t.\end{aligned} \tag{3.6}$$

Finally, the water depth in each cell is integrated using a conventional Euler step, i.e.,

$$h_{ijt} = h_{ij,t-1} + \frac{\Delta V_{ijt}}{\Delta x \Delta y}. \tag{3.7}$$

For completeness, we also apply the naive reflective boundary conditions

$$h_{ijt} = 0,\ \mathbf{q}_{ijt} = 0,\ \tilde{\mathbf{q}}_{ijt} = 0 \tag{3.8}$$

along the four boundaries (left, right, top, and bottom) of the spatial domain.

3.2. Linear Approximations of the Pipe Flow Model

The pipe flow model 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 (3.4). We next present two intuitive approximations to remove these nonlinearities. For convenience, we refer to them as the "lower" and "upper" approximations because they *underestimate* and *overestimate* the amount of water being sent from a cell to its neighbors instead of applying the corrective volume scaling factor K.

Lower Approximation The lower approximation is based on the following intuition: if the estimated outgoing flow from a cell exceeds the available water content within that cell, the outgoing flow is approximated as zero. That is, when the following is satisfied:

$$h_{ij,t-1}\Delta x\Delta y < \left(\tilde{q}^W_{ijt} + \tilde{q}^E_{ijt} + \tilde{q}^N_{ijt} + \tilde{q}^S_{ijt}\right)\Delta t, \tag{3.9}$$

$\mathbf{q}_{ijt}$ is approximated as zero. This bypasses the need for Equations (3.4) and (3.5). Intuitively, this approximation of $\mathbf{q}_{ijt}$ implies that, "when there is not enough water to be transported" from a cell, the water is *held back* within that cell, and so $\mathbf{q}_{ijt} = 0$.

Upper Approximation The upper approximation implements another intuitive idea: if the estimated outgoing flow from a cell exceeds the available water content within that cell, the model instead assumes *there is* enough water, and no scaling of the flow occurs. This again bypasses the need for the volume-corrective Equations (3.4) and (3.5).

It is important to note that, in the case of positive flows calculated as a result of differing dry topographies (and thus differing water surface elevations), Equations (3.4) and (3.5) provide an additional correction beside scaling. When the available water content within a cell is equal to zero, K_{ijt} is also zero, and the resultant flows $\mathbf{q}_{ijt}$ are thus zero. Although this correction is achieved automatically by the lower approximation, it is necessary to impose the constraint $\mathbf{q}_{ijt} = 0$ when $h_{ij,t-1} = 0$ in the upper approximation.

3.3. Optimal Flood Mitigation Problem

This section describes two optimization models based on the lower and upper pipe flow approximations, respectively. Both optimization models aim at protecting 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 overall topographic elevation. A fixed number of barriers, n, are available for that purpose. The models are highly similar, differing only in the pipe flow approximations used. We present them both to give a global view of the lower and upper approximations. Note that boundary conditions are omitted in the optimization models for simplicity.

Lower Approximation Optimization Model The lower approximation optimization model is presented in Model 3.1. The objective function in Line (3.12a) minimizes temporally maximal water depths over the set $\mathcal{A}$ of grid cells containing assets. Constraints (3.12b) and (3.12c) limit the number of barriers, n_{ij}, that may be placed in each cell. Each number must be no greater than M, the maximum allowable number of barriers per cell, as specified in Constraints (3.12b). The budget of barriers is limited by Constraint (3.12d). Constraints (3.12e) define each water surface elevation as the sum of topographic elevation (i.e., base elevation and barrier additions, each with height ΔB) and water depth. These are applied for every non-boundary cell ($\mathcal{X} \times \mathcal{Y}$) and time step $\{2, 3, \dots |\mathcal{T}|\}$. For conciseness, the Cartesian product of these sets is denoted by

$$\mathcal{F} := \mathcal{X} \times \mathcal{Y} \times \{2, 3, \dots |\mathcal{T}|\}. \tag{3.10}$$

Including the direction index set, a similar Cartesian product is denoted by the set

$$\mathcal{F}' := \mathcal{X} \times \mathcal{Y} \times \{2, 3, \dots |\mathcal{T}|\} \times \{W, E, N, S\}. \tag{3.11}$$

Constraints (3.12f) define estimated outgoing flow values, which must always be greater than or equal to zero. Constraints (3.12g) and (3.12h) define the outgoing flow values as prescribed by the lower approximation. Constraints (3.12i) provide convenient definitions for q^{in}_{ijt}, i.e., the sum of all incoming flow for each grid cell. Finally, the time integration of each water depth is defined using an Euler step in Constraints (3.12j).

Upper Approximation Optimization Model The upper approximation optimization model is presented in Model 3.2. Compared to the lower approximation model, the only differences are in Constraints (3.13g) and (3.13h), which ensure that outgoing flows are nonzero only when the water depth within a cell is greater than zero, and in Con-

Model 3.1 Lower approximation optimization model for the OFMP.

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} \max_{t\in\mathcal{T}}\{h_{ijt}\} \tag{3.12a}$$

$$\text{subject to} \quad n_{ij} \in [0, M],\ \forall (i,j) \in \mathcal{X} \times \mathcal{Y} \tag{3.12b}$$

$$n_{ij} = 0,\ \forall (i,j) \in \mathcal{A} \subset \mathcal{X} \times \mathcal{Y} \tag{3.12c}$$

$$\sum_{(i,j)\in\mathcal{X}\times\mathcal{Y}} n_{ij} = n \tag{3.12d}$$

$$w_{ijt} = (B_{ij} + n_{ij}\Delta B) + h_{ijt},\ \forall (i,j,t) \in \mathcal{F} \tag{3.12e}$$

$$\tilde{q}^k_{ijt} = \max\left(0, q^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}(w_{ij,t-1} - w^k_{ij,t-1})\right),\ \forall (i,j,t) \in \mathcal{F}' \tag{3.12f}$$

$$q^k_{ijt} = \tilde{q}^k_{ijt} \ \text{ if } \ h_{ij,t-1}\Delta x\Delta y \geq \Delta t \sum_k \tilde{q}^k_{ijt},\ \forall (i,j,t) \in \mathcal{F}' \tag{3.12g}$$

$$q^k_{ijt} = 0 \ \text{ if } \ h_{ij,t-1}\Delta x\Delta y < \Delta t \sum_k \tilde{q}^k_{ijt},\ \forall (i,j,t) \in \mathcal{F}' \tag{3.12h}$$

$$q^{\text{in}}_{ijt} = q^E_{i,j-1,t} + q^W_{i,j+1,t} + q^N_{i+1,j,t} + q^S_{i-1,j,t},\ \forall (i,j,t) \in \mathcal{F} \tag{3.12i}$$

$$h_{ijt} = h_{ij,t-1} + \Delta t \frac{q^{\text{in}}_{ijt} - \sum_k q^k_{ijt}}{\Delta x \Delta y},\ \forall (i,j,t) \in \mathcal{F} \tag{3.12j}$$

straints (3.13j), which ensure nonnegative depths. That is, if the predicted flow results in a transfer of water greater than what is contained within a cell, its depth is zero.

3.4. Computational Experiments

This section reports empirical results regarding the proposed flood approximations and the associated optimization models. First, Section 3.4.1 evaluates the accuracy of the various pipe flow approximations of the 2D shallow water equations in a simulation setting. Then, Section 3.4.2 explores the potential of using optimization to determine optimal flood mitigation strategies, as well as the computational challenges encountered.

3.4.1. 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 mod-

Model 3.2 Upper approximation optimization model for the OFMP.

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} \max_{t\in\mathcal{T}}\{h_{ijt}\} \tag{3.13a}$$

$$\text{subject to} \quad n_{ij} \in [0, M],\ \forall (i,j) \in \mathcal{X} \times \mathcal{Y} \tag{3.13b}$$

$$n_{ij} = 0,\ \forall (i,j) \in \mathcal{A} \subset \mathcal{X} \times \mathcal{Y} \tag{3.13c}$$

$$\sum_{(i,j)\in\mathcal{X}\times\mathcal{Y}} n_{ij} = n \tag{3.13d}$$

$$w_{ijt} = (B_{ij} + n_{ij}\Delta B) + h_{ijt},\ \forall (i,j,t) \in \mathcal{F} \tag{3.13e}$$

$$\tilde{q}^k_{ijt} = \max\left(0, q^k_{ij,t-1} + \frac{Ag\Delta t}{\Delta s}(w_{ij,t-1} - w^k_{ij,t-1})\right),\ \forall (i,j,t) \in \mathcal{F}' \tag{3.13f}$$

$$q^k_{ijt} = \tilde{q}^k_{ijt} \ \text{ if } \ h_{ij,t-1} > 0,\ \forall (i,j,t) \in \mathcal{F}' \tag{3.13g}$$

$$q^k_{ijt} = 0 \ \text{ if } \ h_{ij,t-1} \leq 0,\ \forall (i,j,t) \in \mathcal{F}' \tag{3.13h}$$

$$q^{\text{in}}_{ijt} = q^E_{i,j-1,t} + q^W_{i,j+1,t} + q^N_{i+1,j,t} + q^S_{i-1,j,t},\ \forall (i,j,t) \in \mathcal{F} \tag{3.13i}$$

$$h_{ijt} = \max\left(0, h_{ij,t-1} + \Delta t \frac{q^{\text{in}}_{ijt} - \sum_k q^k_{ijt}}{\Delta x \Delta y}\right),\ \forall (i,j,t) \in \mathcal{F} \tag{3.13j}$$

els. 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 [117]. 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, [91] does not necessarily recommend using constant cross-sectional pipe areas nor constant time steps. However, this chapter's intent is to simplify the 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. Note that this is different from evacuation settings, where estimates of the flood arrival time at various locations may be critical. Figure 3.2 compares images of maximum depth results from a 2D shallow water equations model (**SWE**) similar to [30], as well as pipe flow (**P**), lower approximation (**L**),

and upper approximation (**U**) models, which use various pipe areas and time steps.

The top row of Figure 3.2 compares **SWE** with **P**, **L**, and **U** using a parameterization calibrated to minimize the root-mean-square error between **P** and **SWE**. Here, **P** and **L** appear to be good approximators of **SWE**, but **U** overestimates maximum flood depths. This is because, in **U**, the large pipe area of 500 m^2 results in unrestricted large flows and poor volume conservation. In the second row, the pipe area is substantially decreased, and the pipe flow and lower approximation models overestimate **SWE**, although the approximation of **U** appears to improve. Finally, in the third row, as Δt is decreased, **U** begins to converge upon **P** and **L**. Most model parameterizations provide similar simulated flood extents when compared to those found in the literature [78], [117].

Finally, Figure 3.3 reports volume conservation error for selected upper approximation parameterizations. As anticipated, the pipe flow and lower approximation models conserve volume well, with volume conservation error on the order of machine epsilon. The upper approximation accumulates error more rapidly, although it displays good convergence as the time step is decreased, similar to the behavior shown in Figure 3.2.

It is important to note a unique difference between pipe flow simulations and 2D hydrodynamic simulations based on the shallow water equations. When using a 2D shallow water model, the Taum Sauk scenario can be 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 3.2 results in fast propagation, i.e., the flood is fully propagated in less than an hour. The smaller pipe areas used in the second and third rows result in slower propagation, i.e., a time extent of roughly three hours is required. In general, as the cross-sectional pipe area decreases, a longer time extent is required for full propagation. Nonetheless, since flood mitigation is primarily concerned with protecting assets, and thus maximum water depths, we consider differences in propagation speed acceptable for this application.

3.4.2. The Potential of Optimization

This section examines a number of small case studies to highlight the potential and challenges in using optimization to determine flood mitigation strategies. More generally, the section highlights these properties for the integration of simulation and optimization.

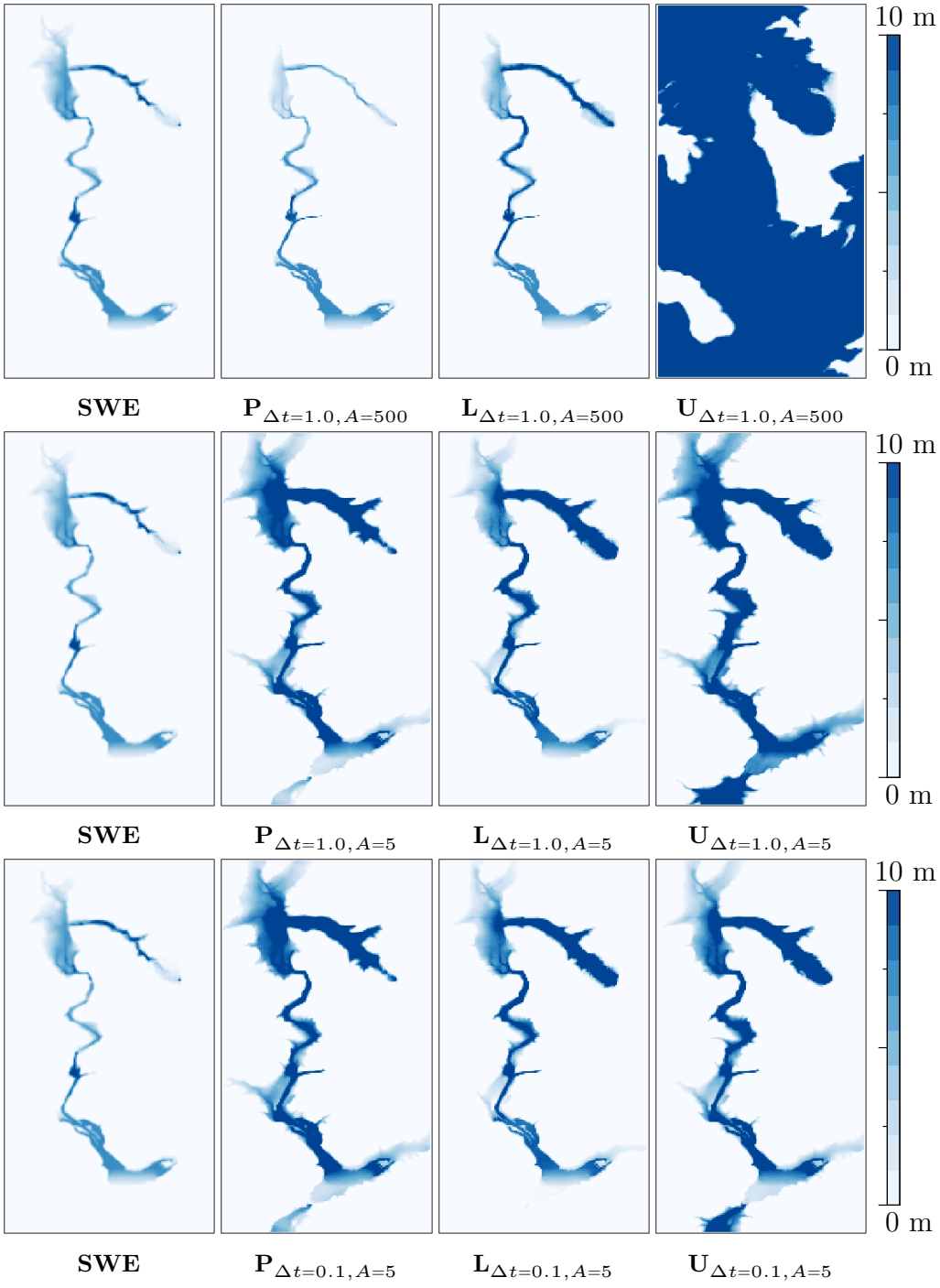


Figure 3.2: Comparison of pipe flow model accuracy on the Taum Sauk dam failure. Here, maximum depths are shown for ten-hour simulations of the dam failure using shallow water equations (SWE), pipe flow (P), lower approximation (L), and upper approximation (U) models.

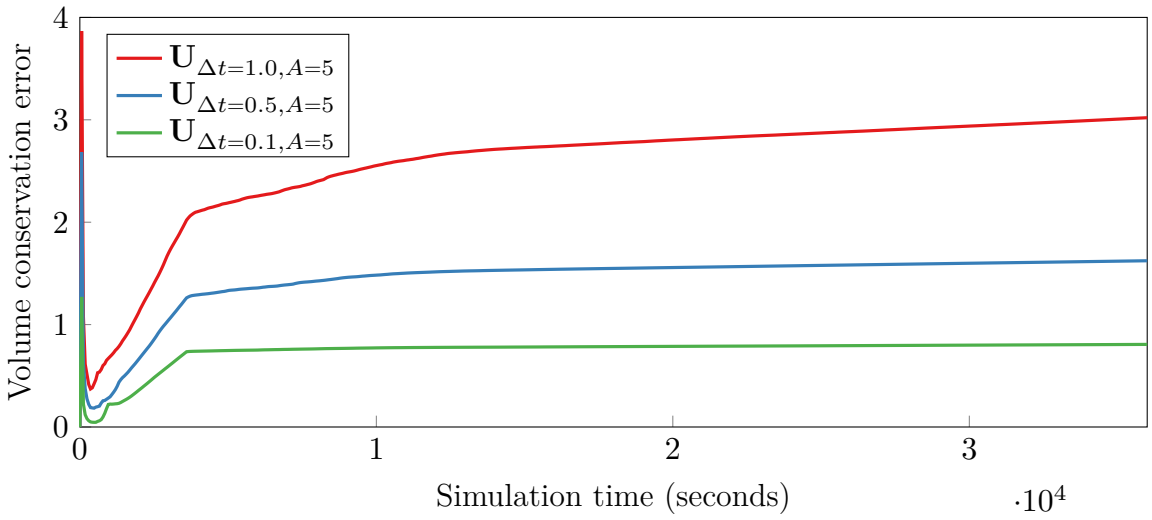


Figure 3.3: Volume conservation error for upper pipe flow approximation models. Ten hour simulations of the Taum Sauk dam break were considered using various time steps. Error is computed as $(V_{\mathbf{computed}} - V_{\mathbf{added}})/V_{\mathbf{added}}$.

Experimental Setting The lower and upper approximation MILP models were implemented using the C++ CPLEX interface and executed on twenty Intel Xeon E5-2660 v3 cores at 2.60 GHz, with 128 GB of memory. Conditional expressions and min/max functions were reformulated using big-M transformations. In our MILP implementation, no attempt was made to further strengthen formulations or exploit problem structure.

A Simple Case Study To validate the optimization model, an 8×8 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. Under the influence of gravity and in the presence of the topographic gradient, the water was forced down the spatial 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.

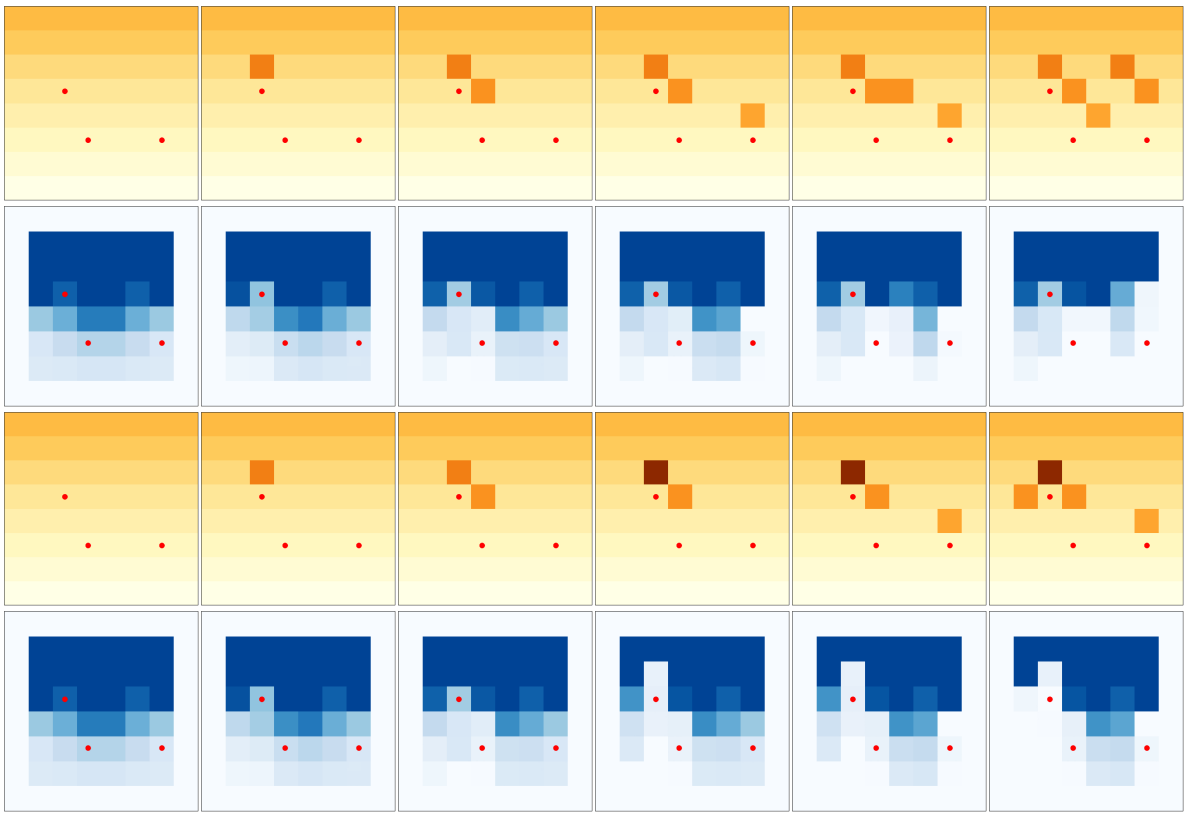


Figure 3.4: Optimal flood mitigation results using the lower pipe flow approximation. Optimal elevation fields and maximum depths are depicted. The allowable number of barriers per cell is one (first two rows) and two (bottom two rows), and the total number of barriers ranges from zero to five. Darker orange and blue colors correspond to larger topographic elevations and maximum depths, respectively. Red circles correspond to asset grid cells $(i, j) \in \mathcal{A}$ that are to be protected.

Optimal Asset Protection Figure 3.4 displays optimization results from the lower optimization models. Observe that, when only one barrier is allowed per cell, the optimization model tries 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 are allowed per cell, it clearly becomes preferential to protect the topmost asset, which receives a large amount of water over the duration of the simulation. Figure 3.5 displays optimization results from the upper optimization models. These show similarly interesting outcomes. In the one barrier per cell case, the optimization decides to protect the topmost asset less in favor of protecting the bottom assets. When two barriers are

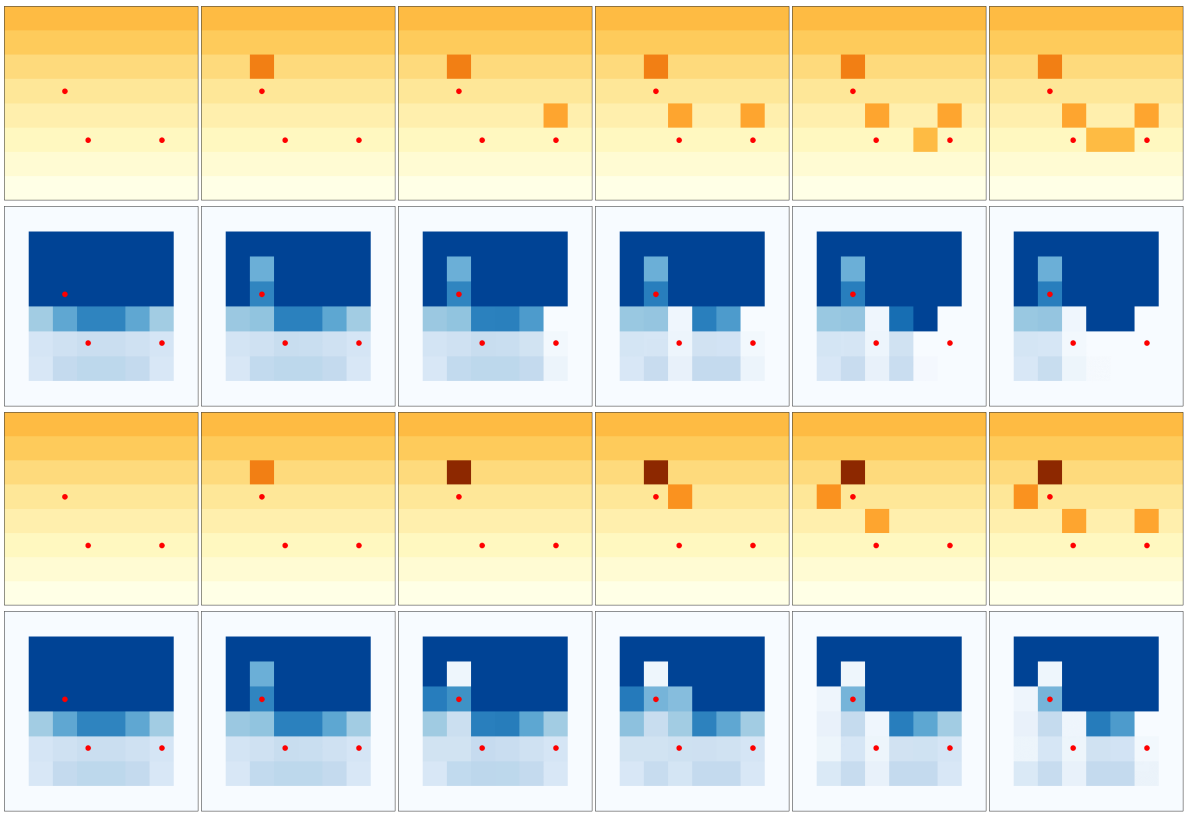


Figure 3.5: Optimal flood mitigation results using the upper pipe flow approximation. In this figure, the same setting as Figure 3.4 is assumed.

allowed per cell and enough barriers are available, it is beneficial to protect the topmost asset as much as possible from the water above it, which reduces the objective value.

It is also interesting to observe the nonincremental behavior of the optimization results. Allowing more barriers sometimes changes their optimal positioning. For example, this is the case when moving from three to four barriers in the bottom row. Furthermore, since the barrier placements sometimes differ in both models, it is important to study how the strategies affect flooding using the other model. These results are shown in the last two columns of Table 3.1. Column η^* gives the optimal solutions, and the last column describes the objective value obtained when the optimal solution of the upper optimization model was used in the lower optimization model and vice versa. This column sometimes shows significant differences in objective values. In practice, solutions could be evaluated using full 2D shallow water model simulations for various scenarios.

Model	t_{CPU} (s)	n_{nodes}	n_{var}	n_{con}	n_{bin}	η^* (m)	η_{com} (m)
$\mathbf{L}_3^1$	49.65	37,808	2,280	4,410	840	0.111607	0.134935
$\mathbf{L}_4^1$	83.66	71,528	2,280	4,410	840	0.0878997	0.123499
$\mathbf{L}_5^1$	91.53	66,485	2,280	4,410	840	0.0837807	0.112588
$\mathbf{L}_3^2$	82.74	86,964	2,646	4,998	975	0.0739754	0.134935
$\mathbf{L}_4^2$	134.58	121,477	2,651	5,004	978	0.0484597	0.155655
$\mathbf{L}_5^2$	80.61	56,586	2,651	5,004	978	0.0248889	0.13016
$\mathbf{U}_3^1$	55.56	41,487	2,027	3,840	840	0.178971	0.239858
$\mathbf{U}_4^1$	46.17	25,037	2,027	3,840	840	0.17244	0.234095
$\mathbf{U}_5^1$	123.50	83,850	2,027	3,840	840	0.166794	0.225759
$\mathbf{U}_3^2$	77.29	55,398	2,379	4,292	1,018	0.178971	0.197646
$\mathbf{U}_4^2$	263.72	203,168	2,382	4,297	1,019	0.161937	0.163313
$\mathbf{U}_5^2$	234.85	108,909	2,382	4,297	1,019	0.127602	0.157531

Table 3.1: Solution statistics of lower and upper pipe flow flood mitigation models.

Evolution of the Objective Value Figure 3.6 depicts the value of the objective function as the number of available barriers increases for cases where the models allow for one or two barriers per cell. The key takeaway is the importance of using multiple barriers at a specific location. This appears to bring significant benefits as the number of barriers is increased. 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 behave comparably as the number of maximum barriers is increased and, as expected, the upper objective value is greater than the lower objective value.

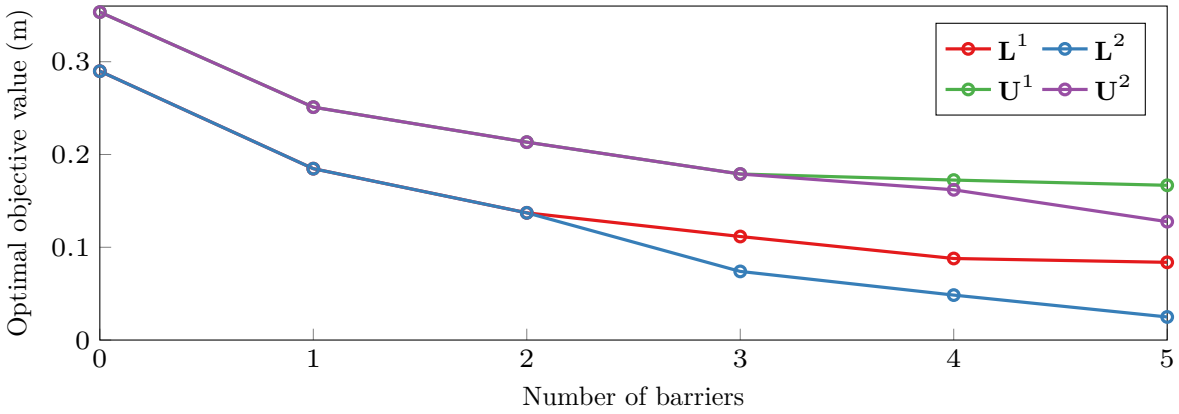


Figure 3.6: Objectives of lower and upper pipe flow flood mitigation models. Also compared are differences among objective values when one barrier is allowed per cell ($\mathbf{L}^1$ and $\mathbf{U}^1$) versus two barriers per cell ($\mathbf{L}^2$ and $\mathbf{U}^2$).

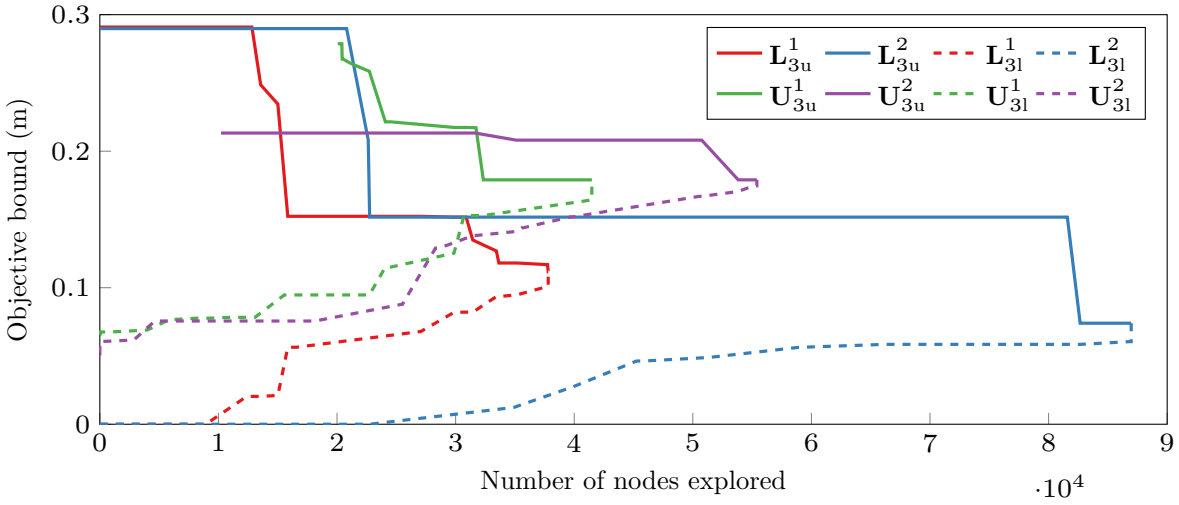


Figure 3.7: Convergence of lower and upper pipe flow flood mitigation models. Here, "l" and "u" are lower and upper objective bounds for $n = 3$.

Computational Results Finally, Table 3.1 tabulates experimental 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 explored in the search tree. The fourth, fifth, and sixth columns describe the number of variables, constraints,

and binary variables after presolving. Column η^* 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 is used in the lower model and vice versa.

As mentioned previously, no attempt was made to strengthen formulations or exploit problem structure. The instances have about 2,000 (mostly binary) variables and 4,000 constraints, and they can typically be solved within a few minutes. In general, CPLEX is not able to find feasible solutions quickly, which substantially increases computation times. This is illustrated in Figure 3.7, where CPLEX spends much time improving the upper bound. Integrating primal heuristics should improve convergence significantly.

3.5. Conclusion

Each year, flood-related disasters cause billions of dollars in damage, loss of life, and significant human suffering. Structural mitigation measures, such as levees and berms, are often used to lessen the consequences of these events. The design of these mitigation efforts, however, is sometimes suboptimal 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 chapter was to establish the foundation for a more principled approach to flood mitigation. It introduced the OFMP, which aims at integrating simulation and optimization by including flood simulation equations as part of the optimization model. To ensure the tractability of the approach, the main contribution of the chapter is the development of linear, physics-based approximations of shallow water flood 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 within optimization models was tested on several small cases, demonstrating the potential of optimization in this context.

Future work should focus on addressing the primary computational challenges raised by the OFMP. Surprisingly, state-of-the-art MILP solvers are not capable of exploiting the structure of this problem. In particular, they do not recognize that, once barriers are placed, the solution 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) may have potential in addressing this challenge for MILP-based

optimization approaches. It may also be useful to consider if dominance relationships hold among mitigation solutions, which would reduce the size of the search space.

More generally, exploiting the natural separation between mitigation decisions and flood propagation appears to be key when scaling to realistic problems. To this end, Chapter 4 develops an alternative approach to the MILP optimization method developed in this chapter. Specifically, Chapter 4 combines a fast simulator of the 2D shallow water equations, a metaheuristic-based optimization technique, and a novel search space reduction method to efficiently compute high-quality hard and soft structural flood mitigation strategies for scenarios with realistically-sized spatiotemporal domains.

Chapter 4

Optimization of Structural Flood Mitigation Strategies Using Physics-based Metaheuristics

Chapter 3 explored the use of linear approximations and MILP problem formulations in the attempt to solve challenging OFMPs. As the empirical results demonstrate, however, the technique does not scale to large instances for two reasons. First, even linearized approximations of the shallow water equations require conditional expressions to model important conservation relationships. Second, MILP solvers struggle to exploit the natural separation between the fixing of mitigation decisions and the resultant flood dynamics. These properties suggest that a simulation-based metaheuristic optimization approach may hold promise when scaling the OFMP to realistically-sized flood scenarios. That is, although such an approach may not provide optimality guarantees, it may be able to more readily exploit the clear separation between simulation and optimization.

To this end, this chapter, which is based on [133], explores a more scalable approach to the problem of designing structural FRM strategies over PDE constraints. Specifically, it develops a problem discretization amenable to simulation-based derivative-free optimization. Moreover, the chapter shows that metaheuristics alone are insufficient for obtaining quality solutions in reasonable time. As a result, it presents several innovative computational and physics-based techniques to limit the search space and increase convergence to high-quality solutions. The efficiency of the proposed approach is compared using hypothetical dam break scenarios of varying complexity under multiple mitiga-

tion budgets. Experimental results show that the proposed algorithm results in a 65% improvement in solution quality compared to a direct algorithm implementation.

Chapter Overview The rest of this chapter proceeds as follows: Section 4.1 discusses the background of flood modeling and formalization of the OFMP; Section 4.2 describes solution methods for a specific OFMP; Section 4.3 compares methods using fictional dam break scenarios, with both simplistic (Section 4.3.3) and realistic (Section 4.3.4) topographies, and multiple mitigation budgets; and Section 4.4 concludes the chapter.

4.1. Background and Optimization Model

In this chapter, it is assumed that flood scenarios are modeled using the 2D shallow water equations. Rewriting Equations (2.24) and (3.1) with volumetric, bed slope, and *Manning-Strickler* bed shear stress source terms, these equations are expressed as

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = R(x, y, t), \tag{4.1a}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial (huv)}{\partial y} = -gh\frac{\partial B}{\partial x} - g\frac{n^2}{h^{1/3}}|u|u, \tag{4.1b}$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial B}{\partial y} - g\frac{n^2}{h^{1/3}}|v|v, \tag{4.1c}$$

where n is the Manning's roughness coefficient [30]. For greater conciseness, the shallow water equations can be rewritten in vector form by first introducing the definitions

$$\begin{aligned} \mathbf{U} &:= (h, hu, hv)\,,\ \mathbf{F}(\mathbf{U}) := \left(hu, hu^2 + \frac{1}{2}gh^2, huv\right), \\ \mathbf{G}(\mathbf{U}) &:= \left(hv, huv, hv^2 + \frac{1}{2}gh^2\right), \mathbf{S}_R(R) := (R(x, y, t), 0, 0)\,, \\ \mathbf{S}_B(\mathbf{U}, B) &:= \left(0, -gh\frac{\partial B}{\partial x}, -gh\frac{\partial B}{\partial y}\right),\ \mathbf{S}_n := \left(0, -g\frac{n^2}{h^{1/3}}|u|u, -g\frac{n^2}{h^{1/3}}|v|v\right), \end{aligned} \tag{4.2}$$

where $\mathbf{U}$ is the vector of conserved variables; $\mathbf{F}$ and $\mathbf{G}$ are fluxes in the x- and y-directions, respectively; and $\mathbf{S}_R$, $\mathbf{S}_B$, and $\mathbf{S}_n$ are the volumetric, bed slope, and bed shear stress source terms, respectively. This allows the 2D shallow water Equations

(4.1a), (4.1b), and (4.1c) to be rewritten more concisely using vector notation as

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \mathbf{S}_B + \mathbf{S}_n, \tag{4.3}$$

where t, x, and y indicate partial differentiation with respect to those variables.

In this chapter, we expand upon the simplified OFMP described in Chapter 3 to construct a more general form. Specifically, the OFMP in this chapter is assumed to be constrained by the shallow water equations and allows for both elevation additions *and* surface roughness modifications to mitigate the flood. Similar to Chapter 3, the OFMP in this chapter considers a flood scenario (e.g., a dam failure) and a set of 2D regions (or "assets") to protect. To minimize flooding at asset locations, the model must produce optimal topographic elevation and roughness fields using a set of m mitigation measures. For each measure $i \in \{1, 2, \dots, m\}$, the functions $\delta_B(\omega_i)$ and $\delta_n(\omega_i)$ define continuous 2D fields of height and roughness for a given tuple of field parameters ω_i. Measures can first additively modify the elevation field B to return a new field $\tilde{B}$, defined as

$$\tilde{B}(B, (\omega_1, \omega_2, \dots, \omega_m)) := B + \sum_{i=1}^{m} \delta_B(\omega_i). \tag{4.4}$$

Similarly, measures can modify the roughness field n to return a new field defined as

$$\tilde{n}(n, (\omega_1, \omega_2, \dots, \omega_m)) := n + \left\{ \max_i \{\delta_n(\omega(i))(x, y)\} : (x, y) \in \mathbb{R}^2 \right\}, \tag{4.5}$$

i.e., a field of maximum roughness. For notational ease, hereafter, $\tilde{B}$ refers to Equation (4.4), $\tilde{n}$ refers to Equation (4.5), and the tuple $(\omega_1, \omega_2, \dots, \omega_m)$ is referred to as the "parametric configuration," or simply the "configuration." With these definitions and shorthand notations, the *modified* bed slope source term is then defined as

$$\tilde{\mathbf{S}}_B(\mathbf{U}, B, (\omega_1, \omega_2, \dots, \omega_m)) := \left(0, -gh\frac{\partial \tilde{B}}{\partial x}, -gh\frac{\partial \tilde{B}}{\partial y}\right). \tag{4.6}$$

The change in elevation may be a result of permanent structures such as levees or

temporary measures such as sandbags. Similarly, the modified bed shear stress is

$$\tilde{\mathbf{S}}_n\left(\mathbf{U}, n, (\omega_1, \omega_2, \dots, \omega_m)\right) := \left(0, -g\frac{\tilde{n}^2}{h^{1/3}}|u|u, -g\frac{\tilde{n}^2}{h^{1/3}}|v|v\right). \tag{4.7}$$

Hereafter, $\tilde{\mathbf{S}}_B := \tilde{\mathbf{S}}_B\left(\mathbf{U}, B, (\omega_1, \omega_2, \dots, \omega_m)\right)$ and $\tilde{\mathbf{S}}_n := \tilde{\mathbf{S}}_n\left(\mathbf{U}, n, (\omega_1, \omega_2, \dots, \omega_m)\right)$ are used to concisely denote these two source terms that vary with the configuration.

The OFMP is then written in a form that embeds the 2D shallow water equations as constraints and optimizes the tuple $(\omega_1, \omega_2, \dots, \omega_m)$ (i.e., the configuration) via

$$\underset{\omega_1,\omega_2,\dots,\omega_m}{\text{minimize}} \qquad \eta\left(\omega_1, \omega_2, \dots, \omega_m\right) = \sum_{a\in\mathcal{A}} \iint_a \max_t h(x, y, t)\,\mathrm{d}x\,\mathrm{d}y \tag{4.8a}$$

$$\text{subject to} \qquad \mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \tilde{\mathbf{S}}_B + \tilde{\mathbf{S}}_n \tag{4.8b}$$

$$\delta_B(\omega_i)(x, y) = 0,\ \forall i \in \{1, 2, \dots, m\},\ \text{for } (x, y) \in \bigcup \mathcal{A} \tag{4.8c}$$

$$\delta_n(\omega_i)(x, y) = 0,\ \forall i \in \{1, 2, \dots, m\},\ \text{for } (x, y) \in \bigcup \mathcal{A} \tag{4.8d}$$

$$(\omega_1, \omega_2, \dots, \omega_m) \in \mathcal{F}. \tag{4.8e}$$

Here, $\mathcal{A}$ denotes the set of asset regions to be protected and η denotes the objective function. This function is defined in Equation (4.8a) and captures the maximum water volume over all asset locations and times. Constraint (4.8b) denotes the solution to the shallow water equations in the presence of the m mitigation measures. Constraints (4.8c) prohibit measures from being constructed "underneath" an asset. Similarly, Constraints (4.8d) prohibit the roughness at an asset location from being modified. Finally, Constraint (4.8e) ensures $(\omega_1, \omega_2, \dots, \omega_m)$ resides within the set of all feasible parametric configurations $\mathcal{F}$. That is, $\mathcal{F}$ distinguishes valid and invalid mitigation designs.

For simplicity of presentation, this chapter considers only two types of structural measures, although the approach can easily be generalized to include other soft and hard measures, both temporary and permanent. The first type is an immovable wall of fixed length (ℓ), width (w), and height ($\bar{b}_i$). Each wall is defined using three continuously-defined, bounded parameters: latitudinal position of the wall centroid (λ_i), longitudinal position of the wall centroid (ϕ_i), and angle of the wall formed with respect to the

longitudinal axis (θ_i). In this chapter, the centroid position is bounded by the scenario domain's spatial extent, and $\theta_i \in [0, \pi]$. The second structural type is a revegetation project defined by a 2D circular region with center (λ_i, ϕ_i) and fixed radius r that increases the area's Manning's roughness coefficient based on a fixed field $\bar{n}_i$. Under these assumptions, the OFMP aims at deciding $\omega_i = (\lambda_i, \phi_i, \theta_i, \bar{b}_i, \bar{n}_i)$ for each measure $i \in \{1, 2, \dots, m\} = \mathcal{M}$, where $\bar{b}_i$ and $\bar{n}_i$ are decided a priori for each measure. More specifically, these mitigation properties produce an OFMP of the specialized form

$$\underset{\omega_1, \omega_2, \dots, \omega_m}{\text{minimize}} \quad \eta(\omega_1, \omega_2, \dots, \omega_m) = \sum_{a \in \mathcal{A}} \iint_a \max_t h(x, y, t) \, \mathrm{d}x \, dy \tag{4.9a}$$

$$\text{subject to} \quad \mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \tilde{\mathbf{S}}_B + \tilde{\mathbf{S}}_n \tag{4.9b}$$

$$\delta_B(\omega_i)(x, y) = 0, \text{ for } (x, y) \in \bigcup \mathcal{A}, \ \forall i \in \mathcal{M} \tag{4.9c}$$

$$\delta_n(\omega_i)(x, y) = 0, \text{ for } (x, y) \in \bigcup \mathcal{A}, \ \forall i \in \mathcal{M} \tag{4.9d}$$

$$\delta_B(\omega_i)(x, y) = \begin{cases} \bar{b}_i \text{ for } \begin{cases} |(x - \phi_i)\cos\theta_i - (y - \lambda_i)\sin\theta_i| \leq \frac{\ell}{2} \\ |(x - \phi_i)\sin\theta_i + (y - \lambda_i)\cos\theta_i| \leq \frac{w}{2} \end{cases} \forall i \in \mathcal{M} \\ 0 \text{ otherwise} \end{cases} \tag{4.9e}$$

$$\delta_n(\omega_i)(x, y) = \begin{cases} \bar{n}_i(x, y) & \text{for } (x - \phi_i)^2 + (y - \lambda_i)^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases} \forall i \in \mathcal{M} \tag{4.9f}$$

$$\lambda_{lb} \leq \lambda_i \leq \lambda_{ub}, \ \phi_{lb} \leq \phi_i \leq \phi_{ub}, \ 0 \leq \theta_i \leq \pi, \ \forall i \in \mathcal{M}. \tag{4.9g}$$

Using this formulation, i is a wall when $\bar{b}_i > 0$ and $\bar{n}_i = 0$, and i is a revegetation project when $\bar{b}_i = 0$ and $\bar{n}_i > 0$. Constraints (4.9c) and (4.9d) emphasize that modifications cannot be made within asset regions; Constraints (4.9e) impose the wall height $\bar{b}_i$ within each rotated rectangle defined using the parameters λ_i, ϕ_i, and θ_i and a standard 2D rotation matrix; and Constraints (4.9f) impose additions to roughness within each revegetation circle defined by the center (λ_i, ϕ_i). Finally, Constraints (4.9g) replace Constraint (4.8e) of the more general OFMP. Here, λ_{lb} and λ_{ub} (ϕ_{lb} and ϕ_{ub}) are the

lower and upper latitudinal (longitudinal) boundaries of the flood scenario domain.

Constraints (4.9g) imply a large feasible region, as the spatial extent is typically much larger than the flood's extent. To reduce the solution space, the notion of a *restricted region* $\mathcal{P}$ is thus introduced, where structure centroids must reside in $\mathcal{P}$. That is,

$$(\lambda_i, \phi_i) \in \mathcal{P}, \ \forall i \in \{1, 2, \dots, m\} \tag{4.9h}$$

is appended to the problem above, completing the primary model used in this chapter.

4.2. Optimization Algorithm

The OFMP at the end of Section 4.1 is difficult to solve directly. However, with recent improvements in both numerical discretizations of the shallow water equations (e.g., [30]) and high-performance implementations thereof (e.g., [20], [135]), efficient solutions of the PDEs described in Constraint (4.9b) are possible. With this intuition, in Algorithm 4.1, a time-limited search-based method is introduced to find a near-optimal solution $(\omega_1^*, \omega_2^*, \dots, \omega_m^*)$ to the problem defined by Equations (4.9a) through (4.9h).

Algorithm 4.1 SolveOFMP: Solves the OFMP of Equations (4.9a) through (4.9h).

function SolveOFMP($B, n, \mathcal{A}, m, T_{\max}, \alpha$)
  $\tilde{\mathcal{P}} \leftarrow$ InitializeRestriction($B, n, \mathcal{A}, \alpha$)
  $(\omega_1^*, \omega_2^*, \dots, \omega_m^*) \leftarrow$ InitializeSolution($m, \tilde{\mathcal{P}}$), $\Omega \leftarrow \emptyset$
  while Clock $< T_{\max}$ **do**
    $(\omega_1, \omega_2, \dots, \omega_m) \leftarrow$ GenerateSolution($m, \tilde{\mathcal{P}}, \Omega$)
    Solve $\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \tilde{\mathbf{S}}_B + \tilde{\mathbf{S}}_n$
    $\Omega \leftarrow \Omega \cup \{(\omega_1, \omega_2, \dots, \omega_m)\}$
    if $\eta(\omega_1, \omega_2, \dots, \omega_m) < \eta(\omega_1^*, \omega_2^*, \dots, \omega_m^*)$ **then**
      $(\omega_1^*, \omega_2^*, \dots, \omega_m^*) \leftarrow (\omega_1, \omega_2, \dots, \omega_m)$
      $\tilde{\mathcal{P}} \leftarrow$ UpdateRestriction($\mathbf{U}, \mathcal{A}, \tilde{\mathcal{P}}, \alpha$)
    end if
  end while
  return $(B + \sum_{i=1}^{m} \delta_B(\omega_i^*), \ n + \{\max_i\{\delta_n(\omega_i^*)(x, y)\} : (x, y) \in \mathbb{R}^2\})$
end function

Here, B and n denote the initial topographic elevation and Manning's roughness coefficient fields; $\mathcal{A}$ denotes the set of assets; m denotes the number of mitigation

measures being configured; $T_{\max}$ denotes the maximum clock time; and α is a parameter used for computing restrictions. The function Clock returns the current clock time. Since a useful definition of $\mathcal{P}$ is difficult to compute a priori, $\tilde{\mathcal{P}}$ serves as an iterative approximation of some desired $\mathcal{P}$. In Line 2, $\tilde{\mathcal{P}}$ is initialized; it is later modified in Line 10 using UpdateRestriction. Both functions are described in Section 4.2.1. In Line 3, the best solution and the historical solution set Ω are initialized. In Line 5, a configuration is generated via some history-dependent function GenerateSolution, described in Section 4.3.1. In Line 6, the shallow water equations are solved. In Line 7, the historical solution set is updated. In Lines 8 through 11, the best solution and $\tilde{\mathcal{P}}$ are updated. Finally, in Line 13, the best elevation and roughness fields are returned.

4.2.1. Computation of the Restricted Region

Direct Methodology A globally acceptable method for selecting $\mathcal{P}$ is to assume

$$\mathcal{P} = \mathbb{R}^2, \tag{4.10}$$

where, of course,

$$\left\{(x, y) \in \mathbb{R}^2 : \lambda_{lb} \leq x \leq \lambda_{ub},\ \phi_{lb} \leq y \leq \phi_{ub}\right\} \subset \mathcal{P}, \tag{4.11}$$

indicating the bounds within Constraints (4.9g) involving λ_i and ϕ_i dominate those imposed by $\mathcal{P}$. This method for selecting $\mathcal{P}$ is hereafter referred to as the direct method. In practice, this method is used to define the direct implementations of the functions InitializeRestriction and UpdateRestriction, both of which return the set $\mathbb{R}^2$.

Pathline Methodology A *pathline* is the trajectory an individual fluid element follows over time, beginning at position (x_0, y_0) and time t_0. In 2D, a pathline satisfies

$$x(t) = x_0 + \int_{t_0}^{t} u(x(t'), y(t'), t')dt', \tag{4.12a}$$

$$y(t) = y_0 + \int_{t_0}^{t} v(x(t'), y(t'), t')dt', \tag{4.12b}$$

where u and v are velocities in the x- and y-directions. To compute the pathline from a flood wave to an initially dry point (x_0, y_0), the definition of $t_{\text{wet}}(x_0, y_0)$ is introduced as the time at which the depth at (x_0, y_0) exceeds some threshold. More concisely,

$$t_{\text{wet}}(x_0, y_0) := \min \left\{ t \in [t_0, t_f] : h(x_0, y_0, t) \geq \epsilon_h \right\}, \tag{4.13}$$

where ϵ_h is an arbitrarily small depth, taken in this chapter to be one millimeter. Using this definition, the pathline equations may be temporally integrated *in reverse*, giving

$$x_{\text{wet}}(x_0, y_0, t) = x_0 + \int_{t_{\text{wet}}}^{t} u(x_{\text{wet}}(t'), y_{\text{wet}}(t'), t')dt', \tag{4.14a}$$

$$y_{\text{wet}}(x_0, y_0, t) = y_0 + \int_{t_{\text{wet}}}^{t} v(x_{\text{wet}}(t'), y_{\text{wet}}(t'), t')dt', \tag{4.14b}$$

where it is assumed that $t \leq t_{\text{wet}}$. The above equations approximate a path to flooding.

In this chapter, a *pathtube* is defined as a set of pathlines satisfying Equations (4.14a) and (4.14b). For a region $\mathcal{R}$, the pathtube $\mathcal{S}$ encompassing $\mathcal{R}$ with a start time of t_0 is

$$\mathcal{S}(\mathbf{U}, \mathcal{R}) = \left\{ (x_{\text{wet}}(x_0, y_0, t), y_{\text{wet}}(x_0, y_0, t)) \in \mathbb{R}^2 : (x_0, y_0) \in \mathcal{R},\ t \in [t_0, t_{\text{wet}}(x_0, y_0)] \right\}. \tag{4.15}$$

This region encompasses approximate paths of least resistance from a flood to $\mathcal{R}$. It is clear that good locations for structural mitigation measures are likely to reside in $\mathcal{S}$.

A robust selection of $\mathcal{P}$ would account for the change in $\mathbf{U}$ with respect to a large set of feasible configurations. In an ideal setting, a good selection for $\mathcal{P}$ would thus be

$$\mathcal{P} = \bigcup_{\omega \in \mathcal{F}} \bigcup_{a \in \mathcal{A}} \left\{ (x, y) \in \mathcal{S}(\mathbf{U}, a) : \mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \tilde{\mathbf{S}}_B + \tilde{\mathbf{S}}_n \right\}. \tag{4.16}$$

In practice, defining $\mathcal{P}$ as per Equation (4.16) is nontrivial. First, each $a \in \mathcal{A}$ may be a set of infinitely many points. There are also infinitely many moments t in a solution $\mathbf{U}$ to the shallow water equations. Most importantly, the union over all feasible configurations $(\omega_1, \omega_2, \ldots, \omega_m) = \omega \in \mathcal{F}$ assumes knowledge of $\mathbf{U}$ for any such feasible configuration $(\omega_1, \omega_2, \ldots, \omega_m)$. For these reasons, an iteratively-constructed definition of the pathtube-like region $\tilde{\mathcal{P}}$ is instead proposed, which approximately captures the features of some unknown larger spatial set $\mathcal{P}$ that is relevant to the OFMP (e.g., Equation (4.16)).

From a numerical perspective, each $a \in \mathcal{A}$ is actually a polygon whose exterior connects a set of points P_a. Solutions to the OFMP are likely to intersect the pathlines from a flood to each of these points. Also, in practice, numerical solutions to the shallow water equations are discrete in space and time. Assuming that solutions are obtained for a set of timestamps $\mathcal{T}$ on a rectangular grid G, t_{wet} is first redefined as

$$t_{\text{wet}}(x_0, y_0) := \min\{t \in \mathcal{T} : h_{i_0,j_0,t} \geq \epsilon_h\}, \tag{4.17}$$

where (i_0, j_0) is the unique index of the cell in grid G that contains the point (x_0, y_0).

For each point along an asset exterior, a representation of the pathline leading *to* that point is desired. To accomplish this, it is assumed that a pathline can be approximated as a set $\mathcal{L}$ of discrete points. These points can be generated by solving Equations (4.14a) and (4.14b) using any suitable ordinary differential equation (ODE) technique. Here, suggestions from [136] (initially described for *streamlines*, which trace a static field) are used to compute pathlines according to the function ComputePathline($\mathbf{U}, x_0, y_0$), whose arguments are a solution $\mathbf{U}$ to the shallow water equations and the x- and y-positions of a seed point. This algorithm is detailed in the appendix (Algorithm A.1).

The definition of ComputePathline enables the computation of a set of points Q approximating the *pathtube* leading to a set of exterior asset points $P_a \in a \in \mathcal{A}$ via

$$Q(\mathbf{U}, P_a) = \bigcup_{(x_0,y_0)\in P_a} \text{ComputePathline}(\mathbf{U}, x_0, y_0). \tag{4.18}$$

Since pathtubes are curvilinear, typical geometries that envelope Q (e.g., the convex hull) do not effectively summarize this set. For this reason, the notion of an *alpha shape* is introduced, which minimally encompasses points of Q using straight lines. A discussion on alpha shapes can be found in [56]. In this study, Edelsbrunner's algorithm [51], presented in the appendix (Algorithm A.2), is used to compute alpha shapes. The function AlphaShape(Q, α) computes this shape for a set Q and alpha value α.

The definition of the function AlphaShape finally allows for definition of the functions InitializeRestriction and UpdateRestriction for the pathline method. Both assume restrictions are the unions of alpha shapes approximating the pathtubes leading to each asset. The functions are described in Algorithms 4.2 and Equation (4.19), respectively. In Algorithm 4.2, Line 2, the shallow water equations are solved

Algorithm 4.2 InitializeRestriction: Returns the initial restricted positional set.

1: **function** InitializeRestriction$(B, n, \mathcal{A}, \alpha)$
2: Solve $\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \mathbf{S}_B + \mathbf{S}_n$
3: **return** $\bigcup_{a\in\mathcal{A}} \text{AlphaShape}\left(Q(\mathbf{U}, P_a), \alpha\right) \setminus \bigcup \mathcal{A}$
4: **end function**

without the presence of structural mitigation measures. In Line 3, the union of alpha shapes for all pathtubes leading to the assets $a \in \mathcal{A}$ is computed. Asset regions are then subtracted from this set to ensure measures do not overlap with asset locations.

The function UpdateRestriction using the pathline approach is defined as

$$\text{UpdateRestriction}(\mathbf{U}, \mathcal{A}, \mathcal{P}, \alpha) = \mathcal{P} \cup \left(\bigcup_{a\in\mathcal{A}} \text{AlphaShape}(Q(\mathbf{U}, P_a), \alpha) \right) \setminus \bigcup \mathcal{A}. \tag{4.19}$$

The majority of this function resembles Algorithm 4.2, although the union of the current set and previous $\mathcal{P}$ is computed to encourage exploration of a more representative (i.e., spatially expanded) search space. Moreover, as per Algorithm 4.1, this function is only called as better solutions to the OFMP are obtained. This decreases the potential burden of computing pathtubes and alpha shapes on each iteration of the algorithm.

4.2.2. Sequential Optimization Algorithm

Due to the nonlinear sensitivity of flooding behavior with respect to mitigation efforts, predictable and incremental changes to solutions of the OFMP while increasing the number of mitigation measures, m, are not ensured. This may be undesirable from a planning perspective. A separate algorithm is thus proposed to induce a sequential solution to the OFMP, whereby solutions with $m = 2$ include those of $m = 1$, solutions with $m = 3$ include those of $m = 2$, and so on. This ensures increasing utility for configurations of increasing sizes. It also allows policymakers to more clearly understand the effects of budgetary constraints with respect to the overall structural flood mitigation efforts. The recursive relationship used to compute sequential solutions is defined as

$$(B_i, n_i) = \text{SolveOFMP}\left(B_{i-1}, n_{i-1}, \mathcal{A}, 1, \frac{T_{\max}}{m}, \alpha\right), \tag{4.20}$$

where $B_0 = B$, $n_0 = n$, and the time for each subproblem is an equal portion of $T_{\max}$. In this chapter, Line 10 is eliminated from Algorithm 4.1 when using the sequential approach, as the best placement for a *single* structural mitigation measure is likely to reside within the *initial* $\tilde{\mathcal{P}}$ computed on Line 2. As a consequence, for each structural measure positioned using the sequential approach, pathtubes are constructed only once.

4.3. Computational Experiments

4.3.1. Model Relaxation

The approach uses the open source `scipy.optimize.differential_evolution` (DE) and RBFOpt software libraries in Python to produce two separate implementations of GenerateSolution in Algorithm 4.1 [41], [128]. Both only include support for simple bounds like those indicated in Constraints (4.9g). Thus, these implementations of GenerateSolution may generate configurations that are infeasible with respect to Constraints (4.9c) and (4.9f). To overcome this, the OFMP defined by Equations (4.9a) through (4.9h) is replaced with the penalty-based relaxed problem formulation

$$\underset{\omega_1,\omega_2,\ldots,\omega_m}{\text{minimize}} \quad \eta\left(\omega_1,\omega_2,\ldots,\omega_m\right) = p_1 + p_2 + \sum_{a\in\mathcal{A}} \iint_a \max_t h(x,y,t)\,\mathrm{d}x\,\mathrm{d}y \tag{4.21a}$$

$$\text{subject to} \quad p_1 = c_1 \sum_{i=1}^{m} \min\left\{\|(x,y)-(\lambda_i,\phi_i)\| : (x,y)\in\mathcal{P}\right\} \tag{4.21b}$$

$$p_2 = c_2 \sum_{i=1}^{m}\sum_{a\in\mathcal{A}} \iint_a \delta_B(\omega_i)\,\mathrm{d}x\,\mathrm{d}y + c_3 \sum_{i=1}^{m}\sum_{a\in\mathcal{A}} \iint_a \delta_n(\omega_i)\,\mathrm{d}x\,\mathrm{d}y \tag{4.21c}$$

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}_R + \tilde{\mathbf{S}}_B + \tilde{\mathbf{S}}_n \tag{4.21d}$$

$$\delta_B(\omega_i)(x,y) = \begin{cases} \bar{b}_i \text{ for } \begin{cases} |(x-\phi_i)\cos\theta_i - (y-\lambda_i)\sin\theta_i| \le \frac{\ell}{2} \\ |(x-\phi_i)\sin\theta_i + (y-\lambda_i)\cos\theta_i| \le \frac{w}{2} \end{cases} \forall i \in \mathcal{M} \\ 0 \text{ otherwise} \end{cases} \tag{4.21e}$$

$$\delta_n(\omega_i)(x,y) = \begin{cases} \bar{n}_i & \text{for } (x-\phi_i)^2 + (y-\lambda_i)^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{M} \tag{4.21f}$$

$$\lambda_{lb} \leq \lambda_i \leq \lambda_{ub},\ \phi_{lb} \leq \phi_i \leq \phi_{ub},\ 0 \leq \theta_i \leq \pi,\ \forall i \in \mathcal{M}. \tag{4.21g}$$

In Equation (4.21a), two penalty terms are included in the objective to capture infeasibilities in Constraints (4.9c) through (4.9f). The first penalty, p_1, is defined in Constraint (4.21b) and denotes the sum of all minimum distances between each measure's centroid and the nearest point of the restricted positional set $\mathcal{P}$. This term is scaled by the constant c_1, taken in this study to be equal to one. The second penalty, p_2, is defined in Constraint (4.21c). Here, the first term denotes the net modified elevation volume over all asset regions, and the second term denotes the net change in roughness over all asset regions. These terms are scaled by the constants c_2 and c_3, respectively. Herein, both are taken to be $(\Delta r)^{-2}$, where Δr is the spatial resolution of the discretization.

4.3.2. Experimental Setting

For simplicity, Sections 4.3.3 through 4.3.4 focus on OFMPs where only wall-type measures are considered (i.e., $\bar{b}_i > 0$), while Section 4.3.4 presents an algorithmic proof of concept where only revegetation-type measures are considered (i.e., $\bar{n}_i > 0$). For each experiment, Algorithm 4.1 was limited to one day of wall-clock time. When using DE, population sizes of $45m$ ($\bar{b}_i > 0$) and $30m$ ($\bar{n}_i > 0$) were employed; trial solutions were computed as the best solution plus scaled contributions of two random candidates; the mutation constant varied randomly within $[0.5, 1.0)$; and the recombination constant was set to 0.9. When using the direct InitializeRestriction and UpdateRestriction methods, Latin hypercube sampling was used to initialize the population. When using the pathline-based methods, the population was initialized via random sampling over the initial restricted set (i.e., $\tilde{\mathcal{P}}$) and $\theta_i \in [0, \pi]$. When using RBFOpt, the `sampling` method was used; most other parameters were left unchanged.

For computational considerations, if the configuration proposed by GenerateSolution was feasible, the shallow water equations (i.e., Constraint (4.21d)) were solved using the proposed configuration. Otherwise, a solution containing *no structural mitigation measures* was referenced. That is, $\tilde{S}_B$ was replaced with S_B, and $\tilde{S}_n$ was replaced

with S_n. To solve the PDEs, the open-source surface water modeling software NUFLOOD [135] was used, where the shallow water equations are discretized according to [83].

Each experiment was conducted on one Intel Xeon E5-2695 v4 CPU containing eighteen cores at 2.1 GHz and 125 GB of RAM. NUFLOOD was compiled in single-precision mode using the Intel C++ Compiler, version 17.0.1. The remainder of Algorithm 4.1 was implemented in Python 3.6. Compared to the shallow water PDE evaluations, these other portions of the algorithm were measured to be computationally negligible.

4.3.3. Simplified Circular Dam Break Scenarios

To compare the two positional restriction methodologies described in Section 4.2.1, six simple OFMP scenarios were constructed. All were intended to have human intuitive solutions, i.e., optimal placement of structural mitigation measures could be inferred from a basic understanding of flood propagation. These scenarios are displayed pictorially in Figure 4.1. In each scenario, under the influence of gravity, the initial volume of water (colored with blue) is propagated outward. Without mitigation measures, this water easily comes into contact with critical assets (colored with red), flooding them.

Each of the six scenarios was modeled using a spatial resolution of one meter and 64×64 grid cells. The ground surface was assumed to be frictionless; critical depth boundary conditions were employed; and a simulation duration of one hundred seconds was used. When necessary to compute pathlines, intermediate PDE solution data was reported for every one second of simulation time. In the experiments performed, each of the corresponding OFMPs was solved with the number of walls, m, ranging from one to five. Wall widths, lengths, and heights were fixed to 2.5, 8.0, and 1.0 meters, respectively. Finally, all experiments were performed using a single fixed random seed.

In Figure 4.2, for each optimization experiment, the objective behavior is plotted against the number of PDE evaluations required to reach that objective. These behaviors are compared for the direct differential evolution solver (DE-D) and its pathline-based counterpart (DE-PL). The DE-PL solver was generally able to find good solutions faster and improve upon them more rapidly, especially for configurations involving larger numbers of walls. However, there were some instances where the DE-D solver produced higher quality solutions than the DE-PL solver, e.g., when optimizing the configuration of five walls in Scenario 4. These anomalies could be a consequence of the random na-

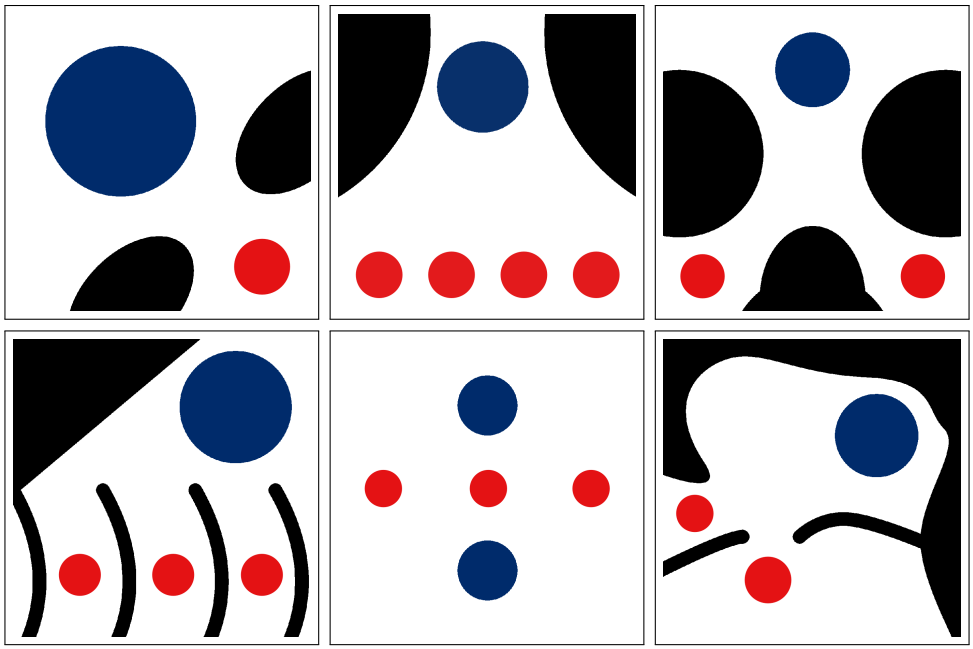


Figure 4.1: Pictorial depictions of six simplified flood scenarios. They are ordered numerically (e.g., one in the upper left). Black represents nonzero topographic elevation (of height one meter); blue represents nonzero initial water depth (of height one meter); and red represents assets.

ture of the DE algorithm; they could also be due to the DE algorithm implementation's tendency to terminate once the population of candidates has sufficiently stabilized.

In Figure 4.3, the best obtained wall configurations using DE-PL are displayed pictorially for all pairs of scenarios and numbers of mitigation measures. Here, as well as in similar subsequent figures of this chapter, darker blue corresponds to larger maximum depths; black corresponds to nonzero portions of the initial topographic elevation field; green corresponds to elevation additions via the placement of walls; and red corresponds to asset locations. Finally, orange lines represent the exteriors of the final computed restricted positional sets $\widetilde{\mathcal{P}}$ in Algorithm 4.1. As observed in this figure, the configurations resemble what might be intuited by a human. When applicable, configurations are non-overlapping and well-connected. As the number of walls varies, configurations also show interesting nonincremental behavior. For example, in the first scenario, walls are initially placed close to the asset. As the number of walls increases, they are placed farther away to form connections with existing topographic features. However, we note that such non-sequential behavior may be undesirable from a planning perspective.

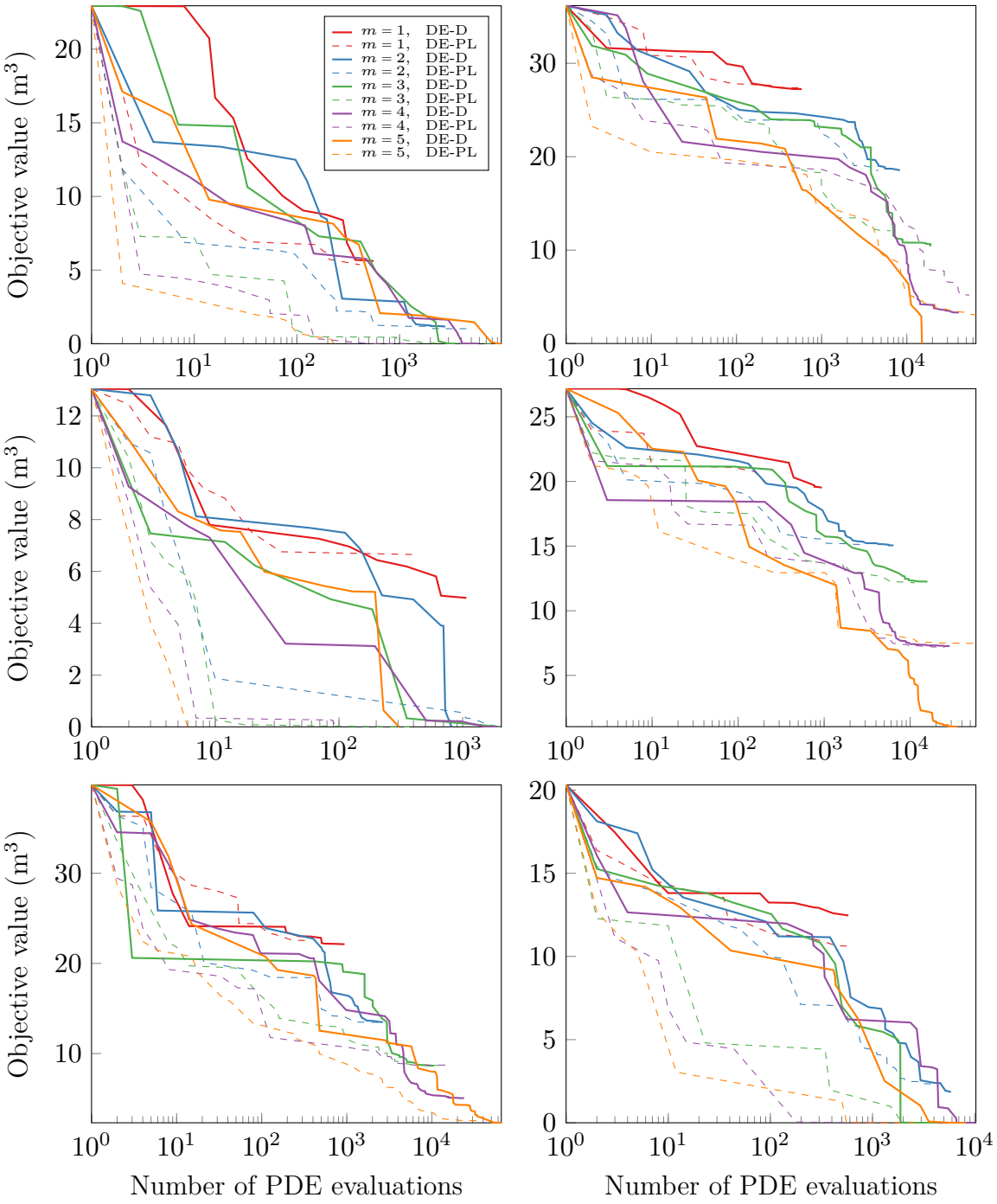


Figure 4.2: Algorithm convergence on the six simplified flood scenarios. Here, the objective values versus numbers of PDE evaluations are compared for OFMP Scenarios 1 through 6, respectively, using DE-D and DE-PL to optimize mitigation configurations of one through five flood walls.

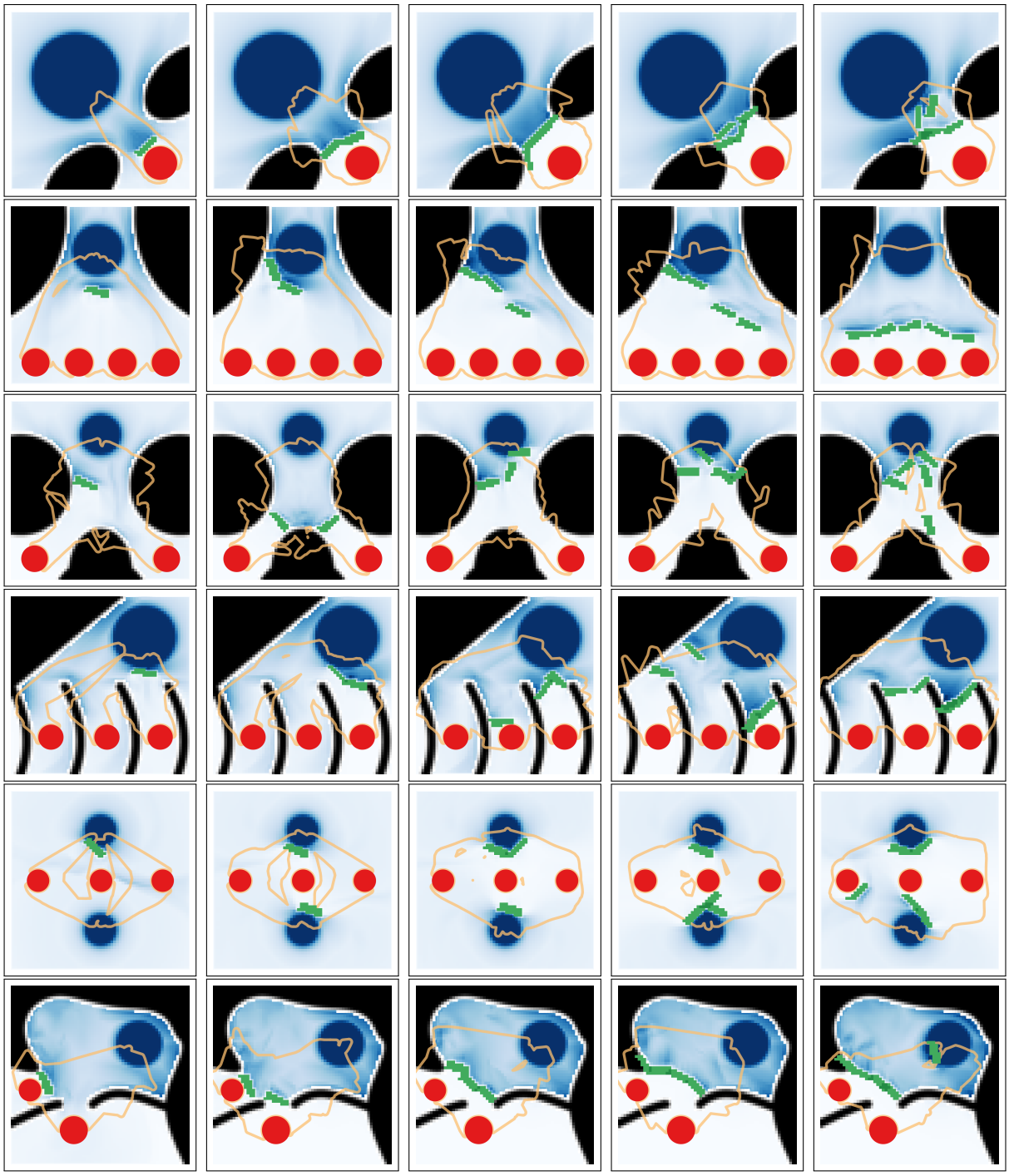


Figure 4.3: Solution illustrations for the six simplified flood scenarios. The best obtained elevations and maximum depths are shown for configurations of one through five walls and reference Scenarios 1 through 6 (vertically). In these illustrations, orange lines represent the exteriors of the final computed restricted positional sets $\tilde{\mathcal{P}}$ used in Algorithm 4.1.

4.3.4. Dam Break Scenario from Theme C of the 12th International Benchmark Workshop on Numerical Analysis of Dams

This section focuses on demonstrating the merits of the sequential optimization algorithm using the dam break defined in Theme C of the 12th International Benchmark Workshop on Numerical Analysis of Dams (ICOLD 2013) [77]. To simulate this scenario, the dam break was modeled as a point source with time-dependent discharge. The initial topographic elevation field (with the dam excluded) was provided by the workshop and resampled from a resolution of ten to ninety meters to ease computational burden. The Manning's roughness coefficient was set to 0.035; critical depth boundary conditions were employed; a duration of twelve hours was used; and, when necessary to compute pathlines, PDE solution data was reported every ten minutes of simulation time.

Asset locations and sizes were selected in an attempt to increase the overall difficulty of the OFMP, with two assets placed near the primary channel of the scenario and three placed farther away. The experimental setup remained similar to that described in Section 4.3.3. However, in this case, the number of walls ranged from one to ten, while wall widths, lengths, and heights were fixed to 250, 1,000, and ten meters, respectively. To compare differences in OFMP solver performance, each solver was executed using ten different random seeds for each possible number of walls. In total, the experiments described in this subsection thus required nearly six hundred days of compute time.

Pathline-based algorithm results

To confirm the effectiveness of the pathline-based solvers, two implementations of Algorithm 4.1 using RBFOpt were benchmarked. In Table 4.1, the objective behavior of the pathline-based solver (RBFOpt-PL) is compared against its direct counterpart (RBFOpt-D). The pathline-based solver clearly outperforms RBFOpt-D in nearly all instances, e.g., it results in smaller minima, means, and standard deviations. The single exception appears to be for $m = 1$, where the direct solver produces an equivalent minimum to the pathline-based solver. Nonetheless, on average, the pathline-based solver provides a 45% improvement over the direct solver, with generally larger improvements

	RBFOpt-D				RBFOpt-PL				Mean
m	Mean	Min	Max	SD	Mean	Min	Max	SD	Imprvmt.
1	165.93	159.63	170.25	3.36	162.81	159.63	167.59	2.61	1.88%
2	147.96	105.34	166.60	20.14	111.02	95.98	130.88	10.74	24.96%
3	144.93	111.87	164.10	15.44	89.22	77.56	93.62	5.40	38.44%
4	128.52	105.81	159.12	14.21	81.33	51.63	97.83	13.47	36.72%
5	135.94	128.25	144.42	5.03	62.38	26.40	80.40	15.67	54.11%
6	122.24	98.38	140.19	14.22	59.13	41.40	71.25	9.07	51.63%
7	119.14	81.14	145.80	18.65	51.11	29.13	66.20	13.08	57.10%
8	102.08	78.65	122.69	16.22	43.28	27.82	57.35	9.40	57.60%
9	107.40	81.93	127.53	15.52	42.73	19.00	53.26	11.19	60.21%
10	104.90	77.17	124.13	16.75	34.72	21.75	42.74	7.60	66.90%

Table 4.1: Comparison of RBFOpt-D and RBFOpt-PL solver solutions. Each entry considers objectives over ten seeds, with the number of walls (m) ranging from one to ten, as in Section 4.3.4. Values are scaled by 10^{-4}.

for scenarios with greater numbers of walls. This improvement was computed as

$$\text{Percentage Improvement} = \left(\frac{a-b}{a}\right) 100\%, \tag{4.22}$$

where a and b represent the mean objective values obtained from the RBFOpt-D and RBFOpt-PL solvers. The same metric is also used throughout Tables 4.2, 4.3, and 4.4.

A similar comparison is made between DE-D and DE-PL in Table 4.2. Again, the pathline-based solver (DE-PL) outperforms its direct counterpart (DE-D) in nearly all metrics, providing an overall mean improvement of 59%. The pathline-based solver also displays mostly monotonic decreases in the objective as the number of walls increases, while the objectives associated with the direct solver generally *increase* as the number of walls increases. However, note that for small numbers of walls (i.e., one and two), the direct DE solver outperforms its pathline-based counterpart. This could be a consequence of the more complicated objective penalty in Constraint (4.21b) when $\mathcal{P}$ is restricted. For example, [47] describes various means by which penalty-based derivative-free optimization algorithms can result in suboptimal solutions. Nonetheless, overall, the direct penalization method considered in this chapter appears to provide good solutions.

It is important to note the differences between the RBFOpt-based and DE-based solvers benchmarked in Tables 4.1 and 4.2, respectively. In general, DE-PL greatly

	DE-D				DE-PL				Mean
m	Mean	Min	Max	SD	Mean	Min	Max	SD	Imprvmt.
1	<u>162.18</u>	**158.59**	167.59	3.85	163.86	159.63	170.07	4.52	−1.04%
2	99.49	**84.87**	104.24	6.29	102.61	94.24	105.39	3.19	−3.13%
3	66.78	39.56	119.20	27.76	<u>50.75</u>	**33.36**	65.93	14.80	24.01%
4	101.06	79.57	134.61	20.57	31.84	17.51	56.87	13.20	68.49%
5	115.43	100.56	145.39	16.04	22.74	12.82	36.39	8.04	80.30%
6	124.15	101.22	145.75	15.85	24.79	7.24	60.66	14.87	80.03%
7	129.17	110.94	153.14	12.52	18.14	5.59	26.77	6.64	85.96%
8	125.51	96.68	144.92	16.47	14.02	4.03	19.74	5.59	88.83%
9	129.23	99.39	146.58	14.02	17.83	8.56	22.46	4.85	86.20%
10	119.09	86.16	140.87	17.91	19.68	10.43	30.38	5.61	83.47%

Table 4.2: Comparison of DE-D and DE-PL solver solutions. This table assumes the same reporting conventions as Table 4.1. Best objective values over all seeds and solvers considered in Tables 4.1 through 4.4 are denoted in bold, while the best *mean* objective values are denoted with underlines.

outperforms both RBFOpt-based solvers. For example, DE-PL provides a 47% mean improvement over RBFOpt-PL. These differences could be for multiple reasons. For example, there are many more hyperparameters associated with RBFOpt than DE; more careful tuning may have increased RBFOpt's convergence. Furthermore, RBFOpt's `sampling` search strategy was used to show the efficacy of the pathline-based approach when applied to other (possibly non-evolutionary) heuristic search techniques; the RBFOpt solver software may indeed perform more favorably using some other strategy.

Figure 4.4 displays the best obtained wall configuration for each possible number of walls using the DE-PL solver. Structure placement appears highly nonincremental as the number of walls increases, especially for smaller numbers of walls. Also, when optimizing for a number of walls greater than eight, solutions generally deteriorate, indicating the search space becomes prohibitively large. Interestingly, the size of the restricted set $\widetilde{\mathcal{P}}$ does not increase substantially as the configuration size grows. Finally, in Figure 4.5, the best obtained solution for ten walls using DE-D is displayed. This underscores the difficulty of such a problem when applying a conventional algorithm.

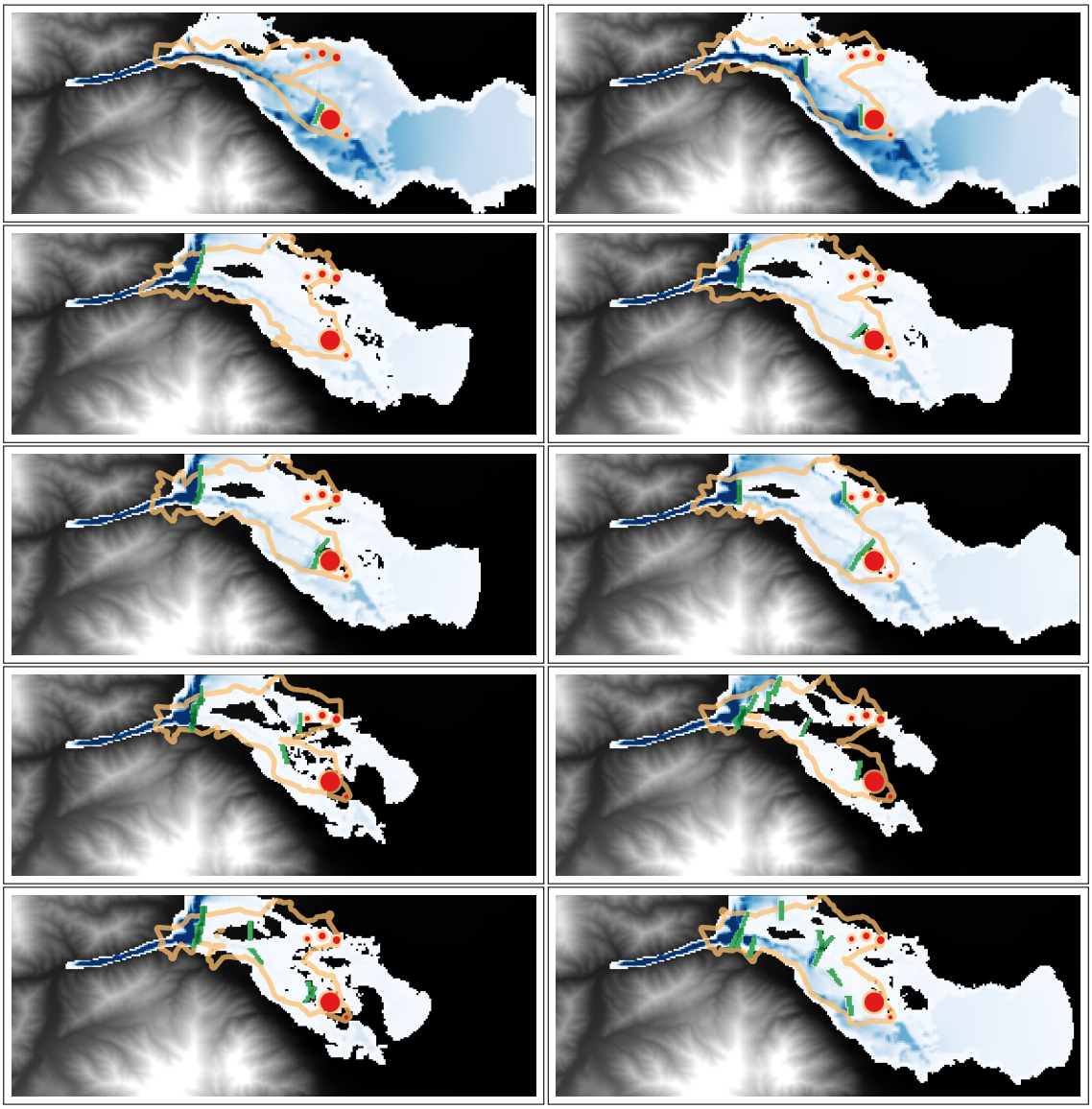


Figure 4.4: Pathline-based solutions for the realistic dam failure. Here, *best* elevations and maximum depths when using DE-PL are shown.

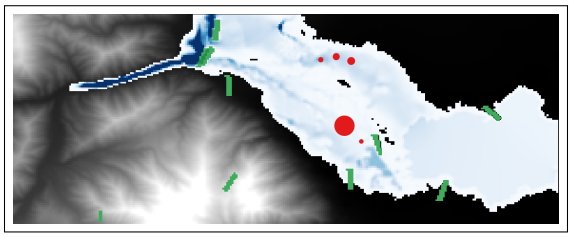


Figure 4.5: Direct ten-wall solution for the realistic dam failure.

	DE-PL				DE-D-S				Mean
m	Mean	Min	Max	SD	Mean	Min	Max	SD	Imprvmt.
1	163.86	159.63	170.07	4.52	<u>162.18</u>	**158.59**	167.59	3.85	1.02%
2	102.61	94.24	105.39	3.19	102.38	100.97	103.68	0.87	0.22%
3	<u>50.75</u>	**33.36**	65.93	14.80	64.98	48.97	81.93	14.71	−28.06%
4	31.84	17.51	56.87	13.20	41.04	26.07	58.60	14.88	−28.89%
5	22.74	12.82	36.39	8.04	23.22	17.45	34.87	6.18	−2.11%
6	24.79	7.24	60.66	14.87	14.94	11.56	18.39	2.26	39.76%
7	18.14	5.59	26.77	6.64	11.14	4.10	14.53	2.87	38.60%
8	14.02	4.03	19.74	5.59	9.42	6.02	16.64	3.30	32.81%
9	17.83	8.56	22.46	4.85	4.72	**0.00**	9.91	3.74	73.52%
10	19.68	10.43	30.38	5.61	3.07	**0.00**	9.29	3.10	84.38%

Table 4.3: Comparison of DE-PL and DE-D-S solver solutions. This table assumes the same reporting conventions as Table 4.2.

Sequential algorithm results

To counteract the degradation of solutions for larger configurations, the sequential approach presented in Section 4.2.2 was benchmarked in a similar setting. In Table 4.3, performance of the direct sequential DE solver (DE-D-S) is compared against DE-PL. Interestingly, DE-D-S performs much better than DE-PL for configurations containing many walls, providing improvements as large as 84%. This result indicates the difficulty in optimizing configurations of multiple structural mitigation measures simultaneously, which may lead to a worse objective when executing the previous algorithms with more measures. Note, however, that the sequential approach generally does not provide improvements over DE-PL for configurations consisting of three, four, and five walls. These results indicate that sequential optimization is most beneficial when the number of possible structural measures becomes much larger (e.g., greater than five).

Finally, a comparison between DE-D-S and the sequential DE-PL solver (DE-PL-S) is made in Table 4.4. On average, DE-PL-S provides a 24% improvement over its direct counterpart. The sequential DE-PL solver is also capable of finding a solution which completely mitigates the flood using a smaller structural budget. That is, the direct sequential solver finds a totally mitigating solution at $m = 9$, but DE-PL-S accomplishes this for $m = 8$. Interestingly, however, for $m = 10$, DE-D-S finds a totally mitigating solution, whereas DE-PL-S only finds a *nearly* mitigating solution. This again may be

	DE-D-S				DE-PL-S				Mean
m	Mean	Min	Max	SD	Mean	Min	Max	SD	Imprvmt.
1	162.18	**158.59**	167.59	3.85	163.61	159.63	167.59	4.19	−0.88%
2	102.38	100.97	103.68	0.87	98.25	86.78	105.84	7.45	4.04%
3	64.98	48.97	81.93	14.71	56.74	35.73	86.12	13.92	12.69%
4	41.04	26.07	58.60	14.88	27.18	**14.55**	56.37	11.81	33.77%
5	23.22	17.45	34.87	6.18	16.32	**8.38**	25.35	5.25	29.73%
6	14.94	11.56	18.39	2.26	10.63	**3.92**	16.96	5.29	28.82%
7	11.14	4.10	14.53	2.87	7.02	**0.13**	15.00	5.08	36.97%
8	9.42	6.02	16.64	3.30	4.53	**0.00**	9.68	3.81	51.90%
9	4.72	**0.00**	9.91	3.74	3.36	**0.00**	7.86	3.43	28.83%
10	3.07	**0.00**	9.29	3.10	2.59	0.00	6.30	2.60	15.76%

Table 4.4: Comparison of DE-D-S and DE-PL-S solver solutions. This table assumes the same reporting conventions as Table 4.2.

a consequence of the relatively small number of experiments performed. Overall, except for small m (i.e., $m = 1$), the pathline-based sequential approach appears to be highly superior to the direct sequential approach. This result indicates that DE-PL-S could serve as a good general purpose OFMP solver for realistically-sized flood scenarios.

Figure 4.6 displays the ten incremental configurations obtained via DE-PL-S for $m = 10$ and the random seed that gave the minimum corresponding objective in Table 4.4. The ultimate solution for $m = 10$ shows remarkable similarity to the solution obtained via DE-PL for $m = 8$, as shown in Figure 4.4. That is, both solutions appear to exploit the critical depth boundary condition to divert water outside of the domain's uppermost boundary. However, the sequential solution appears to place a larger number of walls in more intuitive locations. Similarly, as displayed by the solution for $m = 10$ shown in Figure 4.7, DE-D-S also produces a configuration which diverted flow out of the domain's uppermost boundary, although one wall is placed extraordinarily near this boundary. We remark that such mitigation solutions may not be possible when using the pathline approach, as pathlines typically do not reside near the domain boundaries.

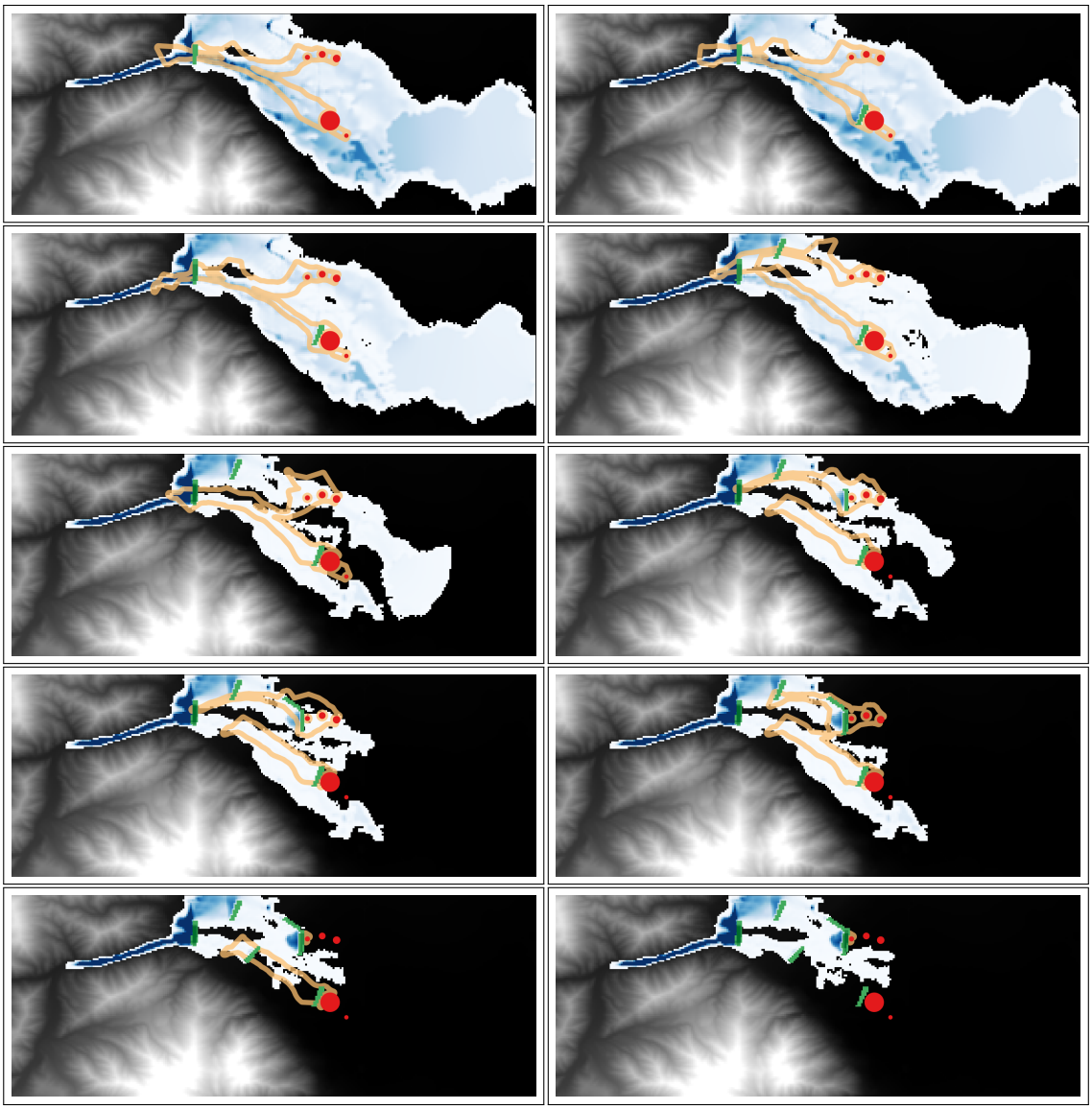


Figure 4.6: Pathline-sequential solutions for the realistic dam failure. *Best* elevations and maximum depths when using DE-PL-S are shown.

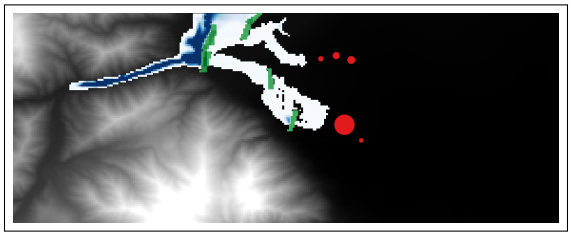


Figure 4.7: Direct-sequential ten-wall solution for the realistic dam failure.

Summary of algorithm comparisons

Tables 4.1 through 4.4 compare the performance of solvers against one another. Within these tables, the best objectives over *all* seeds and solvers are denoted in bold, while the best *mean* objectives are underlined. It is first apparent that for $m \in \{1, 2\}$, minimum objectives are obtained through use of DE-D. Good mean objectives are also obtained using this solver. This result indicates that direct local search algorithms are capable of performing well on OFMPs that contain a small number of structural measures. It also implies that more careful tuning of these solution algorithms may hold great promise.

For $m = 3$, DE-PL performs most favorably, providing the best overall and best mean objectives. This implies for a moderate number of structural measures, DE-PL effectively uses pathlines to restrict the search space. Moreover, if the optimal solution is nonincremental, it is capable of finding solutions that sequential approaches cannot. However, for $m > 3$, DE-PL-S performs most favorably, indicating a combination of pathline-based and sequential approaches are needed to solve challenging problems.

Proof of concept for soft structural mitigation measures

Sections 4.3.3 through 4.3.4 focus on OFMPs designed to configure the placement of *hard* structural mitigation measures (i.e., $\bar{b}_i > 0$ and $\bar{n}_i = 0$). However, it is important to emphasize that the problem formulations and techniques described throughout Sections 4.1, 4.2, and 4.3.1 are not limited to such measures. To exemplify this, a proof of concept employing only soft structural measures is assessed. In particular, an OFMP taking the form of Equations (4.21a) through (4.21g) is proposed that optimizes the configuration of m revegetation projects (i.e., $\bar{n}_i > 0$ and $\bar{b}_i = 0$), differing from previous experiments.

Using the ICOLD 2013 scenario, the above problem was constructed for a number of revegetation projects ranging from one to ten. Each revegetation project was assumed to have a radius of 250 meters and increased the Manning's roughness coefficient in the project region from 0.035 to 0.123. An experimental setting equivalent to that described in Section 4.3.2 was used. However, in these experiments, only the DE and DE-PL solvers were compared. Furthermore, only a single random seed was used.

In Figure 4.8, for each experiment, the objective behavior is plotted against the number of PDE evaluations required to reach that objective value. The DE-PL solver is generally able to improve upon solutions more rapidly, especially for configurations in-

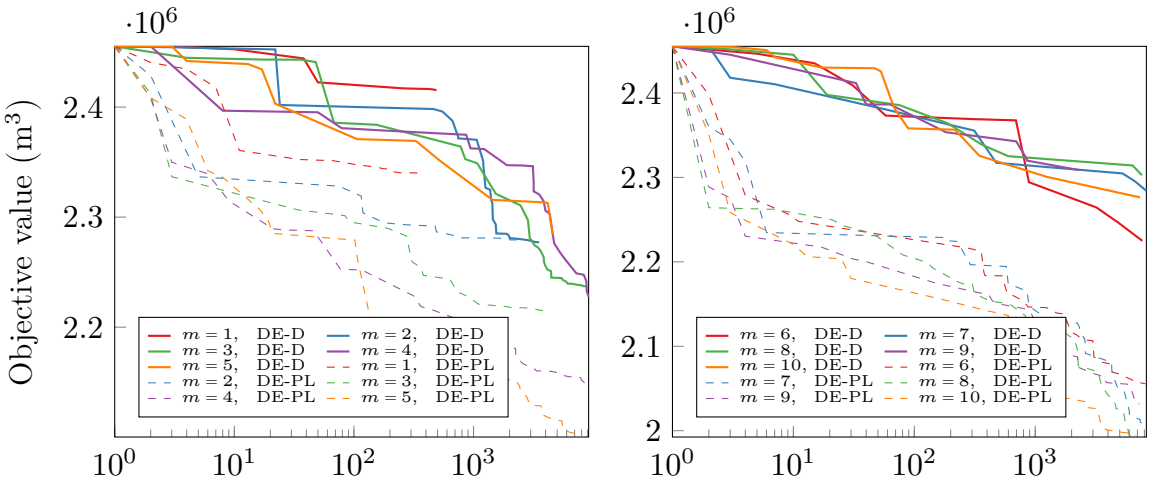


Figure 4.8: Revegetation algorithm convergence for the realistic dam failure. Here, the objective values versus numbers of PDE evaluations are compared for the OFMP scenario in Section 4.3.4, using DE-D and DE-PL to optimize positions of one through ten revegetation projects.

volving larger numbers of revegetation projects. These results mimic the behaviors of Figure 4.2, Table 4.1, and Table 4.2. That is, for smaller numbers of projects, the direct algorithm is sufficient, but for larger numbers of projects, the pathline-based algorithm is needed to obtain meaningful mitigation solutions in a reasonable amount of time.

Finally, in Figure 4.9, the configurations using DE-PL are displayed pictorially for all pairs of scenarios and numbers of projects. The results are highly intuitive upon greater inspection. First, many of the projects appear to be placed in locations that interdict the initial flood wave. More interestingly, many are located along the primary channels of the scenario domain, where larger velocities would occur. This makes sense, as the bed shear stress source terms in Equations (4.1) are proportional to the square of velocity. Measures that increase roughness are thus highly beneficial in these regions.

4.4. Conclusion

This chapter addressed the difficult problem of designing structural flood risk management strategies for use within the risk assessment process. To this end, an optimization-based decision support approach was proposed for designing mitigation strategies. A number of numerical methodologies were developed that generally function through

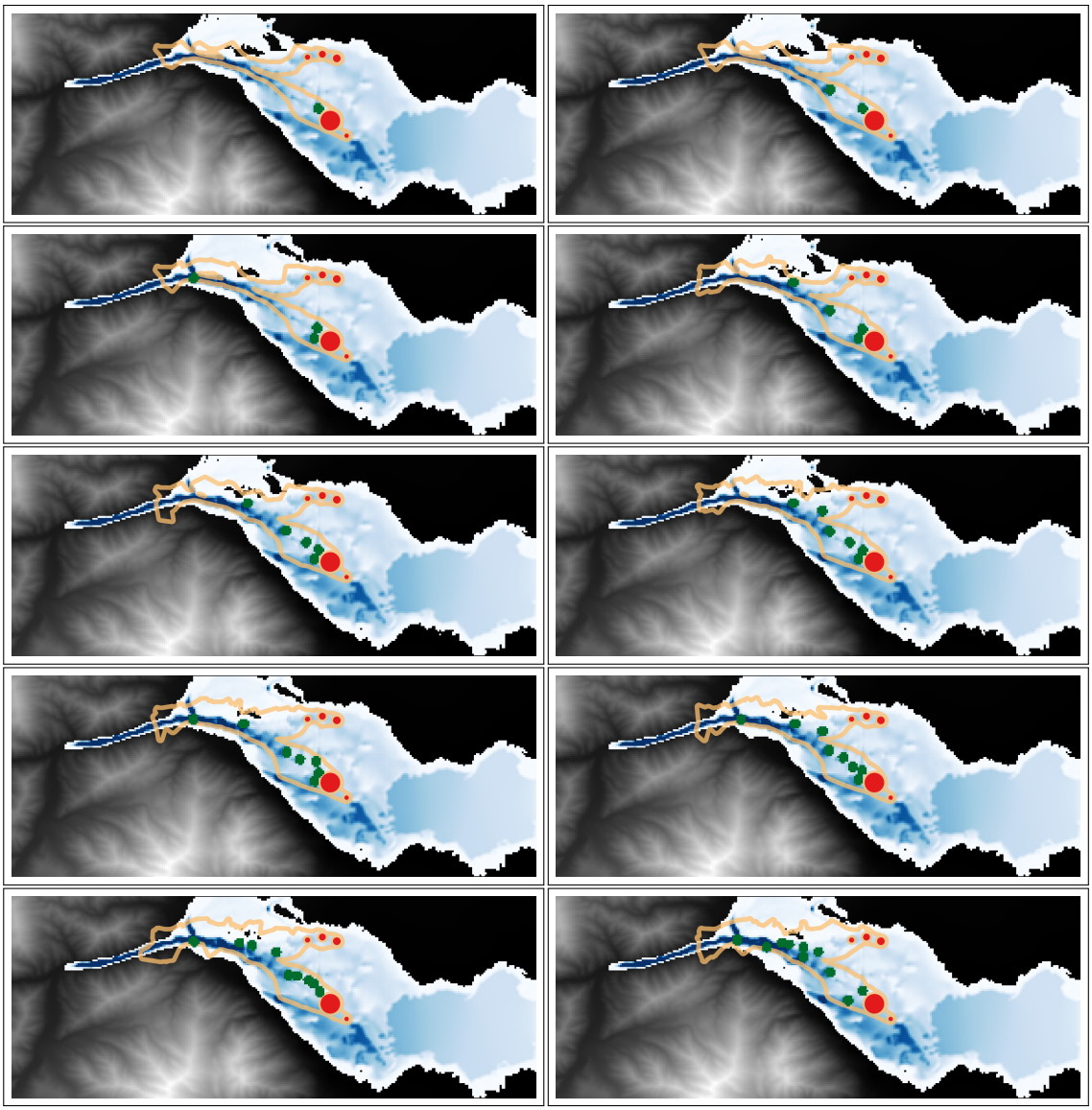


Figure 4.9: Pathline-based revegetation solutions for the realistic dam failure.

modifying the bed slope and bed shear stress source terms of the 2D shallow water equations. However, the methodologies are sufficiently general to modify other source terms (e.g., adjustment of soil properties that affect $\mathbf{S}_R$ via infiltration) or even supplant the shallow water equations with a different (perhaps approximation-based) physical model.

To formalize the mitigation task, the OFMP of Chapter 3 was reformulated. To solve practical problems of this type, a time-limited search-based optimization algorithm was

developed. Within this algorithm, three approaches to generate solutions were explored: a direct approach using only derivative-free optimization, an augmented approach using pathlines to restrict the search space, and a sequential optimization approach. The latter two were largely successful, depending on the number of mitigation measures defined in the OFMP. Overall, the non-sequential and sequential pathline-based differential evolution approaches provided average improvements of 59% and 65% over their direct counterpart, respectively. Results illustrate the first meaningful solutions to large-scale optimization problems of this type, vastly improving on the methodology of Chapter 3. However, unlike Chapter 3, in this chapter, we note that the metaheuristic nature of the algorithm that was developed ultimately provides no solution bounds nor guarantees.

Future work should seek to increase and prove the applicability of the approach to realistic flood scenarios. First, it should seek to generalize the approach by benchmarking performance on a greater number of real-world flood scenarios. Second, it should address the inherent uncertainty in flood scenario parameterizations (e.g., topographic elevation, dam breach parameterization, bed friction). To this end, a stochastic optimization approach could be developed to ensure solutions are distributionally robust from a planning perspective. Third, a human behavioral study should be conducted to compare the utility of the optimization approach presented herein with the typically manual process used in simulation-based mitigation design. Fourth, algorithmic enhancements should be made to increase the realism of mitigation designs. For example, flood walls used in the numerical experiments were considered overtoppable. This may not be realistic from a flood risk management perspective. Such realism can be embedded within the optimization problem in the form of additional penalties (e.g., when walls are overtopped, a penalty is introduced) or additional constraints. Finally, the approach should be extended to solve OFMPs for scenarios that require modeling at finer spatial resolutions. To accomplish this, a multi-resolution approach could be developed, where the spatial resolution of a flood scenario is iteratively refined as optimization progresses. Such a contribution would be valuable for realistic scenarios, where fine resolution details are sometimes necessary to accurately predict flooding behavior.

Chapter 5

Exact Mixed-integer Convex Programming Formulation for Optimization of Potable Water Distribution Network Design

Chapters 3 and 4 exemplify the challenges encountered when solving practical infrastructure optimization problems constrained by the transient dynamics that govern large-scale systems of fluids. However, as discussed in Chapters 1 and 2, many important infrastructures comprise *networked* systems of fluids that can be well-approximated using *steady-state* fluid dynamical models. Although these fluid models are simpler, optimization problems involving these systems are still challenging due to (i) discrete decisions that must be determined in many problem classes; (ii) controllable system components that must be modeled using discrete variables; and (iii) the nonlinear nonconvexity of Bernoulli-inspired energy equations. For the remaining technical chapters of this dissertation (i.e., Chapters 5 through 8), we forgo the PDE-constrained problems of Chapters 3 and 4 and instead consider problems involving networked fluid systems at steady state.

The first such systems we consider are potable water distribution networks. The efficient transport of potable water has challenged human civilization for millenia. To address this challenge, the development of distribution networks has played a vital role in supplying water to domestic demands since the third millenium BCE [90]. With the perpetual growth and migration of populations, new water distribution systems

are continually being developed, and existing systems are similarly being expanded. Designing and expanding these systems, however, are often highly complex tasks that involve selecting sizes, locations, and operational statuses of system components while also considering their physical limitations and minimizing cost [87]. These decisions are highly combinatorial and, furthermore, constrained by the nonconvex physics that describe the transport of water through circular pipes subject to frictional losses. These properties pose considerable challenges for water network planning, even in modernity.

This chapter considers the problem of optimal water network *design*, where the layout of the network is known, and the diameter and material of each pipe must be selected from a discrete set to minimize cost while satisfying fixed demand. The canonical design problem presented throughout the literature excludes common operational components (e.g., pumps), and all demand is assumed to be gravity-fed. This chapter focuses on the development of mathematical programming (nonheuristic) solution techniques for this problem. However, because of the nonconvexities that appear in the physics of water systems, naive mathematical programming formulations quickly become intractable. To address this, the problem is typically relaxed via convexification. Unfortunately, solutions to relaxed problems may be infeasible with respect to the full nonconvex physics.

To address the challenges associated with nonconvex network analysis, [29] introduced a convex program for determining the feasibility of a fixed network governed by linear conservation laws and nonlinear loss relationships. This was later revived by [36], who developed the convex "Content" and "Co-Content" Models for the analysis of pipe networks. Later, [137] developed a gradient method for the Content Model, a version of which is still employed by the de facto standard for water network analysis, EPANET [116]. These studies have established the importance of convexity in network *simulation*.

From a network *optimization* perspective, [110] and others (as reviewed by [87]) have used Content-like models within simulation-based techniques, whereby candidate solutions generated during optimization are *checked* for feasibility. However, with respect to *formulating* network decision problems, a convex Content-like model has never been directly embedded within a different network problem. This has likely been for three reasons: (i) the Content and Co-Content Models are in the space of two different variable sets; (ii) the models *do not* include engineering bounds on these variables; and (iii) the models require minimization of a convex objective. In this chapter, we present a novel convex system for network analysis that eliminates these limitations. This allows

the system to be convexly embedded within other nonlinear network decision problems.

This chapter explores the potential of this convex network analysis reformulation and applies it within the context of water network design. Its primary contributions are

- A convex embedding of *all* constraints and bounds for a gravity-fed pipe network;
- An *exact* mixed-integer convex formulation of the optimal network design problem;
- Application of the formulation within a relaxation-based network design algorithm;
- A comparison of this algorithm with the previous state of the art (i.e., [110]);
- New objective bounds for open benchmark instances throughout the literature.

Broadly, these contributions could be generalized to network design and expansion planning problems for other important network types exhibiting nonlinear relationships among nodal potentials and flows (e.g., natural gas and oil transmission networks).

Chapter Overview As this chapter builds heavily upon previous techniques developed over several decades, it is divided into two parts: the first reviews foundational theoretical and algorithmic work, and the second details our new contributions. As background, Section 5.1 reviews optimization techniques for water and other similar network problems, Section 5.2 formulates the water network design problem as a MINCP, and Section 5.3 summarizes the contributions of [110]: (i) their MICP relaxation, (ii) the further relaxation that forms a MILP, and (iii) the outline of a MILP-based algorithm. The remaining sections detail this chapter's contributions: Section 5.4 derives a convex description of feasible designs, then reformulates the optimal design problem *exactly* as an MICP; Section 5.5 augments an algorithm similar to [110] with novel outer approximations based on the new MICP; Section 5.6 compares the new and previous algorithms using standard instances from the literature; and Section 5.7 concludes the chapter.

5.1. Literature Review

Nonlinear networks refer to a class of networks in which (i) flow is driven by potentials and (ii) potential loss along an edge is a nonlinear function of flow. Network types with these properties include potable water, natural gas, and crude oil. Because of their mathematically similar descriptions, optimization methods developed for any of these networks are often easily adapted to the others. For over fifty years, a variety of techniques have been employed to solve optimization problems that involve these

network types [110]. Comprehensive literature reviews of solution techniques used for optimal water network operation and design are provided by [88] and [87], respectively, both of which are dominated by metaheuristic methods. A similar review of solution techniques for problems involving natural gas networks is provided by [115]. Finally, [118] review methods for optimizing various aspects of the crude oil supply chain.

Outside of mathematical programming, the predominant approaches used for solving water network design problems have been heuristic techniques based on simulation optimization. Indeed, as expressed in [87], "... research [has] been trapped, to some extent, in applying new metaheuristic [optimization] methods to relatively simple (from an engineering perspective) design problems, without understanding the principles behind algorithm performance." The same review specifies that 84% of the 124 studies compared use "stochastic" methods (e.g., evolutionary and genetic algorithms), 9% use "deterministic" methods (e.g., linear and nonlinear programming), and 7% use hybridized methods. This observation is further reinforced by [86], who describe the prevalence of metaheuristic optimization techniques in the water resources literature and the associated performance inconsistencies inherent with using such techniques. For these reasons, as well as due to the lack of optimality guarantees associated with metaheuristic techniques, this chapter focuses on mathematical programming (specifically, globally optimal mixed-integer nonlinear) approaches for solving water network design problems.

A number of recent studies have developed mathematical programming techniques that have proven to be effective on various nonlinear network problems. In [16], a relaxation-based method for natural gas network expansion planning is developed. In [42], a survey of methods used throughout water system optimization is provided, which includes both approximation- and relaxation-based techniques for optimal water network operation and design. In [110], a relaxation-based approach for nonlinear network design is developed, showcasing the efficacy of the method on water network design instances. Our work considers the algorithm therein to be the state of the art for global nonlinear network design. As such, the algorithm of [110] serves as a foundation for the one developed in this chapter, where only the selection of cutting planes (or "cuts") differs.

In conjunction with recent advances in relaxation-based methods for optimizing over nonlinear network constraints, similar developments have been made in formulating cuts that strengthen these relaxations. For example, [72] develop a number of valid inequalities for nonlinear network design problems and briefly describe their potential

for water networks. Similarly, [73] extend this work for natural gas networks. In both cases, valid cuts are derived from the solution of a nonconvex program and several auxiliary problems. In contrast, the cuts in this chapter are trivially derived from the novel mixed-integer convex reformulation of the original water network design problem.

The notion of a convex feasibility problem for nonlinear network analysis, originating with [29], applied by [36], and exploited computationally by [137] and [110], is also discussed by [45]. Here, a convex reformulation of the optimal natural gas transmission problem under restrictive assumptions is presented. A number of properties of this problem are described, including solution uniqueness and physical interpretation. Unlike this chapter, the problem they consider is not discrete, and their contributions appear to have gone unused throughout the natural gas optimization literature. Moreover, as elaborated upon in Section 5.4.1, our convex reformulation for network analysis is more general than prior studies, as it includes all physical variables, bounds, and no objective.

Finally, during the writing of this chapter, it was brought to our attention that similar but unpublished independent results appear in [111, Chapter 5], with the author of [110] as a collaborator. The thesis begins by recasting a bilevel formulation of the design problem as a single-level optimization problem. From this single-level problem, a convex inequality is derived that enables the writing of the design problem exactly as an MICP. Finally, the thesis proposes an outer approximation of their inequality within an algorithm similar to that of [110], finding positive results on smaller design instances, but concluding their method is only marginally better than, or on par with, [110].

5.2. Problem Formulation

5.2.1. Notation for Sets

A water distribution network is represented by a directed graph $\mathcal{G} := (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes (i.e., junctions and reservoirs) and $\mathcal{A}$ is the set of arcs (i.e., pipes). Herein, the set of reservoirs (i.e., source nodes with fixed potentials) is denoted by $\mathcal{S} \subset \mathcal{N}$ and the set of junctions by $\mathcal{J} \subset \mathcal{N}$. Junctions are modeled as demand nodes (where the demand for flow is nonnegative) and, without loss of generality, $\mathcal{S} \cap \mathcal{J} = \emptyset$. The set of arcs incident to node $i \in \mathcal{N}$ where i is the tail (respectively, head) of the arc is denoted by $\delta_i^+ := \{(i,j) \in \mathcal{A}\}$ (respectively, $\delta_i^- := \{(j,i) \in \mathcal{A}\}$). All arcs incident to

a source node $i \in \mathcal{S}$ are assumed to be outgoing, i.e., $\delta_i^- = \emptyset$ and $\delta_i^+ \neq \emptyset$, and arcs with tails at $i \in \mathcal{J}$ have heads also at demand nodes. Finally, the design problem of Section 5.2.3 involves selecting from a set of resistances $\mathcal{R}_a := \left\{p_1, p_2, \ldots, p_{|\mathcal{R}_a|}\right\}$ for each $a \in \mathcal{A}$.

5.2.2. Physical Feasibility

This section describes the variables and constraints required to model the physics of gravity-fed water distribution networks given a *fixed* selection of pipe resistances r_a, $a \in \mathcal{A}$. In the constraints that follow, q_a, $a \in \mathcal{A}$, denote variables representing the flow of water across each arc (expressed as a volumetric flow rate in m^3/s). Nodal potentials are denoted by the variables h_i, $i \in \mathcal{N}$, where each represents the total hydraulic head in units of length (m). The total hydraulic head (hereafter referred to as "head") assimilates elevation and pressure heads at a node, while the velocity head is neglected.

Flow Bounds When q_a is positive (negative), flow on arc $a := (i, j)$ travels from node i to j (j to i). Flow is often bounded by physical capacity or network analysis. Herein,

$$\underline{q}_a \leq q_a \leq \overline{q}_a, \ \forall a \in \mathcal{A}. \tag{5.1}$$

The maximum speed of flow along arc $a \in \mathcal{A}$, $\overline{v}_a$, is often used to estimate the bounds $\underline{q}_a = -\frac{\pi}{4}\overline{v}_a D_a^2$ and $\overline{q}_a = \frac{\pi}{4}\overline{v}_a D_a^2$, where D_a is the fixed diameter of the pipe $a \in \mathcal{A}$.

Head Bounds For each reservoir $i \in \mathcal{S}$, the head is fixed at a constant value h_i^s, i.e.,

$$h_i = h_i^s, \ \forall i \in \mathcal{S}. \tag{5.2}$$

For each junction $i \in \mathcal{J}$, a predefined minimum head $\underline{h}_i$ must be satisfied. Upper bounds on heads can also be provided or implied by network data. For example, we assume

$$\underline{h}_i \leq h_i \leq \overline{h}_i = \max_{j \in \mathcal{S}}\{h_j^s\}, \ \forall i \in \mathcal{J}. \tag{5.3}$$

Conservation of Flow at Demand Nodes Flow must be delivered throughout the network in order to satisfy fixed demand, d_i, at all demand nodes $i \in \mathcal{J}$. That is,

$$\sum_{a \in \delta_i^-} q_a - \sum_{a \in \delta_i^+} q_a = d_i, \ \forall i \in \mathcal{J}. \tag{5.4}$$

Head Loss Relationships Flow along an arc is induced by the difference in head between the two nodes connected by that arc. The relationships that link flow and head are commonly referred to as the "head loss equations" and are generally of the form

$$h_i - h_j = \phi_a(q_a), \ \forall a := (i,j) \in \mathcal{A}, \tag{5.5}$$

where $\phi_a : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function with rotational symmetry about the origin. The most common head loss relationships include the Darcy-Weisbach equation,

$$h_i - h_j = \frac{8 L_a \tau_a q_a |q_a|}{\pi^2 g D_a^5}, \tag{5.6}$$

and the Hazen-Williams equation (where the constant 10.7 is in standard units),

$$h_i - h_j = \frac{10.7 L_a q_a |q_a|^{0.852}}{\kappa_a^{1.852} D_a^{4.8704}}. \tag{5.7}$$

Here, L_a is the pipe length, τ_a is the friction factor, g is gravitational acceleration, and κ_a is the roughness, which depends on the pipe material. A more thorough discussion of the history of these equations is presented in Chapter 2. In Equation (5.6), τ_a depends on q_a in a nonlinear manner. However, in the mathematical programming literature, τ_a is often fixed to a constant, which removes the term's nonlinearity in q_a [61], [140].

When all terms *except* h_i, h_j, and q_a are fixed, both head loss equations reduce to

$$h_i - h_j = L_a r_a q_a |q_a|^{\alpha - 1}, \ \forall a := (i,j) \in \mathcal{A}. \tag{5.8}$$

Here, α denotes the exponent required by Equation (5.6) or (5.7), and r_a, $a \in \mathcal{A}$, denotes the resistance per unit length. The resistance per unit length comprises all non-length constant terms appearing in Equations (5.6) and (5.7) and is in units of $(\mathrm{m}^3/\mathrm{s})^{-\alpha}$.

For fixed resistances r, the nonconvex formulation for water network feasibility is thus

$$\begin{aligned} &\text{Physical bounds: Constraints (5.1), (5.2), (5.3)} \\ &\text{Flow conservation: Constraints (5.4)} \\ &\text{Head loss relationships: Constraints (5.8).} \end{aligned} \qquad (\text{NLP}(r))$$

5.2.3. Optimal Network Design

To formulate the optimal water network design problem, (NLP(r)) is coupled with the combinatorial problem of selecting one resistance from a predefined set of resistances for each arc, $\mathcal{R}_a$ for $a \in \mathcal{A}$, while minimizing the overall cost of network design. To model the disjunction representing discrete resistance choices, each q_a is first decomposed into $|\mathcal{R}_a|$ binary variables $z_{ap} \in \{0,1\} = \mathbb{B}$ and continuous variables $q_{ap} \in \mathbb{R}$. This implies

$$q_a := \sum_{p \in \mathcal{R}_a} q_{ap}, \ \forall a \in \mathcal{A}, \tag{5.9}$$

$$\underline{q}_{ap} z_{ap} \leq q_{ap} \leq \overline{q}_{ap} z_{ap}, \ z_{ap} \in \mathbb{B}, \ \forall a := (i,j) \in \mathcal{A}, \ \forall p \in \mathcal{R}_a. \tag{5.10}$$

Here, $z_{ap} = 1$ when $p \in \mathcal{R}_a$ is selected as the resistance and is zero otherwise. From Constraint (5.10), it follows that q_{ap} is nonzero only when $z_{ap} = 1$. Also note that $\underline{q}_{ap} = -\frac{\pi}{4}\overline{v}_a D_{ap}^2$ and $\overline{q}_{ap} = \frac{\pi}{4}\overline{v}_a D_{ap}^2$ are often used as flow bounds, as each $p \in \mathcal{R}_a$ is typically derived from a unique pipe diameter D_{ap}. Furthermore, since only one resistance may be selected per pipe indexed by $a \in \mathcal{A}$, we include the additional constraints

$$\sum_{p \in \mathcal{R}_a} z_{ap} = 1, \ \forall a \in \mathcal{A}. \tag{5.11}$$

The head loss Constraints (5.8) are then expanded to formulate the constraints

$$h_i - h_j = L_a \sum_{p \in \mathcal{R}_a} p q_{ap} |q_{ap}|^{\alpha - 1}, \ \forall a := (i,j) \in \mathcal{A}. \tag{5.12}$$

Finally, the objective function, $\eta(x)$, for the optimal design problem is written as

$$\eta(x) = \sum_{a \in \mathcal{A}} L_a \sum_{p \in \mathcal{R}_a} c_{ap} z_{ap}, \tag{5.13}$$

where c_{ap} is the cost per unit length of installing a pipe along $a \in \mathcal{A}$ with $p \in \mathcal{R}_a$.

These modifications allow the optimal water network design problem to be written as

$$\begin{array}{ll} \text{minimize} & \text{Objective function: } \eta(x) \text{ of Equation (5.13)} \\ \text{subject to} & \text{Physical bounds: Constraints (5.2), (5.3), (5.10)} \\ & \text{Flow conservation: Constraints (5.4)} \\ & \text{Resistance selection: Constraints (5.11)} \\ & \text{Head loss relationships: Constraints (5.12).} \end{array} \tag{MINLP}$$

Here, Constraints (5.4) employ the definitions of q_a described in Equations (5.9). Note that (MINLP) is mixed-integer *nonconvex* because of the nonconvex Constraints (5.12). In many similar network problems, the challenge of nonconvexity is often addressed via a convex relaxation of the complicating constraints. As one example, [110] addresses this challenge via convex relaxation followed by linearization of the resulting MICP. Their method is reviewed in Section 5.3 both for context and because we ultimately employ a similar algorithm. This chapter, on the other hand, addresses this challenge via an *exact* convex reformulation of (NLP(r)) to describe feasibility, followed by an *exact* MICP reformulation of (MINLP). These are further described in Section 5.4.

5.3. Review of a Relaxation-based Algorithm

This section reviews the relaxation-based formulation and global optimization algorithm of [110] for optimal network design, which are crucial for two reasons: (i) they establish the context for this chapter's *exact* convex reformulations described in Section 5.4, and (ii) they provide useful algorithmic techniques which are later applied in Section 5.5. First, their relaxed MICP of the design problem is presented in Section 5.3.1. Then, an exact MILP reformulation of (MINLP), based on an outer approximation of the MICP, is presented in Section 5.3.2. Convex programs for determining the feasibility of a design are presented in Section 5.3.3. Finally, a simple global algorithm, which leverages the relaxed MILP formulation described in Section 5.3.2, is presented in Section 5.3.4.

5.3.1. Mixed-integer Convex Relaxation of (MINLP)

This section relaxes the nonconvex head loss constraints of (MINLP) via an outer convexification to form a relaxed MICP of the original optimal water network design problem. This is accomplished by partitioning Constraints (5.12) into their symmetric positive and negative components. To begin, the directed flow variables $q_{ap}^{\pm} \geq 0$ are introduced, denoting nonnegative flows in the two directions along arc $a \in \mathcal{A}$, with

$$q_a := \sum_{p \in \mathcal{R}_a} \left(q_{ap}^{+} - q_{ap}^{-} \right), \ \forall a \in \mathcal{A} \tag{5.14}$$

replacing Equations (5.9). Next, bound Constraints (5.10) in (MINLP) are rewritten as

$$0 \leq q_{ap}^{\pm} \leq \overline{q}_{ap}^{\pm} z_{ap}, \ z_{ap} \in \mathbb{B}, \ \forall a \in \mathcal{A}, \ \forall p \in \mathcal{R}_a, \tag{5.15}$$

where $\overline{q}_{ap}^{+} = \max\{0, \overline{q}_{ap}\}$ and $\overline{q}_{ap}^{-} = \max\{0, -\underline{q}_{ap}\}$ replace the bounds of Constraints (5.10). Nonnegative head difference variables $\Delta h_a^{\pm}$ are similarly introduced to denote head loss in the two possible flow directions. These are related to the h_i, $i \in \mathcal{N}$, via

$$\Delta h_a^{+} - \Delta h_a^{-} = h_i - h_j, \ \forall a := (i,j) \in \mathcal{A}. \tag{5.16}$$

Next, the variables $y_a \in \mathbb{B}$, $a \in \mathcal{A}$, are used to model the direction of flow along each arc, where, for $a := (i,j)$, $y_a = 1$ implies flow from i to j and $y_a = 0$ from j to i, i.e.,

$$0 \leq q_{ap}^{+} \leq \overline{q}_{ap}^{+} y_a, \ 0 \leq q_{ap}^{-} \leq \overline{q}_{ap}^{-} (1 - y_a), \ y_a \in \mathbb{B}, \ \forall a \in \mathcal{A}, \ \forall p \in \mathcal{R}_a \tag{5.17a}$$

$$0 \leq \Delta h_a^{+} \leq \Delta \overline{h}_a^{+} y_a, \ 0 \leq \Delta h_a^{-} \leq \Delta \overline{h}_a^{-} (1 - y_a), \ y_a \in \mathbb{B}, \ \forall a \in \mathcal{A}. \tag{5.17b}$$

Here, each bound $\Delta \overline{h}_a^{\pm}$ is derived from the lower and upper bounds on h in Constraints (5.3). Next, the right-hand sides in Equations (5.8) are decomposed into two convex functions representing head loss in the positive and negative directions. This implies

$$\Delta h_a^{\pm} = L_a \sum_{p \in \mathcal{R}_a} p \left(q_{ap}^{\pm} \right)^{\alpha}, \ \forall a \in \mathcal{A}. \tag{5.18}$$

Recalling that $q_{ap}^{\pm}$s are nonzero for only one $p \in \mathcal{R}_a$, Equations (5.18) are relaxed as

$$L_a p \left(q_{ap}^{\pm}\right)^{\alpha} \leq \Delta h_a^{\pm}, \ \forall a \in \mathcal{A}, \ \forall p \in \mathcal{R}_a. \tag{5.19}$$

Note that $(0,0)$ and $\left(\overline{q}_{ap}^{\pm}, L_a p \left(\overline{q}_{ap}^{\pm}\right)^{\alpha}\right)$ are endpoints of the lines that upper-bound the convex right-hand terms of Equations (5.18), with the slopes calculated as $\frac{L_a p \left(\overline{q}_{ap}^{\pm}\right)^{\alpha} - 0}{\overline{q}_{ap}^{\pm} - 0} = L_a p \left(\overline{q}_{ap}^{\pm}\right)^{\alpha-1}$. The convex relaxations in Constraints (5.19) are then upper-bounded via

$$\Delta h_a^{\pm} \leq L_a \sum_{p \in \mathcal{R}_a} \left[p \left(\overline{q}_{ap}^{\pm}\right)^{\alpha-1} q_{ap}^{\pm} \right], \ \forall a \in \mathcal{A}. \tag{5.20}$$

These constraints ultimately give rise to an MICP *relaxation* of (MINLP), namely,

minimize	Objective function: $\eta(x)$ of Equation (5.13)	(MICP-R)
subject to	Physical bounds: Constraints (5.2), (5.3), (5.15)	
	Flow conservation: Constraints (5.4)	
	Resistance selection: Constraints (5.11)	
	Head difference relationships: Constraints (5.16), (5.19), (5.20)	
	Direction-related inequalities: Constraints (5.17).	

Constraints (5.4) employ the definitions of q_a described in Equations (5.14). The validity of this relaxed formulation was proven by [110]. However, note that because (MICP-R) is a relaxation of (MINLP), network design solutions that are feasible to (MICP-R) may not be feasible with respect to the original, nonconvex head loss constraints of (MINLP).

5.3.2. Mixed-integer Linear Reformulation of (MINLP)

In [110], (MINLP) is solved via a global, relaxation-based MILP algorithm. The algorithm leverages much of (MICP-R), linear outer approximations of Constraints (5.19), and linear feasibility cuts for integer solutions (i.e., network designs) $\hat{z}$ that are infeasible to (MINLP). This section restates these cuts, then develops a MILP reformulation of (MINLP), representing the relevant aspects of the theoretical contributions from [110].

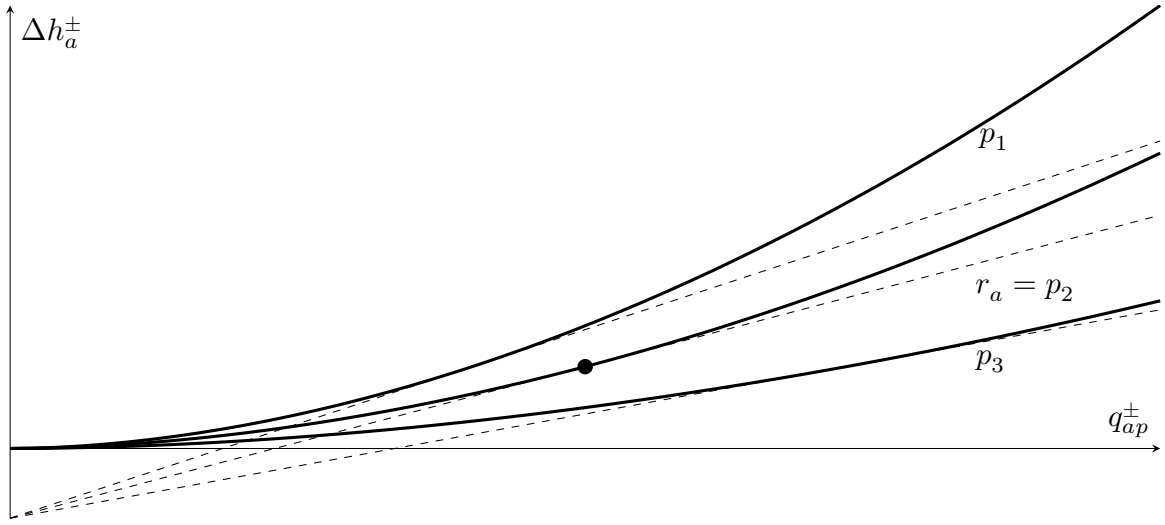


Figure 5.1: Illustration of equal intercept head loss outer approximations. Here, dashed lines correspond to equal intercept outer approximations and solid lines are convexified head loss relations for a hypothetical case where $|\mathcal{R}_a| = 3$, $r_a = p_2$, and where $\tilde{q}_{ar_a}$ is represented by the point.

Outer Approximation Cutting Planes Since Constraints (5.19) are convex, they are easily linearized via outer approximation. However, instead of applying traditional first-order outer approximations for *each* Constraint (5.19), [110] derives aggregate outer approximations based on the notion of outer-approximating lines with *equal intercepts* for all $p \in \mathcal{R}_a$. This is illustrated in Figure 5.1 for a hypothetical instance where $|\mathcal{R}_a| = 3$ and $r_a = p_2$. Note that a standard outer approximation of Constraint (5.19) for $p \in \mathcal{R}_a$, based on the first-order Taylor expansion of its left-hand side at $\tilde{q}_{ap}^{\pm}$, is

$$(1-\alpha)\, L_a p \left(\tilde{q}_{ap}^{\pm}\right)^{\alpha} + \alpha L_a p \left(\tilde{q}_{ap}^{\pm}\right)^{\alpha-1} q_{ap}^{\pm} \leq \Delta h_a^{\pm}. \tag{5.21}$$

Thus, if we fix a point $\tilde{q}_{ar_a}^{\pm} \in [0, \overline{q}_{ar_a}^{\pm}]$ for some $r_a \in \mathcal{R}_a$ and $a \in \mathcal{A}$, for the remaining $p \in \mathcal{R}_a$, the outer approximations will have the *same intercept* for $\tilde{q}_{ap}^{\pm} := \tilde{q}_{ar_a}^{\pm}(r_a/p)^{1/\alpha}$.

Recall that variables $q_{ap}^{\pm}$ become nonzero only when the corresponding resistance p is active (i.e., $z_{ap} = 1$). Also note that, since only one flow direction is chosen per arc in (MICP-R), cuts can be strengthened through multiplication of constants with affine expressions of y_a. Exploiting these properties, aggregating over all $r_a \in \mathcal{R}_a$ and $\tilde{q}_{ar_a}^{\pm}$

allows the rewriting of Constraints (5.19) in the resistance-aggregated form as

$$\tilde{\tau}^{+}_{ar_a} y_a + \alpha \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{+}_{ap}\right)^{\alpha-1} q^{+}_{ap} \leq \frac{\Delta h^{+}_a}{L_a}, \; \forall a \in \mathcal{A}, \; \forall r_a \in \mathcal{R}_a, \; \forall \tilde{q}^{+}_{ar_a} \in \mathcal{Q}^{+}_{ar_a} \tag{5.22a}$$

$$\tilde{\tau}^{-}_{ar_a} (1 - y_a) + \alpha \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{-}_{ap}\right)^{\alpha-1} q^{-}_{ap} \leq \frac{\Delta h^{-}_a}{L_a}, \; \forall a \in \mathcal{A}, \; \forall r_a \in \mathcal{R}_a, \; \forall \tilde{q}^{-}_{ar_a} \in \mathcal{Q}^{-}_{ar_a}, \tag{5.22b}$$

where $\tilde{\tau}^{\pm}_{ar_a} := (1-\alpha)\, r_a \left(\tilde{q}^{\pm}_{ar_a}\right)^{\alpha}$ and $\mathcal{Q}^{\pm}_{ar_a} = [0, \overline{q}^{\pm}_{ar_a}]$. As in [110], each Constraint (5.22) is *stronger* than the set of standard disaggregated cuts that outer-approximate Constraints (5.19). Also, as shown in Section 5.3.4, algorithmically, $\mathcal{Q}^{\pm}_{ar_a}$ can be replaced by the finite sets $\tilde{\mathcal{Q}}^{\pm}_{ar_a}$ to linearly *approximate* (MICP-R) rather than reproduce it.

Feasibility Cutting Planes Since (MICP-R) is a relaxation of (MINLP), a solution to (MICP-R) is not guaranteed to be physically feasible. To address this, let $\bar{\mathcal{X}}$ denote the set of designs represented by binary vectors $\bar{z}$ satisfying Constraints (5.11) that are *not* physically feasible. Then, any infeasible designs will be excluded by the set of cuts

$$\sum_{a \in \mathcal{A}} \left[\left(\sum_{p \in \mathcal{R}_a : \bar{z}_{ap} = 1} z_{ap} \right) - \left(\sum_{p \in \mathcal{R}_a : \bar{z}_{ap} = 0} z_{ap} \right) \right] \leq |\mathcal{A}| - 1, \; \forall \bar{z} \in \bar{\mathcal{X}}. \tag{5.23}$$

Each Constraint (5.23) is a combinatorial "no good" cut, which removes *one* combination of resistances (i.e., one network design) from the space of solutions feasible to (MICP-R).

Mixed-integer Linear Reformulation Combining much of (MICP-R) with the cuts described in this section, the infinite MILP reformulation of (MINLP), which abstractly

describes the theoretical contributions of and foundational formulation used by [110], is

$$\begin{array}{ll} \text{minimize} & \text{Objective function: } \eta(x) \text{ of Equation (5.13)} \\ \text{subject to} & \text{Physical bounds: Constraints (5.2), (5.3), (5.15)} \\ & \text{Flow conservation: Constraints (5.4)} \\ & \text{Resistance selection: Constraints (5.11)} \\ & \text{Head difference relationships: Constraints (5.16), (5.20), (5.22)} \\ & \text{Direction-related inequalities: Constraints (5.17)} \\ & \text{Feasibility cutting planes: Constraints (5.23).} \end{array} \tag{MIP-R}$$

Algorithmically, the sets $\mathcal{Q}^{\pm}_{ar_a}$ and $\bar{\mathcal{X}}$ of Constraints (5.22) and (5.23) are replaced by the initially empty finite sets, $\tilde{\mathcal{Q}}^{\pm}_{ar_a}$ and $\tilde{\mathcal{X}}$, and progressively augmented during the algorithm's execution. Hereafter, we refer to the relaxation of (MIP-R) as (MIP-RR). To refine this relaxation, the convex oracle algorithmically used for determining whether a design is contained in $\bar{\mathcal{X}}$ (i.e., whether a design is feasible) is described in Section 5.3.3.

5.3.3. Convex Method for Determining Design Feasibility

This section reviews the work of [29] and [36], who provide convex programs for determining the feasibility of a nonlinear network with fixed resistances r. These resistances correspond to an integer solution $\hat{z}$ satisfying Constraints (5.11) via the relationship

$$r_a \in \{p \in \mathcal{R}_a : \hat{z}_{ap} = 1\}, \ \forall a \in \mathcal{A}. \tag{5.24}$$

Thus, this feasibility-testing method can be used to determine whether $\hat{z}$ is contained in $\bar{\mathcal{X}}$. Letting $q^{\pm}_a$ denote nonnegative directed flow variables along $a \in \mathcal{A}$ and using the definition $q_a := q^+_a - q^-_a$, the studies propose a convex programming problem similar to

$$\begin{array}{ll} \underset{q^{\pm} \geq 0}{\text{minimize}} & \displaystyle\sum_{a \in \mathcal{A}} \frac{L_a r_a}{1+\alpha} \left[(q^+_a)^{1+\alpha} + (q^-_a)^{1+\alpha} \right] - \sum_{i \in \mathcal{S}} h^s_i \sum_{a \in \delta^+_i} q_a \\ \text{subject to} & \displaystyle\sum_{a \in \delta^-_i} q_a - \sum_{a \in \delta^+_i} q_a = d_i, \ \forall i \in \mathcal{J}. \end{array} \tag{P(r)}$$

Note that this problem is, at its essence, a rewriting of the Content Model of [36].

We next summarize that [36] and [110] prove three important properties of (P(r)): (i) its flow solution $\hat{q}$ is unique; (ii) the duals of flow conservation constraints correspond to unique heads $\hat{h}$; and (iii) the primal-dual solution $(\hat{q}, \hat{h})$ satisfies flow conservation and head loss Constraints (5.4) and (5.8). The ability to find a $(\hat{q}, \hat{h})$ satisfying these physical equations by solving (P(r)) is appealing because the feasibility of r can be tested by checking whether $(\hat{q}, \hat{h})$ satisfies Constraints (5.1) and (5.3). In the work of [110], this method is used to guarantee global convergence of their relaxation-based algorithm.

Note the objective of (P(r)) is convex and all constraints are affine. The linearity constraint qualification thus implies the existence of a strong dual. For completeness, this dual is rederived in Appendix B.1 using Lagrangian duality. It is expressed here as

$$\begin{aligned} \underset{\Delta h^{\pm} \geq 0}{\text{maximize}} \quad & \frac{-\alpha}{1+\alpha} \sum_{a \in \mathcal{A}} \frac{1}{\sqrt[\alpha]{L_a r_a}} \left[(\Delta h_a^+)^{1+\frac{1}{\alpha}} + (\Delta h_a^-)^{1+\frac{1}{\alpha}} \right] - \sum_{i \in \mathcal{J}} h_i d_i \\ \text{subject to} \quad & \Delta h_a^+ - \Delta h_a^- = h_i^s - h_j, \ \forall a := (i,j) \in \mathcal{A} : i \in \mathcal{S} \\ & \Delta h_a^+ - \Delta h_a^- = h_i - h_j, \ \forall a := (i,j) \in \mathcal{A} : i \in \mathcal{J}. \end{aligned} \tag{D(r)}$$

We note that this problem is essentially a rewriting of the Co-Content Model of [36].

5.3.4. Global Optimization Algorithm

This section outlines a simple algorithm for solving (MINLP) to global optimality via the iterative solution and augmentation of (MIP-RR). Algorithm 5.1 leverages the outer approximation and feasibility cuts established in Section 5.3.2. In Line 1, the outer approximation point sets and infeasible design set, $\tilde{\mathcal{Q}}^{\pm}$ and $\tilde{\mathcal{X}}$, are initialized as empty. In Line 2, the relaxed problem is solved, and the solution components $(\hat{q}, \hat{h}, \hat{z})$ are stored. In Line 3, feasibility of the design $\hat{z}$ is determined (e.g., via the method of Section 5.3.3). If the design is found to be physically infeasible, outer approximations of head loss constraints are added in Line 4. A feasibility cut that excludes $\hat{z}$ is then added in Line 5. Finally, a solution to the new relaxed problem, with the aforementioned cuts, is obtained in Line 6. These steps are repeated until a solution $\hat{z}$ to (MINLP) is identified. Since $\hat{z}$ is discovered via the solution of sequential *relaxations* of (MINLP) and the cuts do not exclude feasible solutions, the design is guaranteed to be optimal.

In practice, the algorithm developed by [110] is more sophisticated than Algorithm 5.1. Specifically, it exploits the linearization-based linear programming/nonlinear program-

Algorithm 5.1 A MILP relaxation-based global optimization algorithm for (MINLP).

1: $\tilde{\mathcal{Q}}^{\pm}_{ap} \leftarrow \emptyset,\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a$; $\tilde{\mathcal{X}} \leftarrow \emptyset$.
2: $(\hat{q}, \hat{h}, \hat{z}) \leftarrow$ Solve (MIP-RR).
3: **while** $\hat{z}$ is infeasible to (MINLP) **do**
4: $\quad \tilde{\mathcal{Q}}^{\pm}_{ap} \leftarrow \tilde{\mathcal{Q}}^{\pm}_{ap} \cup \{\hat{q}^{\pm}_{ap}\},\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a$.
5: $\quad \tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}} \cup \{\hat{z}\}$.
6: $\quad (\hat{q}, \hat{h}, \hat{z}) \leftarrow$ Solve (MIP-RR).
7: **end while**

ming branch and bound (LP/NLP-BB) framework developed by [108]. This permits more flexibility than Algorithm 5.1 through the use of "callbacks" available in MILP solvers. First, outer approximations are added in other parts of the search, not just integer solutions to (MIP-RR), as in Line 4. Moreover, outer approximation points $\tilde{\mathcal{Q}}^{\pm}$ are more thoughtfully selected. Second, [110] develops heuristics internal to the search that recover feasible solutions from fractional and integer solutions *infeasible* to (MINLP).

In this chapter, contributions from [110] are used to devise a new algorithm, which is described in Section 5.5. The two algorithms primarily differ in the selection of outer approximation cuts. Prior to developing this algorithm, Section 5.4 describes the novel contributions that eventually lead to these new cuts, which are derived and applied in Section 5.5. Note that a thorough algorithmic description is presented in Appendix B.4.

5.4. Convex Reformulations for Analysis and Design

Whereas this chapter has thus far discussed important *past* contributions in water network analysis and design, the remainder of this chapter describes *our new* contributions. For context, Section 5.3.3 summarized a convex system for determining the feasibility of a nonlinear network with fixed resistances. This method could be exploited within a bilevel programming formulation for optimal design, whereby resistance selections obtained in the outer level must satisfy constraints on the corresponding solution of (P(r)) (i.e., the inner level). Indeed, such a bilevel method was developed by [149] for the design of pipe networks. Interestingly, however, no study has fully examined the relationship between (P(r)) and its (strong) dual (D(r)), which we show leads to exact convex reformulations of the original feasibility and design problems. This section describes two

of our novel contributions, both of which serve as foundations for the remainder of this chapter. The first subsection derives an *exact* convex reformulation of (NLP(r)). The second subsection extends this to derive an *exact* MICP reformulation of (MINLP).

5.4.1. Convex Reformulation of (NLP(r))

Note that (P(r)) and (D(r)) have three limitations for our purposes: (i) they are *either* in the variable space of flows *or* in the space of heads; (ii) they do not include variable bounds, which are required for modeling pipe capacities and minimum pressures; and (iii) they require minimization or maximization of an objective. This makes directly embedding (P(r)) or (D(r)) within a separate decision problem (e.g., the optimal design problem) a challenging task. To address this limitation, this section extends Section 5.3.3 to derive an *exact* convex reformulation of (NLP(r)) based on the strong duality conditions between (P(r)) and (D(r)) and the further addition of physical bounds.

Theorem 5.1. *Let $f_P(q)$ and $f_D(h)$ denote the objective functions of* (P(r)) *and* (D(r)), *respectively, and $q_a := q_a^+ - q_a^-,\ \forall a \in \mathcal{A}$. The following system is equivalent to* (NLP(r)):

$$
\begin{aligned}
& f_P(q) - f_D(h) \leq 0 \\
& \sum_{a \in \delta_i^-} q_a - \sum_{a \in \delta_i^+} q_a = d_i,\ \forall i \in \mathcal{J} \\
& \Delta h_a^+ - \Delta h_a^- = h_i^s - h_j,\ \forall a := (i,j) \in \mathcal{A} : i \in \mathcal{S} \\
& \Delta h_a^+ - \Delta h_a^- = h_i - h_j,\ \forall a := (i,j) \in \mathcal{A} : i \in \mathcal{J} \\
& \underline{h}_i \leq h_i \leq \overline{h}_i,\ \forall i \in \mathcal{J},\ 0 \leq q_a^{\pm} \leq \overline{q}_a^{\pm},\ \Delta h_a^{\pm} \geq 0,\ \forall a \in \mathcal{A}.
\end{aligned}
\tag{CP(r)}
$$

Proof. By weak duality, it follows that $f_P(q) \geq f_D(h)$ for any feasible solutions to (P(r)) and (D(r)), with equality holding for optimal solutions by strong duality. As a result, optimality is equivalently imposed by combining constraints of (P(r)) and (D(r)) with the *convex* constraint $f_P(q) - f_D(h) \leq 0$. Next, [110] shows that head loss Constraints (5.8) of (NLP(r)) are equivalent to a portion of the first order optimality conditions for (P(r)). Moreover, [17, Section 5.5.3] ensures that, since (P(r)) is a convex differentiable problem with a strong dual, the dual multipliers h appearing in its optimality conditions are optimal solutions of the dual problem (D(r)). That is, any (q, h) that satisfies

constraints of (P(r)), (D(r)), and strong duality will satisfy Constraints (5.4) and (5.8). Finally, appending bound constraints on q and h ensures equivalence to (NLP(r)). □

A physical interpretation of the convex strong duality constraint is discussed in Appendix B.2. Summarily, the constraint implies the conservation of power (or energy), with an inequality replacing the traditional equality. Nonetheless, it is known via the strong duality argument above that any solution to (CP(r)) will indeed satisfy this constraint with equality, implying that (CP(r)) possesses an empty interior. This is different than (P(r)) and (D(r)), which themselves separately *do* possess strict interiors. We remark that conventional nonlinear programming algorithms (e.g., most interior point methods) will thus typically provide no guarantees for the solution of (CP(r)).

Theoretically and numerically, a strict interior could be achieved by employing the relaxation $f_P(q) - f_D(h) \leq \epsilon$, where ϵ is an arbitrarily small constant. It could also be achieved by minimizing $f_P(q) - f_D(h)$ instead of constraining it. However, in the context of this chapter and the algorithm of Section 5.5, (CP(r)) is never solved in practice, so we forgo these challenges. Rather, (CP(r)) is used as a theoretical tool to construct the novel MICP reformulation of the optimal design problem presented in Section 5.4.2.

Finally, we remark that the convex strong duality constraint above, $f_P(q) - f_D(h) \leq 0$, is similar to the convex valid inequality derived in the unpublished thesis of [111]. There, a convex constraint is obtained directly from a bilevel formulation of the design problem and proven to be valid using a strong duality argument. Our result differs in two ways: (i) we obtain a complete convex rewriting of the network analysis problem in (CP(r)), and (ii) their convex inequality includes nonlinear terms in the space of head differences, whereas our convex inequality includes nonlinearities over flows and head differences.

5.4.2. Mixed-integer Convex Reformulation of (MINLP)

To reformulate (MINLP) using (CP(r)), we first introduce continuous flow variables $q^{\pm}_{ap}$ and resistance selection variables z_{ap}, subject to Constraints (5.15). Next, we introduce continuous variables $\Delta h^{\pm}_{ap}$ and derive their bounds from Constraints (5.3), i.e.,

$$0 \leq \Delta h^{\pm}_{ap} \leq \Delta \overline{h}^{\pm}_{ap} z_{ap}, \;\; z_{ap} \in \mathbb{B}, \;\; \forall a \in \mathcal{A}, \;\; \forall p \in \mathcal{R}_a. \tag{5.25}$$

The constraints involving head differences in (CP(r)) are next rewritten as

$$\sum_{p\in\mathcal{R}_a}\left(\Delta h^+_{ap}-\Delta h^-_{ap}\right)=h_i-h_j,\ \forall a:=(i,j)\in\mathcal{A}:i\in\mathcal{J} \tag{5.26a}$$

$$\sum_{p\in\mathcal{R}_a}\left(\Delta h^+_{ap}-\Delta h^-_{ap}\right)=h^s_i-h_j,\ \forall a:=(i,j)\in\mathcal{A}:i\in\mathcal{S}. \tag{5.26b}$$

Finally, the strong duality constraint in (CP(r)) is expanded over all $p\in\mathcal{R}_a$ as

$$\begin{aligned}&\frac{1}{1+\alpha}\sum_{a\in\mathcal{A}}L_a\sum_{p\in\mathcal{R}_a}p\left[(q^+_{ap})^{1+\alpha}+(q^-_{ap})^{1+\alpha}\right]-\sum_{i\in\mathcal{S}}h^s_i\sum_{a\in\delta^+_i}q_a\\&+\frac{\alpha}{1+\alpha}\sum_{a\in\mathcal{A}}\sum_{p\in\mathcal{R}_a}\frac{1}{\sqrt[\alpha]{L_ap}}\left[(\Delta h^+_{ap})^{1+\frac{1}{\alpha}}+(\Delta h^-_{ap})^{1+\frac{1}{\alpha}}\right]+\sum_{i\in\mathcal{J}}h_id_i\le 0.\end{aligned} \tag{5.27}$$

These expansions of (CP(r)) give rise to the *exact* MICP reformulation of (MINLP), where Constraints (5.4) employ the definitions of q_a in Equations (5.14). Namely,

minimize	Objective function: $\eta(x)$ of Equation (5.13)
subject to	Physical bounds: Constraints (5.3), (5.15), (5.25)
	Flow conservation: Constraints (5.4)
	Resistance selection: Constraints (5.11)
	Head difference equalities: Constraints (5.26)
	Strong duality: Constraint (5.27).

(MICP-E)

We note that a similar MICP formulation is derived in the unpublished work of [111].

Theorem 5.2. *A $\hat{z}$ is feasible for* (MICP-E) *if and only if it is feasible for* (MINLP).

Proof. For any discrete network design solution $\hat{z}$ satisfying Constraints (5.11), (MINLP) reduces to (NLP(r)) and (MICP-E) to (CP(r)), with r given by Equations (5.24). For any fixed set of resistances r, (NLP(r)) and (CP(r)) are equivalent, as required by Theorem 5.1. Thus, the sets of all feasible $\hat{z}$ for (MINLP) and (MICP-E) are equal. □

As in our discussion of (CP(r)), when the integer variables of (MICP-E) are fixed, the resulting subproblem has an empty interior. In the classical presentation of many techniques that are used for solving MICPs, e.g., the generalized Benders decomposition

of [58], it is assumed that after fixing complicating variables, the resulting subproblem possesses a *nonempty* interior. For this reason, we provide no guarantees on the solution of (MICP-E) obtained via conventional direct MICP methods. Nonetheless, in Appendix B.3, we provide the convergence profile of one such method on a small instance of (MICP-E). Although it converges to the known global optimum, its convergence is slow. This inefficiency further motivates the development of the MILP relaxation-based algorithm presented in Section 5.5. As such, in practice, the algorithmic side effects of empty interiors are again avoided throughout the experiments presented in this study.

Constraint (5.27) of (MICP-E) can be viewed as a convex embedding of Constraints (5.12). Although convexity is desirable, Constraint (5.27) is also highly aggregated. In this sense, the disaggregated nonconvex Constraints (5.12) of (MINLP) or convex Constraints (5.19) of (MICP-R) may be more numerically useful. Section 5.5 uses this observation to construct a relaxation-based global algorithm based on (MICP-E) and (MICP-R) that outer-approximates both Constraint (5.27) *and* Constraints (5.19).

5.5. Cutting Planes and Algorithmic Enhancements

As discussed in Section 5.4.2, (MICP-E) is an exact reformulation of the design problem. However, it was also noted that conventional MICP solvers (e.g., BONMIN [12]) may provide no optimality guarantees on the solution of (MICP-E). Furthermore, as Appendix B.3 shows, even when direct methods are applied successfully, they converge slowly, as modern MICP solvers do not efficiently handle nonquadratic nonlinear relationships (e.g., Constraint (5.27)). On the other hand, modern MILP solvers are highly efficient but require conscientious linearizations of (MICP-E). This section pursues the latter technique, following a structure similar to Section 5.3. Specifically, it introduces novel outer approximation cuts while developing a MILP reformulation and relaxation of (MICP-E), then summarizes this chapter's extensions to the algorithm of [110].

Flow Direction-based Inequalities Although (MICP-E) does not require direction variables y_a, we nonetheless incorporate y_a to strengthen inequalities throughout our reformulation. Similar to Constraints (5.17), flows and head differences are bounded via

$$0 \leq q_{ap}^{+} \leq \overline{q}_{ap}^{+} y_a,\ 0 \leq q_{ap}^{-} \leq \overline{q}_{ap}^{-}(1 - y_a),\ y_a \in \mathbb{B},\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a \tag{5.28a}$$

$$0 \leq \Delta h_{ap}^{+} \leq \Delta \overline{h}_{ap}^{+} y_a,\ 0 \leq \Delta h_{ap}^{-} \leq \Delta \overline{h}_{ap}^{-}(1 - y_a),\ y_a \in \mathbb{B},\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a. \tag{5.28b}$$

We also employ valid inequalities to exploit a priori knowledge concerning flow directionality throughout the network. The inequalities proposed here are similar to those described by [16] in the context of natural gas network planning. The first are

$$\sum_{a \in \delta_i^{+}} y_a \geq 1,\ \forall i \in \mathcal{S}, \tag{5.29}$$

which model that at least one pipe must send water *away* from a source. The next are

$$\sum_{a \in \delta_i^{-}} y_a + \sum_{a \in \delta_i^{+}} (1 - y_a) \geq 1,\ i \in \mathcal{J} : d_i > 0, \tag{5.30}$$

which model that at least one pipe must provide water *to* each demand node. Finally,

$$\sum_{a \in \delta_i^{-}} y_a - \sum_{a \in \delta_i^{+}} y_a = 0,\ i \in \mathcal{J} : (d_i = 0) \wedge \left(\deg_i^{+} = \deg_i^{-} = 1\right) \tag{5.31a}$$

$$\sum_{a \in \delta_i^{-}} y_a + \sum_{a \in \delta_i^{+}} y_a = 1,\ i \in \mathcal{J} : (d_i = 0) \wedge \left(\deg_i^{\pm} = 2\right) \wedge \left(\deg_i^{\mp} = 0\right) \tag{5.31b}$$

model flow directionality at junctions with zero demand and degree two, with the implication that the direction of incoming flow must equal the direction of outgoing flow.

Head Loss Outer Approximation Cutting Planes Although also not required by (MICP-E), head loss relationships are used to strengthen linearized reformulations. Similar to Constraints (5.22) in (MIP-R), we express head loss outer approximations as

$$\tilde{\tau}^{+}_{ar_a} y_a + \alpha \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{+}_{ap}\right)^{\alpha-1} q^{+}_{ap} \leq \sum_{p \in \mathcal{R}_a} \frac{\Delta h^{+}_{ap}}{L_a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \tilde{q}^{+}_{ar_a} \in \mathcal{Q}^{+}_{ar_a} \tag{5.32a}$$

$$\tilde{\tau}^{-}_{ar_a} (1 - y_a) + \alpha \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{-}_{ap}\right)^{\alpha-1} q^{-}_{ap} \leq \sum_{p \in \mathcal{R}_a} \frac{\Delta h^{-}_{ap}}{L_a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \tilde{q}^{-}_{ar_a} \in \mathcal{Q}^{-}_{ar_a}. \tag{5.32b}$$

Furthermore, similar to Constraints (5.19), we upper-bound each head difference with

$$\Delta h^{\pm}_{ap} \leq L_a p \left(\bar{q}^{\pm}_{ap}\right)^{\alpha-1} q^{\pm}_{ap},\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a. \tag{5.33}$$

Strong Duality Cutting Planes A primary contribution of this chapter is the strong duality Constraint (5.27) of (MICP-E). To linearize it, we use $q^{\text{NL}}_a, \Delta h^{\text{NL}}_a \geq 0$ to approximate sums of nonlinear terms with $q^{\pm}_{ap}$ and $\Delta h^{\pm}_{ap}$ in the original constraint, i.e.,

$$q^{\text{NL}}_a = \frac{1}{1+\alpha} \sum_{p \in \mathcal{R}_a} p \left[\left(q^{+}_{ap}\right)^{1+\alpha} + \left(q^{-}_{ap}\right)^{1+\alpha}\right],\ \forall a \in \mathcal{A} \tag{5.34a}$$

$$\Delta h^{\text{NL}}_a = \frac{\alpha}{1+\alpha} \sum_{p \in \mathcal{R}_a} \frac{1}{\sqrt[\alpha]{p}} \left[(\Delta h^{+}_{ap})^{1+\frac{1}{\alpha}} + (\Delta h^{-}_{ap})^{1+\frac{1}{\alpha}}\right],\ \forall a \in \mathcal{A}. \tag{5.34b}$$

This permits a purely linear rewriting of the original strong duality constraint, namely

$$\sum_{a \in \mathcal{A}} L_a q^{\text{NL}}_a - \sum_{i \in \mathcal{S}} h^s_i \sum_{a \in \delta^{+}_i} \sum_{p \in \mathcal{R}_a} (q^{+}_{ap} - q^{-}_{ap}) + \sum_{a \in \mathcal{A}} \frac{\Delta h^{\text{NL}}_a}{\sqrt[\alpha]{L_a}} + \sum_{i \in \mathcal{J}} h_i d_i \leq 0. \tag{5.35}$$

Observe that the right-hand sides of Equations (5.34) are convex. Similar to the head loss outer approximations derived in Section 5.3.2, we follow the notion of *equal intercept* linear outer approximations to compose cuts similar to Constraints (5.32) for q^{NL}_a and Δh^{NL}_a. For q^{NL}_a, let the intercept be determined by the linear approximation to the term corresponding to $r_a \in \mathcal{R}_a$ at point $\tilde{q}^{\pm}_{ar_a}$. For each remaining $p \in \mathcal{R}_a$, the same intercept is achieved by the linear approximation at point $\tilde{q}^{\pm}_{ap} := \tilde{q}^{\pm}_{ar_a}(r_a/p)^{1/(1+\alpha)}$. With these

values in mind, the outer approximations of the convex nonlinear flow terms are then

$$\tilde{\zeta}^{+}_{ar_a} y_a + \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{+}_{ap}\right)^{\alpha} q^{+}_{ap} \leq q^{\mathrm{NL}}_{a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \tilde{q}^{+}_{ar_a} \in \mathcal{Q}^{\mathrm{NL}^{+}}_{ar_a} \tag{5.36a}$$

$$\tilde{\zeta}^{-}_{ar_a} (1 - y_a) + \sum_{p \in \mathcal{R}_a} p \left(\tilde{q}^{-}_{ap}\right)^{\alpha} q^{-}_{ap} \leq q^{\mathrm{NL}}_{a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \tilde{q}^{-}_{ar_a} \in \mathcal{Q}^{\mathrm{NL}^{-}}_{ar_a}, \tag{5.36b}$$

where $\mathcal{Q}^{\mathrm{NL}\pm}_{ar_a} = [0, \overline{q}^{\pm}_{ar_a}]$, and $\tilde{\zeta}^{\pm}_{ar_a} := \left(\frac{1}{1+\alpha} - 1\right) r_a \left(\tilde{q}^{\pm}_{ar_a}\right)^{1+\alpha}$ is defined for conciseness.

An analogous procedure is followed for $\Delta h^{\mathrm{NL}}_{a}$. Similar to Constraints (5.36), assuming $\Delta\tilde{h}^{\pm}_{ap}$ have coinciding outer approximation intercepts, the outer approximations are

$$-\xi^{\pm}_{ar_a} y_a + \sum_{p \in \mathcal{R}_a} \left(\sqrt[\alpha]{\frac{\Delta\tilde{h}^{+}_{ap}}{p}} \right) \Delta h^{+}_{ap} \leq \Delta h^{\mathrm{NL}}_{a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \Delta\tilde{h}^{+}_{ar_a} \in \mathcal{H}^{\mathrm{NL}^{+}}_{ar_a} \tag{5.37a}$$

$$-\xi^{\pm}_{ar_a} (1 - y_a) + \sum_{p \in \mathcal{R}_a} \left(\sqrt[\alpha]{\frac{\Delta\tilde{h}^{-}_{ap}}{p}} \right) \Delta h^{-}_{ap} \leq \Delta h^{\mathrm{NL}}_{a},\ \forall a \in \mathcal{A},\ \forall r_a \in \mathcal{R}_a,\ \forall \Delta\tilde{h}^{-}_{ar_a} \in \mathcal{H}^{\mathrm{NL}^{-}}_{ar_a}, \tag{5.37b}$$

where $\mathcal{H}^{\mathrm{NL}^{\pm}}_{ar_a} = [0, \Delta\overline{h}^{\pm}_{ar_a}]$, and $\xi^{\pm}_{ar_a} := \frac{\left(\Delta\tilde{h}^{\pm}_{ar_a}\right)^{1+\frac{1}{\alpha}}}{(1+\alpha)\sqrt[\alpha]{r_a}}$ is similarly defined for conciseness.

Mixed-integer Linear Reformulation With the above variables and constraints, a MILP reformulation of (MICP-E) may be written in a way similar to (MIP-R), that is,

minimize	Objective function: $\eta(x)$ of Equation (5.13)
subject to	Physical bounds: Constraints (5.3), (5.15), (5.25)
	Flow conservation: Constraints (5.4)
	Resistance selection: Constraints (5.11)
	Head difference relationships: Constraints (5.26), (5.32), (5.33)
	Direction-related inequalities: Constraints (5.28)–(5.31)
	Strong duality: Constraints (5.35)–(5.37)
	Feasibility cutting planes: Constraints (5.23).

(MIP-E)

Similarly to (MIP-R), algorithmically, the sets $\mathcal{Q}^{\pm}$, $\mathcal{Q}^{\text{NL}\pm}$, $\mathcal{H}^{\text{NL}\pm}$, and $\bar{\mathcal{X}}$ of Constraints (5.32), (5.36), (5.37), and (5.23) are replaced by the initially empty finite sets $\tilde{\mathcal{Q}}^{\pm}$, $\tilde{\mathcal{Q}}^{\text{NL}\pm}$, $\tilde{\mathcal{H}}^{\text{NL}\pm}$, and $\tilde{\mathcal{X}}$, respectively. These changes give rise to the finite relaxation (MIP-ER).

Algorithmic Enhancements Our algorithm, which is omitted for brevity but exploits (MIP-ER), is similar to that of [110] but differs in a few important respects. Primarily, the algorithm of [110] only applies outer approximations similar to Constraints (5.32), which correspond to head loss relationships. Our algorithm extends this by adding outer approximations of terms appearing in the strong duality Constraint (5.27). A more detailed comparison of the two algorithms is presented in Appendix B.4. We finally acknowledge similarity with the unpublished algorithm of [111], which also extends the algorithm of [110] with outer approximations of their valid inequality.

5.6. Computational Experiments

This section compares the convergence of our new algorithm and an algorithm based on [110]. Both were implemented in the Julia programming language using JuMP, version 0.20 [50], and version 0.1 of WaterModels, an open-source Julia package for water distribution network optimization [131]. Section 5.6.1 describes the instances, computational resources, and parameters used in the experiments; Section 5.6.2 compares the efficacy of the two algorithms by examining convergence; and Section 5.6.3 compares their performance on a set of instances obtained by scaling all demands.

5.6.1. Experimental Setup

The numerical experiments consider instances of varying sizes that appear in the water network design literature and are summarized in Table 5.1 [42], [87]. All use the Hazen-Williams head loss relationship originally defined by Equation (5.7). The set of *diameters* in each problem thus gives a set of *resistances*, each being proportional to $D_{ap}^{-4.8704}$.

The instances of Table 5.1 are divided into two classes: *moderate* instances, comprising `shamir`, `blacksburg`, `hanoi`, `foss_poly_0`, and `foss_iron`; and *large* instances, comprising `foss_poly_1`, `pescara`, and `modena`. Generally, moderate instances are solvable to optimality with both algorithms given a sufficient amount of time (i.e., sec-

Network	# Nodes	# Arcs	# Resistances	# Binary Vars.	# Designs
`shamir`	8	8	14	120	1.48×10^{9}
`blacksburg`	32	23	14	369	2.30×10^{26}
`hanoi`	33	34	6	238	2.87×10^{26}
`foss_poly_0`	38	58	7	464	1.04×10^{49}
`foss_iron`	38	58	13	812	4.06×10^{64}
`foss_poly_1`	38	58	22	1334	7.25×10^{77}
`pescara`	74	99	13	1386	1.91×10^{110}
`modena`	276	317	13	4438	1.32×10^{353}

Table 5.1: Summary of optimal water network design instances from the literature. Here, "# Arcs" is the number of arcs with $|\mathcal{R}_a| \neq 1$; "# Resistances" is $|\mathcal{R}_a|$ per *variable* pipe; "# Binary Vars." is $|z| + |y|$ for MILPs; and "# Designs" is the number of designs satisfying Constraints (5.11).

onds to hours), while large instances cannot be solved, even given substantial time (i.e., days). Each experiment began each algorithm with equivalent data, initial feasible solutions, and outer approximation points. Parameters of the two algorithms were chosen to coincide with those used by [110] and are further detailed in Appendix B.4 and B.5.

Each experiment was executed on a node containing two Intel Xeon E5-2695 v4 processors, each with 18 cores at 2.10 GHz, and 125 GB of memory. Excluding the small amount of time required by the heuristic procedure of [110] to obtain an initial feasible solution (seconds), each experiment was provided a wall-clock time of 171,900 seconds (approximately two days). For solutions of the MILPs, Gurobi 9.0.3 was used with `Cuts=0`, which disables all of Gurobi's internal cutting plane methods. For moderate instances, `Heuristics=0.0` was used, which disables Gurobi's internal heuristics. For large instances, `MIPFocus=1` was used, which places a focus on finding feasible solutions.

For convex subproblems, e.g., the solution of (P(r)), Ipopt version 3.13 was used [142]. As per [134], since these problems are small, the linear solver MA57 was employed. The settings `warm_start_init_point="yes"` and `nlp_scaling_method="none"` were also used. Heuristically, these parameters computed solutions to (P(r)) most efficiently.

5.6.2. Comparison of Algorithms on Standard Benchmarks

Figure 5.2 illustrates the lower and upper bound convergence of the two algorithms on a representative subset of instances (i.e., `hanoi`, `foss_poly_0`, `foss_iron`, and `pescara`).

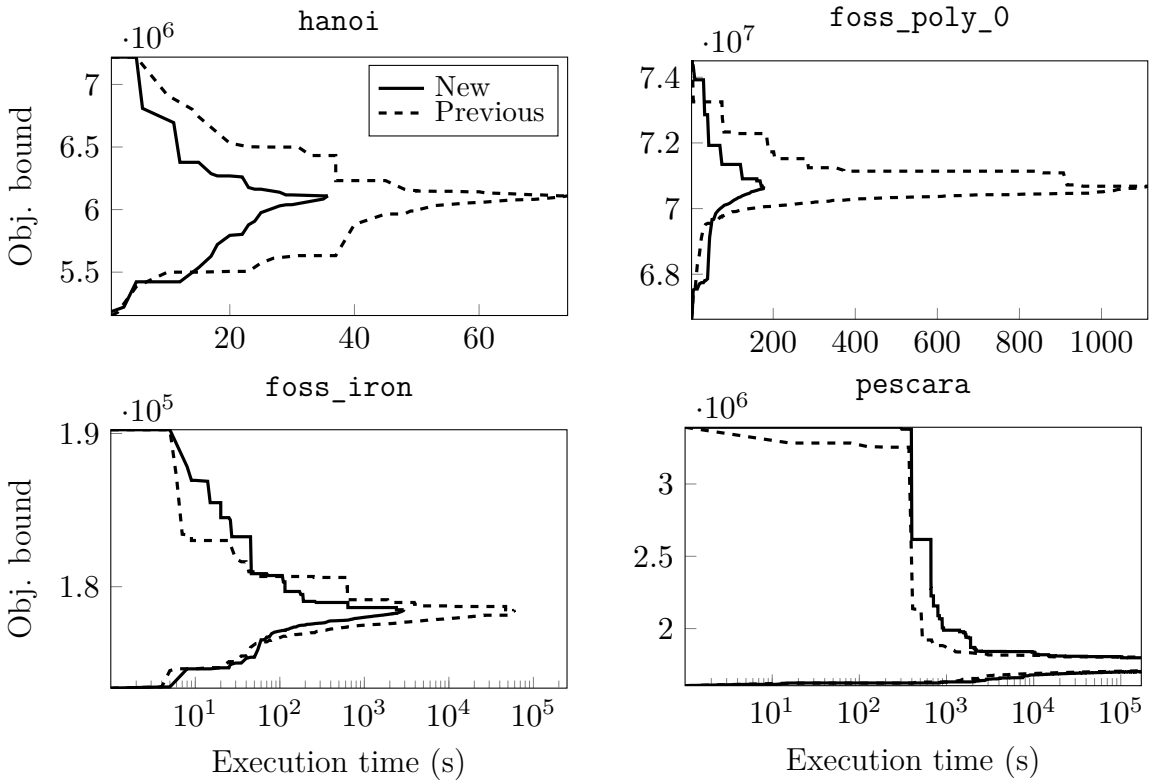


Figure 5.2: Convergence of objective bounds on select water network design instances. Compared are the new algorithm with an algorithm similar to [110] (Previous). Note the uses of linear and logarithmic x-axes.

Here, both algorithms converge to global optimality on the three moderate instances, with the new algorithm displaying more favorable performance, i.e., lower and upper bounds converging more quickly. However, both algorithms can solve `hanoi` in relatively short amounts of times, taking just over a minute to reach global optimality in the worst case. For larger moderate instances (i.e., `foss_poly_0` and `foss_iron`), the differences in convergence behavior are more dramatic. In the case of `foss_poly_0`, the new algorithm converges nearly an order of magnitude more quickly. For `foss_iron`, the difference is further emphasized, with the new algorithm converging more than an order of magnitude faster. *To highlight this point, throughout the literature, neither of these two `foss` instances appear to have been solved to global optimality. Here, both algorithms solve both problems, but ours does so around an order of magnitude faster.* In large cases (e.g., `pescara`), the algorithm of [110] converges more quickly. This could be for many

	Previous Algorithm			New Algorithm		
Problem	Gap (%)	Nodes Expl.	Time (s)	Gap (%)	Nodes Expl.	Time (s)
shamir	0.00	12,098	12.27	0.00	**2,567**	**7.11**
hanoi	0.00	32,024	74.21	0.00	**24,765**	**35.74**
blacksburg	0.00	16,009	**14.05**	0.00	**15,971**	29.25
foss_poly_0	0.00	144,226	1,112.78	0.00	**63,120**	**177.08**
foss_iron	0.00	1,307,123	59,088.98	0.00	**282,202**	**2,923.24**
foss_poly_1	**4.19**	48,320,343	Limit	4.90	21,858,635	Limit
pescara	**5.26**	5,633,461	Limit	5.29	2,010,998	Limit
modena	**33.77**	329,614	Limit	41.65	55,592	Limit

Table 5.2: Comparison of water network design algorithm convergence. Specifically, optimality gaps, nodes explored, and solution times for the new algorithm and one similar to [110] (Previous) are compared. Here, bold denotes better (smaller) times, nodes explored, and optimality gaps.

	Solutions from the Literature		Solutions from This Study	
Problem	Lower Bnd.	Upper Bnd.	Lower Bnd.	Upper Bnd.
shamir	**419,000**[1*]	**419,000**[1*]	**419,000**[*]	**419,000**[*]
hanoi	**6,109,620.09**[1*]	**6,109,620.09**[1*]	**6,109,620.90**[*]	**6,109,620.90**[*]
blacksburg	**118,251.09**[1*]	**118,251.09**[1*]	**118,251.09**[*]	**118,251.09**[*]
foss_poly_0	70,063,161.90[1]	**70,680,507.90**[2]	**70,680,507.90**[*]	**70,680,507.90**[*]
foss_iron	177,512.42[1]	**178,494.14**[1]	**178,494.14**[*]	**178,494.14**[*]
foss_poly_1	26,240.84[1]	29,202.99[2]	**27,269.65**	**28,462.34**
pescara	1,700,517.06[1]	≈ **1,790,000**[3]	**1,708,090.52**	1,798,252.52
modena	**2,206,914.89**[1]	≈ **2,560,000**[4]	2,198,756.06	3,319,652.71

Table 5.3: Comparison of water network design bounds with the literature. Bold denotes best bounds, asterisks denote proven optimality, and blue denotes instances closed for the first time. References from the literature are labeled according to the numbers [1][110], [2][18], [3][150], and [4][3].

reasons, the most likely being the more frequent addition of outer approximations in the new algorithm, creating larger BB linear subproblems. We finally remark that these numerical findings are additionally supported by the unpublished independent results of [111], who find favorable performance on small and moderate instances, close the gap on foss_poly_0 and foss_iron, and observe marginal benefits on larger instances.

Table 5.2 provides relevant convergence data for the instances. This table further supports the trends of Figure 5.2. For moderate instances, the new algorithm explores

fewer nodes to reach optimality. In most of these cases, this node reduction translates to smaller execution times, except for the case of `blacksburg`. For large instances, node exploration appears important for finding new feasible solutions. Thus, the dramatic reduction in the number of nodes explored by the new algorithm has a negative impact on the optimality gap reached by the time limit. Generally, the new algorithm appears useful for instances where optimality can be proven quickly but tends to suffer on large instances because of the increased master problem size — a topic of future work.

Table 5.3 compares the results from the water network design literature with the best results obtained in this chapter, including where our implementation of [110] outperformed the new algorithm. Specifically, the best objective lower and upper bounds are compared. The table depicts many new bounds discovered using the algorithms herein, especially on outstanding instances. (For the three *large* cases, our implementation of the algorithm in [110], not the new algorithm, discovered the bounds shown. The exception is for our upper bound on `pescara`, which the new algorithm discovered.) Note that here, literature solutions are not differentiated between those obtained heuristically and those obtained via algorithms that can also provide lower bounds (e.g., our algorithm). We finally remark that the slight change in the optimal solution of `hanoi` could either be a typographical error of [110] or a small difference in our definition of resistance.

5.6.3. Comparison of Algorithms on Demand-scaled Instances

This section compares the efficacy of the two algorithms on a set of moderate instances extending those described in Section 5.6.1. For each junction in each network, the original demand was scaled by a factor between 0.5 and 1.5 in steps of 0.05, generating 21 instances per network and producing 105 instances in total. For `hanoi`, instances with scalings greater than one were discovered to be infeasible, reducing the final set to 95 instances. For each instance, the time required to prove optimality was measured.

Figure 5.3 compares the times required to reach optimality across demand-scaled versions of `foss_poly_0` and `foss_iron`. These instances display the most dramatic differences between the two algorithms. For the `foss_poly_0` network, the new algorithm always outperforms the previous algorithm, occasionally by over an order of magnitude. For demand scalings in the set $\{1.2, 1.35, 1.4, 1.45, 1.5\}$, neither algorithm can solve the design instances within the nearly two day time limit. The comparison of

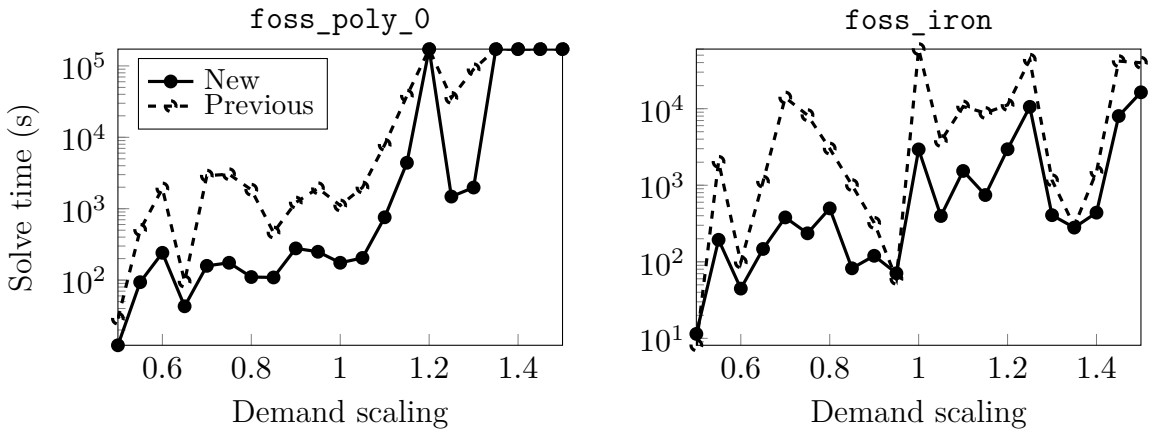


Figure 5.3: Comparison of times to reach optimality on two design instances. Specifically, the times (log scaled) are shown for ten demand-scaled versions of `foss_poly_0` and `foss_iron` design instances using our new algorithm (New) with an algorithm similar to [110] (Previous).

solve times in the `foss_iron` plot implies similar behavior. For all instances except two (0.5 and 0.95), the new algorithm outperforms the previous algorithm. Moreover, the new algorithm occasionally outperforms the previous by nearly two orders of magnitude.

Figure 5.4 compares the times to reach optimality across all demand-scaled instances. The figure excludes the five instances where optimality could not be proven. For instances requiring roughly one hundred seconds or less to reach optimality, the times required by both algorithms are similar, as shown by their placement around the dashed identity line. For instances requiring roughly one hundred seconds or more, however, the new algorithm always outperforms the previous algorithm, often by one or two orders of magnitude. This supports Section 5.6.2: for challenging problems where optimality can be proven, our new algorithm outperforms the previous algorithm's implementation.

5.7. Conclusion

This chapter presented the derivation and algorithmic application of a novel, *exact* MICP formulation for the global optimization of potable water distribution network design. Construction of this problem began with an *exact* convex reformulation for network analysis, extending the models of [29], [36], and [110] using nonlinear duality. Then, using the new MICP formulation as a foundation, the linear relaxation-based

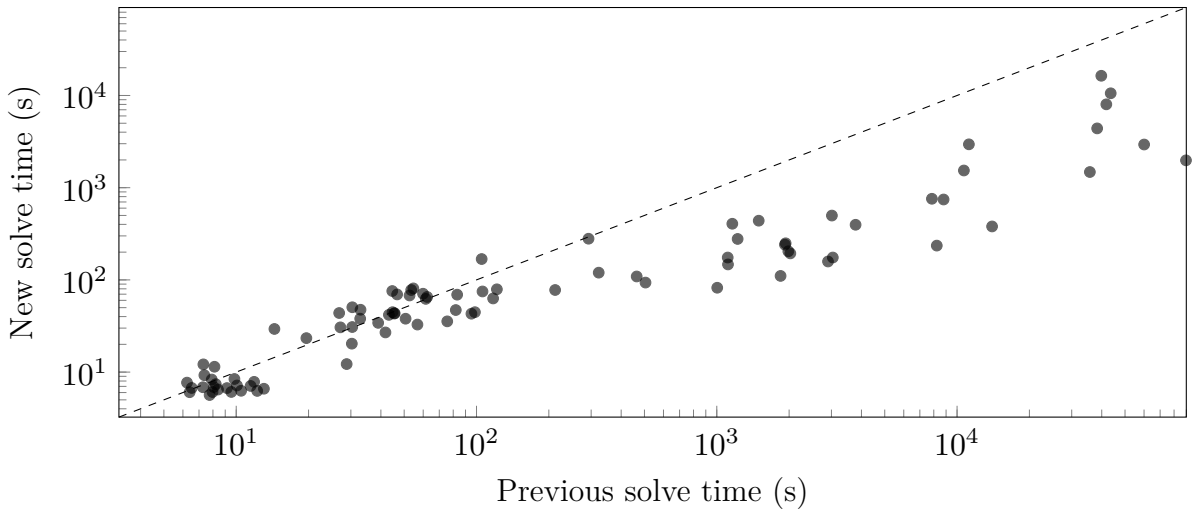


Figure 5.4: Comparison of times to reach optimality on scaled design instances. Instances where optimality was *not* proven by either are excluded.

algorithm of [110] was augmented to use outer approximations derived from the MICP.

To measure its efficacy, our global optimization method was compared against the previous state of the art on standard network design instances. Then, new moderately-sized instances were generated by scaling demands throughout the original networks. The combination of results implies significant speedups in convergence (i.e., one to two orders of magnitude) can be achieved on moderately-sized instances for which optimality can be proven within a modest amount of time (i.e., hours). These results imply that our convex reformulation technique may carry substantial algorithmic benefits.

This chapter provides a number of novel and useful contributions to the field of water system design and, more broadly, nonlinear network optimization. First and foremost, the formulation of a purely *convex* system for determining design feasibility appears to be the first in the literature. Notably, this system differs from the well-known Content and Co-Content Models of [36] in three respects: (i) it is in the variable space of flows *and* heads; (ii) it includes variable bounds; and (iii) it does not require an objective. These features allow direct embedding of the system within network decision problems. Second and perhaps just as importantly, the exact MICP reformulation of the network design problem establishes a new paradigm for approaching problems of this type. Third, an algorithm based on the novel features of this MICP is presented and appears capable of

proving optimality on several challenging instances. In fact, our computational results close the optimality gap on two outstanding instances (`foss_poly_0` and `foss_iron`).

Future work should focus on extending the optimization approaches developed in this chapter. First, the convex and mixed-integer convex reformulations appear to be immediately applicable to other application areas, including natural gas network expansion planning and crude oil network optimization. Such reformulations may have even greater benefits when considered in these new problem contexts. In Chapter 6, along with the development of several other new algorithmic techniques, we also explore the use of a similar convexification approach in the area of water network *operation*.

Chapter 6

Mixed-integer Convex Relaxations for Optimization of Potable Water Distribution Network Operation

Chapter 5 developed MICP reformulations and relaxations for the optimal *design* of *gravity-fed* water distribution networks. The canonical design problem, however, neglects many important practical aspects of modern water distribution network *operation*. The most important missing aspects are (i) the presence of pumps, which ensure adequate pressures are maintained in a real system; (ii) the presence of tanks, which act as points of storage for the system; and (iii) temporal evolution of the system, which results from the presence of tanks as well as variations in demand and electricity price over a predetermined time horizon. In this chapter, we thus instead consider the task of optimal *pump scheduling*, which we term the Optimal Water Flow (OWF) problem.

The most common OWF variants aim at minimizing the cost of a water system's energy consumption – an important practical objective. Estimates of energy required for general water use are variable throughout the literature but typically range from 4% to 16% of total U.S. consumption [62], [121], [138]. For example, one study estimates that the energy demand of water utilities is 1.0% of U.S. consumption [31]. The oft-quoted estimate for the consumption of California's water sector, on the other hand, is 19% [99]. Of the energy required for water use, one third is estimated to be from the pumping and treatment of water and wastewater [62]. These processes thus comprise large portions of a water system's total energy consumption and, due to more stringent environmental

regulations and increasing energy costs, stand to benefit from more efficient operations.

There are around 52,000 community water systems in the United States alone, defined as serving twenty-five or more year-round residents. However, nearly 85% of the total U.S. population is served by only 5% of these systems, while the remaining 95% comprise small and very small systems that service 3,300 people or fewer. Since they often lack economies of scale, these smaller utilities pay more per unit of water produced than larger utilities. Furthermore, in these smaller systems, nearly all of the energy consumed is via electricity, and 80% is consumed by motors used for pumping operations [40]. Finally, since these smaller systems often lack smart controls and computational infrastructure, they tend to rely on local control and ad hoc rules at the expense of system efficiency [119]. As such, the ability to quickly generate cost-optimal or nearly-optimal short-term (e.g., one day) pump schedules for small water distribution systems could greatly benefit the reliability and efficiency of water infrastructure in the United States and beyond.

To address this, this chapter aims at developing tractable, relaxation-based solution techniques for the OWF problem. The OWF is highly challenging for several reasons, many of which are separate from the complexities encountered in the network design problem of Chapter 5. First, similar to the design problem, the OWF is constrained by the head loss equations for pipes, which act as nonconvex nonlinear constraints. Second, the addition of pumps introduces new dynamical constraints in the form of quadratic equations. Third, the addition of pumps and valves requires the use of discrete variables for modeling the statuses of these components, which introduces new combinatorial complexity. Finally, the addition of tanks and time-evolving quantities establishes a temporal dimension to the problem, which increases the problem size considerably.

To alleviate these difficulties, this chapter attempts to address some limitations of current state-of-the-art OWF methods. The novel contributions of this chapter include

- A tight polyhedral MILP relaxation-based formulation of the OWF problem;
- A single-step approximation of the OWF that can be leveraged algorithmically;
- An optimization-based bound tightening (OBBT) algorithm using the relaxation;
- An optimization-based cut generation (OBCG) algorithm using the relaxation;
- New valid inequalities based on the convex inequalities of Chapter 5;
- A thorough empirical investigation of the above algorithmic techniques.

These contributions primarily aim at addressing a notable deficiency in the current state of the art: the slow improvement of OWF dual bounds. Although the development of

good primal-bounding techniques is not a primary focus of this chapter, the expectation is that better dual-bounding techniques will also ultimately improve primal bounds, as physically infeasible OWF solutions will be fathomed from the search more efficiently.

Chapter Overview The remainder of this chapter proceeds as follows: Section 6.1 reviews optimization techniques for optimal water network *operation*, which extends the literature review of Chapter 5; Section 6.2 formulates the OWF problem as a MINCP; Section 6.3 develops MICP and MILP relaxations of the original OWF problem; Section 6.4 derives a valid convex inequality for the OWF problem using techniques similar to Chapter 5; Section 6.5 develops a number of preprocessing methods to strengthen relaxation-based formulations; Section 6.6 empirically evaluates our network modeling, formulation, and algorithmic improvements; and Section 6.7 concludes the chapter.

6.1. Literature Review

Much of the nonlinear network optimization literature was reviewed in Chapter 5, along with the most relevant literature related to optimal water network *design*. Here, we extend this literature review to optimal water network *operation*, specifically. A thorough and recent review of this topic area is given by [88]. Here, they note that the majority of studies in this field are related to the control of pumps and valves, similar to the problem considered in this chapter. Moreover, they recognize that the majority of studies apply "stochastic" optimization methods (e.g., evolutionary and genetic algorithms), while fewer use "deterministic methods" (i.e., mathematical programming). Finally, they observe that computational studies typically consider networks with relatively small numbers of nodes, with 80% concerning networks of one hundred nodes or fewer. In this chapter, we develop deterministic methods for similarly small networks.

As in Chapter 5, we again develop mathematical programming methods for the OWF problem due to the drawbacks associated with stochastic metaheuristic techniques. First, metaheuristics provide no guarantess of optimality, whereas relaxation-based mathematical programming methods provide dual bounds on the true optimal cost of network operation in the OWF. Second, and perhaps more importantly, many metaheuristics (and indeed some deterministic methods) rely on relaxations of important constraints via penalizing infeasibilities in the objective. This can result in schedules

of pump activations that are, in reality, physically infeasible. Solutions obtained via such methods could thus have deleterious effects when implemented in practice (e.g., allowance of important pressure violations or lack of sufficient water volume in tanks).

Recently, mathematical programming techniques for optimal water network operation have seen a resurgence in the literature. For example, [57] consider a variant of the OWF similar to the one considered in this chapter. To solve the problem, they develop a mixed-integer second-order cone programming (SOCP) relaxation and propose a specialized algorithm for a problem subclass. In [126], a penalty-based SOCP relaxation is developed. In [14], a mathematical programming heuristic, using a similar relaxation strategy, is developed. Although these and other similar studies appear to sometimes rely on the assumption of water networks with a particular structure, they establish the importance of *convex relaxation* of nonlinearities that complicate the OWF problem.

A number of even more recent studies have established more general formulations for the OWF, independent of network structure. Many of these studies benefit from further MILP relaxations of the previously-described convex relaxations. One example of such a study is by [141], who develop a piecewise-linear relaxation of complicating nonlinearities, which is similarly done in this chapter. They also introduce a number of practical network simplifications and valid inequalities, also used in this chapter, as well as an algorithm that recovers a valid feasible solution from the relaxation's solution using EPANET. Another relevant example is given by [85], who develop a MILP formulation for a similar water network demand response application and consider a heuristic bound-tightening procedure using randomized network simulation results as bound proxies.

The study most similar to this chapter, however, is [15]. There, the authors develop a MILP outer approximation of the OWF, as well as an LP/NLP-BB algorithm that checks solution feasibility at integer-feasible nodes, similar to the algorithm used for network design in [110] and Chapter 5. They also develop an OBBT preprocessing routine to improve the strength of the MILP formulation. Finally, they develop a primal heuristic that repairs physically infeasible solutions by allowing for *continuous*-duration pump activations. As such, heuristic solutions are not necessarily feasible to the full OWF.

Unlike their study, this chapter develops a *piecewise* MILP relaxation instead of an outer approximation, which is intended to more accurately approximate nonlinear equations that constrain pipes and pumps. It also leverages a parallel piecewise relaxation-based OBBT routine, whereas [15] employs a more computationally challenging MINCP-

based OBBT routine. Furthermore, this chapter computes and applies novel valid inequalities for the OWF, whereas [15] provides no such valid inequalities. Finally, this chapter does not assume that solutions with *continuous*-duration pump activations provide valid primal bounds for the OWF. Thus, neither the heuristic employed by [15] nor any other heuristic are used in this chapter. Although this implies that OWF primal bounds may improve more slowly, it does ensure that solutions are feasible with respect to the discreteness of pump activations, which is a common feature of the OWF.

6.2. Problem Formulation

Operation of a water distribution network involves the use of pumps to increase pressure and guarantee flow over a given planning horizon. Together, pumps and tanks can be temporally coordinated to meet variable demand while strategically achieving a desired objective. As one example, the typical OWF problem aims at minimizing the total cost of energy consumed by the water network while satisfying physical and operational constraints. An optimal "pump schedule" leverages the variation in energy price over the planning horizon. In this section, we formulate the OWF exactly as a MINCP.

6.2.1. Network Modeling

Notation for Sets

We remark that notation in this chapter differs from Chapter 5 to accomodate a larger variety of network components and the temporal aspects related to scheduling. Here, a water network is represented by a directed graph $\mathcal{G} := (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of node-connecting components (or arcs). Since the OWF problem considers operational decisions across a fixed time horizon, $\mathcal{K} = \{1, 2, \dots, K\}$ is used to denote the set of instantaneous points in time that describe the network's evolution. The set of time *intervals* that connect adjacent time points in $\mathcal{K}$ is similarly denoted by $\mathcal{K}' = \{1, 2, \dots, K' := K - 1\}$. We further define the sets of demands $\mathcal{D}$, reservoirs $\mathcal{R}$, and tanks $\mathcal{T}$ as disjoint subsets of the nodes $\mathcal{N}$ in the network, where $\mathcal{D} \cup \mathcal{R} \cup \mathcal{T} = \mathcal{N}$. The set of node-connecting components $\mathcal{L}$ in the network comprises pipes $\mathcal{A} \subset \mathcal{L}$, short pipes $\mathcal{S} \subset \mathcal{L}$, valves $\mathcal{V} \subset \mathcal{L}$, and pumps $\mathcal{P} \subset \mathcal{L}$. For convenience, the set of node-connecting components incident to $i \in \mathcal{N}$ where i is the tail (respectively, head) of the

component is denoted by $\delta_i^+ := \{(i,j) \in \mathcal{L}\}$ (respectively, $\delta_i^- := \{(j,i) \in \mathcal{L}\}$).

Next, we examine each of the components individually, define their corresponding decision variables, and present constraints that each component enforces on water network operations. Specifically, for each component, we present two types of constraints: (i) operational limits and (ii) physical constraints. We begin by examining the nodes, $\mathcal{N}$.

Nodes

As in Chapter 5, nodal potentials are denoted by the variables h_i^k, $i \in \mathcal{N}$, $k \in \mathcal{K}$, where each variable represents the total hydraulic head in units of length. For each $i \in \mathcal{N}$, $k \in \mathcal{K}$, a minimum head $\underline{h}_i^k$, determined a priori, must first be satisfied. Additionally, an upper bound $\overline{h}_i^k$ can often be inferred from network data. This implies the bounds

$$\underline{h}_i^k \leq h_i^k \leq \overline{h}_i^k, \ \forall i \in \mathcal{N}, \ \forall k \in \mathcal{K}. \tag{6.1}$$

Node-connecting Components

Every node-connecting component $(i,j) \in \mathcal{L}$ is associated with a variable, q_{ij}^k, which denotes the volumetric flow rate across that component. Assuming lower and upper bounds of $\underline{q}_{ij}^k$ and $\overline{q}_{ij}^k$, respectively, these variables must satisfy the capacity constraints

$$\underline{q}_{ij}^k \leq q_{ij}^k \leq \overline{q}_{ij}^k, \ \forall (i,j) \in \mathcal{L}, \ \forall k \in \mathcal{K}'. \tag{6.2}$$

As in Chapter 5, flow can assume both positive and negative values, where a positive (respectively, negative) value of q_{ij}^k implies flow from node i to j (respectively, j to i).

Pipes

As discussed in Chapter 5, in water networks, flow along a pipe is induced by the difference in head between the two nodes connected by that pipe. The head loss relationships that associate flow and head are extended over all time intervals to give the constraints

$$h_i^k - h_j^k = L_{ij} r_{ij} q_{ij}^k \left|q_{ij}^k\right|^{\alpha-1}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.3}$$

As in Chapter 5, α is the exponent required by the Darcy-Weisbach or Hazen-Williams relationship (i.e., 2 or 1.852, respectively), L_{ij} is the pipe length, and r_{ij} is the pipe resistance per unit length. In this chapter, the resistance is also assumed to be fixed.

Short Pipes

Short pipes are components that model resistance-less transport of flow between two nodes. This is similar to treating the length of a pipe as negligible. Short pipes thus ensure the equality of heads at two nodes, i and j, connected by that component, i.e.,

$$h_i^k - h_j^k = 0, \ \forall (i,j) \in \mathcal{S}, \ \forall k \in \mathcal{K}'. \tag{6.4}$$

In this chapter, we assume pipes with small maximum head loss magnitudes (less than ten centimeters), derived from network data, can instead be categorized as short pipes.

Valves

Valves are used to route the flow of water to certain portions of the network or to block flow during maintenance of subnetworks. In this chapter, valves are considered controllable elements that are either open or closed. The status of each valve $(i,j) \in \mathcal{V}$ is indicated using a binary variable $z_{ij}^k \in \{0,1\}$, where $z_{ij}^k = 1$ corresponds to an open valve and $z_{ij}^k = 0$ to a closed valve. These variables limit the flow across each valve as

$$\underline{q}_{ij}^k z_{ij}^k \leq q_{ij}^k \leq \overline{q}_{ij}^k z_{ij}^k, \ z_{ij}^k \in \{0,1\}, \ \forall (i,j) \in \mathcal{V}, \ \forall k \in \mathcal{K}'. \tag{6.5}$$

Furthermore, when a valve is open, the pressures at the nodes connected by that valve are equal, similar to a short pipe. When the valve is closed, however, these pressures are decoupled. This phenomenon is modeled via the following set of disjunctive constraints:

$$(1 - z_{ij}^k)\left(\underline{h}_i^k - \overline{h}_j^k\right) \leq h_i^k - h_j^k \leq (1 - z_{ij}^k)\left(\overline{h}_i^k - \underline{h}_j^k\right), \ \forall (i,j) \in \mathcal{V}, \ \forall k \in \mathcal{K}', \tag{6.6}$$

i.e., if $z_{ij}^k = 1$, then $h_i^k = h_j^k$. Otherwise, if $z_{ij}^k = 0$, then h_i^k and h_j^k are decoupled.

Pumps

Each pump $(i,j) \in \mathcal{P}$ increases the head from node i to j when active and permits only unidirectional flow. Here, we consider only fixed-speed pumps, where each pump is assumed to be either on or off. When the pump is off, there is zero flow along the pump, and heads at adjacent nodes are decoupled. When the pump is on, there is appreciably positive flow (greater than or equal to some fixed ϵ_{ij}), and the head increase from i to j is described by a nonlinear function. The variable $z_{ij}^k \in \{0,1\}$ indicates the status of each pump, where $z_{ij}^k = 1$ if $q_{ij}^k \geq \epsilon_{ij}$ and $z_{ij}^k = 0$ if $q_{ij}^k < \epsilon_{ij}$. This implies the bounds

$$\underline{q}_{ij}^k = 0 \leq \epsilon_{ij} z_{ij}^k \leq q_{ij}^k \leq \overline{q}_{ij}^k z_{ij}^k, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.7}$$

The variable $G_{ij}^k \geq 0$ is introduced for each pump to denote the head gain that results from that pump. Modeling a pump's head gain as a quadratic function, when active, is an established practice that is well-documented in the literature [139]. This chapter assumes that each head gain is modeled via a strictly concave function of the form

$$a_{ij} \left(q_{ij}^k\right)^2 + b_{ij} q_{ij}^k + c_{ij} z_{ij}^k = G_{ij}^k, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.8}$$

In these constraints, a_{ij} is negative, and the head gain function is offset by some positive constant (i.e., $c_{ij} > 0$). Note further that q_{ij}^k and $c_{ij} z_{ij}^k$ are restricted to zero when z_{ij}^k is zero. This ensures that, when a pump is off, the corresponding head gain G_{ij}^k is also zero. To ensure the decoupling of hydraulic heads when a pump is off, the conventional head loss relationships may instead be rewritten as head *gain* relationships. That is,

$$h_i^k - h_j^k + G_{ij}^k \leq \left(1 - z_{ij}^k\right) \max\left(0, \overline{h}_i^k - \underline{h}_j^k\right), \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}' \tag{6.9a}$$

$$h_i^k - h_j^k + G_{ij}^k \geq \left(1 - z_{ij}^k\right) \min\left(0, \underline{h}_i^k - \overline{h}_j^k\right), \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.9b}$$

Note that when $z_{ij}^k = 1$, the pump is on, and the head gain between two nodes is G_{ij}^k.

In some water networks, pumps with identical properties are installed in parallel. The symmetry of possible pump activations in these identical groups can add unnecessary combinatorial complexity. As suggested by [61], the symmetry-breaking constraints are

$$z_{ij,1}^k \leq z_{ij,2}^k \leq \cdots \leq z_{ij,n}^k, \ \forall (i,j) \in \mathcal{P}_G, \ \forall k \in \mathcal{K}', \tag{6.10}$$

where $\mathcal{P}_G \subset \mathcal{P}$ is the set of identical pumps linking $i \in \mathcal{N}$ and $j \in \mathcal{N}$, which are ordered lexicographically by some arbitrary indices ($1, 2, \ldots, n$, above). For such a pump group, this drastically reduces the number of feasible choices for active pumps from 2^n to $n+1$.

Demands

Demands are nodes in the network where water is typically withdrawn. Each demand is associated with a constant $\overline{q}_i^k$ that denotes the demanded flow, expressed as a volumetric flow rate, at demand $i \in \mathcal{D}$, time $k \in \mathcal{K}'$. Note that, without loss of generality, this chapter allows demands to be negative. A negative demand indicates a point of injection of water into the water distribution network. For convenience, the variables $q_i^k \in \mathbb{R}$, $i \in \mathcal{D}$, $k \in \mathcal{K}'$, are introduced to denote the amount of water consumed by each demand node. (We further remark that, when demand is considered fixed, $q_i^k = \overline{q}_i^k$.)

Reservoirs

Reservoirs are nodes in the network where water is supplied. Each reservoir is assumed to be an infinite source of flow with zero pressure and constant elevation at a point in time (i.e., $\underline{h}_i^k = \overline{h}_i^k$ is assumed at every reservoir $i \in \mathcal{R}$). Furthermore, the variables $q_i^k \geq 0$, $i \in \mathcal{R}$, $k \in \mathcal{K}'$, denote the outflow of water from each reservoir at time k.

Tanks

Tanks are nodes in the network that serve as means for storing and discharging water over time. In this chapter, all tanks are assumed to be cylindrical with a fixed diameter D_i, where $i \in \mathcal{T}$, which implies a cross-sectional area $A_i := \frac{\pi}{4} D_i^2$. The bottom of each tank is assumed to be located at or below the minimum elevation of the associated node, $b_i^k \leq \underline{h}_i^k$, $i \in \mathcal{T}$, and the maximum elevation of water in the tank is assumed to be $\overline{h}_i^k$. The variables q_i^k, $i \in \mathcal{T}$, $k \in \mathcal{K}'$, denote the outflow (positive) or inflow (negative) through each tank. With these variables, the expressions for all tanks' volumes are first

$$V_i^k := A_i(h_i^k - b_i^k), \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}. \tag{6.11}$$

The Euler steps for integrating all tank volumes across intervals are then imposed via

$$V_i^{k+1} = V_i^k - \Delta t^k q_i^k, \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}', \tag{6.12}$$

where Δt^k is the length of the time interval that connects times $k \in \mathcal{K}$ and $k+1 \in \mathcal{K}$.

Flow Conservation

Satisfaction of flow conservation throughout the network requires nodal flow balance constraints to be enforced at every node $i \in \mathcal{N}$ across all time intervals $\mathcal{K}'$. That is,

$$\sum_{(j,i)\in\delta_i^-} q_{ji}^k - \sum_{(i,j)\in\delta_i^+} q_{ij}^k = q_i^k, \ \forall i \in \mathcal{D}, \ \forall k \in \mathcal{K}' \tag{6.13a}$$

$$\sum_{(j,i)\in\delta_i^-} q_{ji}^k - \sum_{(i,j)\in\delta_i^+} q_{ij}^k = -q_i^k, \ \forall i \in \mathcal{R} \cup \mathcal{T}, \ \forall k \in \mathcal{K}'. \tag{6.13b}$$

6.2.2. Physical Feasibility

Given the previously-described constraints, the MINCP for feasibility is then defined as

Head and flow bounds: Constraints (6.1), (6.2) (6.14a)

Pipe dynamics: Constraints (6.3) (6.14b)

Short pipe dynamics: Constraints (6.4) (6.14c)

Valve dynamics: Constraints (6.5), (6.6) (6.14d)

Pump dynamics: Constraints (6.7), (6.8), (6.9), (6.10) (6.14e)

Tank dynamics: Constraints (6.12) (6.14f)

Flow conservation: Constraints (6.13). (6.14g)

The goal of System (6.14) is to determine if water can be routed through the network while satisfying the operational limits and physical constraints imposed by each component of the network. The nonconvexities of System (6.14) arise from two sources: (i) the discreteness of controllable components and (ii) nonlinear equations, i.e., Constraints

(6.3) and (6.8). Section 6.3 describes mixed-integer convex relaxations that address (ii).

6.2.3. Optimal Water Flow

Aside from the feasibility requirements described in Sections 6.2.1 and 6.2.2, the notion of an *optimal* control schedule (i.e., a solution to the OWF) should often satisfy a number of other practical criteria. These criteria are discussed in the following paragraphs.

Tank Volume Recovery

One important practical criterion for an OWF solution is that, by the end of the scheduling horizon, the volume of water within each tank will be at least as large as its initial volume. This ensures that (i) the proposed component schedule (e.g., of pump activations) can be repeatedly applied to subsequent planning periods with similar demand profiles and (ii) tank volumes at the end of the planning period are not small enough to discourage efficient operations in subsequent periods. This implies the constraints

$$V_i^1 \leq V_i^K, \ \forall i \in \mathcal{T}, \tag{6.15}$$

where, in most of the literature, as well as in this chapter, V_i^1 is assumed to be fixed.

Pump Switching Limits

Maintenance accounts for roughly 10% of the net present value lifecycle cost of a pump [98]. Frequent activation and deactivation (i.e., switching) of a pump generally reduces the pump's lifetime and increases its overall maintenance cost. As such, water distribution system operators typically require operational constraints on pump schedules to limit the number of activations and deactivations of pumps over a given time horizon. Assuming that each pump in the network must *remain on* for a minimum time of τ^{on}, *off* for a minimum time of τ^{off}, and *switched on* no more than M_{ij} times, the following

operational *pump switching* constraints are often imposed in the OWF problem [59]:

$$z_{ij}^{k} - z_{ij}^{k-1} \leq z_{ij}^{\text{on},k}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}' \setminus \{1\} \tag{6.16a}$$

$$z_{ij}^{\text{on},k} \leq z_{ij}^{k'}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}', \ \forall k' \in t(k) \leq t(k') \leq t(k) + \tau^{\text{on}} \tag{6.16b}$$

$$z_{ij}^{k-1} - z_{ij}^{k} \leq z_{ij}^{\text{off},k}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}' \setminus \{1\} \tag{6.16c}$$

$$z_{ij}^{k'} \leq 1 - z_{ij}^{\text{off},k}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}', \ \forall k' \in t(k) \leq t(k') \leq t(k) + \tau^{\text{off}} \tag{6.16d}$$

$$\sum_{k \in \mathcal{K}'} z_{ij}^{\text{on},k} \leq M_{ij}, \ \forall (i,j) \in \mathcal{P}. \tag{6.16e}$$

Here, $z_{ij}^{\text{on},k} \in \{0,1\}$ and $z_{ij}^{\text{off},k} \in \{0,1\}$ are discrete variables indicating whether pump $(i,j) \in \mathcal{P}$ has been switched *on* or *off* at time $k \in \mathcal{K}'$, respectively. Constraints (6.16a) ensure that each *on* switching variable is equal to one when a pump *becomes* active at some time index k. Constraints (6.16b) ensure that, if a pump has been switched on, it remains on for at least a duration of τ^{on}. Constraints (6.16c) ensure that each *off* switching variable is equal to one when a pump *becomes inactive* at some time index k. Constraints (6.16d) ensure that, if a pump has been switched off, it remains off for at least a duration of τ^{off}. Finally, Constraints (6.16e) limit the number of *on* switches to respect the maximum number of switches, M_{ij}, over the duration of the pump schedule.

Cost Minimization

The final common criterion for an OWF solution is that the schedule of operations minimizes the total cost of electrical power consumption over the planning horizon. This cost arises primarily from the energy requirements of pumps. The power consumption of pump $(i,j) \in \mathcal{P}$ at time $k \in \mathcal{K}'$ is most accurately defined via the relationship

$$P_{ij}^{k}(q_{ij}^{k}, G_{ij}^{k}) := \frac{\rho g G_{ij}^{k} q_{ij}^{k}}{\eta_{ij}(q_{ij}^{k})}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.17}$$

Here, P_{ij}^{k} is the power consumed, ρ is the density of water, g is the acceleration due to gravity, and $\eta_{ij}(\cdot)$ is the pump's dimensionless efficiency, which is often a nonlinear function of flow and dictated by a predefined curve provided by the pump manufacturer. Note that each equation for a pump's power consumption is thus highly nonlinear. As such, some optimization studies have resorted to power curves for fixed speed pumps

that are linear as a function of flow. This results in power functions parameterized as

$$P_{ij}^k(q_{ij}^k, z_{ij}^k) := \lambda_{ij} q_{ij}^k + \mu_{ij} z_{ij}^k, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}', \tag{6.18}$$

where λ_{ij} and μ_{ij} are constants that define the power consumption of pump $(i,j) \in \mathcal{P}$.

No matter the form of a pump's power curve, the pump's energy consumption is

$$E_{ij}^k(q_{ij}^k, \cdot) := \Delta t^k P_{ij}^k(q_{ij}^k, \cdot), \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.19}$$

Letting the (fixed) cost per Joule of energy consumed over time step $k \in \mathcal{K}'$ be denoted by π^k, the cost of operating each pump (i,j) at each time step $k \in \mathcal{K}'$ is then given by

$$C_{ij}^k(q_{ij}^k, \cdot) := \pi^k E_{ij}^k(q_{ij}^k, \cdot), \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.20}$$

This implies that the total cost of water network operation is defined by the function

$$f(q, \cdot) := \sum_{k \in \mathcal{K}'} \sum_{(i,j) \in \mathcal{P}} C_{ij}^k(q_{ij}^k, \cdot). \tag{6.21}$$

Combining feasibility, tank recovery, pump switching limits, and minimization of the total operational cost gives rise to an exact form of the typical OWF problem,

$$\begin{array}{ll} \text{minimize} & \text{Objective function: } f(q,\cdot) \text{ of Equation (6.21)} \\ \text{subject to} & \text{Operational feasibility: Constraints (6.14)} \\ & \text{Tank volume recovery: Constraints (6.15)} \\ & \text{Pump switching limits: Constraints (6.16).} \end{array} \tag{MINLP}$$

Similar to System (6.14), this is a challenging MINCP that is difficult to solve directly.

6.3. Convex Relaxation

6.3.1. Mixed-integer Convex Relaxation

As discussed in Section 6.2.2, there are two sources of nonlinear nonconvexity that render both System (6.14) and (MINLP) extremely difficult to solve directly. Specifically, these

are Constraints (6.3) and (6.8), which model each pipe's head loss and each pump's head gain, respectively. A number of studies have addressed these challenges via convex outer approximations of these constraints [15], [85], [141]. Here, we elect to use similar outer approximations as a starting point to ultimately develop a tighter MILP relaxation.

To formulate the relaxations, as in Chapter 5, we introduce the variables $q_{ij}^{k\pm}$ to denote nonnegative flows in the two directions along link $(i,j) \in \mathcal{L}$. This decomposition implies

$$q_{ij}^{k} := q_{ij}^{k+} - q_{ij}^{k-}, \ \forall (i,j) \in \mathcal{L}, \ \forall k \in \mathcal{K}'. \tag{6.22}$$

Next, the flow bound Constraints (6.2) are decomposed into directed components as

$$0 \leq q_{ij}^{k\pm} \leq \overline{q}_{ij}^{k\pm}, \ \forall (i,j) \in \mathcal{L}, \ \forall k \in \mathcal{K}', \tag{6.23}$$

where $\overline{q}_{ij}^{k+} = \max\{0, \overline{q}_{ij}^{k}\}$ and $\overline{q}_{ij}^{k-} = \max\{0, -\underline{q}_{ij}^{k}\}$ replace the original flow bounds. This also requires the discrete variables $y_{ij}^{k} \in \{0,1\}$, $(i,j) \in \mathcal{L}$ to model flow direction, where $y_{ij}^{k} = 1$ implies flow from i to j and $y_{ij}^{k} = 0$ implies flow from j to i. That is,

$$0 \leq q_{ij}^{k+} \leq \overline{q}_{ij}^{k+} y_{ij}^{k}, \ \forall (i,j) \in \mathcal{L}, \ \forall k \in \mathcal{K}' \tag{6.24a}$$

$$0 \leq q_{ij}^{k-} \leq \overline{q}_{ij}^{k-} (1 - y_{ij}^{k}), \ \forall (i,j) \in \mathcal{L}, \ \forall k \in \mathcal{K}'. \tag{6.24b}$$

We further note that the direction-related inequalities in Chapter 5, i.e., Constraints (5.29)–(5.31), can also be formulated for the current setting. We omit them here for brevity, although all such inequalities are used in the direction-based formulations herein.

Convexification of Pipe Head Loss Constraints

As in Chapter 5, the above direction-based modifications allow for a rewriting and convex relaxation of the pipe head loss Constraints (6.3). To accomplish this, nonnegative head difference variables $\Delta h_{ij}^{k\pm}$ are first introduced to denote the head loss in the two possible

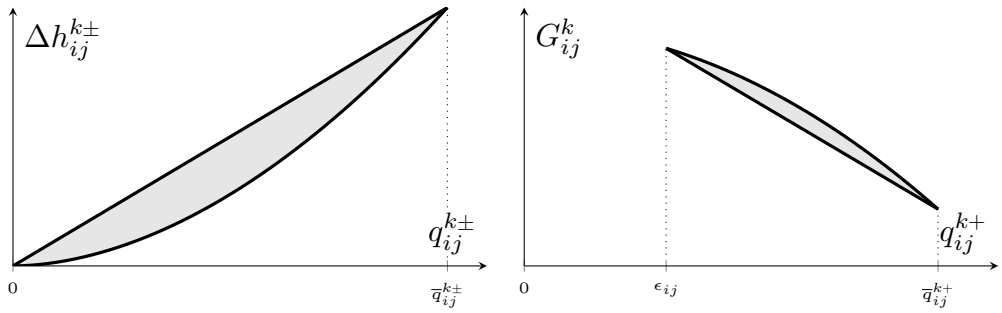


Figure 6.1: Example convex relaxations of pipe/pump head loss/gain constraints.

flow directions. These losses are related to the original head variables h_i^k, $i \in \mathcal{N}$, via

$$\Delta h_{ij}^{k+} - \Delta h_{ij}^{k-} = h_i^k - h_j^k, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.25a}$$

$$0 \leq \Delta h_{ij}^{k+} \leq \Delta \overline{h}_{ij}^{k+} y_{ij}^k, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.25b}$$

$$0 \leq \Delta h_{ij}^{k-} \leq \Delta \overline{h}_{ij}^{k-} (1 - y_{ij}^k), \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.25c}$$

Here, head difference bounds $\Delta \overline{h}_{ij}^{k\pm}$ are derived from the bounds on h in Constraints (6.1). Next, the right-hand sides in Equations (6.3) are decomposed into two convex functions representing loss in the positive and negative directions. This splitting implies

$$\Delta h_{ij}^{k\pm} = L_{ij} r_{ij} \left(q_{ij}^{k\pm} \right)^{\alpha}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.26}$$

Since only one direction will be selected per pipe, Equations (6.26) are then relaxed as

$$L_{ij} r_{ij} \left(q_{ij}^{k\pm} \right)^{\alpha} \leq \Delta h_{ij}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.27}$$

As in Chapter 5, the relaxations in Constraints (6.27) are linearly upper-bounded by

$$\Delta h_{ij}^{k\pm} \leq L_{ij} r_{ij} \left(\overline{q}_{ij}^{k\pm} \right)^{\alpha - 1} q_{ij}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.28}$$

For illustration, an example feasible region defined by the relaxation Constraints (6.27) and (6.28) is depicted in Figure 6.1 for a pipe with equal directed flow bounds $\overline{q}_{ij}^{k\pm}$.

Convexification of Pump Head Gain Constraints

Similar to the *convex lower approximations* of the head loss Constraints (6.3), *concave upper approximations* of the head gain Constraints (6.8) are easily established. Observing that Constraints (6.8) describe each G_{ij}^k as a concave function of q_{ij}^k, standard convex relaxations are achieved via transformations from equations to inequalities, i.e.,

$$G_{ij}^k \leq a_{ij} \left(q_{ij}^{k+}\right)^2 + b_{ij} q_{ij}^{k+} + c_{ij} z_{ij}^k, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.29}$$

Note that $\overline{q}_{ij}^{k-} = 0$ for all $(i,j) \in \mathcal{P}$, $k \in \mathcal{K}'$, and that $G_{ij}^k = 0$ when z_{ij}^k (and thus q_{ij}^{k+}) is equal to zero. Similar to the *linear upper-bounding* Constraints (6.28) for head loss convexifications, *linear lower bounds* for the pump head gain convexifications are

$$a_{ij} \left[(\overline{q}_{ij}^{k+} + \epsilon_{ij}) q_{ij}^{k+} + \epsilon_{ij}^2 z_{ij}^k\right] + b_{ij}(q_{ij}^{k+} + \epsilon_{ij} z_{ij}^k) + c_{ij} z_{ij}^k \leq G_{ij}^k, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}'. \tag{6.30}$$

For illustration, an example feasible region defined by the pump gain relaxation Constraints (6.29) and (6.30) is depicted in Figure 6.1. Note that the size of the feasible region is highly dependent on the flow bounds present for an active pump, ϵ_{ij} and $\overline{q}_{ij}^k$.

Mixed-integer Convex Relaxation for Physical Feasibility

To formulate the standard MICP relaxation of System (6.14), the head and flow bound Constraints (6.1) and (6.2) are replaced by Constraints (6.1), (6.23), (6.24), and (6.28). Assuming the definition of q_{ij}^k in Equation (6.22) and replacing the head loss and gain Constraints (6.3) and (6.8) with Constraints (6.27), (6.28), (6.29), and (6.30), this is

Head and flow bounds: Constraints (6.1), (6.23), (6.24), (6.25) (6.31a)

Pipe dynamics: Constraints (6.27), (6.28) (6.31b)

Short pipe dynamics: Constraints (6.4) (6.31c)

Valve dynamics: Constraints (6.5), (6.6) (6.31d)

Pump dynamics: Constraints (6.7), (6.9), (6.10), (6.29), (6.30) (6.31e)

Tank dynamics: Constraints (6.12) (6.31f)

Flow conservation: Constraints (6.13). (6.31g)

Mixed-integer Convex Relaxation for Optimal Water Flow

Combining System (6.31) with the common OWF requirements of Section 6.2.3 gives

$$\begin{array}{ll} \text{minimize} & \text{Objective function: } f(q,\cdot) \text{ of Equation (6.21)} \\ \text{subject to} & \text{Operational feasibility: Constraints (6.31)} \\ & \text{Tank volume recovery: Constraints (6.15)} \\ & \text{Pump switching limits: Constraints (6.16).} \end{array} \tag{MICP}$$

Compared to (MINLP), this is a more tractable problem. In practice, however, (MICP) is of questionable practical value *and* still challenging to solve directly. First, note that a solution to (MICP) may not be feasible to (MINLP). Second, most solvers capable of solving (MICP), e.g., Juniper [82], are not guaranteed to do so efficiently, primarily due to the fractional exponent ($\alpha = 1.852$) required to model Hazen-Williams head loss relationships. This property deprives us of the ability to directly use mixed-integer quadratic programming (MIQP) techniques. Finally, the strength of the convex relaxation is ultimately determined by the tightness of pipe and pump flow bounds, as illustrated in Figure 6.1. These observations of model strength further motivate the development of a *mixed-integer piecewise linear* relaxation, developed in Section 6.3.2.

6.3.2. Mixed-integer Piecewise Linear Relaxation

To increase the relaxation strength and tractability of (MICP), this section develops a mixed-integer piecewise linear relaxation of (MINLP) that leverages lower and upper approximations of nonlinear constraints. Compared to (MICP), the overall relaxation approach is similar but forms tighter piecewise envelopes of the nonlinear constraints illustrated in Figure 6.1. Additionally, whereas (MICP) uses *nonlinear* convex outer approximations, this section applies polyhedral (i.e., linear) outer approximations of the nonlinearities. This polyhedral approximation technique further allows for the relaxation to be solved using modern MILP solvers (e.g., Gurobi), which are typically more efficient and numerically reliable than conventional MICP solvers (e.g., Juniper).

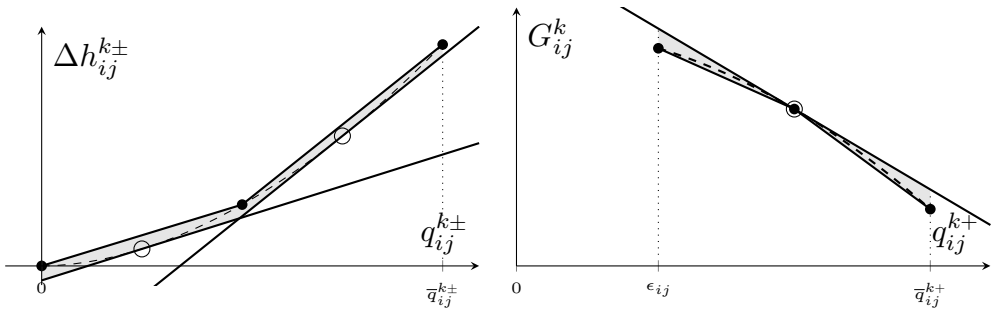


Figure 6.2: Example polyhedral relaxations of pipe/pump head loss/gain constraints. Dashed lines indicate the direction-decomposed nonlinear head loss and gain constraint functions, respectively. Filled and clear circles indicate inner and outer approximation points, respectively.

Linearization of Pipe Head Loss Constraints

To begin, for each pipe and pump, a set of lower approximation breakpoints, $\underline{Q}_{ij}^{k\pm}$, and a set of upper approximation breakpoints, $\overline{Q}_{ij}^{k\pm}$, $(i,j) \in \mathcal{A} \cup \mathcal{P}$ parameterize the relaxation. As in Chapter 5, the standard outer approximations of Constraints (6.27) are first

$$r_{ij}\left[\left(\hat{q}_{ij}^{k\pm}\right)^{\alpha} + \alpha\left(\hat{q}_{ij}^{k\pm}\right)^{\alpha-1}\left(\hat{q}_{ij}^{k\pm} - q_{ij}^{k}\right)\right] \leq \frac{\Delta h_{ij}^{k\pm}}{L_{ij}},\ \forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{K}',\ \forall \hat{q}_{ij}^{k\pm} \in \underline{Q}_{ij}^{k\pm}. \tag{6.32}$$

Constraints (6.32) are strengthened via multiplying constants with flow directions, i.e.,

$$r_{ij}\left[\left(\hat{q}_{ij}^{k+}\right)^{\alpha} y_{ij}^{k} + \alpha\left(\hat{q}_{ij}^{k+}\right)^{\alpha-1}\left(\hat{q}_{ij}^{k+} y_{ij}^{k} - q_{ij}^{k}\right)\right] \leq \frac{\Delta h_{ij}^{k+}}{L_{ij}}, \quad \forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{K}',\ \forall \hat{q}_{ij}^{k+} \in \underline{Q}_{ij}^{k+} \tag{6.33a}$$

$$r_{ij}\left[\left(\hat{q}_{ij}^{k-}\right)^{\alpha}\left(1 - y_{ij}^{k}\right) + \alpha\left(\hat{q}_{ij}^{k-}\right)^{\alpha-1}\left(\hat{q}_{ij}^{k-}(1 - y_{ij}^{k}) - q_{ij}^{k}\right)\right] \leq \frac{\Delta h_{ij}^{k-}}{L_{ij}}, \quad \forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{K}',\ \forall \hat{q}_{ij}^{k-} \in \underline{Q}_{ij}^{k-}. \tag{6.33b}$$

In addition, piecewise *inner* (or upper) approximations of these constraints, which strengthen the upper bounds described in Constraints (6.28), are developed using a convex combination approach. Letting $0 \leq \lambda_{ijp}^{k\pm} \leq 1$, $p \in \{1, 2, \ldots, |\overline{Q}_{ij}^{k\pm}|\}$, $(i,j) \in \mathcal{A}$, $k \in \mathcal{K}'$ denote the continuous convex combination variables and $x_{ijp}^{k\pm} \in \{0,1\}$, $p \in$

$\{2, \ldots, |\overline{\mathcal{Q}}_{ij}^{k\pm}|\}$ denote the binary convex combination variables, the upper bounds are

$$\frac{\Delta h_{ij}^{k\pm}}{L_{ij}} \leq \sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k\pm}|} r_{ij}(\hat{q}_{ijp}^{k\pm})^{\alpha} \lambda_{ijp}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.34}$$

The directed flow variables are similarly constrained by the convex combination, i.e.,

$$q_{ij}^{k\pm} = \sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k\pm}|} \hat{q}_{ijp}^{k\pm} \lambda_{ijp}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.35}$$

The activations of directed convex combination variables are of course limited by

$$\sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k+}|} \lambda_{ijp}^{k+} = y_{ij}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.36a}$$

$$\sum_{p=2}^{|\overline{\mathcal{Q}}_{ij}^{k+}|} x_{ijp}^{k+} = y_{ij}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.36b}$$

$$\sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k-}|} \lambda_{ijp}^{k-} = 1 - y_{ij}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.36c}$$

$$\sum_{p=2}^{|\overline{\mathcal{Q}}_{ij}^{k-}|} x_{ijp}^{k-} = 1 - y_{ij}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}'. \tag{6.36d}$$

Finally, the continuous and binary convex combination variables are related via

$$\lambda_{ij,1}^{k\pm} \leq x_{ij,2}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.37a}$$

$$\lambda_{ij,|\overline{\mathcal{Q}}_{ij}^{k\pm}|}^{k\pm} \leq x_{ij,|\overline{\mathcal{Q}}_{ij}^{k\pm}|}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}' \tag{6.37b}$$

$$\lambda_{ijp}^{k\pm} \leq x_{ijp}^{k\pm} + x_{ij,p+1}^{k\pm}, \ \forall (i,j) \in \mathcal{A}, \ \forall k \in \mathcal{K}', \ \forall p \in \{2, 3, \ldots, |\overline{\mathcal{Q}}_{ij}^{k\pm}| - 1\}. \tag{6.37c}$$

Compared to Figure 6.1, an illustration of the polyhedral sets that result from relaxing head loss constraints as per Constraints (6.33)–(6.37) is depicted on the left of Figure 6.2. The relaxation improves upon the previous upper-bounding technique at the expense

of a slightly weaker (i.e., linearly lower-approximated) lower bound. Nonetheless, even given the small number of lower and upper approximation breakpoints (i.e., two and three, respectively) in this illustration, the reduction in feasible area for each head loss relaxation implies possible relaxation benefits at the cost of additional binary variables.

Linearization of Pump Head Gain Constraints

Similarly, for pumps, the standard outer approximations of Constraints (6.29) are

$$G_{ij}^{k} \leq a_{ij}(\hat{q}_{ij}^{k+})^2 + b_{ij}\hat{q}_{ij}^{k+} + c_{ij}z_{ij}^{k} + \left(2a_{ij}\hat{q}_{ij}^{k+} + b_{ij}\right)(q_{ij}^{k} - \hat{q}_{ij}^{k}z_{ij}^{k}), \\ \forall (i,j) \in \mathcal{P},\ \forall k \in \mathcal{K}',\ \forall \hat{q}_{ij}^{k+} \in \overline{\mathcal{Q}}_{ij}^{k+}. \tag{6.38}$$

Following the intuition of Constraints (6.34)–(6.37), the piecewise *lower bounds* are

$$\sum_{p=1}^{|\underline{\mathcal{Q}}_{ij}^{k+}|} [a_{ij}(\hat{q}_{ijp}^{k+})^2 + b_{ij}\hat{q}_{ijp}^{k+} + c_{ij}]\lambda_{ijp}^{k+} \leq G_{ij}^{k},\ \forall (i,j) \in \mathcal{P},\ \forall k \in \mathcal{K}'. \tag{6.39}$$

Nonnegative flow variables are similarly constrained by the convex combination, i.e.,

$$q_{ij}^{k+} = \sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k+}|} \hat{q}_{ijp}^{k+}\lambda_{ijp}^{k+},\ \forall (i,j) \in \mathcal{P},\ \forall k \in \mathcal{K}'. \tag{6.40}$$

The activation of convex combination variables are then limited by pump status, i.e.,

$$\sum_{p=1}^{|\overline{\mathcal{Q}}_{ij}^{k+}|} \lambda_{ijp}^{k+} = z_{ij},\ \forall (i,j) \in \mathcal{P},\ \forall k \in \mathcal{K}' \tag{6.41a}$$

$$\sum_{p=2}^{|\overline{\mathcal{Q}}_{ij}^{k+}|} x_{ijp}^{k+} = z_{ij},\ \forall (i,j) \in \mathcal{P},\ \forall k \in \mathcal{K}'. \tag{6.41b}$$

Finally, the continuous and binary convex combination variables are related via

$$\lambda^{k+}_{ij,1} \leq x^{k+}_{ij,2}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}' \tag{6.42a}$$

$$\lambda^{k+}_{ij,|\underline{\mathcal{Q}}^{k+}_{ij}|} \leq x^{k+}_{ij,|\underline{\mathcal{Q}}^{k+}_{ij}|}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}' \tag{6.42b}$$

$$\lambda^{k+}_{ijp} \leq x^{k+}_{ijp} + x^{k+}_{ij,p+1}, \ \forall (i,j) \in \mathcal{P}, \ \forall k \in \mathcal{K}', \ \forall p \in \{2,3,\dots,|\underline{\mathcal{Q}}^{k+}_{ij}| - 1\}. \tag{6.42c}$$

An illustration of the polyhedral sets resulting from relaxing head gain constraints as per Constraints (6.38)–(6.42) is depicted on the right of Figure 6.2. Similar to the polyhedral head loss relaxations, the *lower-bounding* technique for head gain is improved at the expense of a slightly weaker *upper* bound due to the linear *upper* approximation.

Mixed-integer Linear Relaxation for Physical Feasibility

Letting Constraints (6.27)–(6.28) be replaced by Constraints (6.33)–(6.37) and Constraints (6.29)–(6.30) by Constraints (6.38)–(6.42), System (6.14)'s MILP relaxation is

Head and flow bounds: Constraints (6.1), (6.23), (6.24), (6.25) (6.43a)

Pipe dynamics: Constraints (6.33)−(6.37) (6.43b)

Short pipe dynamics: Constraints (6.4) (6.43c)

Valve dynamics: Constraints (6.5), (6.6) (6.43d)

Pump dynamics: Constraints (6.7), (6.9), (6.10), (6.38)−(6.42) (6.43e)

Tank dynamics: Constraints (6.12) (6.43f)

Flow conservation: Constraints (6.13). (6.43g)

Mixed-integer Linear Relaxation for Optimal Water Flow

Combining System (6.43) with the common OWF requirements of Section 6.2.3 gives

$$\begin{array}{ll} \text{minimize} & \text{Objective function: } f(q,\cdot) \text{ of Equation (6.21)} \\ \text{subject to} & \text{Operational feasibility: Constraints (6.43)} \\ & \text{Tank volume recovery: Constraints (6.15)} \\ & \text{Pump switching limits: Constraints (6.16).} \end{array} \tag{MILP}$$

Compared to (MICP), (MILP) is typically both more tractable *and* more accurate, as (i) it can be solved efficiently using modern commercial MILP solvers (e.g., Gurobi) and (ii) depending on the number of breakpoints used, the piecewise approximation technique provides a controllably accurate relaxation of the nonlinearities in (MINLP).

6.4. Convex Reformulation and Valid Inequality

Section 6.2 introduced the OWF as a MINCP, and Section 6.3 presented convex relaxations that generalize some typical features of those encountered in the literature. In this section, we extend the convex reformulation approach detailed in Chapter 5 in an attempt to convexify the OWF in a similar manner. First, in Section 6.4.1, we present convex "Content" and "Co-Content" models for water networks that also include pumps. Then, in Section 6.4.2, we extend these results to derive valid convex inequalities for the OWF problem. These inequalities are similar to the "strong duality" constraints of Chapter 5, although they are not *exact* models of the complicating physical dynamics.

6.4.1. Instantaneous Physical Feasibility

In the context of network analysis (e.g., the simulation of a network with fixed control decisions using EPANET), a relaxed version of System (6.14) can be solved via an explicit sequential method while neglecting flow and head bounds. At the core of such methods is a technique that solves a strictly convex programming problem and provides a solution satisfying Constraints (6.3), (6.8), and (6.13). For pipe networks, these convex "Content" and "Co-Content" model approaches were previously described in Chapter 5.

Here, we extend the conventional Content Model for pipes from [36] to instead write

$$\begin{aligned}
& & & \sum_{(i,j)\in\mathcal{A}} \frac{L_{ij} r_{ij}}{1+\alpha} \left[(q_{ij}^{k+})^{1+\alpha} + (q_{ij}^{k-})^{1+\alpha} \right] \\
& \text{minimize} & & - \sum_{(i,j)\in\mathcal{P}_1^k} \left[\frac{a_{ij}}{3} (q_{ij}^{k+})^3 + \frac{b_{ij}}{2} (q_{ij}^{k+})^2 + c_{ij} q_{ij}^{k+} \right] && \text{(6.44a)} \\
& & & - \sum_{i\in\mathcal{R}\cup\mathcal{T}} h_i^k \left[\sum_{(i,j)\in\delta_i^{k+}} (q_{ij}^{k+} - q_{ij}^{k-}) - \sum_{(j,i)\in\delta_i^{k-}} (q_{ji}^{k+} - q_{ji}^{k-}) \right] \\
& \text{subject to} & & \sum_{(j,i)\in\delta_i^{k-}} q_{ji}^k - \sum_{(i,j)\in\delta_i^{k+}} q_{ij}^k = \overline{q}_i^k, \ \forall i \in \mathcal{D} && \text{(6.44b)} \\
& & & q_{ij}^{k\pm} \geq 0, \ \forall (i,j) \in \mathcal{A} \cup \mathcal{S} \cup \mathcal{V}_1^k && \text{(6.44c)} \\
& & & q_{ij}^{k+} \geq 0, \ q_{ij}^{k-} = 0, \ \forall (i,j) \in \mathcal{P}_1^k, && \text{(6.44d)}
\end{aligned}$$

where $\mathcal{V}_1^k \subset \mathcal{V}$ and $\mathcal{P}_1^k \subset \mathcal{P}$ denote the set of valves and pumps that are *active* at time index $k \in \mathcal{K}'$, respectively (i.e., all inactive components are assumed to be "removed"). Similarly, δ_i^{k+} and δ_i^{k-} include only the *active* set of components at time step $k \in \mathcal{K}'$. Note that there is only one primary difference from conventional formulations of the Content Model (e.g., [110]). This is the inclusion of the second line of Equation (6.44a), which models the nonlinear head gain behavior associated with active pumps. We do remark, however, that similar treatments of head gain appear in the computational experiments of [36]; the extension of the Content Model to pump networks in [37]; and the hydraulic analysis study of [95]. All stress the importance of head gain functions that are strictly concave to ensure solution uniqueness. The relevant summation in Equation (6.44a) is strictly concave when $2a_{ij}q_{ij}^{k+} + b_{ij} < 0$ for $(i,j) \in \mathcal{P}_1^k$, $k \in \mathcal{K}'$, making its negation strictly convex and the overall objective convex. Also note that in many models, $a_{ij} < 0$, $|b_{ij}| \ll |a_{ij}|$, and $q_{ij}^{k+} > -b_{ij}/(2a_{ij})$ for an active pump, making the convexity assumption valid. Nonetheless, for simplicity, we assume only pump curves where all $b_{ij} = 0$, which guarantees the strict convexity of each function over $q_{ij}^{k+} > 0$.

Assuming fixed statuses of pumps and valves, as well as fixed heads at reservoirs and tanks, the strictly convex Problem (6.44) can thus be employed to solve for the unique set of flows that correspond to the given network configuration. Given the values of flows, a linear system can then be solved to obtain the corresponding values of remaining heads.

Thus, similar to the feasibility checking routine of Chapter 5, a feasibility checking routine that leverages Problem (6.44) will (i) remove inactive components from the network model, (ii) solve the corresponding convex program, (iii) solve a linear system to obtain non-fixed nodal heads, and (iv) check for flow and head bound satisfaction.

For the purpose of this chapter, we next derive a corresponding Co-Content variant of Problem (6.44) that will be ultimately be used to derive the inequalities in Section 6.4.2. First, pumps can be regarded mathematically as pipes with negative resistances or as pipes with head losses directed from j to i (instead of i to j). The remainder of the problem formulation remains functionally equivalent to the Content Model, and short pipes and active valves furthermore only ensure equality of heads at connecting nodes. For an even closer mapping, these components could similarly be interpreted as pipes with negligible resistance, which would contribute nothing to the content objective.

With these observations in mind, extending the Lagrangian dual derivation method of Appendix B.1, we are interested in finding the maximizers of the Lagrangian terms

$$\cdots \; - \sum_{(i,j)\in\mathcal{P}_1^k} \left(\frac{a_{ij}}{3}[q_{ij}^{k+}]^3 + \frac{b_{ij}}{2}[q_{ij}^{k+}]^2 + c_{ij}q_{ij}^{k+} - [\lambda_i^k - \lambda_j^k]q_{ij}^{k+} \right). \tag{6.45}$$

Note that here, all terms are of the form $\tilde{a}_{ij}(q_{ij}^{k+})^3 + \tilde{b}_{ij}(q_{ij}^{k+})^2 + \tilde{c}_{ij}q_{ij}^{k+} - tq_{ij}^{k+}$, where $f(q_{ij}^{k+}) = \tilde{a}_{ij}(q_{ij}^{k+})^3 + \tilde{b}_{ij}(q_{ij}^{k+})^2 + \tilde{c}_{ij}q_{ij}^{k+}$ is a concave function. There are two possibilities: if $t \geq 0$, the component is nonincreasing in q_{ij}^{k+} over $q_{ij}^{k+} \geq 0$ and thus attains its maximum at $q_{ij}^{k+} = 0$. Otherwise, if $t < 0$, the function is increasing at $q_{ij}^{k+} = 0$, attains its maximum, then starts decreasing. This maximum is attained at the stationary point

$$\hat{q}_{ij}^{k+} = \frac{-b_{ij} - \sqrt{b_{ij}^2 - 4a_{ij}(c_{ij} - t)}}{2a_{ij}}. \tag{6.46}$$

After simplifying, then, the maximum value of the corresponding component is

$$\begin{aligned} g_{ij}(t) = \frac{b_{ij}^3}{12a_{ij}^2} + \frac{b_{ij}^2\sqrt{b_{ij}^2 - 4a_{ij}(c_{ij} - t)}}{12a_{ij}^2} + \frac{t\sqrt{b_{ij}^2 - 4a_{ij}(c_{ij} - t)}}{3a_{ij}} \\ - \frac{c_{ij}\sqrt{b_{ij}^2 - 4a_{ij}(c_{ij} - t)}}{3a_{ij}} - \frac{b_{ij}c_{ij}}{2a_{ij}} + \frac{b_{ij}t}{2a_{ij}}. \end{aligned} \tag{6.47}$$

Note that the first, fifth, and sixth terms are constants or linear in t. Using this chapter's assumption of $b_{ij} = 0$ and our additional assumptions on a_{ij} and c_{ij}, the second term can be dropped, and the third and fourth terms can combine to create a function proportional to the form $(d+w)^{3/2}$, where $w \geq 0$ is variable and d is a positive constant. Note that this is a strictly convex function over the domain of interest. This implies the negated sum of components in Equation (6.45) is strictly concave over $\Delta h_{ij}^{-} := -t$.

Using the above interpretation and results, our Co-Content Model is written as

$$\begin{aligned}
\text{maximize} \quad & \frac{-\alpha}{1+\alpha} \sum_{(i,j)\in\mathcal{A}} \frac{1}{\sqrt[\alpha]{L_{ij} r_{ij}}} \left[(\Delta h_{ij}^{k+})^{1+\frac{1}{\alpha}} + (\Delta h_{ij}^{k-})^{1+\frac{1}{\alpha}} \right] - \\
& - \sum_{(i,j)\in\mathcal{P}_1^k} g_{ij}(-\Delta h_{ij}^{k-}) - \sum_{i\in\mathcal{D}} h_i \overline{q}_i^k && \text{(6.48a)} \\
\text{subject to} \quad & \Delta h_{ij}^{k+} - \Delta h_{ij}^{k-} = h_i^k - h_j^k, \ \forall (i,j) \in \mathcal{L}^k && \text{(6.48b)} \\
& \Delta h_{ij}^{k\pm} \geq 0, \ \forall (i,j) \in \mathcal{A} \cup \mathcal{S} \cup \mathcal{V}_1^k && \text{(6.48c)} \\
& \Delta h_{ij}^{k+} = 0, \ \Delta h_{ij}^{k-} \geq 0, \ \forall (i,j) \in \mathcal{P}_1^k. && \text{(6.48d)}
\end{aligned}$$

where $\mathcal{L}^k$ corresponds to the node-connecting components that are *active* at time index $k \in \mathcal{K}'$. This forms our version of the Co-Content Model with pumps and valves.

6.4.2. Duality-based Valid Convex Inequalities

Letting the objective of Problem (6.44) be denoted by $f_P^k(q^k)$ and the objective of Problem (6.48) be denoted by $f_D^k(h^k)$, the similar *convex* strong duality constraint $f_P^k(q^k) - f_D^k(h^k) \leq 0$ can be constructed to model the nonconvex system dynamics at time step $k \in \mathcal{K}'$. There is an important caveat, however: this convexity assumes that tank heads are fixed, along with the typical assumption that reservoir heads and demand flows are also fixed. That is, in the context of the OWF, unlike the water network design problem, the strong duality constraint cannot be embedded to provide an exact convex reformulation. However, a convex *relaxation* can be formed by using a standard McCormick relaxation of the complicating nonconvex terms. That is, products of flows and heads at tanks that appear in $f_P^k(q^k)$ and $f_D^k(h^k)$, denoted here as $q_i^k h_i^k$ for

brevity, can be relaxed in the strong duality constraint by introducing the constraints

$$w_i^k \geq \underline{q}_i^k h_i^k + \underline{h}_i^k q_i^k - \underline{q}_i^k \underline{h}_i^k, \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}' \tag{6.49a}$$

$$w_i^k \geq \overline{q}_i^k h_i^k + \overline{h}_i^k q_i^k - \overline{q}_i^k \overline{h}_i^k, \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}' \tag{6.49b}$$

$$w_i^k \leq \underline{q}_i^k h_i^k + \overline{h}_i^k q_i^k - \underline{q}_i^k \overline{h}_i^k, \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}' \tag{6.49c}$$

$$w_i^k \leq \overline{q}_i^k h_i^k + \underline{h}_i^k q_i^k - \overline{q}_i^k \underline{h}_i^k, \ \forall i \in \mathcal{T}, \ \forall k \in \mathcal{K}', \tag{6.49d}$$

where w_i^k, $i \in \mathcal{T}$, $k \in \mathcal{K}'$, denotes the relaxed product of head and flow at a tank.

Using the above relaxation in the constraints $f_P^k(q^k) - f_D^k(h^k) \leq 0$ ultimately yields

$$\begin{aligned} &\sum_{(i,j)\in\mathcal{A}} \frac{L_{ij} r_{ij}}{1+\alpha} \left[(q_{ij}^{k+})^{1+\alpha} + (q_{ij}^{k-})^{1+\alpha} \right] - \sum_{(i,j)\in\mathcal{P}} \left[\frac{a_{ij}}{3} (q_{ij}^{k+})^3 + \frac{b_{ij}}{2} (q_{ij}^{k+})^2 + c_{ij} q_{ij}^{k+} \right] \\ &\quad - \sum_{i\in\mathcal{R}} h_i^k \left[\sum_{(i,j)\in\delta_i^+} (q_{ij}^{k+} - q_{ij}^{k-}) - \sum_{(j,i)\in\delta_i^-} (q_{ji}^{k+} - q_{ji}^{k-}) \right] - \sum_{i\in\mathcal{T}} w_i^k \\ &\quad + \frac{\alpha}{1+\alpha} \sum_{(i,j)\in\mathcal{A}} \frac{1}{\sqrt[\alpha]{L_{ij} r_{ij}}} \left[(\Delta h_{ij}^{k+})^{1+\frac{1}{\alpha}} + (\Delta h_{ij}^{k-})^{1+\frac{1}{\alpha}} \right] \\ &\quad + \sum_{(i,j)\in\mathcal{P}} z_{ij}^k g_{ij}(-\Delta h_{ij}^{k-}) + \sum_{i\in\mathcal{D}} h_i^k \overline{q}_i^k \leq 0, \ \forall k \in \mathcal{K}'. \end{aligned} \tag{6.50}$$

Note that, in place of using the set $\mathcal{P}_1^k$, the original set $\mathcal{P}$ is used in the summations, where terms in the second sum are equal to zero when pumps are deactivated in the OWF, and terms in the second to last sum are similarly zero when pumps are deactivated. (Note that although $z_{ij}^k g_{ij}(\cdot)$ is a bilinear product, it can be exactly represented using a McCormick representation since z_{ij}^k is binary. In practice, nonlinearities in Constraints (6.50) will be algorithmically outer-approximated, which also mitigates this.)

6.5. Strengthening Convex Relaxations

The relaxations of the OWF developed in previous sections depend strongly on the tightness of variable bounds, especially pipe and pump flow bounds. In the case of (MILP), the relaxation also depends on the number of breakpoints used to model inner and outer approximations of the nonlinear constraints. Although outer approximations

introduce little computational complexity to the formulation, inner approximations are *piecewise*-linear and require a number of discrete variables to model accurately. Thus, the tightness of the relaxation also depends strongly on the piecewise-linear discretization of nonlinearities, which has an important tradeoff with computational complexity.

We are interested in developing OWF relaxations that reduce or limit the combinatorial sizes of piecewise-linear discretizations to maintain problem tractability. In this section, there are two methods we consider: (i) improvement of variable bounds and (ii) derivation of linear inequalities that strengthen the relaxation. To this end, Section 6.5.1 first introduces a *single-step approximation* of the OWF, which can then be used to efficiently derive time-independent properties of an OWF instance. Section 6.5.2 then develops an OBBT method that leverages this single-step approximation. Section 6.5.3 extends the intuition of the OBBT method to computationally derive valid inequalities to the OWF. Finally, Section 6.5.4 modifies network-based (or physics-inspired) valid inequalities that have been successfully used in a previous relaxation-based OWF study.

6.5.1. Single-step Approximation

To formulate a more tractable approximation of the OWF problem for the purpose of computing tighter variable bounds and new valid inequalities, the temporal properties of each network component are first collapsed into a single-step approximated model of that component. The resultant network approximation can thus serve as a *relaxation* of any independent steady-state model that is to be satisfied within the sequence of time steps $k \in \mathcal{K}'$. To begin, the original nodal head variables h_i^k, $i \in \mathcal{N}$, $k \in \mathcal{K}$, are temporally collapsed to form the nodal head variables h_i, $i \in \mathcal{N}$. These are bounded as

$$\underline{h}_i \leq h_i \leq \overline{h}_i, \ \forall i \in \mathcal{N}, \tag{6.51}$$

where it is assumed that $\underline{h}_i = \min\{\underline{h}_i^k : k \in \mathcal{K}\}$ and $\overline{h}_i = \max\{\overline{h}_i^k : k \in \mathcal{K}\}$. That is, the head at each node in the network is constrained between its temporally extremal bounds. This notion is also used to bound temporally collapsed flows, q_{ij}, $(i,j) \in \mathcal{L}$, as

$$\underline{q}_{ij} \leq q_{ij} \leq \overline{q}_{ij}, \ \forall (i,j) \in \mathcal{L}, \tag{6.52}$$

where it is similarly assumed that $\underline{q}_{ij} = \min\{\underline{q}_{ij}^k : k \in \mathcal{K}'\}$ and $\overline{q}_{ij} = \max\{\overline{q}_{ij}^k : k \in \mathcal{K}'\}$.

For each pipe, a single head loss relationship is enforced for the component, i.e.,

$$h_i - h_j = L_{ij} r_{ij} q_{ij} \left| q_{ij} \right|^{\alpha - 1}, \ \forall (i,j) \in \mathcal{A}. \tag{6.53}$$

The dynamics for each short pipe are approximated in a similar single-step manner as

$$h_i - h_j = 0, \ \forall (i,j) \in \mathcal{S}. \tag{6.54}$$

For valves, single-step indicator variables $z_{ij} \in \{0,1\}$ are first introduced. These variables are then used with the extremal flow bounds to constrain each valve's flow as

$$\underline{q}_{ij} z_{ij} \leq q_{ij} \leq \overline{q}_{ij} z_{ij}, \ \forall (i,j) \in \mathcal{V}. \tag{6.55}$$

The heads at the nodes connected by each valve are then limited by the constraints

$$(1 - z_{ij}) \left(\underline{h}_i - \overline{h}_j \right) \leq h_i - h_j \leq (1 - z_{ij}) \left(\overline{h}_i - \underline{h}_j \right), \ \forall (i,j) \in \mathcal{V}. \tag{6.56}$$

For pumps, single-step indicator variables $z_{ij} \in \{0,1\}$ are also introduced. These variables are again used with the extremal flow bounds to constrain each pump's flow as

$$\epsilon_{ij} z_{ij} \leq q_{ij} \leq \overline{q}_{ij} z_{ij}, \ z_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{P}. \tag{6.57}$$

Each pump's single-step approximated head gain is then modeled via the constraints

$$a_{ij} \left(q_{ij} \right)^2 + b_{ij} q_{ij} + c_{ij} z_{ij} = G_{ij}, \ \forall (i,j) \in \mathcal{P}. \tag{6.58}$$

The heads at the nodes connected by each pump are then constrained as

$$h_i - h_j + G_{ij} \leq \left(1 - z_{ij}\right) \max \left(0, \overline{h}_i - \underline{h}_j \right), \ \forall (i,j) \in \mathcal{P} \tag{6.59a}$$

$$h_i - h_j + G_{ij} \geq \left(1 - z_{ij}\right) \min \left(0, \underline{h}_i - \overline{h}_j \right), \ \forall (i,j) \in \mathcal{P}. \tag{6.59b}$$

To relax the temporal variation in demand, the variables q_i, $i \in \mathcal{D}$, are introduced to denote the single-step approximated demanded (or injected) flow at each demand point.

Whereas in the original problem, demands were fixed, demands are now bounded as

$$\min\{\overline{q}_i^k : k \in \mathcal{K}'\} \leq q_i \leq \max\{\overline{q}_i^k : k \in \mathcal{K}'\}, \ \forall i \in \mathcal{D}. \tag{6.60}$$

Single-step approximation variables q_i, $i \in \mathcal{R} \cup \mathcal{T}$, are also introduced to denote the outgoing flow from reservoirs and tanks. Since the tank volume integration Constraints (6.12) are temporal in nature, they are omitted from the single-step approximation. The remaining constraints are similar to the flow conservation Constraints (6.13), written as

$$\sum_{(j,i)\in\delta_i^-} q_{ji} - \sum_{(i,j)\in\delta_i^+} q_{ij} = q_i, \ \forall i \in \mathcal{D} \tag{6.61a}$$

$$\sum_{(j,i)\in\delta_i^-} q_{ji} - \sum_{(i,j)\in\delta_i^+} q_{ij} = -q_i, \ \forall i \in \mathcal{R} \cup \mathcal{T}. \tag{6.61b}$$

Given the previously-described constraints that model each component in the water network, the so-called *single-step approximation* of System (6.14) is then written as

Head and flow bounds: Constraints (6.51), (6.52) (6.62a)

Pipe dynamics: Constraints (6.53) (6.62b)

Short pipe dynamics: Constraints (6.54) (6.62c)

Valve dynamics: Constraints (6.55), (6.56) (6.62d)

Pump dynamics: Constraints (6.10), (6.57), (6.58), (6.59) (6.62e)

Flow conservation: Constraints (6.60), (6.61). (6.62f)

This approximation serves as an easier-to-solve variant of System (6.14), primarily due to the elimination of challenging temporal aspects related to feasibility. Because this problem is substantially easier to solve, it can serve as a reliable and efficient means for discovering relationships that strengthen the formulation or relaxations of (MINLP).

Algorithm 6.1 Optimization-based bound tightening for water networks.

Input: System (6.62) (or any relaxation thereof) that comprises a feasible set Ω
Output: Valid bounds for System (6.14) and (MINLP): $\underline{h}$, $\overline{h}$, $\underline{q}$, $\overline{q}$, $\underline{z}$, $\overline{z}$, ϵ

1: **repeat**
2: $(\underline{h}^f, \overline{h}^f, \underline{q}^f, \overline{q}^f, \underline{z}^f, \overline{z}^f, \epsilon^f) \leftarrow \underline{h}, \overline{h}, \underline{q}, \overline{q}, \underline{z}, \overline{z}, \epsilon$
3: $\Omega \leftarrow$ Relaxed feasibility system given $\underline{h}^f, \overline{h}^f, \underline{q}^f, \overline{q}^f, \underline{z}^f, \overline{z}^f, \epsilon^f$
4: **for all** $i \in \mathcal{N}$ **do**
5: $\underline{h}_i \leftarrow$ minimize h_i subject to Ω
6: $\overline{h}_i \leftarrow$ maximize h_i subject to Ω
7: **end for**
8: **for all** $(i, j) \in \mathcal{L}$ **do**
9: $\underline{q}_{ij} \leftarrow$ minimize q_{ij} subject to Ω
10: $\overline{q}_{ij} \leftarrow$ maximize q_{ij} subject to Ω
11: **end for**
12: **for all** $(i, j) \in \mathcal{P} \cup \mathcal{V}$ **do**
13: $\underline{z}_{ij} \leftarrow$ minimize z_{ij} subject to Ω
14: $\overline{z}_{ij} \leftarrow$ maximize z_{ij} subject to Ω
15: **end for**
16: **for all** $(i, j) \in \mathcal{P}$ **do**
17: $\epsilon_{ij} \leftarrow$ minimize q_{ij} subject to $\Omega \cup \{z_{ij} = 1\}$
18: **end for**
19: **until** $(\underline{h}^f, \overline{h}^f, \underline{q}^f, \overline{q}^f, \underline{z}^f, \overline{z}^f, \epsilon^f) = (\underline{h}, \overline{h}, \underline{q}, \overline{q}, \underline{z}, \overline{z}, \epsilon)$

6.5.2. Optimization-based Bound Tightening

Note that System (6.62) is a valid relaxation of the constraints that must be satisfied at *each* time step $k \in \mathcal{K}'$ of System (6.14), independently. Among these steady states, the tank volume integration Constraints (6.12) and tank recovery Constraints (6.15) are the only conditions that link them, which are notably excluded from System (6.62). Because of these properties, one of the clearest benefits that System (6.62) provides is the capability of being embedded within an efficient OBBT algorithm. The bounds discovered from this procedure, since valid for the single-step approximation, are thus also valid for any steady state $k \in \mathcal{K}'$ modeled in System (6.14). We remark that a related OBBT approach has been successfully used by [15] in a similar OWF context.

Algorithm 6.1 presents a simplified variant of the OBBT algorithm used in this study. Here, Lines 1 and 19 give the algorithm's iteration and termination conditions, respec-

tively, i.e., repeat the interior of the algorithm (Lines 2–18) until variable bounds no longer improve. Line 2 sets the variable bounds for the current iteration of the algorithm. Line 3 constructs a new relaxation, Ω, using these variable bounds. Lines 4–7 derive new bounds for heads in the network. Here, Line 5 solves a minimization problem that yields a lower bound $\underline{h}_i$, and Line 6 solves a maximization problem that yields an upper bound $\overline{h}_i$. In a similar manner, Lines 8–11 compute new lower and upper bounds for flow variables in the network and Lines 12–15 compute new bounds for indicator variables. Finally, Lines 16–18 compute lower bounds for variable pump flows *when the pump is active* (i.e., $z_{ij} = 1$). This conditional bound-tightening allows for further strengthening of relaxations for each pump's head gain curve by improving the fixed ϵ_{ij}.

In practice, Algorithm 6.1 can be used to derive even tighter bounds depending on the relaxation ultimately employed to solve the OWF problem. For example, if a direction-based formulation is used, head loss relaxations can be strengthened by computing tighter lower flow bounds that are conditional on the flow direction of a pipe, i.e.,

$$\underline{q}^{+}_{ij} \leftarrow \text{minimize } q^{+}_{ij} \text{ subject to } \Omega \cup \{y_{ij} = 1\} \tag{6.63a}$$

$$\underline{q}^{-}_{ij} \leftarrow \text{minimize } q^{-}_{ij} \text{ subject to } \Omega \cup \{y_{ij} = 0\}. \tag{6.63b}$$

Tighter bounds can also be computed for other variables used in direction-based formulations, e.g., y_{ij} can be tightened in a manner similar to z_{ij} in Lines 12–15 of Algorithm 6.1. The notion that such *conditional* bounds can be derived via an OBBT framework inspires our next novel contribution presented in Section 6.5.3, in which valid inequalities between two variables (binary-binary and binary-continuous) are computed similarly.

6.5.3. Optimization-based Valid Inequalities

Aside from OBBT, System (6.62) also provides an efficient mechanism by which globally valid inequalities can be derived for its temporal counterpart, System (6.14). The procedure for computationally deriving these inequalities (i.e., cuts) is similar to the OBBT Algorithm 6.1. For this reason, we refer to the procedure as OBCG. Cuts derived from the OBCG procedure further strengthen relaxation-based formulations of the OWF problem, potentially enabling faster improvement of solution lower bounds.

Algorithm 6.2 presents a simplified version of the OBCG algorithm used in this chap-

ter. Here, Line 1 instantiates the set of binary variables to consider in the cut generation procedure. Line 2 defines the binary-binary cut generation loop, where all unique, ordered pairs of binary variables are considered. Lines 3 and 4 minimize and maximize the first variable in the pair, respectively, subject to the relaxation constraints and a fixing of the second variable to zero. On Line 5, if both minimization and maximization yield optimal objectives of zero, a cut can be derived, which is added to the set of cuts $\bar{\mathcal{X}}$ on Line 6. Otherwise, if minimization and maximization both yield one, a similar process is followed on Lines 6–9. On Lines 10–16, the same process is repeated while fixing the second variable to one instead of zero. This generates the set of all binary-binary cuts.

The second loop begins on Line 18, where variable pairs comprise *continuous* variables of flow and head as well as *binary* variables of controllable component statuses. The goal of this loop is to derive cuts that improve variable bounds *depending on controllable component statuses*. For example, Line 19 minimizes a continuous variable subject to the relaxation constraints and a fixing of the binary variable to zero. In turn, the optimal objective provides a potentially tighter *lower* bound for the continuous variable *when the binary variable is zero*. This conditional inequality is added to the set of cuts $\bar{\mathcal{X}}$ on Line 20. Similar cuts are derived for other bounds and variable fixings on Lines 21–26.

Note that, since the cuts are derived from the single-step approximation, System (6.62), they can be applied to every time $k \in \mathcal{K}'$ within an OWF formulation. We also remark that, like the OBBT algorithm described in Section 6.5.2, Algorithm 6.2 can be used to derive even more cuts depending on the relaxation employed to solve the OWF problem. These additional cuts present new opportunities to strengthen relaxed OWF formulations. For example, if a direction-based formulation is used, all direction variables $\{y_{ij} : (i,j) \in \mathcal{L}\}$ can also be included in the set $\mathcal{B}$. Similarly, directed flow variables $\{q^{\pm}_{ij} : (i,j) \in \mathcal{L}\}$ can be included in the unioned set of continuous variables.

6.5.4. Network-based Valid Inequalities

An interesting intertemporal cut for the OWF is formulated by [141] for networks where all tank and nodal demands were ultimately served by reservoir-drawn pumps. In some networks, however, some portions of demand can also be served via gravity-fed distri-

Algorithm 6.2 Optimization-based cut generation for water networks.

Input: System (6.62) (or any relaxation thereof) that comprises a feasible set Ω
Output: Valid inequalities (or cuts) for System (6.14) and (MINLP), denoted by $\bar{\mathcal{X}}$

$\mathcal{B} \leftarrow \{z_{ij} : (i,j) \in \mathcal{P} \cup \mathcal{V}\}$, $\bar{\mathcal{X}} = \emptyset$
for all $(x_1, x_2) \in (\mathcal{B} \times \mathcal{B}) \setminus \{(x,x) : x \in \mathcal{B}\}$ **do**
  $\underline{x}_1^0 \leftarrow$ minimize x_1 subject to $\Omega \cup \{x_2 = 0\}$
  $\overline{x}_1^0 \leftarrow$ maximize x_1 subject to $\Omega \cup \{x_2 = 0\}$
  if $\underline{x}_1^0 = \overline{x}_1^0 = 0$ **then**
    $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 \leq x_2\}$
  else if $\underline{x}_1^0 = \overline{x}_1^0 = 1$ **then**
    $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 + x_2 \geq 1\}$
  end if
  $\underline{x}_1^1 \leftarrow$ minimize x_1 subject to $\Omega \cup \{x_2 = 1\}$
  $\overline{x}_1^1 \leftarrow$ maximize x_1 subject to $\Omega \cup \{x_2 = 1\}$
  if $\underline{x}_1^1 = \overline{x}_1^1 = 0$ **then**
    $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 + x_2 \leq 1\}$
  else if $\underline{x}_1^1 = \overline{x}_1^1 = 1$ **then**
    $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 \geq x_2\}$
  end if
end for
for all $(x_1, x_2) \in \big(\{q_{ij} : (i,j) \in \mathcal{L}\} \cup \{h_i : i \in \mathcal{N}\}\big) \times \mathcal{B}$ **do**
  $\underline{x}_1^0 \leftarrow$ minimize x_1 subject to $\Omega \cup \{x_2 = 0\}$
  $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{\underline{x}_1^0(1 - x_2) + \underline{x}_1 x_2 \leq x_1\}$
  $\overline{x}_1^0 \leftarrow$ maximize x_1 subject to $\Omega \cup \{x_2 = 0\}$
  $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 \leq \overline{x}_1^0(1 - x_2) + \overline{x}_1 x_2\}$
  $\underline{x}_1^1 \leftarrow$ minimize x_1 subject to $\Omega \cup \{x_2 = 1\}$
  $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{\underline{x}_1^1 x_2 + \underline{x}_1(1 - x_2) \leq x_1\}$
  $\overline{x}_1^1 \leftarrow$ maximize x_1 subject to $\Omega \cup \{x_2 = 1\}$
  $\bar{\mathcal{X}} \leftarrow \bar{\mathcal{X}} \cup \{x_1 \leq \overline{x}_1^1 x_2 + \overline{x}_1(1 - x_2)\}$
end for

bution. Thus, more general cuts than those provided by [141] are derived here as

$$\sum_{i \in \mathcal{T}} (V_i^1 - V_i^k) + \sum_{k'=k}^{K'} \sum_{i \in \mathcal{D}} \Delta t^{k'} d_i^{k'} \leq \sum_{i \in \mathcal{R}} \sum_{k'=k}^{K'} \Delta t^{k'} q_i^{k'}, \ \forall k \in \mathcal{K}'. \tag{6.64}$$

First, each tank's volume at the end of the schedule, V_i^K, must be greater than or equal to V_i^1. This implies that $V_i^1 - V_i^k$, $k \in \mathcal{K}'$, represents the volume of water that *must be*

Network	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}\rvert$
`Simple FSD`	4	2	0	0	3	2	1	1
`AT(M)`	22	41	0	0	3	19	1	2
`Poormond`	52	43	1	4	7	46	1	5

Table 6.1: Summary of optimal water flow networks derived from the literature.

restored to a tank $i \in \mathcal{T}$ by the end of the time horizon. Thus, $\sum_{i \in \mathcal{T}} (V_i^1 - V_i^k)$ represents the total volume that must be restored to all tanks by time index K, beginning from k. Similarly, $\sum_{k'=k}^{K'} \sum_{i \in \mathcal{D}} \Delta t^{k'} d_i^{k'}$ represents the total volume of water that *remains to be delivered* by the end of the time horizon. Since all tank and nodal demands are ultimately fed by reservoirs, $\sum_{i \in \mathcal{R}} \sum_{k'=k}^{K'} \Delta t^{k'} q_i^{k'}$ represents the total volume contributed by all reservoirs in the network between time step k and K, which completes the inequality.

6.6. Computational Experiments

This section empirically evaluates our formulation and algorithmic contributions to the OWF problem. All formulations and algorithms were implemented in the Julia programming language using JuMP, version 0.21 [50], and version 0.7 of WaterModels, an open-source Julia package for water distribution network optimization [131]. First, Section 6.6.1 describes the common instances, computational resources, and parameters used in the experiments. Section 6.6.2 empirically measures the effects of OBBT on the lower bounds of relaxation-based polyhedral OWF formulations. Section 6.6.3 does the same for the OBCG-generated inequalities using the algorithm of Section 6.5.3, as well as the volume-inspired inequalities of Section 6.5.4. Section 6.6.4 evaluates the effects of the content-based inequalities of Section 6.4.2. Section 6.6.5 empirically evaluates the upper bound tradeoffs of various parameterizations of polyhedral OWF formulations.

6.6.1. Experimental Setup

The numerical experiments consider three small networks of various sizes that appear in the pump scheduling literature and are summarized in Table 5.1 [15], [87]. All use the Hazen-Williams head loss relationship for pipes. Although `Simple FSD` and `Poormond` are similar (but not equivalent) to the networks studied in [15], our version of `AT(M)`

was modified to ensure feasibility in *our* writing of the OWF, where pump statuses are required to be fixed across a time interval. In this case, relaxation feasibility was ensured by increasing maximum water levels of the two tanks in the system from 71.53 meters to 72.00 meters. Note that all pumps in `Simple FSD` and `AT(M)` are identical and leverage the symmetry-breaking Constraints (6.10), but all pumps in `Poormond` are unique. Additionally, eighteen switches per pump (M_{ij}) are permitted for `Simple FSD` and `AT(M)`, while six pump switches are permitted for `Poormond`. Furthermore, τ^{on} and τ^{off} are equal to 3600 and 1800 seconds, respectively, for all OWF instances considered. Finally, all pump power consumptions are modeled linearly, as done in Equations (6.18).

Each of the three networks have correspondence with fifteen unique OWF instances, which were also derived from [15] and that study's corresponding open-source data sets. Each instance differs in two ways. First, although each instance assumes a twenty-four hour scheduling period, the patterns of demands and electricity prices are specified in two hour, one hour, and thirty minute intervals. That is, instances have varying temporal dimensions, where $K' \in \{12, 24, 48\}$. Second, for one K', electricity price patterns differ across five different daily estimates. In summary, the instances study variations in temporal resolution of the network model as well as variations in electricity pricing.

Each experiment was executed on a node containing two Intel Xeon E5-2695 v4 processors, each with 18 cores at 2.10 GHz, and 125 GB of memory. All experiments primarily concerned benchmarking relaxations based on (MILP), although continuous relaxations of (MICP) were also sometimes used to determine breakpoints in (MILP) parameterizations. For the solutions of all MILPs, Gurobi 9.1 was used with the `NumericFocus=1` parameter, which increases the overall numerical accuracy of the solution algorithm. For the solution of convex NLPs, Ipopt 3.12 was used with the linear solver MA57, as per the recommendation of [134], and the tolerance parameter `tol=1.0e-11`.

6.6.2. Effects of Optimization-based Bound Tightening

This section evaluates the efficacy of the OBBT algorithm of Section 6.5.2 and its effects on the objective lower bounds of MILP relaxations. In each execution of Algorithm 6.1, the single-step approximation System (6.43) was used (i.e., as Ω), with each head loss and head gain function piecewise-linearly relaxed using ten equally-spaced breakpoints for both the inner- and outer-approximations of the functions. Trivially parallelizable

	$K' = 12$			$K' = 24$			$K' = 48$		
Network	$\mathcal{N}$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{A}$	$\mathcal{P}$
`Simple FSD`	44%	30%	85%	40%	33%	85%	34%	37%	85%
`AT(M)`	11%	63%	2%	8%	73%	2%	3%	81%	2%
`Poormond`	62%	2%	13%	62%	2%	13%	62%	2%	13%

Table 6.2: Bound improvements for water network instances using OBBT.

portions of the algorithm were parallelized over 70 threads. Even with the tight (MILP)-like relaxation, the OBBT procedure completed in under 19 seconds for all `Simple FSD` instances, 88 seconds for all `AT(M)` instances, and 47 seconds for all `Poormond` instances.

Table 6.2 shows the overall improvement in physical bounds for select component sets of the problem instances. Here, each network is divided into the three possible temporal categories, $K' \in \{12, 24, 48\}$. Within each category, the $\mathcal{N}$ column lists the mean overall improvement between the initial and bound-tightened ranges of nodal head bounds (i.e., $\overline{h}_i^k - \underline{h}_i^k$). Similarly, the $\mathcal{A}$ column corresponds to the overall improvement in initial and bound-tightened ranges of pipe flow bounds (i.e., $\overline{q}_{ij}^k - \underline{q}_{ij}^k$). Finally, the $\mathcal{P}$ column corresponds to the improvement in ranges of pump flow bounds (i.e., $\overline{q}_{ij}^k - \epsilon_{ij}^k$).

For `Simple FSD`, the effects of OBBT are dramatic: head bounds are improved by around 40%, pipe flow bounds are improved by over 30%, and pump flow bounds are improved by 85%. For `AT(M)`, the effects on head and pump flow bounds are less dramatic, although pipe flow bounds are improved substantially. Finally, for `Poormond`, flow bounds are only marginally improved, but head bounds are improved significantly. We remark that, for the `Simple FSD` instances, no additional bound information was provided by the data set from which these instances were derived. On the contrary, the data set from which `AT(M)` was derived included initial lower head bounds for three nodes, and `Poormond` included initial lower and upper flow bounds for all pipes and all pumps. This could explain the small improvements for some bounds in the `AT(M)` and `Poormond` instances. It also implies that OBBT may be especially useful for networks that lack additional bound or capacity data, as illustrated in the results for `Simple FSD`.

Next, Table 6.3 shows the effects of using tightened bounds when solving relaxed variants of the OWF. Specifically, the (MILP) relaxation of the OWF was used, where two and ten flow breakpoints modeled each head loss/head gain inner and outer approximation. The Gurobi parameter `MIPFocus=3`, which emphasizes improving the

		$K' = 12$			$K' = 24$			$K' = 48$		
	Day	LB	LB^{BT}	% Diff.	LB	LB^{BT}	% Diff.	LB	LB^{BT}	% Diff.
`Simple FSD`	1	162.2	-	-	153.6	155.3	1.1%	152.3	152.7	0.3%
	2	165.9	-	-	158.2	158.5	0.2%	156.6	157.0	0.3%
	3	179.6	-	-	171.4	172.0	0.4%	169.2	170.4	0.7%
	4	187.9	-	-	179.2	181.3	1.2%	176.7	177.8	0.6%
	5	159.3	-	-	148.2	148.5	0.2%	146.8	147.1	0.2%
`AT(M)`	1	713.9	714.0	0.0%	704.9	706.8	0.3%	697.7	699.3	0.2%
	2	705.2	705.8	0.1%	697.1	700.1	0.4%	688.0	690.8	0.4%
	3	732.9	733.1	0.0%	724.7	725.7	0.1%	715.9	718.0	0.3%
	4	793.0	793.8	0.1%	782.9	784.7	0.2%	774.5	775.5	0.1%
	5	646.3	646.9	0.1%	635.3	637.1	0.3%	628.2	630.0	0.3%
`Poormond`	1	106.6	108.3	1.6%	98.8	99.4	0.6%	95.5	97.3	1.9%
	2	109.8	109.9	0.1%	101.6	102.0	0.4%	98.5	99.7	1.2%
	3	121.9	121.0	-0.7%	113.3	113.3	0.0%	109.1	109.8	0.6%
	4	133.1	133.4	0.2%	123.0	124.2	1.0%	118.8	120.7	1.6%
	5	108.8	111.4	2.4%	86.8	87.9	1.3%	82.1	82.8	0.9%

Table 6.3: Objective lower bound improvements for instances using OBBT.

objective lower bound, was also used. In Table 6.3, each problem instance is categorized with respect to its temporal dimension (K') and its corresponding electricity price profile ("Day"). The "LB" column corresponds to the best lower bound obtained by Gurobi within one hour when using naive variable bounds based on provided network data, while "LB^{BT}" corresponds to the best lower bound when using the OBBT-based bounds. Finally, "% Diff." measures the percent difference between the bounds, i.e.,

$$\% \text{ Diff.} := \left(\frac{\text{LB}^{\text{BT}} - \text{LB}}{\text{LB}} \right) 100\%, \tag{6.65}$$

where positive values indicate a relative improvement in lower bound when using OBBT.

For all instances except one (`Poormond`, $K' = 12$, Day 3), OBBT-based variable bounds result in consistently improved objective lower bounds. For `Simple FSD`, $K' = 12$, the tighter variable bounds allow for proofs of infeasibility for the instances (i.e., having no lower bounds), whereas the naively-bounded instances are classified as feasible. For the remaining `Simple FSD` instances, lower bound improvements are often less than 1%. For `AT(M)` instances, the improvements are not as large and always less than 0.5%.

For `Poormond`, the benefits are more apparent. Here, lower bounds are often improved between 1% and 2%, which could be substantial when attempting to prove optimality.

6.6.3. Effects of Valid Linear Inequalities

This section evaluates the efficacy of the valid inequalities generated via the OBCG method of Section 6.5.3 and the network-based method of Section 6.5.4. As in Section 6.6.2, within the OBCG Algorithm 6.2, the single-step approximation System (6.43) was used (i.e., as Ω), with each head loss and head gain function piecewise-linearly relaxed using ten equally-spaced breakpoints for both the inner- and outer-approximations of the functions. The OBCG routines completed in under 38 seconds for all `Simple FSD` instances, 1,985 seconds for all `AT(M)` instances, and 2,188 seconds for all `Poormond` instances. Similar to the OBBT algorithm, the OBCG algorithm was parallelized over 70 threads. Although OBCG is more intensive than OBBT, note that these cuts can be reused as long as network properties remain within their temporally extremal bounds.

We next compare the overall improvement in the objective lower bounds achieved using the above cuts, similar to the setting of Section 6.6.2. Here, the same parameterizations of (MILP) and Gurobi as in Section 6.6.2 were used. In Table 6.4, the best bounds achieved within one hour are again compared for the various OWF instances, where "LB^{BT}" corresponds to the best lower bound achieved when using *only* improved variable bounds, and "LB^{C}" corresponds to the best lower bound achieved when using improved variable bounds *as well as* the cuts described in Sections 6.5.3 and 6.5.4.

Table 6.4 first shows that, for `Simple FSD` and `AT(M)`, the effects of adding valid inequalities are often negligible or detrimental. In the case of `AT(M)`, the sometimes negative effects of adding cuts could be due to the increased size of the master problem, which might be substantial if a large number of OBCG cuts are computed. However, for `Poormond`, the effects of the valid inequalities are generally beneficial and sometimes dramatic. In most cases, the lower bound improvement is between 1% and 5%, and for the largest instances ($K' = 48$), the improvements are more substantial. These results indicate that, for difficult instances, valid inequalities can strengthen MILP formulations significantly, although the effects appear to be mostly dependent on network structure.

		$K' = 12$			$K' = 24$			$K' = 48$		
	Day	LB^{BT}	LB^{C}	% Diff.	LB^{BT}	LB^{C}	% Diff.	LB^{BT}	LB^{C}	% Diff.
Simple FSD	1	-	-	-	155.3	155.3	0.0%	152.7	152.7	0.0%
	2	-	-	-	158.5	158.7	0.1%	157.0	157.0	0.0%
	3	-	-	-	172.0	172.1	0.1%	170.4	170.4	0.0%
	4	-	-	-	181.3	181.4	0.1%	177.8	177.8	0.0%
	5	-	-	-	148.5	148.7	0.1%	147.1	147.1	0.0%
AT(M)	1	714.0	714.1	0.0%	706.8	707.1	0.0%	699.3	699.7	0.1%
	2	705.8	705.6	0.0%	700.1	699.2	-0.1%	690.8	691.3	0.1%
	3	733.1	733.1	0.0%	725.7	725.4	0.0%	718.0	717.9	0.0%
	4	793.8	794.0	0.0%	784.7	784.6	0.0%	775.5	776.1	0.1%
	5	646.9	646.8	0.0%	637.1	637.2	0.0%	630.0	630.0	0.0%
Poormond	1	108.3	109.1	0.7%	99.4	103.8	4.4%	97.3	101.7	4.5%
	2	109.9	112.3	2.2%	102.0	106.7	4.6%	99.7	105.0	5.3%
	3	121.0	123.9	2.4%	113.3	117.3	3.5%	109.8	115.2	4.9%
	4	133.4	135.5	1.6%	124.2	129.6	4.3%	120.7	127.8	5.9%
	5	111.4	112.7	1.2%	87.9	91.0	3.5%	82.8	86.9	5.0%

Table 6.4: Objective lower bound improvements using valid inequalities.

6.6.4. Effects of Valid Nonlinear Convex Inequalities

This section evaluates the efficacy of the valid inequalities derived in Section 6.4.2. Here, these inequalities were implemented as the linear McCormick Constraints (6.49) and relaxations of Constraints (6.50), where nonlinear terms involving $q^{\pm}$ and $\Delta h^{\pm}$ were piecewise-linearly inner-outer approximated, as with the head loss and gain nonlinearities of (MILP). We forgo an explicit derivation and statement of these inner and outer approximations since they bear such strong similarity with the linearizations described in Chapter 5 and Section 6.3.2. We do, however, remark that the breakpoints used in these linear relaxations coincided with the flow breakpoints used elsewhere in (MILP).

Similar to previous subsections, Table 6.5 evaluates overall improvements in objective lower bounds when using linearizations of the new convex inequalities. The same parameterizations of (MILP) and Gurobi were used as in the lower-bounding experiments of previous subsections. In Table 6.5, LB^{C} corresponds to the objective lower bounds achieved when using OBBT and the valid linear inequalities of Section 6.6.3, and LB^{PD} corresponds to the objective lower bounds achieved when using OBBT, the previous valid linear inequalities, *and* linearizations of the new valid convex inequalities.

		$K' = 12$			$K' = 24$			$K' = 48$		
	Day	LB^C	LB^{PD}	% Diff.	LB^C	LB^{PD}	% Diff.	LB^C	LB^{PD}	% Diff.
Simple FSD	1	-	-	-	155.3	155.3	0.0%	152.7	152.7	0.0%
	2	-	-	-	158.7	158.7	0.0%	157.0	157.0	0.0%
	3	-	-	-	172.1	172.1	0.0%	170.4	170.4	0.0%
	4	-	-	-	181.4	181.4	0.0%	177.8	177.8	0.0%
	5	-	-	-	148.7	148.7	0.0%	147.1	147.1	0.0%
AT(M)	1	714.1	718.1	0.6%	707.1	711.6	0.6%	699.7	704.4	0.7%
	2	705.6	708.4	0.4%	699.2	704.8	0.8%	691.3	696.5	0.8%
	3	733.1	736.8	0.5%	725.4	732.6	1.0%	717.9	724.5	0.9%
	4	794.0	796.4	0.3%	784.6	788.4	0.5%	776.1	781.9	0.7%
	5	646.8	651.2	0.7%	637.2	645.3	1.3%	630.0	636.2	1.0%
Poormond	1	109.1	108.7	-0.4%	103.8	105.8	1.9%	101.7	104.2	2.5%
	2	112.3	113.2	0.8%	106.7	107.6	0.8%	105.0	106.4	1.3%
	3	123.9	125.3	1.1%	117.3	119.2	1.6%	115.2	117.8	2.3%
	4	135.5	136.1	0.4%	129.6	131.4	1.4%	127.8	129.4	1.3%
	5	112.7	113.3	0.5%	91.0	92.4	1.5%	86.9	88.1	1.4%

Table 6.5: Objective lower bound improvements using convex valid inequalities.

Table 6.5 first shows that additional lower bound improvements are often less than 1%. For `Simple FSD`, no improvements are observed, and for `AT(M)`, improvements less than 1% are observed. For all except one `Poormond` instance, lower bounds are improved, often around 1% or 2%. As with the previous results, greater improvements are generally seen as the network size and temporal dimension grows, indicating that better bounds and more valid inequalities can improve objective lower bounds substantially.

6.6.5. Upper-bounding Experiments

Although not the primary focus of this chapter, this section aims at highlighting the difficulty in obtaining feasible *upper* bounds for challenging OWF instances, even with formulation and model strengthening techniques that provide tighter OWF relaxations. Specifically, we compare two relaxation-based formulations and measure their efficacy at determining upper bounds of each relaxed OWF *and* the original nonconvex OWF. The first is a (MILP) formulation that uses two and ten equally-spaced inner- and outer-approximation breakpoints for head loss and gain nonlinearities; assumes variable

bounds from OBBT; and continuously relaxes all flow direction variables y_{ij}^k, $(i,j) \in \mathcal{L}$, $k \in \mathcal{K}'$. This attempts to mimic some features of the relaxed formulation used by [15].

The second formulation is a (MILP) formulation that uses five inner- and outer-approximation breakpoints for pipes connected to nodes with total degree no greater than two, centered around the solution of a continuous relaxation of (MICP); two and five inner- and outer-approximation breakpoints for pipes connected to nodes with total degree greater than two; and five and ten inner- and outer-approximation breakpoints for pumps, again centered around the solution of (MICP)'s continuous relaxation. Additionally, the second formulation leverages OBBT-based bounds; OBCG- and network-based cuts; and the valid inequalities explored in Section 6.6.4. In summary, the second formulation is a stronger but more expensive relaxation when compared to the first.

Relaxation Upper Bounds Table 6.6 evaluates objective upper bounds obtained after one hour when solving each of the relaxed formulations discussed above. Here, UB^{R} corresponds to the first (highly relaxed) formulation, and UB^{P} corresponds to the second (stronger but more expensive) formulation. For `Simple FSD`, we note that all $K' = 24$ instances converge in under 4 seconds, and all $K' = 48$ instances converge in under 497 seconds. Likely due to the small size of the network, the first formulation finds only slightly smaller upper bounds than the second, as indicated by the "% Diff." results. The exception is for $K' = 48$, Day 5, where GUROBI reported a small constraint violation for the second formulation, which enabled the smaller objective upper bound.

For `AT(M)`, no instances are solved within the one hour time limit. For $K' = 12$, the stronger formulation finds smaller upper bounds that are likely nearer to physical feasibility. The stronger formulation also finds a feasible upper bound for one $K' = 24$ instance, whereas the weaker formulation finds no feasible solution. Nonetheless, for the remainder of the $K' \in \{24, 48\}$ OWF instances, no integer relaxation-feasible solutions can be found within the one hour time limit. This observation indicates that *even integer feasible solutions to relaxations of the OWF are often very difficult to obtain.*

For `Poormond`, again, no instances are solved to optimality within the one hour time limit. However, the results here are more intuitive, i.e., upper bounds achieved by the highly-relaxed problem are *smaller* than upper bounds from the stronger formulation. Furthermore, the first formulation produces feasible solutions more often than the second, as evidenced by $K' = 48$ results. Nonetheless, we remark that the feasible solutions

		$K' = 12$			$K' = 24$			$K' = 48$		
	Day	UB^R	UB^P	% Diff.	UB^R	UB^P	% Diff.	UB^R	UB^P	% Diff.
Simple FSD	1	-	-	-	155.3	155.5	0.1%	152.7	152.9	0.1%
	2	-	-	-	158.5	159.0	0.3%	157.0	157.0	0.0%
	3	-	-	-	172.0	172.7	0.4%	170.4	170.5	0.1%
	4	-	-	-	181.3	181.5	0.1%	177.8	177.9	0.1%
	5	-	-	-	148.5	149.1	0.4%	147.1	147.0	-0.1%
AT(M)	1	718.4	714.2	-0.6%	-	-	-	-	-	-
	2	712.6	707.5	-0.7%	-	-	-	-	-	-
	3	742.2	735.4	-0.9%	-	-	-	-	-	-
	4	797.1	792.8	-0.5%	-	-	-	-	-	-
	5	654.2	648.6	-0.9%	-	647.6	-	-	-	-
Poormond	1	112.0	113.9	1.7%	109.6	113.1	3.2%	109.7	-	-
	2	114.0	117.2	2.8%	111.4	114.7	3.0%	112.1	137.3	22.5%
	3	125.3	132.0	5.3%	123.0	130.5	6.1%	123.0	-	-
	4	139.7	142.2	1.8%	136.1	140.2	3.0%	136.2	-	-
	5	113.3	114.9	1.4%	95.3	112.1	17.6%	93.9	-	-

Table 6.6: Relaxation upper bound comparisons using different formulations.

from *either* formulation are not guaranteed to be *physically feasible* since they relax the original (MINLP). The most important takeaways, here, are (i) even loose relaxations of the OWF problem are often difficult to solve to optimality and (ii) solvers often struggle to find integer solutions that are feasible *to the relaxations themselves*.

Physically-feasible Upper Bounds This section expands upon the previous results, this time comparing physically-feasible upper bounds obtained by the relaxations. Here, a feasibility-checking procedure similar to the one in Chapter 5 was used within the MILP search. Specifically, at every integer-feasible node of the search tree, the feasibility-checking routine described in Section 6.4.1 was executed for the corresponding schedule of controllable components. For the solution of each convex subproblem, IPOPT with the same parameterization as discussed in Section 6.6.1 was used. As in [15] and [97], when a bound infeasibility is discovered through the extended period analysis

(via sequential subproblems), the following combinatorial no good cut can be appended:

$$\sum_{k=1}^{K^{\text{inf}}} \left(\sum_{(i,j)\in\mathcal{P}\cup\mathcal{V}:\hat{z}_{ij}^{k}=0} z_{ij}^{k} - \sum_{(i,j)\in\mathcal{P}\cup\mathcal{V}:\hat{z}_{ij}^{k}=1} (1 - z_{ij}^{k}) \right) \geq 1. \tag{6.66}$$

Here, $K^{\text{inf}} \in \mathcal{K}'$ is the first time interval at which infeasibility is detected in the extended period analysis, and $\hat{z}$ corresponds to the solution at the current BB node. This cut implies that at least one component status must change to address the infeasibility. If the integer solution is instead feasible, the solution to the extended period analysis is used to compute a *true* upper bound to (MINLP), which is not necessarily optimal.

In Table 6.7, the best (MINLP)-feasible upper bounds obtained over the search are represented as $\widetilde{\text{UB}}^{\text{R}}$ and $\widetilde{\text{UB}}^{\text{P}}$, respectively. For `Simple FSD`, all except two solutions are equivalent. Note that neither solution technique guarantees a *globally* optimal solution, as the MILP search terminates once the optimality gap *of the relaxation* has been closed. This explains the reason for the smaller upper bounds obtained by the *tighter* relaxations for `Simple FSD`. For `AT(M)`, *no physically-feasible solutions* are discovered for any of the problem instances during the search. This could be a consequence of the rarity in encountering even integer-feasible solutions *of the relaxation*, as discussed earlier. Finally, for `Poormond`, the tighter relaxation finds feasible solutions for more $K' = 12$ instances, but for Days 1 and 2, the weaker relaxation finds lower-cost feasible solutions. This indicates that *exploration* of integer-feasible relaxation solutions may be especially important for some OWF instances. Nonetheless, for the remaining larger `Poormond` instances, no (MINLP)-feasible solutions are found by either MILP relaxation.

6.7. Conclusion

This chapter explored relaxation-based solution techniques for the OWF problem, which aims at minimizing the cost of pump energy consumption over a fixed time horizon. Unlike previous studies, this chapter focused specifically on formulation, preprocessing, and algorithmic techniques that aim at improving objective lower bounds of OWF instances, which has been identified as a weakness of recent mathematical programming techniques (e.g., [15], [141]). First, the OWF was formulated as a MINCP, and a stan-

		$K' = 12$			$K' = 24$			$K' = 48$		
	Day	$\widetilde{\text{UB}}^{\text{R}}$	$\widetilde{\text{UB}}^{\text{P}}$	% Diff.	$\widetilde{\text{UB}}^{\text{R}}$	$\widetilde{\text{UB}}^{\text{P}}$	% Diff.	$\widetilde{\text{UB}}^{\text{R}}$	$\widetilde{\text{UB}}^{\text{P}}$	% Diff.
Simple FSD	1	-	-	-	155.6	155.6	0.0%	153.0	152.9	-0.1%
	2	-	-	-	159.0	159.0	0.0%	157.0	157.0	0.0%
	3	-	-	-	172.7	172.7	0.0%	170.6	170.6	0.0%
	4	-	-	-	182.2	181.5	-0.4%	178.0	178.0	0.0%
	5	-	-	-	149.1	149.1	0.0%	147.3	147.3	0.0%
AT(M)	1	-	-	-	-	-	-	-	-	-
	2	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-
	4	-	-	-	-	-	-	-	-	-
	5	-	-	-	-	-	-	-	-	-
Poormond	1	113.8	116.5	2.4%	-	-	-	-	-	-
	2	119.8	123.6	3.2%	-	-	-	-	-	-
	3	-	138.0	-	-	-	-	-	-	-
	4	-	141.9	-	-	-	-	-	-	-
	5	-	117.3	-	-	-	-	-	-	-

Table 6.7: Objective upper bound comparisons using different formulations.

dard MICP relaxation was introduced. To better leverage modern MILP solvers, a piecewise-linear relaxation-based formulation of the OWF was then developed, whose accuracy and tractability are controlled by the number of breakpoints used to model nonlinearities. These contributions generalize approaches used in prior OWF literature.

Expanding upon the previous literature, a duality-based convex valid inequality, similar to the one derived for water network design in Chapter 5, was described for use in the OWF problem. Additionally, a number of preprocessing techniques were developed to increase the tightness of relaxed OWF problem specifications. First, a single-step approximation of the OWF was formalized, which can be used in efficient preprocessing techniques. Then, similar to the OBBT method used by [15], a relaxation-based OBBT method was outlined. Finally, a novel OBCG method and generalizations of prior valid OWF inequalities were described that aim at further strengthening OWF relaxations.

The above model strengthening procedures were then sequentially evaluated by measuring objective lower bound improvements over small OWF instances. The empirical results indicate that these techniques are generally effective at increasing the lower bounds of particularly challenging instances, although the magnitude of these improvements

varies based on network structure and temporal complexity. Finally, upper-bounding experiments indicate that finding integer-feasible solutions, even to OWF relaxations, is a difficult task. Even more difficult is the task of finding (MINLP)-feasible solutions on large OWF instances. This indicates the need for better OWF heuristic algorithms.

Based on results of the upper-bounding experiments, future work should focus on the development of heuristic techniques capable of generating feasible solutions to OWF relaxations and (MINLP). As mentioned in Section 6.1, many existing techniques rely on metaheuristics, which allow network infeasibilities, or mathematical programming methods, which rely on specific assumptions concerning network structure. One promising general mathematical programming heuristic, however, has been recently proposed by [13], in which low-cost feasible solutions are obtained by solving a relaxed problem with preprocessed exact surrogate data that assume particular tank levels. Nonetheless, success of the approach appears highly dependent on the selection of tank levels during preprocessing and variations in OWF demand and price profiles. Another promising heuristic is used in the LP/NLP-BB method of [15], although it allows for continuous-duration pump activations, and thus solutions are generally not feasible to (MINLP).

A possible approach for finding good feasible solutions to (MINLP) could be one that efficiently leverages quickly-solvable partially-continuous relaxations of (MILP). Sections 6.6.2 and 6.6.3 highlighted the speed at which *temporal* relaxations of the OWF can be solved within preprocessing applications. This same intuition could be applied to a heuristic that *continuously relaxes* binary variables outside a given time window, *fixes* locally optimal operational decisions within the time window, and iteratively *repeats* the process until a (MINLP)-feasible operational schedule is discovered. Such a technique would have the benefit of heuristic subproblems that include a small number of binary variables while maintaining model accuracy in localized time windows of interest.

Chapter 7

Convex Relaxations of Maximal Load Delivery for Multi-contingency Analysis of Natural Gas Transmission Networks

Chapters 5 and 6 highlight the importance of convex reformulation and relaxation in the context of potable water distribution network optimization. However, many other important critical infrastructure networks are modeled using similar nonlinear nonconvex functions that describe energy losses. Natural gas transmission networks are one such network type crucial to the functioning of modern society. In Chapters 7 and 8, we apply convexification techniques to two novel optimization problems that aim at maximizing load delivery in a gas network that has undergone a severe disruption. Compared to Chapters 5 and 6, Chapters 7 and 8 differ in two broad respects. First, Chapters 5 and 6 developed *new techniques* for problems that have existed in the water resources literature for decades, whereas Chapters 7 and 8 leverage existing techniques to solve *new problems* in energy system optimization that are of practical and timely importance. Second, Chapters 5 and 6 placed an emphasis on finding *physically feasible* solutions, whereas Chapters 7 and 8 are concerned with finding *relaxation solutions*. This is motivated by the desire to *bound* properties of damaged gas and power networks.

Between 2012 and 2040, global electric power generation capacity is predicted to increase from 21.6 million gigawatt-hours (GWh) to 36.5 million GWh. Of this, gas-

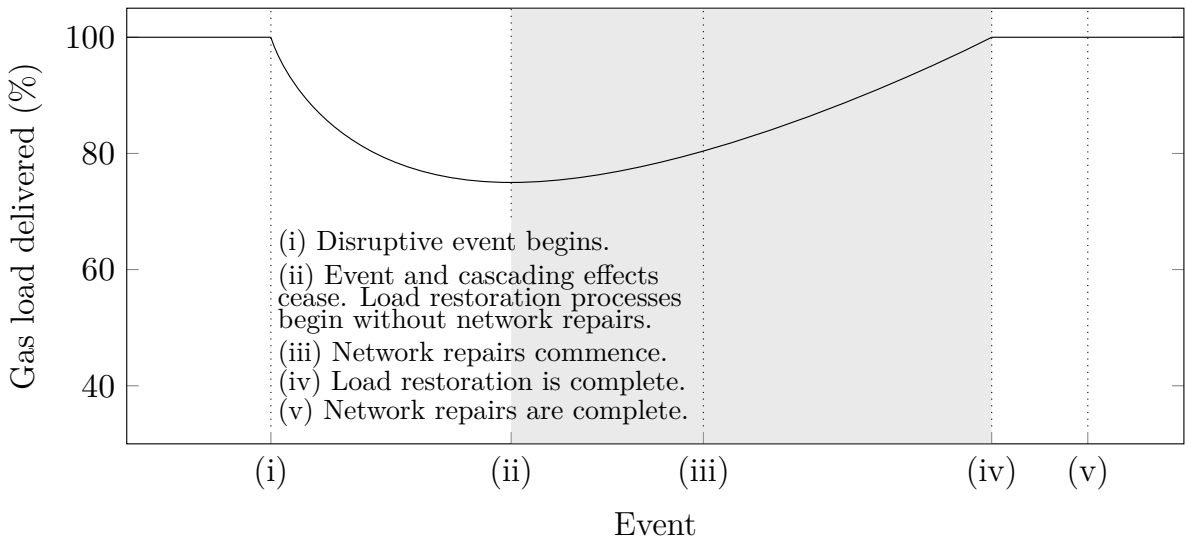


Figure 7.1: Illustration of a natural gas network's response to a severe disruption. The shaded region indicates the points in the disruption and restoration timeline that are studied in this chapter using an optimization-based assessment of damaged natural gas network capacities.

fired generation is expected to increase from 22% to 28% [39]. This growing dependence underscores the increasing sensitivity of power systems to upstream disruptions in gas pipelines. The most recent example is the February 2021 Texas power crisis, where the Electric Reliability Council of Texas experienced a loss of nearly 52.3 GW (48.6%) of its generation capacity. Nearly half of the loss was attributed to a lack of gas-fired power generation [112]. Other examples include the 2014 polar vortex, where curtailments in gas delivery resulted in roughly 25% of generation outages throughout the Pennsylvania-New Jersey-Maryland Interconnection [104]. Aside from such downstream effects on the power grid, these disruptions further inhibit the transport of fuel for residential heating, which provides essential temperature control to many individual homes during winter months. Mitigating the effects of these disruptions is thus critical to the resilience of gas and power delivery networks. To that end, this chapter examines how to compute the optimal response to a large-scale multi-contingency gas pipeline network disruption, whose origin (e.g., a natural hazard or sophisticated attack) is treated agnostically.

The scope of the multi-contingency response measures considered in this chapter is illustrated at a high level in Figure 7.1. When a severe disruption begins, (i) delivery of

load decreases as gas network components are damaged and cascading effects begin, and (ii) cascading effects cease, and a new stable operating point is found. After (ii), some amount of load can be restored through operational methods until (iii) network repairs commence. These repairs are conducted in accord with other restoration processes until (iv) all load can be delivered. Repairs continue until (v) all network components are again operational. Addressing the complete scope of Figure 7.1 is a substantial and complex task. In this chapter, we focus on determining optimal steady-state operating points between events of types (ii) and (iv), i.e., operational restoration decisions that enable the delivery of a maximal amount of load in the damaged gas pipeline network.

Several commercial tools exist for analyzing the operation of gas pipelines in the steady-state and transient regimes, including the NextGen pipeline simulation suite [53], Energy Solutions' gas management software [52], and Atmos Pipe [4]. These tools are designed for capacity planners to simulate the operation of a gas pipeline network under various physical conditions and flow nominations rather than for resilience analysis. When one or more components are set to be nonoperational (e.g., to simulate damage caused by a natural disaster) and must be removed from the network model, this must typically be done manually in such tools. Therefore, evaluating the impact of many thousands of multi-component outage scenarios using existing tools is too labor-intensive to be practical. Furthermore, because some deliveries (demand points) and receipts (supply points) may need to be adjusted (or omitted) because of component outages, a new feasible operating solution, including flow allocation and compressor settings, needs to be quickly obtained for each scenario. It is therefore unclear how existing commercial tools can be used for probabilistic risk assessment in this setting. This motivates a mathematical approach that can determine a feasible natural gas pipeline operating point that maximizes delivery subject to the many multi-contingency considerations.

In this chapter, we formalize this task as the steady-state MLD problem. Informally, the problem is as follows: given a severely damaged gas pipeline network in which a number of components have become nonoperational, we seek to maximize the amount of prioritized load that can be served in the damaged network subject to steady-state pipeline physical flow, capacity limits, pressure bounds, and other operating requirements. The nonlinear physics of gas transport and the discrete nature of operations in the network (i.e., the opening and closing of valves) render this problem a challenging MINCP. To address this, we introduce an MICP relaxation of the problem, which is

found to be a reliable means for determining bounds on the maximum deliverable load.

Recent interest in large-scale gas network planning and control has led to a variety of optimization-based applications and methods. A summary of studies connected to the optimization-based evaluation of gas network capacities is provided by [69]. In [125], detailed steady-state models and approximations of network components for use in optimization applications are presented. In [65], [64], and [70], mixed-integer programming approaches for optimization problems involving gas transport in the transient regime are proposed. Finally, [66], [67] describe relaxation methods for similar control problems.

The methodology of this chapter is primarily inspired by the success of approaches developed for multi-contingency analysis of power transmission networks. A similar optimization study on damaged power grids was motivated by the analysis of natural disaster vulnerabilities [34]. In that study, the MLD problem was formulated for alternating current (AC) power networks, and convex relaxations were developed that allow for the problem's efficient solution on large-scale instances. Another study introduced convex relaxations for gas network expansion planning [16]. Similar convex relaxations for gas pipeline flow have been explored subsequently [28], [145]. These relaxations inspire the convex reformulation of the MLD problem described in this chapter. Extending these studies, this chapter also presents nonconvex models and convex relaxations for modeling components that were previously not explicitly considered, e.g., resistors.

A number of studies consider problems similar to the one developed here. In [129] and [2], the problems of identifying the k critical components of a power transmission network and gas pipeline network are examined, respectively, whose simultaneous failure maximizes disruption to the network. Their studies use MLD problems as inner portions of broader bilevel interdiction problems. In contrast, we seek to determine an optimal operating point for a given disruption rather than find the worst-case disruption. In [7], the joint expansion planning problem for gas and power networks is considered. Their study, which links power generation to gas delivery, serves as inspiration for Chapter 8.

To solve the gas MLD problem, this chapter provides new contributions, including

- The first formulation of the steady-state MLD problem for gas networks;
- A mixed-integer nonconvex quadratic programming (MINQP) MLD reformulation;
- A mixed-integer convex quadratic programming (MICQP) MLD relaxation;
- A rigorous benchmarking of the formulations on instances of various sizes;
- A proof-of-concept MLD analysis for spatially distributed natural hazards.

Chapter Overview The remaining sections of this chapter proceed as follows: Section 7.1 formulates the requirements for gas pipeline feasibility as a MINCP; Section 7.2 formulates the MLD problem as a MINCP, then proposes an MICP relaxation; Section 7.3 rigorously benchmarks the nonconvex and convex formulations across several gas network data sets of various sizes and examines the use of the MLD method for probabilistic risk assessment for natural disasters; and Section 7.4 concludes the chapter.

7.1. Network Modeling

Section 2.7 rigorously derives the pressure-flow relationship for modeling losses in high-pressure natural gas pipelines. This relationship serves as the basis for modeling gas pipelines at a steady state. As in Equation (2.49), in this chapter, we assume all pipes are level. Additionally, while in practice the gas compressibility factor Z depends significantly on pressure and temperature in the physical regime of high pressure gas pipeline flow, in this chapter we use an ideal gas equation of state and suppose Z to be fixed. We further assume that the temperature T is constant along each pipe. This implies that the resistance in Equation (2.49) can be treated as fixed for our application. Aside from pipes, gas networks include a variety of other components [81], each of which is modeled using different sets of variables and constraints. The remainder of this section describes the nonconvex models of components considered when modeling a gas network.

Notation for Sets A natural gas transmission network is represented by a directed graph $\mathcal{G} := (\mathcal{J}, \mathcal{A})$, where $\mathcal{J}$ is the set of junctions (nodes) and $\mathcal{A}$ is the set of node-connecting components (arcs). The set of node-connecting components in the network includes horizontal pipes, short pipes, resistors, loss resistors, valves, regulators, and compressors. The set of receipts (producers) is denoted by $\mathcal{R}$ and deliveries (consumers) by $\mathcal{D}$. Receipts and deliveries are attached to existing junctions $i \in \mathcal{J}$. Furthermore, we let the subset of receipts attached to $i \in \mathcal{J}$ be denoted by $\mathcal{R}_i$ and the subset of deliveries by $\mathcal{D}_i$. The sets of horizontal and short pipes in the network are denoted by $\mathcal{P} \subset \mathcal{A}$ and $\mathcal{S} \subset \mathcal{A}$, respectively; the sets of regular and constant loss resistors by $\mathcal{T} \subset \mathcal{A}$ and $\mathcal{U} \subset \mathcal{A}$, respectively; the sets of valves and regulating (i.e., control) valves by $\mathcal{V} \subset \mathcal{A}$ and $\mathcal{W} \subset \mathcal{A}$, respectively; and the set of compressors by $\mathcal{C} \subset \mathcal{A}$. Finally, the set of node-connecting components incident to junction $i \in \mathcal{J}$ where i is the tail (respectively,

head) of the arc is denoted by $\delta_i^+ := \{(i,j) \in \mathcal{A}\}$ (respectively, $\delta_i^- := \{(j,i) \in \mathcal{A}\}$). We now examine each of the aforementioned components individually, define the decision variables, and present constraints that each component enforces on the gas network's operations. In particular, for each component, we present two types of constraints: (i) operational limits and (ii) physical constraints. We begin by examining junctions.

Junctions Each junction $i \in \mathcal{J}$ in the network is associated with a pressure variable, p_i. Operational limits require that this pressure resides between predefined lower and upper bounds, denoted by $\underline{p}_i$ and $\overline{p}_i$, respectively. This implies the first set of constraints,

$$0 \leq \underline{p}_i \leq p_i \leq \overline{p}_i, \ \forall i \in \mathcal{J}. \tag{7.1}$$

We note that typically, for a subset of predefined "slack junctions" $\mathcal{J}^s \subset \mathcal{J}$, $\underline{p}_i = \overline{p}_i$.

Node-connecting Components Every node-connecting component $(i,j) \in \mathcal{A}$ is associated with a decision variable, f_{ij}, which denotes the mass flow rate across that component. These variables satisfy the component capacity and flow velocity constraints

$$\underline{f}_{ij} \leq f_{ij} \leq \overline{f}_{ij}, \ \forall (i,j) \in \mathcal{A}. \tag{7.2}$$

As in the modeling of water networks, a positive (respectively, negative) value of f_{ij} implies mass flow that is directed from node i to j (respectively, j to i). In the forthcoming paragraphs, we present the constraints required for modeling the operational limits and physics of node-connecting components in the network, each of which has a flow f_{ij}.

Pipes Pipes transport gas throughout a pipeline network. We suppose that in steady-state flow, each pipe satisfies the pressure-flow Equation (2.49), which relates the mass flow rate f_{ij} through the cross-sectional area of the pipe to the pressures p_i and p_j at the two end-points of the pipe. With slight changes in notation, these relationships are

$$p_i^2 - p_j^2 = w_{ij} f_{ij} |f_{ij}|, \ \forall (i,j) \in \mathcal{P}, \tag{7.3}$$

where w_{ij} denotes the overall mass flow resistance of the pipe (including pipe length).

Short Pipes Short pipes in gas networks serve a similar purpose to short pipes in water networks, i.e., they ensure equality of pressures at the two connecting junctions:

$$p_i - p_j = 0, \ \forall (i,j) \in \mathcal{S}. \tag{7.4}$$

Resistors Aside from pipe pressure losses, a variety of other phenomena can also induce pressure loss. Examples include turbulence in shaped components, effects of measurement devices, curvature of piping, and partially closed valves. Resistors serve as surrogate modeling tools for representing these other forms of pressure loss. As in [81], losses across resistors are modeled via the Darcy-Weisbach-inspired equation

$$p_i - p_j = \tau_{ij} f_{ij} |f_{ij}|, \ \forall (i,j) \in \mathcal{T}. \tag{7.5}$$

Here, the resistance is defined by $\tau_{ij} := (8\kappa_{ij})/(\pi^2 D_{ij}^4 \rho_s)$, where κ_{ij} is the resistor's unitless drag factor, D_{ij} is the resistor's diameter, which may be an artificial quantity, and ρ_s is the average standard density of gas throughout the pipeline network.

Loss Resistors Loss resistors serve as an alternate form of the previous resistors, where instead of satisfying the Darcy-Weisbach equation, a fixed pressure loss $\xi_{ij} \geq 0$ is incurred across the component [81]. Modeling these pressure losses requires

$$f_{ij}(p_i - p_j) \geq 0, \ \forall (i,j) \in \mathcal{U} \tag{7.6a}$$

$$(p_i - p_j)^2 = \xi_{ij}^2, \ \forall (i,j) \in \mathcal{U}. \tag{7.6b}$$

Constraints (7.6a) ensure that each mass flow is in the direction of the pressure loss, and Constraints (7.6b) relate each pressure loss magnitude ξ_{ij} to the difference of pressures.

Valves Gas network valves serve similar purposes to the water network valves of Chapter 6. In practice, valves can also be partially closed to control the gas velocity, but in these cases, we choose to model the valve as a resistor [81]. Letting $z_{ij} \in \{0,1\}$ denote the operating status of each valve $(i,j) \in \mathcal{V}$, mass flows are first constrained as

$$\underline{f}_{ij} z_{ij} \leq f_{ij} \leq \overline{f}_{ij} z_{ij}, \ z_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{V}. \tag{7.7}$$

Furthermore, when a valve is open, the pressures at the junctions connected by that valve are equal. When the valve is closed, these pressures are decoupled. That is,

$$p_i \leq p_j + (1 - z_{ij})\overline{p}_i, \ \forall (i,j) \in \mathcal{V} \tag{7.8a}$$
$$p_j \leq p_i + (1 - z_{ij})\overline{p}_j, \ \forall (i,j) \in \mathcal{V}. \tag{7.8b}$$

Regulators Large pipes are usually operated at higher pressures than other portions of the network. As such, interconnection of large pipes with smaller pipes often requires the use of pressure regulators (i.e., control valves) to reduce pressure between differently-sized pipes. Regulators can also be used as an additional means of controlling flow throughout the network [81]. The operating status of a regulator is given using a binary variable $z_{ij} \in \{0,1\}$, where $z_{ij} = 1$ and $z_{ij} = 0$ indicate active and inactive statuses, respectively. The mass flow across each regulator is governed by bounding constraints

$$\underline{f}_{ij} z_{ij} \leq f_{ij} \leq \overline{f}_{ij} z_{ij}, \ z_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{W}. \tag{7.9}$$

Furthermore, each regulator (i,j) is associated with a multiplicative scaling factor, α_{ij}, that defines the relationship between p_i and p_j when the regulator is active, i.e., $\alpha_{ij} p_i = p_j$. This factor is constrained by operating limits as $\underline{\alpha}_{ij} = 0 \leq \alpha_{ij} \leq \overline{\alpha}_{ij} = 1$. As for valves, the pressures at the junctions connected by a regulator are decoupled when the regulator is inactive. This disjunctive relationship is modeled via the constraints

$$f_{ij}(p_i - p_j) \geq 0, \ \forall (i,j) \in \mathcal{W} \tag{7.10a}$$
$$\underline{\alpha}_{ij} p_i \leq p_j + (1 - z_{ij})\underline{\alpha}_{ij}\overline{p}_i, \ \forall (i,j) \in \mathcal{W} \tag{7.10b}$$
$$p_j \leq \overline{\alpha}_{ij} p_i + (1 - z_{ij})\overline{p}_j, \ \forall (i,j) \in \mathcal{W}. \tag{7.10c}$$

Here, Constraints (7.10a) ensure that each mass flow is in the same direction as the pressure loss, while Constraints (7.10b) and (7.10c) ensure the pressure at node j resides within the scaled bounds of the upstream pressure, p_i, when the control valve is open.

Compressors Each compressor $(i,j) \in \mathcal{C}$ increases the pressure at the downstream junction $j \in \mathcal{J}$ by a variable scalar α_{ij} and is assumed to have negligible length. For the benchmark networks considered in this chapter, bidirectional compressors do not exist, although each compressor may or may not allow for *uncompressed* flow in the reverse direction, i.e., from j to i. In this chapter, these distinct behaviors of compressors are captured by three conditional sets of constraints. The first set of constraints describes the behavior of compressors where uncompressed reverse flow is prohibited. That is,

$$\underline{\alpha}_{ij} p_i \leq p_j \leq \overline{\alpha}_{ij} p_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} \geq 0, \tag{7.11}$$

where $\underline{\alpha}_{ij}$ and $\overline{\alpha}_{ij}$, $(i,j) \in \mathcal{C}$, are minimum and maximum compression ratio bounds.

The second set of constraints describes the behavior of compressors where reverse flow *is* allowed and the minimum compression ratio is equal to one. These constraints are

$$\underline{\alpha}_{ij} p_i \leq p_j \leq \overline{\alpha}_{ij} p_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} = 1 \tag{7.12a}$$

$$f_{ij}(p_i - p_j) \leq 0, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} = 1. \tag{7.12b}$$

Here, Constraints (7.12b) ensure that, if $f_{ij} < 0$ (i.e., when there exists flow in the negative direction), then $p_i = p_j$ (i.e., there is no change in pressure across the compressor).

The final set of constraints describes the behavior of compressors where uncompressed reverse flow is allowed and the minimum compression ratio is *not* equal to one. In this case, the behavior of the compressor is modeled disjunctively. To accomplish this, for each compressor, a binary variable $y_{ij} \in \{0,1\}$ models the direction of flow through the compressor, where $y_{ij} = 1$ implies flow from i to j and $y_{ij} = 0$ implies flow from j to i. The pressures at the junctions that connect each compressor are then modeled via

$$y_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \neq 1 \tag{7.13a}$$

$$p_j \leq \overline{\alpha}_{ij} p_i + (1 - y_{ij}) \overline{p}_j, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \neq 1 \tag{7.13b}$$

$$\underline{\alpha}_{ij} p_i \leq p_j + (1 - y_{ij}) \overline{p}_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \neq 1 \tag{7.13c}$$

$$p_i - p_j \leq y_{ij} \overline{p}_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \neq 1 \tag{7.13d}$$

$$p_j - p_i \leq y_{ij} \overline{p}_j, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \neq 1. \tag{7.13e}$$

Here, Constraints (7.13b) and (7.13c) ensure that, when a compressor's flow is positively directed, the pressure at node j is modeled according to the compression ratio bounds, $\underline{\alpha}_{ij}$ and $\overline{\alpha}_{ij}$. Constraints (7.13d) and (7.13e) ensure that, when flow is traveling in reverse (i.e., when $y_{ij} = 0$), the pressures on both sides of the compressor are equal.

Receipts, Deliveries, and Mass Conservation Receipts and deliveries are points in the network that are attached to junctions where gas can be supplied to and withdrawn from the network, respectively. Each receipt (respectively, delivery) is associated with a nonnegative constant $\overline{s}_k$ (respectively, $\overline{d}_k$) that denotes the fixed mass supply (respectively, demand) at $k \in \mathcal{R}$ (respectively, $k \in \mathcal{D}$). Mass conservation throughout the network requires nodal balance constraints to be enforced at every junction, namely

$$\sum_{(i,j)\in\delta_i^+} f_{ij} - \sum_{(j,i)\in\delta_i^-} f_{ji} = \sum_{k\in\mathcal{R}_i} \overline{s}_k - \sum_{k\in\mathcal{D}_i} \overline{d}_k, \ \forall i \in \mathcal{J}. \tag{7.14}$$

Feasibility Problem Given the previous variables and constraints that model each component in the gas network, the MINCP for steady-state feasibility is defined here as

Pressure and mass flow bounds: Constraints (7.1), (7.2)
Pipe dynamics: Constraints (7.3)
Short pipe dynamics: Constraints (7.4)
Resistor dynamics: Constraints (7.5)
Loss resistor dynamics: Constraints (7.6) (MINCP-F)
Valve dynamics: Constraints (7.7), (7.8)
Regulator dynamics: Constraints (7.9), (7.10)
Compressor dynamics: Constraints (7.11), (7.12), (7.13)
Conservation of mass flow: Constraints (7.14).

The nonconvexities of the system of equations (MINCP-F) arise from three sources: (i) the discreteness of controllable components; (ii) bilinear variable products appearing in flow direction-related inequalities, i.e., Constraints (7.6a), (7.10a), and (7.12b); and (iii) nonlinear equations, i.e., Constraints (7.3), (7.5), and (7.6b). Section 7.2 describes a MINQP formulation addressing (ii), followed by a MICQP relaxation addressing (iii).

7.2. Maximal Load Delivery Formulations

This section formulates the MLD problem, which seeks to determine a feasible operating point for a damaged gas network that maximizes the delivery of load. To the best of our knowledge, this is the first time such a problem has been formulated for gas networks. Three variants of the MLD are presented in order of successive reformulations and relaxations that, though they increase problem complexity, decrease nonconvexity and computational difficulty. First, Section 7.2.1 extends the feasibility constraints of (MINCP-F) to formulate the initial MLD problem. Section 7.2.2 introduces an *exact* MINQP reformulation of this problem. Finally, to alleviate the challenges associated with nonlinear nonconvexity, an MICQP *relaxation* is presented in Section 7.2.3.

Aside from formulating the MLD problem, compared to previous studies (e.g., [2], [7], [16], [28], [145]), this section also (i) presents MIQP reformulations for a broader set of gas pipeline network components and (ii) introduces convex and MICQP relaxations for components that were not explicitly considered (e.g., resistors and loss resistors).

7.2.1. Mixed-integer Nonconvex Formulation

A damaged gas network requires the exclusion of components from the model described in Section 7.1. This set of excluded components comprises both the damaged components themselves as well as any connected components. For example, a damaged junction $i \in \mathcal{J}$ implies a nonoperational status of all node-connecting components $\delta_i^+ \cup \delta_i^-$. To this end, we use a tilde-based notation to denote the sets of components that are *operational* within a damaged gas network, e.g., $\widetilde{\mathcal{P}} \subseteq \mathcal{P}$ denotes the set of operational pipes.

The fundamental motivation for formulating the MLD problem is that a damaged gas network may not be able to satisfy all demands of the original system. This implies a possible imbalance of the mass conservation Constraints (7.14). To address this, in the MLD problem, all receipts and deliveries are treated as dispatchable. This implies the constant supplies and demands, $\overline{s}_k$, $k \in \mathcal{R}$, and $\overline{d}_k$, $k \in \mathcal{D}$, are variables bounded as

$$0 \leq s_k \leq \overline{s}_k, \ \forall k \in \mathcal{R}_i, \ \forall i \in \widetilde{\mathcal{J}} \tag{7.15a}$$

$$0 \leq d_k \leq \overline{d}_k, \ \forall k \in \mathcal{D}_i, \ \forall i \in \widetilde{\mathcal{J}}, \tag{7.15b}$$

where $\overline{s}_k$ and $\overline{d}_k$ denote the original supplies and demands, respectively. Using these

variable supplies and demands, the mass flow conservation Constraints (7.14) become

$$\sum_{(i,j)\in\tilde{\delta}_i^+} f_{ij} - \sum_{(j,i)\in\tilde{\delta}_i^-} f_{ji} = \sum_{k\in\mathcal{R}_i} s_k - \sum_{k\in\mathcal{D}_i} d_k, \ \forall i \in \widetilde{\mathcal{J}}. \tag{7.16}$$

Next, parameters $\beta_k \geq 0$, $k \in \mathcal{D}$, are introduced to denote load restoration priorities. If no load priorities are available, values of one can be used instead. (Indeed, this parameterization is used for all of our experiments in Section 7.3.) Maximization of prioritized load delivered then implies maximization of the linear objective function

$$\eta(d) = \sum_{i\in\widetilde{\mathcal{J}}} \sum_{k\in\mathcal{D}_i} \beta_k d_k. \tag{7.17}$$

The MINCP formulation of the MLD problem is then written as

$$\begin{array}{ll} \text{maximize} & \text{Objective function: } \eta(d) \text{ of Equation (7.17)} \\ \text{subject to} & \text{Supply and demand bounds: Constraints (7.15)} \\ & \text{Conservation of mass flow: Constraints (7.16)} \\ & \text{(MINCP-F) without Constraints (7.14),} \end{array} \tag{MINCP}$$

where in (MINCP-F), all component sets are assumed to be replaced with their tilde-denoted (i.e., undamaged) counterparts, indicating the application of a damage scenario.

7.2.2. Mixed-integer Nonconvex Quadratic Reformulation

As described at the end of Section 7.1, one source of nonconvexity in (MINCP) is the existence of bilinear variable products appearing in flow direction-related inequalities. To address these nonconvexities, this subsection introduces (i) binary direction variables $y_{ij} \in \{0, 1\}$ for all $(i, j) \in \widetilde{\mathcal{A}}$ and (ii) squared pressure variables $\pi_i = p_i^2$ for all $i \in \widetilde{\mathcal{J}}$, which allows for the construction of an *exact* MINQP reformulation of the original problem. That is, the new reformulation contains fewer nonlinearities than (MINCP) but has the same solution set. This new formulation enables *global* solutions to be found with modern MINQP solvers (namely, Gurobi). Later, in Section 7.2.3, this directed formulation is relaxed to form an MICQP relaxation of the original MLD problem.

Junctions Squared pressures must first reside between predefined bounds, i.e.,

$$\underline{\pi}_i \leq \pi_i \leq \overline{\pi}_i, \ \forall i \in \widetilde{\mathcal{J}}. \tag{7.18}$$

where $\underline{\pi}_i$ and $\overline{\pi}_i$, $i \in \widetilde{\mathcal{J}}$ are derived from the original bounds $\underline{p}_i$ and $\overline{p}_i$, respectively.

Node-connecting Components Using the binary flow direction variables, the mass flow bounds of all node-connecting components are restricted by the constraints

$$(1 - y_{ij})\underline{f}_{ij} \leq f_{ij} \leq y_{ij}\overline{f}_{ij}, \ y_{ij} \in \{0, 1\}, \ \forall (i, j) \in \widetilde{\mathcal{A}}. \tag{7.19}$$

Pipes Squared pressure variables and directions allow for the reduction of nonlinearities in the pressure-flow Constraints (7.3) for pipes. These constraints are rewritten as

$$\pi_i - \pi_j \geq w_{ij} f_{ij}^2 - (1 - y_{ij}) \left[w_{ij} \underline{f}_{ij}^2 - (\underline{\pi}_i - \overline{\pi}_j) \right], \ \forall (i, j) \in \widetilde{\mathcal{P}} \tag{7.20a}$$

$$\pi_i - \pi_j \leq w_{ij} f_{ij}^2, \ \forall (i, j) \in \widetilde{\mathcal{P}} \tag{7.20b}$$

$$\pi_j - \pi_i \geq w_{ij} f_{ij}^2 - y_{ij} \left[w_{ij} \overline{f}_{ij}^2 - (\underline{\pi}_j - \overline{\pi}_i) \right], \ \forall (i, j) \in \widetilde{\mathcal{P}} \tag{7.20c}$$

$$\pi_j - \pi_i \leq w_{ij} f_{ij}^2, \ \forall (i, j) \in \widetilde{\mathcal{P}}. \tag{7.20d}$$

This is similar to the reformulation presented by [16]. Here, each Constraint (7.20a) ensures a pressure decrease from i to j when $y_{ij} = 1$. Constraint (7.20b) ensures the pressure-flow equation is satisfied when $y_{ij} = 1$. Constraint (7.20c) ensures a pressure decrease from j to i when $y_{ij} = 0$. Constraint (7.20d) ensures the pressure-flow equation is satisfied when $y_{ij} = 0$. Direction variables bound squared pressure differences via

$$(1 - y_{ij})(\underline{\pi}_i - \overline{\pi}_j) \leq \pi_i - \pi_j, \ \forall (i, j) \in \widetilde{\mathcal{P}} \tag{7.21a}$$

$$\pi_i - \pi_j \leq y_{ij}(\overline{\pi}_i - \underline{\pi}_j), \ \forall (i, j) \in \widetilde{\mathcal{P}}. \tag{7.21b}$$

Short Pipes Squared pressure variables also allow for rewriting Constraints (7.4) as

$$\pi_i - \pi_j = 0, \ \forall (i,j) \in \widetilde{\mathcal{S}}. \tag{7.22}$$

As with pipes, direction variables are used to bound squared pressures via

$$(1 - y_{ij})(\underline{\pi}_i - \overline{\pi}_j) \leq \pi_i - \pi_j, \ \forall (i,j) \in \widetilde{\mathcal{S}} \tag{7.23a}$$
$$\pi_i - \pi_j \leq y_{ij}(\overline{\pi}_i - \underline{\pi}_j), \ \forall (i,j) \in \widetilde{\mathcal{S}}. \tag{7.23b}$$

Resistors For junctions that are connected to a resistor, the squared pressure variables π_i, $i \in \widetilde{\mathcal{J}}$, must first be related to variables denoting nonsquared pressures, i.e.,

$$p_i^2 - \pi_i = 0, \ i \in \widetilde{\mathcal{J}} : \left(\exists (i,j) \in \widetilde{\mathcal{T}}\right) \vee \left(\exists (j,i) \in \widetilde{\mathcal{T}}\right). \tag{7.24}$$

The potential loss relationships are then written in a manner as for pipes, i.e.,

$$p_i - p_j \geq \tau_{ij} f_{ij}^2 - (1 - y_{ij}) \left[\tau_{ij} \underline{f}_{ij}^2 - (\underline{p}_i - \overline{p}_j)\right], \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.25a}$$
$$p_i - p_j \leq \tau_{ij} f_{ij}^2, \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.25b}$$
$$p_j - p_i \geq \tau_{ij} f_{ij}^2 - y_{ij} \left[\tau_{ij} \overline{f}_{ij}^2 - (\underline{p}_j - \overline{p}_i)\right], \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.25c}$$
$$p_j - p_i \leq \tau_{ij} f_{ij}^2, \ \forall (i,j) \in \widetilde{\mathcal{T}}, \tag{7.25d}$$

Direction variables are also used to bound the original pressure variables via

$$(1 - y_{ij})(\underline{p}_i - \overline{p}_j) \leq p_i - p_j, \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.26a}$$
$$p_i - p_j \leq y_{ij}(\overline{p}_i - \underline{p}_j), \ \forall (i,j) \in \widetilde{\mathcal{T}}. \tag{7.26b}$$

Loss Resistors As for resistors, squared pressure variables π_i, $i \in \widetilde{\mathscr{J}}$, for junctions connected to a loss resistor, must be related to nonsquared pressure variables via

$$p_i^2 - \pi_i = 0, \ i \in \widetilde{\mathscr{J}} : \left(\exists (i,j) \in \widetilde{\mathscr{U}}\right) \vee \left(\exists (j,i) \in \widetilde{\mathscr{U}}\right). \tag{7.27}$$

Note that Constraints (7.6) can be rewritten using absolute values as

$$\xi_{ij} = |p_i - p_j|, \ \forall (i,j) \in \widetilde{\mathscr{U}}. \tag{7.28}$$

However, directions y_{ij} allow this disjunctive form to be modeled linearly, i.e.,

$$(2y_{ij} - 1)\xi_{ij} = p_i - p_j, \ \forall (i,j) \in \widetilde{\mathscr{U}}. \tag{7.29}$$

Here, the direction of pressure loss coincides with the direction as required by y_{ij}.

Valves Pressure Constraints (7.8) are written with squared pressures as

$$\pi_i \leq \pi_j + (1 - z_{ij})\overline{\pi}_i, \ \forall (i,j) \in \widetilde{\mathscr{V}} \tag{7.30a}$$
$$\pi_j \leq \pi_i + (1 - z_{ij})\overline{\pi}_j, \ \forall (i,j) \in \widetilde{\mathscr{V}}. \tag{7.30b}$$

Regulators Pressure Constraints (7.10) are written with squared pressures as

$$\pi_j - \overline{\alpha}_{ij}^2 \pi_i \leq (2 - y_{ij} - z_{ij})\overline{\pi}_j, \ \forall (i,j) \in \widetilde{\mathscr{W}} \tag{7.31a}$$
$$\underline{\alpha}_{ij}^2 \pi_i - \pi_j \leq (2 - y_{ij} - z_{ij})\overline{\pi}_i, \ \forall (i,j) \in \widetilde{\mathscr{W}} \tag{7.31b}$$
$$\pi_j - \pi_i \leq (1 + y_{ij} - z_{ij})\overline{\pi}_j, \ \forall (i,j) \in \widetilde{\mathscr{W}} \tag{7.31c}$$
$$\pi_i - \pi_j \leq (1 + y_{ij} - z_{ij})\overline{\pi}_i, \ \forall (i,j) \in \widetilde{\mathscr{W}}. \tag{7.31d}$$

Here, when $y_{ij} = 1$ and $z_{ij} = 1$ (i.e., the regulating valve is open and the flow direction is positive), Constraints (7.31a) and (7.31b) ensure that π_j resides between the scaled value of π_i. When $y_{ij} = 0$ and $z_{ij} = 1$, Constraints (7.31c) and (7.31d) ensure that $\pi_i = \pi_j$, and reverse flow is allowed. Finally, when the valve is closed, $z_{ij} = 0$ and $y_{ij} \in \{0, 1\}$, which ensures that the squared pressures π_i and π_j are decoupled.

Compressors Constraints (7.11), which model compressors where uncompressed reverse flow is prohibited, are first rewritten using squared pressure variables as

$$\underline{\alpha}_{ij}^2 \pi_i \leq \pi_j \leq \overline{\alpha}_{ij}^2 \pi_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} \geq 0. \tag{7.32}$$

For compressors that allow reverse flow, Constraints (7.12) and (7.13) are written as

$$\pi_j \leq \overline{\alpha}_{ij}^2 \pi_i + (1 - y_{ij}) \overline{\pi}_j, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \tag{7.33a}$$

$$\underline{\alpha}_{ij}^2 \pi_i \leq \pi_j + (1 - y_{ij}) \underline{\alpha}_{ij}^2 \overline{\pi}_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \tag{7.33b}$$

$$\pi_i - \pi_j \leq y_{ij} \overline{\pi}_i, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \tag{7.33c}$$

$$\pi_j - \pi_i \leq y_{ij} \overline{\pi}_j, \ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0. \tag{7.33d}$$

Here, when $y_{ij} = 1$, Constraints (7.33a) and (7.33b) require π_j to reside within the scaled bounds of π_i. When $y_{ij} = 0$, Constraints (7.33c) and (7.33d) ensure the equality of pressures when flow is directed negatively from j to i (i.e., there is no compression).

Direction-related Valid Inequalities The formulations of [16] include inequalities that relate node-connecting component directions y_{ij} to nodal conditions in the directed network. These inequalities improve relaxations. Here, we employ constraints that model flow directionality at junctions with zero supply, zero demand, and degree two:

$$\sum_{(i,j) \in \tilde{\delta}_i^-} y_{ij} - \sum_{(i,j) \in \tilde{\delta}_i^+} y_{ij} = 0, \ i \in \widetilde{\mathcal{J}}_0 : \left(\left| \tilde{\delta}_i^{\pm} \right| = 1 \right) \tag{7.34a}$$

$$\sum_{(i,j) \in \tilde{\delta}_i^-} y_{ij} + \sum_{(i,j) \in \tilde{\delta}_i^+} y_{ij} = 1, \ i \in \widetilde{\mathcal{J}}_0 : \left(\left| \tilde{\delta}_i^{\pm} \right| = 2 \right) \wedge \left(\left| \tilde{\delta}_i^{\mp} \right| = 0 \right), \tag{7.34b}$$

where $\widetilde{\mathcal{J}}_0 \subset \widetilde{\mathcal{J}}$ denotes the subset of junctions that satisfy the condition $\sum_{k \in \mathcal{R}_i} \overline{s}_i = \sum_{k \in \mathcal{D}_i} \overline{d}_i = 0$. These valid inequalities imply that, for this particular subset of junctions, the direction of incoming mass flow must be equal to the direction of outgoing mass flow.

Reformulation The preceding changes enable (MINCP) to be reformulated as

maximize	Objective function: $\eta(d)$ of Equation (7.17)	
subject to	Supply and demand bounds: Constraints (7.15)	
	Conservation of mass flow: Constraints (7.16)	
	Pressure bounds: Constraints (7.1), (7.18)	
	Directed mass flow bounds: Constraints (7.19)	
	Pipe dynamics: Constraints (7.20), (7.21)	
	Short pipe dynamics: Constraints (7.22), (7.23)	(MINQP)
	Resistor dynamics: Constraints (7.24), (7.25), (7.26)	
	Loss resistor dynamics: Constraints (7.27), (7.29)	
	Valve dynamics: Constraints (7.7), (7.30)	
	Regulator dynamics: Constraints (7.9), (7.31)	
	Compressor dynamics: Constraints (7.32), (7.33)	
	Direction-related cuts: Constraints (7.34).	

7.2.3. Mixed-integer Convex Quadratic Relaxation

There are a number of nonconvex nonlinear constraints in (MINQP) that render it intractable with increasing network size. Specifically, these are Constraints (7.20b), (7.20d), (7.24), (7.25b), (7.25d), and (7.27). In this section, we apply convex relaxation strategies, as done in [16], [145], and [28], to address this. We also extend these studies by formulating relaxations for components that were not previously considered, e.g., resistors and loss resistors, which require constraints involving explicit pressure variables.

Pipes We first apply convex relaxations to the pressure-flow Constraints (7.20) for pipes. The variables ℓ_{ij} for $(i,j) \in \widetilde{\mathcal{P}}$ are introduced to denote the difference in squared

pressures across each pipe. The convex relaxation constraints are then formulated as

$$\pi_j - \pi_i \leq \ell_{ij} \leq \pi_i - \pi_j, \ \forall (i,j) \in \widetilde{\mathcal{P}} \tag{7.35a}$$
$$\ell_{ij} \leq \pi_j - \pi_i + (2y_{ij})(\overline{\pi}_i - \underline{\pi}_j), \ \forall (i,j) \in \widetilde{\mathcal{P}} \tag{7.35b}$$
$$\ell_{ij} \leq \pi_i - \pi_j + (2y_{ij} - 2)(\underline{\pi}_i - \overline{\pi}_j), \ \forall (i,j) \in \widetilde{\mathcal{P}} \tag{7.35c}$$
$$w_{ij} f_{ij}^2 \leq \ell_{ij}, \ \forall (i,j) \in \widetilde{\mathcal{P}} \tag{7.35d}$$
$$\ell_{ij} \leq w_{ij}\overline{f}_{ij} f_{ij} + (1 - y_{ij}) \left(\left| w_{ij}\overline{f}_{ij}\underline{f}_{ij} \right| + w_{ij}\underline{f}_{ij}^2 \right), \ \forall (i,j) \in \widetilde{\mathcal{P}} \tag{7.35e}$$
$$\ell_{ij} \leq w_{ij}\underline{f}_{ij} f_{ij} + y_{ij} \left(\left| w_{ij}\overline{f}_{ij}\underline{f}_{ij} \right| + w_{ij}\overline{f}_{ij}^2 \right), \ \forall (i,j) \in \widetilde{\mathcal{P}}. \tag{7.35f}$$

Here, Constraints (7.35a) ensure each loss of squared pressures resides between the corresponding differences. Constraints (7.35b) ensure that when $y_{ij} = 0$, each loss is bounded by $\pi_j - \pi_i$, and Constraints (7.35c) imply that each loss is bounded by $\pi_i - \pi_j$ when $y_{ij} = 1$. Constraints (7.35d) are relaxations of the pressure-loss equations. Finally, Constraints (7.35e) and (7.35f) apply linear upper bounds on each variable ℓ_{ij}.

Resistors We next apply convex relaxations to Constraints (7.24), which relate non-squared and squared pressure variables. These convex relaxations yield the constraints

$$p_i^2 \leq \pi_i, \ i \in \widetilde{\mathcal{J}} : \left(\exists (i,j) \in \widetilde{\mathcal{T}} \right) \vee \left(\exists (j,i) \in \widetilde{\mathcal{T}} \right). \tag{7.36}$$

Constraints (7.25) are then convexly relaxed as done for Constraints (7.35). That is,

$$p_j - p_i \leq \ell_{ij} \leq p_i - p_j, \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.37a}$$
$$\ell_{ij} \leq p_j - p_i + (2y_{ij})(\overline{p}_i - \underline{p}_j), \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.37b}$$
$$\ell_{ij} \leq p_i - p_j + (2y_{ij} - 2)(\underline{p}_i - \overline{p}_j), \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.37c}$$
$$\tau_{ij} f_{ij}^2 \leq \ell_{ij}, \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.37d}$$
$$\ell_{ij} \leq \tau_{ij}\overline{f}_{ij} f_{ij} + (1 - y_{ij}) \left(\left| \tau_{ij}\overline{f}_{ij}\underline{f}_{ij} \right| + \tau_{ij}\underline{f}_{ij}^2 \right), \ \forall (i,j) \in \widetilde{\mathcal{T}} \tag{7.37e}$$
$$\ell_{ij} \leq \tau_{ij}\underline{f}_{ij} f_{ij} + y_{ij} \left(\left| \tau_{ij}\overline{f}_{ij}\underline{f}_{ij} \right| + \tau_{ij}\overline{f}_{ij}^2 \right), \ \forall (i,j) \in \widetilde{\mathcal{T}}. \tag{7.37f}$$

Loss Resistors As we have done for resistors, we next apply convex relaxations to Constraints (7.27) relating nonsquared and squared pressure variables. We obtain

$$p_i^2 \leq \pi_i, \ i \in \widetilde{\mathscr{J}} : \left(\exists (i,j) \in \widetilde{\mathscr{U}}\right) \vee \left(\exists (j,i) \in \widetilde{\mathscr{U}}\right). \tag{7.38}$$

Relaxation Using the convexification, the MICQP relaxation of (MINQP) is then

maximize	Objective function: $\eta(d)$ of Equation (7.17)	
subject to	Supply and demand bounds: Constraints (7.15)	
	Conservation of mass flow: Constraints (7.16)	
	Pressure bounds: Constraints (7.1), (7.18)	
	Directed mass flow bounds: Constraints (7.19)	
	Pipe dynamics: Constraints (7.21), (7.35)	
	Short pipe dynamics: Constraints (7.22), (7.23)	(MICQP)
	Resistor dynamics: Constraints (7.26), (7.36), (7.37)	
	Loss resistor dynamics: Constraints (7.29), (7.38)	
	Valve dynamics: Constraints (7.7), (7.30)	
	Regulator dynamics: Constraints (7.9), (7.31)	
	Compressor dynamics: Constraints (7.32), (7.33)	
	Direction-related cuts: Constraints (7.34).	

In Section 7.3, we present the results of a comprehensive computational study performed to compare the tractability and application of the exact and relaxed MLD problem formulations, (MINQP) and (MICQP), respectively. These computations display the practical benefits of employing the relaxation, (MICQP), for decision support applications.

7.3. Computational Experiments

This section experimentally analyzes the applicability and computational performance of the (MINQP) and (MICQP) MLD formulations presented in Section 7.2. This informs us of the practical, analytical, and computational tradeoffs associated with using the exact

and relaxed MLD problem formulations, respectively. To accomplish this, we consider three different types of damage scenarios: (i) $N-1$ or single contingency scenarios, (ii) $N-k$ or multi-contingency scenarios, and (iii) earthquake damage scenarios. Each scenario is intended to simulate common sources of damage to a gas pipeline network. Although these damage models arise from reasonable assumptions, we do not claim to quantify the robustness of the specific networks considered. Rather, these models serve as proofs of concept for studying three aspects of the MLD problem: (i) understanding the tractability of MLD formulations, (ii) highlighting the qualitative insights given by an MLD analysis, and (iii) providing guidelines for applications to real-world scenarios.

Both MLD formulations were implemented in the Julia programming language using the mathematical modeling layer JuMP, version 0.21 [50], and version 0.8 of GasModels, an open-source Julia package for steady-state and transient natural gas network optimization [9]. Section 7.3.1 describes the instances, computational resources, and parameters used throughout these experiments; Section 7.3.2 compares the efficacy of MLD formulations on $N-1$ contingency scenarios for each network; Section 7.3.3 does the same for randomized $N-k$ multi-contingency scenarios, where k corresponds to 15% of node-connecting components in each network; and Section 7.3.4 compares the run-time performance of the formulations over both the $N-1$ and $N-k$ experiment sets. Finally, Section 7.3.5 provides a proof-of-concept application of the MLD problem to hypothetical damage scenarios precipitated by deterministic or stochastic hazards.

7.3.1. Experimental Test Data & Setup

The numerical experiments consider networks of various sizes that appear in the literature of natural gas transmission network modeling or are derivable from open data. These instances are summarized in Table 7.1. Here, the `Belgium-20` network was derived from the application of [46]; the North American 154-junction network (i.e., `NA-154`) was derived by subject matter experts using public data; and `GasLib` networks were obtained directly from [124]. For the `GasLib-582` and `GasLib-4197` networks, the `nomination_freezing_1` and `nomination_mild_0006` delivery and receipt nominations were used, respectively. Generally, steady-state optimization problems involving most networks can be solved to optimality given a small amount of time (e.g., seconds to minutes). The exception is `GasLib-4197`, which requires hours to solve many instances.

Network	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{W}\rvert$	$\lvert\mathcal{C}\rvert$
GasLib-11	11	8	0	0	0	1	0	2
Belgian-20	20	24	0	0	0	0	0	3
GasLib-24	24	19	1	1	0	0	1	3
GasLib-40	40	39	0	0	0	0	0	6
GasLib-134	134	86	45	0	0	0	1	1
GasLib-135	135	141	0	0	0	0	0	29
NA-154	154	140	0	0	0	0	0	12
GasLib-582	582	278	269	8	0	26	23	5
GasLib-4197	4197	3537	343	22	6	426	120	12

Table 7.1: Summary of natural gas transmission networks derived from open data.

Each optimization computation was provided a wall-clock time of one hour on a node containing two Intel Xeon E5-2695 v4 processors, each with 18 cores at 2.10 GHz, and 125 GB of memory. For solutions of the MINQP and MICQP formulations, GUROBI 9.0 was used with the parameter `MIPGap=0.0`. For experiments employing (MINQP), the setting `NonConvex=2` was used, which allows for global optimization of this formulation.

7.3.2. Single Contingency Damage Scenarios

The first damage model considered is the single contingency or $N-1$ model, where N corresponds to the original number of network components and $N-1$ indicates that an individual component is removed (or damaged). This model can be thought of as a method for simulating the effects of an unscheduled component outage. Here, it is assumed that both nodal components (i.e., junctions) and node-connecting components (e.g., pipes) can comprise an $N-1$ damage scenario. In our study, this damage model is intended to demonstrate feasibility of the MLD problem for a broad variety of network structures and to validate our network modeling assumptions for damaged gas networks.

For each network, the set of all such possible $N-1$ scenarios was considered, and the corresponding instances were solved using both the (MINQP) and (MICQP) MLD formulations. The exception is `GasLib-4197`, which was limited to 1,040 scenarios because of computational restrictions. Table 7.2 displays statistics of solver termination statuses across all scenarios for each network and formulation. For all networks except `GasLib-135`, `GasLib-582`, and `GasLib-4197`, optimal solutions to all instances are found

	(MINQP)			(MICQP)		
Network	Opt. (%)	Lim. (%)	Inf. (%)	Opt. (%)	Lim. (%)	Inf. (%)
GasLib-11	100.0	0.0	0.0	100.0	0.0	0.0
Belgium-20	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-24	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-40	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-134	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-135	90.2	9.8	0.0	100.0	0.0	0.0
NA-154	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-582	79.6	20.4	0.0	100.0	0.0	0.0
GasLib-4197	3.2	90.3	6.5	43.8	56.2	0.0

Table 7.2: Comparison of solver statuses over $N-1$ scenarios. Here, "Opt." corresponds to instances where optimality is proven, "Lim." to instances where the solver time limit has been reached, and "Inf." to instances that are claimed by the solver (in this case, Gurobi) to be infeasible.

within the prescribed one hour time limit. For GasLib-135, the (MINQP) formulation is unable to prove global optimality on 98 instances. Notably, GasLib-135 contains the largest number of compressors compared to other benchmark networks, and these additional degrees of freedom are the source of computational complexity. For GasLib-582 and GasLib-4197, the large number of (MINQP) instances that cannot be solved to global optimality is most likely because of the networks' comparatively large sizes.

Comparing the two formulations shows the benefit of using the relaxation-based (MICQP) approach, which is capable of solving much larger proportions of challenging GasLib-135, GasLib-582, and GasLib-4197 instances. This suggests that (MICQP), when compared to the mixed-integer nonconvex (MINQP), is often a better candidate for numerically intensive applications. We also note that, when using the default Gurobi parameters described in Section 7.3.1, 292 of 1,040 GasLib-4197 instances of the (MINQP) MLD formulation are claimed to be infeasible. Using the Gurobi parameter NumericFocus=3 for this subset of 292 instances, however, decreases this number to 68. These claimed infeasibilities are likely related to the numerical properties of the GasLib-4197 data set rather than our MLD formulations. Additional future work to preprocess data and rescale constraints would be warranted to address these issues.

	(MINQP)			(MICQP)		
Network	Opt. (%)	Lim. (%)	Inf. (%)	Opt. (%)	Lim. (%)	Inf. (%)
GasLib-11	100.0	0.0	0.0	100.0	0.0	0.0
Belgium-20	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-24	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-40	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-134	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-135	99.6	0.4	0.0	100.0	0.0	0.0
NA-154	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-582	99.9	0.1	0.0	100.0	0.0	0.0
GasLib-4197	92.7	7.3	0.0	99.2	0.8	0.0

Table 7.3: Comparison of solver statuses over $N-k$ scenarios. Here, k corresponds to a random selection of 15% of all node-connecting components.

7.3.3. Multi-contingency Damage Scenarios

The second damage model is the multi-contingency or $N-k$ model, where k corresponds to the number of components that are simultaneously removed from the network. These scenarios are intended to capture the effects of multimodal network failures. Here, we consider the removal of only node-connecting components within the generated $N-k$ scenarios. In each scenario, a uniformly random selection of 15% node-connecting components were assumed to be nonoperational. Heuristically, this proportion of components seemed to generate challenging scenarios while providing different maximal load distributions across the networks considered. For each network, one thousand such scenarios were generated. If solved, the maximal proportional load delivered in each experiment was then computed as the ratio of the optimal nonprioritized objective in Equation (7.17) and the load capable of being delivered in the undamaged network.

Table 7.3 displays statistics of solver termination statuses across all damage scenarios for each network and formulation. For all except GasLib-135, GasLib-582, and GasLib-4197, globally optimal solutions to MLD instances are found within the one hour time limit. For GasLib-135 and GasLib-582, the (MINQP) formulation is unable to prove optimality on four instances and one instance, respectively. For GasLib-4197, a larger proportion cannot be solved. Comparing the two formulations again shows the benefit of the relaxation-based approach, which solves larger proportions of challenging instances. We note that one GasLib-4197 (MINQP) instance is claimed to be infeasible

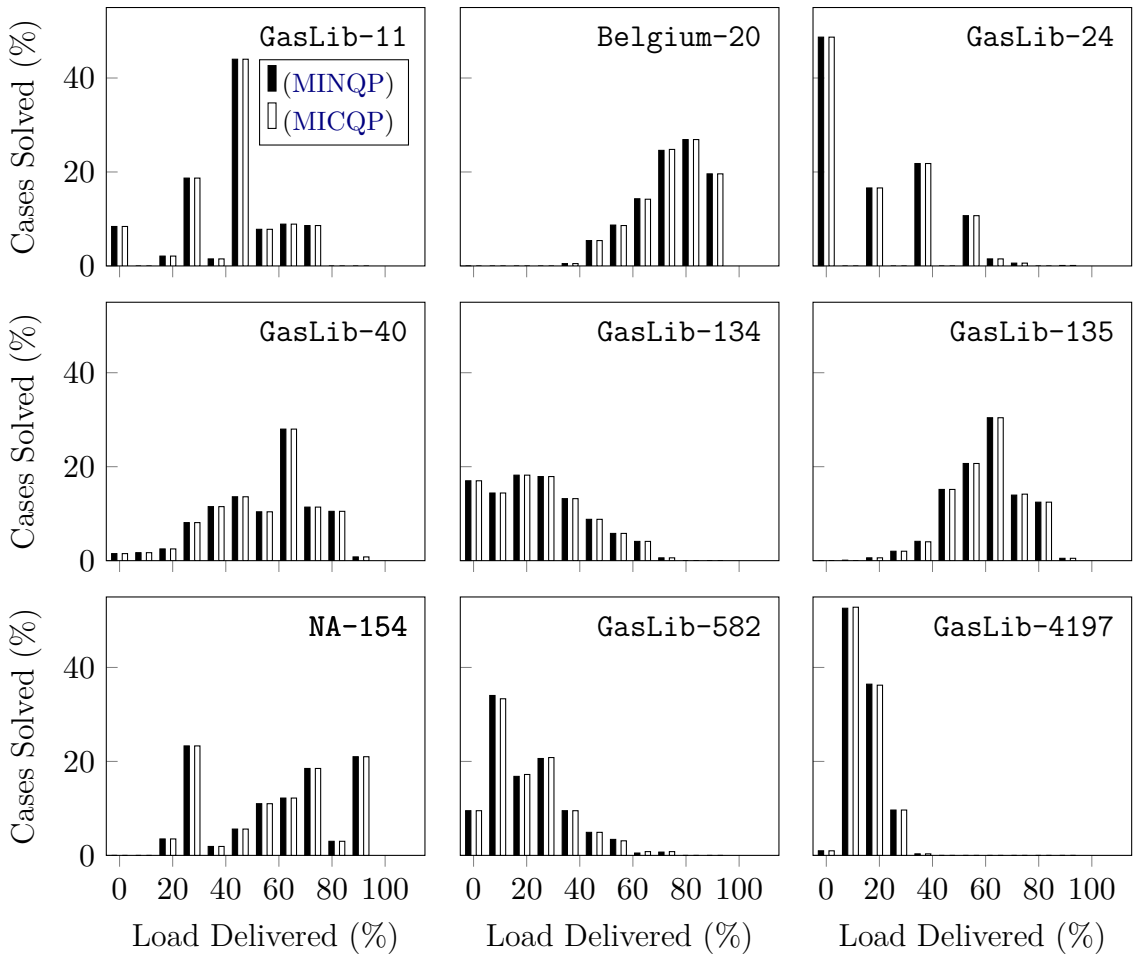


Figure 7.2: Histograms of gas load delivered over randomized $N-k$ scenarios. Here, the (MINQP) and (MICQP) formulations are compared, where k corresponds to 15% of all node-connecting components. Each pair of histograms summarizes instances solved by *both* (MINQP) and (MICQP).

using default Gurobi parameters but is also resolved when using `NumericFocus=3`.

Figure 7.2 displays histograms that compare the proportion of load delivered across *solved* damage scenarios for each network while using the two formulations. Most importantly, these histograms display the similarity of the results achieved while using the relaxation-based formulation. These results also indicate qualitative differences in the hypothetical robustness of each network. For example, larger networks like `GasLib-582` and `GasLib-4197` appear highly sensitive to the 15% damage scenarios, where often only 10% to 30% of load can be delivered. The `Belgium-20` network appears less vulnerable

and is often capable of serving between 70% and 100% of the load under severe contingencies. Finally, for some smaller networks (e.g., `GasLib-11`, `GasLib-24`, `GasLib-134`), many scenarios result in zero or nearly zero deliverable load. This analysis shows the utility of the MLD problem for understanding characteristics of gas network robustness.

7.3.4. Computational Performance

This section compares the performance of (MINQP) and (MICQP) MLD formulations using the instances described in Sections 7.3.2 and 7.3.3. The performance profiles for these instances are shown in Figure 7.3 and are divided into three categories: (S) networks containing tens of nodes; (M) networks containing hundreds of nodes; and (L) networks containing thousands of nodes (i.e., `GasLib-4197`). In all such categories, it is shown that the (MICQP) formulation is able to solve substantially greater numbers of problems than (MINQP) in shorter amounts of time. For networks with tens of nodes, both MLD formulations are capable of solving all instances in less than ten seconds. For networks with hundreds of nodes, (MICQP) is capable of solving most instances within ten seconds, while (MINQP) requires hundreds or thousands of seconds. For networks with thousands of nodes, (MICQP) solves a much greater number of instances within the alotted one hour time limit, although execution times required for solution are much longer than for MLD problems that consider networks of tens or hundreds of nodes.

7.3.5. Synthetic Earthquake Damage Scenarios

This subsection provides a proof-of-concept application to demonstrate the use of the MLD problem in the context of risk assessment for deterministic and uncertain spatial hazards. In each scenario, damage to a network was assumed to be caused by an earthquake with a fixed magnitude and epicenter. For an earthquake, the probability of damage to a pipeline component is commonly represented as a function of peak ground acceleration (PGA) and peak ground velocity (PGV). These relationships for PGA and PGV are typically expressed as functions of earthquake magnitude and distance from seismic rupture. In this chapter, the relationships developed by [24] were used, although their details are omitted here. At a high level, all earthquakes were assumed to arise from strike-slip faulting at a depth of one kilometer in a region of firm soil.

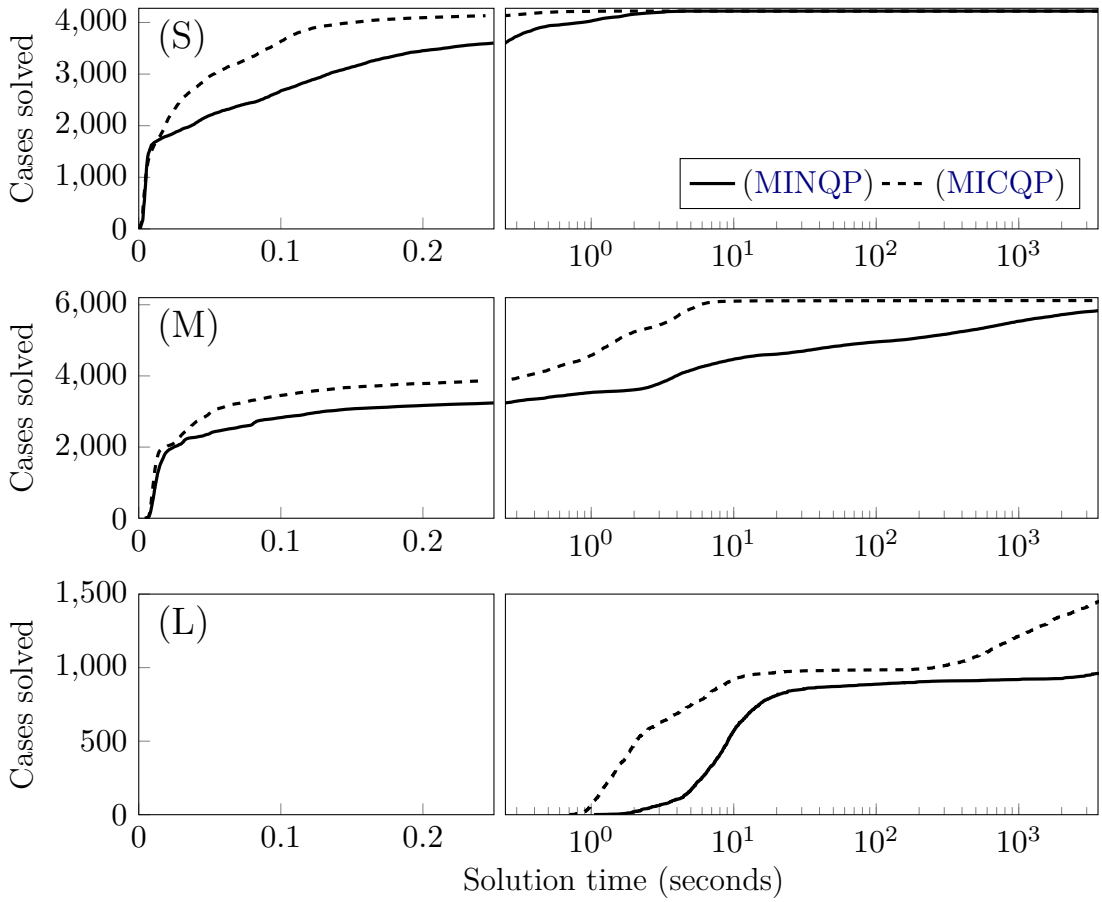


Figure 7.3: Performance profiles comparing the efficiency of (MINQP) and (MICQP). The instances are described in Sections 7.3.2 and 7.3.3. The performance profiles are divided into three categories for (S) networks containing tens of nodes; (M) networks containing hundreds of nodes; and (L) networks containing thousands of nodes (i.e., `GasLib-4197`).

Given the PGA and PGV at a component's point in space, the vulnerability of the component was then modeled probabilistically using the fragility approach of [84] for continuous pipelines in the presence of strong ground shaking. Specifically, the probability of damage was computed as a function of PGV, and node-connecting components exceeding the damage threshold for the first risk state ("very limited loss") were assumed to be nonfunctional. Again, for the purpose of brevity, these relationships are omitted in this chapter. For simplicity, we made three additional assumptions in our model: (i) only node-connecting components are affected by PGA and PGV; (ii) all

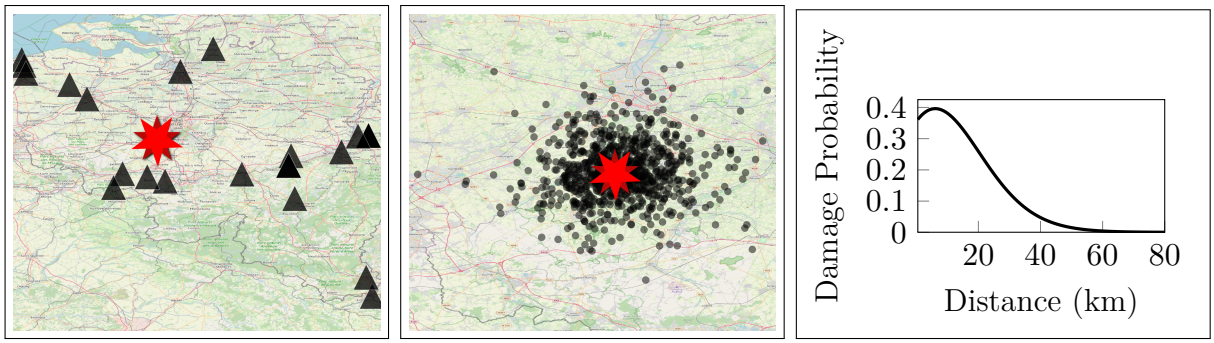


Figure 7.4: Illustrations of earthquake gas network damage scenario properties. The first (left) shows the position of an epicenter (red star) for the `Belgium-20` network and its relation to junctions (black triangles). The second (center) shows the placement of a mean earthquake epicenter (red star) for the `Belgium-20` network and normally distributed epicenters surrounding it (black circles). The last (right) shows a fragility curve derived from [24] and [84] for a magnitude 8.0 earthquake.

node-connecting components are to be assumed structurally equivalent to "continuous pipelines," and (iii) the distance from the epicenter to each node-connecting component is the minimum distance between the earthquake epicenter and the connecting junctions.

Finally, only the six gas networks `Belgium-20`, `GasLib-40`, `GasLib-135`, `NA-154`, `GasLib-582`, and `GasLib-4197` were considered for earthquake damage scenarios, as the remaining three networks do not contain geolocation data for components. Figure 7.4 depicts three illustrations relevant to parameterizing earthquake damage scenarios.

Deterministic Earthquake, Stochastic Fragility Scenarios The first set of earthquake scenarios is intended to demonstrate the applicability of the MLD method when analyzing network vulnerability to a known (i.e., deterministic) natural hazard. To this end, one earthquake was considered per network, each described by a fixed magnitude and epicenter. This type of scenario is illustrated pictorially in the first image of Figure 7.4. The local magnitude of each earthquake was assumed to be 8.0, while each epicenter was assumed to be the center of a k-means cluster containing the largest number of junctions, where $k = 5$. Then, using the PGV model of [24] and the probabilistic fragility approach of [84], one thousand damage scenarios were generated per network, where in each, the operational status per node-connecting component was determined via a uniform random sampling and comparison with the probability of damage. An example

	(MINQP)			(MICQP)		
Network	Opt. (%)	Lim. (%)	Inf. (%)	Opt. (%)	Lim. (%)	Inf. (%)
Belgium-20	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-40	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-135	98.3	1.7	0.0	100.0	0.0	0.0
NA-154	100.0	0.0	0.0	100.0	0.0	0.0
GasLib-582	58.7	41.3	0.0	99.8	0.2	0.0
GasLib-4197	2.9	95.3	1.8	86.0	14.0	0.0

Table 7.4: Comparison of solver statuses over deterministic earthquake scenarios.

fragility curve is depicted in the last image of Figure 7.4, which relates a component's distance from the epicenter of a magnitude 8.0 earthquake to the damage probability.

Table 7.4 displays statistics of solver statuses across all damage scenarios for each network and formulation. For all except the GasLib-135, GasLib-582, and GasLib-4197 networks, optimal solutions to instances are found within the one hour time limit. For GasLib-135, the (MINQP) formulation is unable to prove optimality on 17 instances. For GasLib-582 and GasLib-4197, a larger proportion of (MINQP) instances cannot be solved within the time limit. Even for the convex relaxation, two instances cannot be solved for the GasLib-582 network. Nonetheless, comparing the (MINQP) and (MICQP) formulations shows the benefit of the relaxation-based approach, which is capable of solving a larger proportion of instances. We note that 96 of the thousand GasLib-4197 (MINQP) instances are claimed to be infeasible using default Gurobi parameters. However, this number is reduced to 18 when using NumericFocus=3.

Figure 7.5 displays boxplots comparing the maximal proportion of load delivered across *solved* damage scenarios for each network and formulation. Here, results obtained using the (MINQP) and (MICQP) formulations of the MLD problem are shown to be remarkably similar. This demonstrates the utility of the relaxation-based approach, which provides outcomes comparable to the mixed-integer nonconvex formulation at a smaller computational cost. The boxplots also show substantial qualitative differences in the (hypothetical) vulnerability among networks. For example, the GasLib-40 network is shown to have great variability in maximal load delivered, while GasLib-135, NA-154, and GasLib-582 are mostly unaffected by the hypothetical hazard. Additionally, some natural gas networks (e.g., NA-154, GasLib-582) predict relatively small ranges of load

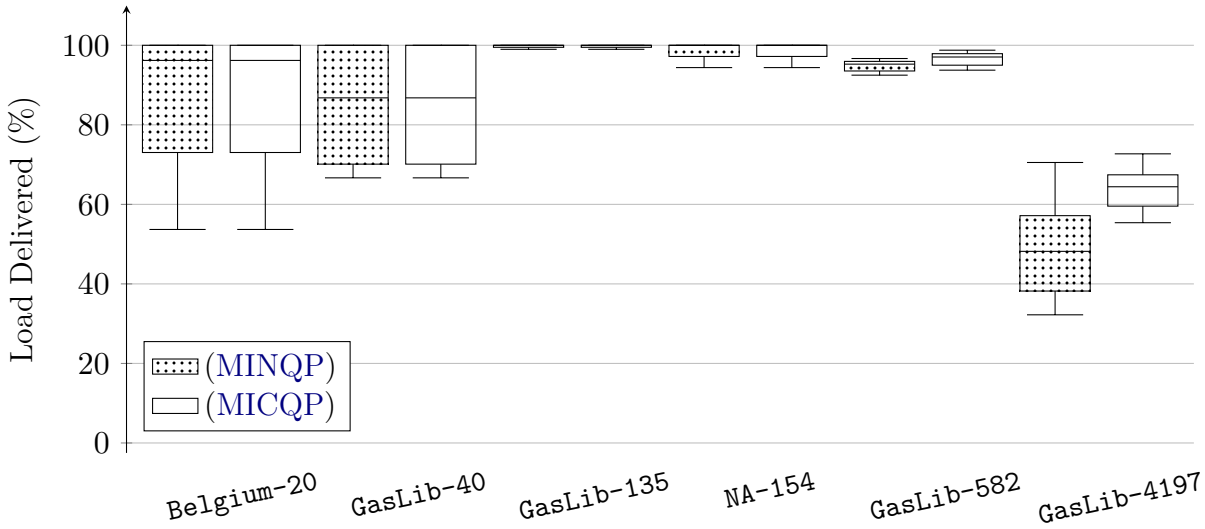


Figure 7.5: Boxplots of deliverable load over deterministic earthquake scenarios. That is, the magnitude and epicenter for each network scenario were assumed to be fixed. Note that each pair of boxplots summarizes only instances solved by *both* (MINQP) and (MICQP).

delivered, while others (e.g., `Belgium-20`) appear to carry relatively greater uncertainty.

The large discrepancies in boxplots for the `GasLib-4197` damage scenarios likely originate from two sources. First, the pair of boxplots is representative of only the subset of instances solved by *both* formulations, which is relatively small. Second, however, are the mathematical differences in nonconvex and relaxed MLD formulations. Notably, the minima across the solvable instances differ by around twenty percent. This could be a consequence of the relaxation, which theoretically predicts maximal load values greater than or equal to the nonconvex formulation. Compared to other networks, these larger discrepancies in predicted deliverable load could be a manifestation of the component relaxations, whose physical errors are further aggregated as the network size grows.

Stochastic Earthquake, Stochastic Fragility Scenarios The second set of earthquake scenarios is intended to demonstrate the applicability of the MLD method when analyzing network vulnerability to a stochastic (i.e., uncertain) natural hazard. To this end, multiple earthquakes were considered per network, where each was randomly sampled assuming normally distributed magnitudes and epicenters. This situation is

	(MINQP)			(MICQP)		
Network	Opt. (%)	Lim. (%)	Inf. (%)	Opt. (%)	Lim. (%)	Inf. (%)
`Belgium-20`	100.0	0.0	0.0	100.0	0.0	0.0
`GasLib-40`	100.0	0.0	0.0	100.0	0.0	0.0
`GasLib-135`	97.3	2.7	0.0	100.0	0.0	0.0
`NA-154`	100.0	0.0	0.0	100.0	0.0	0.0
`GasLib-582`	97.1	2.9	0.0	100.0	0.0	0.0
`GasLib-4197`	4.4	89.7	5.9	82.8	17.2	0.0

Table 7.5: Comparison of solver statuses over stochastic earthquake scenarios.

illustrated in the second image of Figure 7.4. Here, the mean local magnitude of each earthquake was assumed to be 8.0 with a standard deviation of 0.25, and each mean epicenter was again assumed to be the center of a k-means cluster containing the largest number of junctions, where $k = 5$, and where a distance-based standard deviation of ten kilometers was assumed. Using [24] and [84], one thousand random earthquake scenarios were generated per network, where in each, the status per node-connecting component was determined via a uniform sampling and comparison with the probability of damage.

Table 7.5 repeats the format of Table 7.4 to display aggregate statistics of solver termination statuses across all stochastic earthquake scenarios. For most scenarios, globally optimal solutions to MLD instances are found within the prescribed one hour time limit. For scenarios based on the `GasLib-135`, `GasLib-582`, and `GasLib-4197` networks, however, some instances cannot be solved. As in the deterministic scenario analysis, comparing the (MINQP) and (MICQP) formulations shows the benefit of the relaxation-based approach. Furthermore, 133 and one `GasLib-4197` (MINQP) and (MICQP) instances, respectively, are claimed to be infeasible using default Gurobi parameters. These numbers of instances are reduced to 59 and zero, respectively, when using `NumericFocus=3`.

Figure 7.6 displays boxplots comparing the proportion of load delivered across all scenarios. Again, the (MINQP) and (MICQP) formulations provide results that are qualitatively similar. The boxplots also show similar differences in vulnerability as those observed in Figure 7.6. However, the greater variation in epicenters and magnitudes results in greater variation of the effects. That is, the boxplots often have wider ranges.

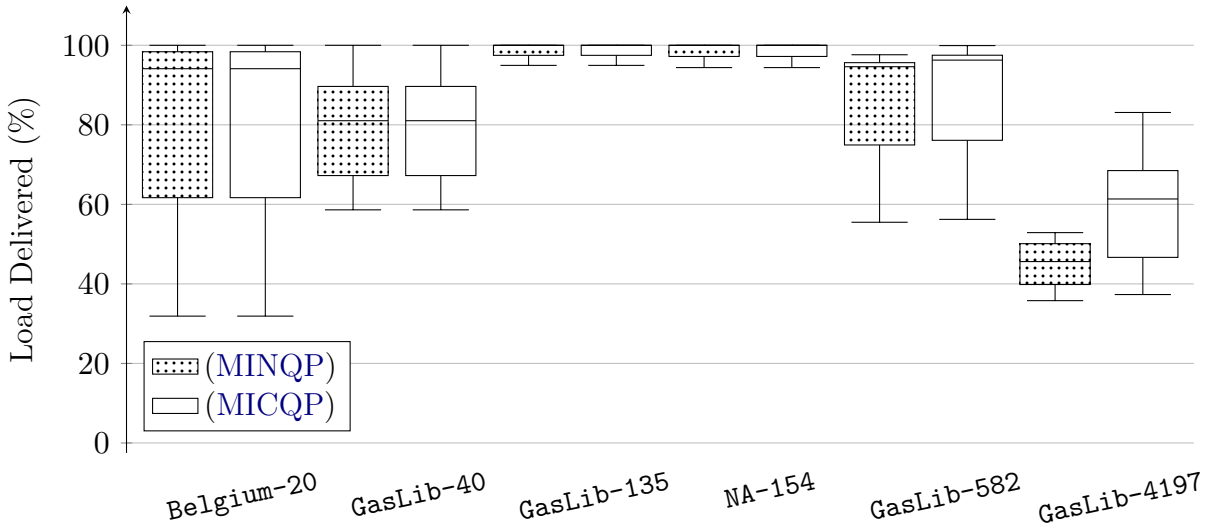


Figure 7.6: Boxplots of deliverable load over stochastic earthquake scenarios. That is, both the magnitude and epicenter for each scenario were assumed to be normally distributed. Note that each pair of boxplots summarizes only instances solved by *both* (MINQP) and (MICQP).

7.4. Conclusion

This chapter introduced the MLD for natural gas transmission networks, which seeks to determine a feasible operating point for a damaged gas network while ensuring the maximal delivery of load. This task was presented using three successive mathematical programming formulations. First, a MINCP was formulated that embeds all physical and engineering requirements for operational feasibility. Second, the first nonconvex program was reformulated *exactly* as a MINQP by introducing new variables. Finally, convex relaxations were applied to all nonconvex relationships in the former problem that involve pressures and mass flows, which results in a relaxed MICQP formulation.

To compare the efficacy of the second and third formulations, a rigorous benchmarking study was conducted over a large number of randomized multi-contingency scenarios on nine networks ranging in size from 11 to 4,197 junctions. First, MLD experiments based on $N-1$ (or single contingency) damage scenarios were conducted. Second, $N-k$ (multi-contingency) damage experiments were performed in order to understand MLD tractability for multimodal failures. A performance comparison of the (MINQP) and

(MICQP) formulations was then conducted using results from the $N-1$ and $N-k$ experiments. Finally, a proof-of-concept application based on network damage from a set of synthetically generated earthquakes demonstrated the application of the MLD problem to probabilistic risk assessment for deterministic and stochastic spatial hazards.

These results lead to three key conclusions. First, the relaxed formulation (MICQP) provides good bounds on the maximal deliverable load obtained from the full mixed-integer nonconvex formulation, (MINQP). Second, the relaxation-based formulation is more computationally robust than the mixed-integer nonconvex formulation and can solve larger proportions of challenging instances in much shorter amounts of time. These observations suggest (MICQP) could be useful in real-time decision support applications. Finally, for the largest network (i.e., `GasLib-4197`), the relaxation-based approach begins to show its limitations. For some challenging scenario types (e.g., $N-1$), large numbers of instances cannot be solved because of the relatively larger network size.

There are several potential studies that could extend the approaches developed in this chapter. First, additional relaxation-based methods could be developed to more accurately and efficiently solve the MLD problem for gas networks containing thousands of nodes. To facilitate this, another useful contribution could be the development of new benchmark instances whose sizes range between the sizes of `GasLib-582` and `GasLib-4197`. Finally, the origin of numerical instabilities associated with the challenging `GasLib-4197` network, and especially the sources of claimed infeasibility by Gurobi for some (MINQP) instances, should be thoroughly investigated. This could involve developing new methods for processing network data or normalizing constraints.

The most important extension of this work in the context of energy systems, however, is the consideration of interdependencies that link power and gas systems. Modeling these interdependencies is critical for determining the effects of multimodal gas and/or power network outages on overall deliverable gas and power loads. To address this, the interdependent gas and power MLD problem is thus thoroughly studied in Chapter 8.

Chapter 8

Convex Relaxations of Maximal Load Delivery for Multi-contingency Analysis of Joint Electric Power and Natural Gas Transmission Networks

As the use of renewable generation has increased, electric power systems have become progressively reliant on natural gas power plants as fast ramping sources for meeting power demands. This dependence has introduced new vulnerabilities to the power grid, including disruptions to gas transmission networks from natural and man-made disasters. To address the operational challenges arising from these disruptions, Chapter 7 considered the task of determining a feasible steady-state operating point for a damaged gas pipeline network *independently* while ensuring the maximal delivery of gas load. However, Chapter 7 *did not* consider the interdependencies that exist between gas and power transmission networks, which could ultimately affect both the operational decisions made by gas network operators as well as the available generation capacity of the interdependent power network. To address this, this chapter proposes the *joint* gas and power MLD problem, which allows for a meaningful exploration of the tradeoffs when maximizing gas versus power delivery in the presence of severe network disruptions.

The contingency response measures considered in this chapter are illustrated in Figure 8.1 and bear similarity to Figure 7.1. Given a severe disruption, (i) gas and/or power load deliveries decrease as gas and/or power network elements are impaired and effects

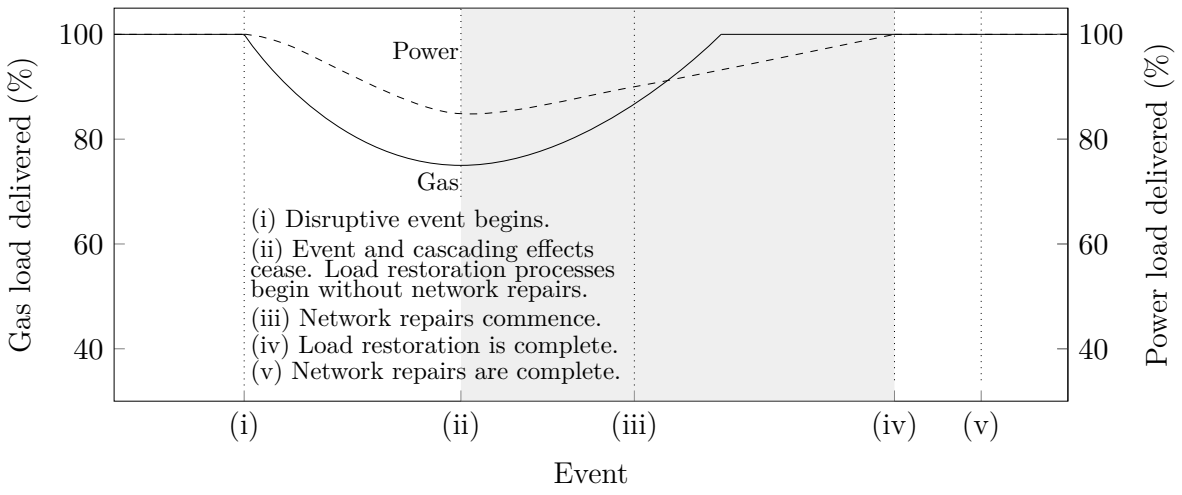


Figure 8.1: Illustration of gas and power network responses to a severe disruption. The shaded region indicates the points in the disruption and restoration timeline that are studied in this chapter via an optimization-based assessment of damaged joint gas-power network capacities.

begin to cascade, and (ii) cascading effects subside, and a new stable operating point is realized. After (ii), load can gradually be restored via operational methods until (iii) network repairs begin. These restorative actions are performed until (iv) all gas and power loads can be delivered. Repairs continue until (v) all gas and power network components are operational. As in Chapter 7, addressing all event types within Figure 8.1 is a substantial task. To make the scope manageable, we focus more narrowly on finding optimal steady-state operating points between events of types (ii) and (iv), i.e., decisions that maximize gas and power load delivery in the surviving gas-power system.

In this chapter, this task is formalized as the steady-state *joint* MLD problem. Similar to the independent gas MLD problem of Chapter 7, this problem is informally stated as follows: given severely damaged gas and power networks in which multiple components have become nonoperational, maximize the amounts of prioritized gas and active power loads that can be served simultaneously in the damaged joint network, subject to steady-state natural gas and AC power network physics. The nonconvex physics and discrete nature of operations in the joint network (e.g., the opening and closing of valves in the gas network) render this a challenging MINCP. To increase its tractability, we develop an MICP relaxation of the MLD problem. As in Chapter 7, the MICP relaxation is

found to be an effective means for bounding maximum deliverable gas and power loads.

This chapter expands upon previous MLD methods for independent gas (Chapter 7) and power networks [34], as well as approaches from joint gas-power network modeling, to formulate and solve the joint gas-power MLD problem. Its contributions include

- The first formulation of the joint gas-power MLD problem;
- A reliable MICP relaxation of the joint gas-power MLD problem;
- Validation of the MLD MICP relaxation's accuracy and computational efficiency;
- Proof-of-concept analytical applications of MLD gas-power tradeoffs.

Chapter Overview The remainder of this chapter proceeds as follows: Section 8.1 reviews gas, power, and joint steady-state optimization models that appear in the literature, then formulates the requirements for AC power and gas pipeline operational feasibility as a MINCP; Section 8.2 formulates the MLD problem as a MINCP, then proposes an MICP relaxation; Section 8.3 benchmarks the formulations across multiple joint gas-power networks of various sizes, then provides proofs of concept for joint multi-contingency analysis using the MLD method; and Section 8.4 concludes the chapter.

8.1. Network Modeling

The past decade has seen remarkable theoretical and algorithmic advances in the independent fields of power and natural gas network optimization. A recent survey of relaxations and approximations used in power system optimization is presented by [94]. The study in power most related to this chapter is by [34], who introduce the AC MLD problem and propose various relaxations to increase its tractability. The method was later extended by [129] to identify the k components that maximize network disruption, as well as by [130] to identify multi-contingency scenarios that would benefit from more detailed cascading analyses. The MLD problem was also exploited by [114], who applied it within a bilevel optimization for balancing wildfire risk and power outages. Finally, an implementation of the power MLD problem is presented by [113], who provide formulations via the PowerModelsRestoration package. Their implementation serves as a computational foundation for the power system modeling portion of this chapter.

As with power, the growing utilization of gas networks has led to a variety of optimization studies, and the relevant gas literature was reviewed in Chapter 7. An even

more recent body of literature has examined the optimal coordination of gas and power infrastructures. A review of joint gas and power planning is given by [54]. Other studies have focused on market coordination and energy pricing problems [23], [93]. Many studies have assumed the networks to be fully coordinated, examining optimal scheduling of generator dispatching and gas compressor operations [151]. Recent studies have expanded upon these earlier joint "optimal gas-power flow" problems, developing specialized formulations and algorithms for related applications [75], [92], [143]. A smaller number of studies have considered joint problems related to restoration, e.g., scheduling of general large-scale interdependent infrastructures in [1] and last-mile restoration of joint gas and power systems in [35]. The remaining subsections build upon previous gas and power studies to develop steady-state requirements for a damaged joint network.

8.1.1. Power Transmission Network Modeling

Notation for Sets A power network is represented by an arbitrarily directed graph $(\mathcal{N}, \mathcal{E} \cup \mathcal{E}^R)$, where $\mathcal{N}$ is the set of buses, $\mathcal{E}$ is the set of forward-directed branches (or lines), and $\mathcal{E}^R$ is the set of branches in their reverse orientation. The set of generators (producers), loads (consumers), and shunts are denoted by $\mathcal{G}$, $\mathcal{L}$, and $\mathcal{H}$, respectively, which are attached to buses $i \in \mathcal{N}$. We let the subset of these components attached to $i \in \mathcal{N}$ be denoted by $\mathcal{G}_i$, $\mathcal{L}_i$, and $\mathcal{H}_i$. We next define the decision variables and constraints required to model a damaged AC power network's steady-state operations.

Power Network Modeling Requirements The MINCP formulation for AC power network feasibility, as defined for AC MLD analysis, is presented in Model 8.1 and detailed by [34]. Here, Constraints (8.1a) and (8.1b) model Ohm's law for lines, where $S_{ij} \in \mathbb{C}$ denotes the variable power along each line; $Y_{ij} \in \mathbb{C}$ and $Y_{ij}^c \in \mathbb{C}$ are constants denoting the line admittance and line charging; $V_i \in \mathbb{C}$ denotes the variable voltage at bus $i \in \mathcal{N}$; and $T_{ij} \in \mathbb{C}$ denotes constant transformer properties. Constraints (8.1c) model power balances from Kirchhoff's current law for each bus, where $S_k^g \in \mathbb{C}$ denotes the variable power supplied by generator $k \in \mathcal{G}$; $S_k^d \in \mathbb{C}$ denotes the maximum power that can be delivered at load $k \in \mathcal{L}$; and Y_k^s denotes the admittance of bus shunt $k \in \mathcal{H}$. Note that z_k^d, $k \in \mathcal{L}$, allows each load to vary between zero and its predefined maximum, and z_k^s, $k \in \mathcal{H}$, allows for shedding fixed bus shunts from the network. These

Model 8.1 Power network modeling requirements

$$S_{ij} = \left(Y_{ij} + Y_{ij}^c\right)^* \frac{|V_i|^2}{|T_{ij}|^2} - Y_{ij}^* \frac{V_i V_j^*}{T_{ij}}, \ \forall (i,j) \in \mathcal{E} \tag{8.1a}$$

$$S_{ji} = \left(Y_{ij} + Y_{ji}^c\right)^* |V_j|^2 - Y_{ij}^* \frac{V_i^* V_j}{T_{ij}^*}, \ \forall (i,j) \in \mathcal{E} \tag{8.1b}$$

$$\sum_{k \in \mathcal{G}_i} S_k^g - \sum_{k \in \mathcal{L}_i} z_k^d S_k^d - \sum_{k \in \mathcal{H}_i} z_k^s Y_k^s |V_i|^2 = \sum_{(i,j) \in \mathcal{E}_i \cup \mathcal{E}_i^R} S_{ij}, \ \forall i \in \mathcal{N} \tag{8.1c}$$

$$|S_{ij}| \leq \overline{S}_{ij}, \ S_{ij} \in \mathbb{C}, \ \forall (i,j) \in \mathcal{E} \cup \mathcal{E}^R \tag{8.1d}$$

$$\underline{\theta}_{ij}^{\Delta} \leq \angle \left(V_i V_j^*\right) \leq \overline{\theta}_{ij}^{\Delta}, \ \forall (i,j) \in \mathcal{E} \tag{8.1e}$$

$$z_i^v \underline{V}_i \leq |V_i| \leq z_i^v \overline{V}_i, \ V_i \in \mathbb{C}, \ \forall i \in \mathcal{N} \tag{8.1f}$$

$$z_i^g \underline{S}_i^g \leq S_i^g \leq z_i^g \overline{S}_i^g, \ S_i^g \in \mathbb{C}, \ \forall i \in \mathcal{G} \tag{8.1g}$$

$$z_i^v \in \{0,1\}, \ \forall i \in \mathcal{N}, \ z_i^g \in \{0,1\}, \ \forall i \in \mathcal{G} \tag{8.1h}$$

$$z_i^d \in [0,1], \ \forall i \in \mathcal{L}, \ z_i^s \in [0,1], \ \forall i \in \mathcal{H} \tag{8.1i}$$

modifications ensure power balance constraints are satisfied in damaged power networks.

Constraints (8.1d)–(8.1i) impose engineering limits and variable bounds. Constraints (8.1d) bound the apparent power flow on each line, representing thermal limits. Constraints (8.1e) ensure that each voltage phase angle difference is limited by predefined lower and upper bounds, $\underline{\theta}_{ij}^{\Delta}$ and $\overline{\theta}_{ij}^{\Delta}$, respectively. Constraints (8.1f) bound the voltage magnitude at each bus, where $\underline{V}_i$ and $\overline{V}_i$ denote lower and upper bounds, respectively. Here, $z_i^v \in \{0,1\}$ is a discrete variable that allows each bus to become de-energized when isolated from load or generation. Similarly, Constraints (8.1g) bound power generation, where $\underline{S}_i^g$ and $\overline{S}_i^g$ denote lower and upper bounds, respectively, and $z_i^g \in \{0,1\}$ allows each generator to become uncommitted when required by Constraints (8.1a) and (8.1b).

8.1.2. Natural Gas Transmission Network Modeling

Notation for Sets Repeating the notation of Chapter 7 for completeness, a gas network is modeled using a directed graph $(\mathcal{J}, \mathcal{A})$, where $\mathcal{J}$ is the set of nodes (i.e., junctions) and $\mathcal{A}$ is the set of components that connect two nodes. The sets of receipts (producers) and deliveries (consumers) are denoted by $\mathcal{R}$ and $\mathcal{D}$, respectively. These

components are considered to be attached to junctions $i \in \mathcal{J}$. The subset of receipts attached to $i \in \mathcal{J}$ is denoted by $\mathcal{R}_i$ and the subset of deliveries by $\mathcal{D}_i$. The sets of horizontal and short pipes are denoted by $\mathcal{P} \subset \mathcal{A}$ and $\mathcal{S} \subset \mathcal{A}$, respectively; the set of resistors by $\mathcal{T} \subset \mathcal{A}$; the set of valves and pressure-reducing regulators by $\mathcal{V} \subset \mathcal{A}$ and $\mathcal{W} \subset \mathcal{A}$, respectively; and the set of compressors by $\mathcal{C} \subset \mathcal{A}$. Loss resistors are not considered in this chapter. Additionally, the set of node-connecting components incident to $i \in \mathcal{J}$ where i is the tail (respectively, head) of the arc is denoted by $\delta_i^+ := \{(i,j) \in \mathcal{A}\}$ (respectively, $\delta_i^- := \{(j,i) \in \mathcal{A}\}$). We next define the decision variables and constraints required to model a damaged natural gas transmission network's steady-state operations.

Gas Network Modeling Requirements The MINCP formulation for gas network feasibility, as defined for MLD analysis, is presented in Model 8.2, which summarizes details previously presented in Chapter 7. First, Constraints (8.2a) model nodal physics, i.e., mass flow conservation at junctions $i \in \mathcal{J}$. Here, $f_{ij} \in \mathbb{R}$ denotes the variable mass flow along each node-connecting component; $s_k \in \mathbb{R}_+$ denotes the variable supply at receipt $k \in \mathcal{R}$; and $d_k \in \mathbb{R}_+$ denotes the variable demand (or load) at delivery $k \in \mathcal{D}$.

Constraints (8.2b)–(8.2u) model the physics of node-connecting components. Constraints (8.2b) model the pressure-flow relationship for steady-state flow in a gas pipeline for each pipe $(i,j) \in \mathcal{P}$. Here, $p_i \in \mathbb{R}_+$ denotes the variable pressure at junction $i \in \mathcal{J}$, and $w_{ij} \in \mathbb{R}_+$ denotes the constant mass flow resistance of the pipe. These constraints are the most frequent sources of nonconvex nonlinearity in modeling the gas system.

Constraints (8.2c) model short pipes in the network, which provide resistance-less mass transport between two junctions. Constraints (8.2d) model resistors in the network, which act as surrogate components capable of modeling pressure losses elsewhere from pipes. Here, pressure loss is modeled according to the Darcy-Weisbach equation, where $\tau_{ij} \in \mathbb{R}_+$ is the resistance, which is a function of the resistor's unitless drag factor and diameter. Note that like Constraints (8.2b), these constraints are also nonconvex.

Constraints (8.2e)–(8.2g) model valves in the network. Here, the operating status of each valve $(i,j) \in \mathcal{V}$ is modeled using a discrete variable $z_{ij} \in \{0,1\}$, where $z_{ij} = 1$ indicates an open valve and $z_{ij} = 0$ indicates a closed valve. Constraints (8.2e) prohibit flow across each valve when $z_{ij} = 0$. Constraints (8.2f) and (8.2g) model, when a valve is open, that the pressures at connecting junctions are equal. They also model the decoupling of pressures at junctions connected by the valve when the valve is closed.

Model 8.2 Gas network modeling requirements

$$\sum_{(i,j)\in\delta_i^+} f_{ij} - \sum_{(j,i)\in\delta_i^-} f_{ji} = \sum_{k\in\mathcal{R}_i} s_k - \sum_{k\in\mathcal{D}_i} d_k,\ \forall i \in \mathcal{J} \tag{8.2a}$$

$$p_i^2 - p_j^2 = w_{ij} f_{ij}|f_{ij}|,\ \forall (i,j) \in \mathcal{P} \tag{8.2b}$$

$$p_i - p_j = 0,\ \forall (i,j) \in \mathcal{S} \tag{8.2c}$$

$$p_i - p_j = \tau_{ij} f_{ij}|f_{ij}|,\ \forall (i,j) \in \mathcal{T} \tag{8.2d}$$

$$\underline{f}_{ij} z_{ij} \le f_{ij} \le \overline{f}_{ij} z_{ij},\ z_{ij} \in \{0,1\},\ \forall (i,j) \in \mathcal{V} \tag{8.2e}$$

$$p_i \le p_j + (1 - z_{ij})\overline{p}_i,\ \forall (i,j) \in \mathcal{V} \tag{8.2f}$$

$$p_j \le p_i + (1 - z_{ij})\overline{p}_j,\ \forall (i,j) \in \mathcal{V} \tag{8.2g}$$

$$\underline{f}_{ij} z_{ij} \le f_{ij} \le \overline{f}_{ij} z_{ij},\ z_{ij} \in \{0,1\},\ \forall (i,j) \in \mathcal{W} \tag{8.2h}$$

$$f_{ij}(p_i - p_j) \ge 0,\ \forall (i,j) \in \mathcal{W} \tag{8.2i}$$

$$\underline{\alpha}_{ij} p_i \le p_j + (1 - z_{ij})\underline{\alpha}_{ij}\overline{p}_i,\ \forall (i,j) \in \mathcal{W} \tag{8.2j}$$

$$p_j \le \overline{\alpha}_{ij} p_i + (1 - z_{ij})\overline{p}_j,\ \forall (i,j) \in \mathcal{W} \tag{8.2k}$$

$$\underline{\alpha}_{ij} p_i \le p_j \le \overline{\alpha}_{ij} p_i,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} \ge 0 \tag{8.2l}$$

$$\underline{\alpha}_{ij} p_i \le p_j \le \overline{\alpha}_{ij} p_i,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} = 1 \tag{8.2m}$$

$$f_{ij}(p_i - p_j) \le 0,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} = 1 \tag{8.2n}$$

$$y_{ij} \in \{0,1\},\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \ne 1 \tag{8.2o}$$

$$p_j \le \overline{\alpha}_{ij} p_i + (1 - y_{ij})\overline{p}_j,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \ne 1 \tag{8.2p}$$

$$\underline{\alpha}_{ij} p_i \le p_j + (1 - y_{ij})\overline{p}_i,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \ne 1 \tag{8.2q}$$

$$p_i - p_j \le y_{ij}\overline{p}_i,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \ne 1 \tag{8.2r}$$

$$p_j - p_i \le y_{ij}\overline{p}_j,\ \forall (i,j) \in \mathcal{C} : \underline{f}_{ij} < 0 \wedge \underline{\alpha}_{ij} \ne 1 \tag{8.2s}$$

$$\underline{f}_{ij} \le f_{ij} \le \overline{f}_{ij},\ \forall (i,j) \in \mathcal{A} \tag{8.2t}$$

$$0 \le \underline{p}_i \le p_i \le \overline{p}_i,\ \forall i \in \mathcal{N} \tag{8.2u}$$

$$0 \le s_k \le \overline{s}_k,\ \forall k \in \mathcal{R},\ 0 \le d_k \le \overline{d}_k,\ \forall k \in \mathcal{D} \tag{8.2v}$$

Constraints (8.2h)–(8.2k) model regulators (i.e., pressure-reducing valves) in the network. Similar to valves, the status of each regulator is modeled using a discrete variable

$z_{ij} \in \{0,1\}$, where $z_{ij} = 1$ and $z_{ij} = 0$ indicate active and inactive statuses, respectively. Constraints (8.2h) prohibit mass flow across each regulator when $z_{ij} = 0$. Constraints (8.2i) ensure that mass flow across each regulator is in the same direction as the loss in pressure. Constraints (8.2j) and (8.2k) model the remaining pressure dynamics. Here, each regulator has a corresponding scaling factor, α_{ij}. This models the relationship between junction pressures when the regulator is active, i.e., $\alpha_{ij} p_i = p_j$. The factor is limited by the bounds $\underline{\alpha}_{ij} = 0 \leq \alpha_{ij} \leq \overline{\alpha}_{ij} = 1$. Constraints (8.2j) and (8.2k) require that, when a regulator is active, pressures are defined according to the scaling relationship. Otherwise, pressures at the junctions connected by the regulator are decoupled.

Constraints (8.2l)–(8.2s) model compressors in the network. Each compressor $(i,j) \in \mathcal{C}$ models an increase in pressure at junction $j \in \mathcal{J}$ by a variable scalar α_{ij}. Without loss of generality, bidirectional compression is not considered, although each compressor may allow for *uncompressed* flow in the opposite direction. These different behaviors of compressors are modeled by employing three different sets of constraints. The first are Constraints (8.2l) for compressors that *prohibit* reverse flow, where $\underline{\alpha}_{ij}$ and $\overline{\alpha}_{ij}$ are minimum and maximum pressure ratios. The second are Constraints (8.2m) and (8.2n) for compressors where reverse flow *is allowed* and $\underline{\alpha}_{ij} = 1$. Note that here, if $f_{ij} < 0$, then $p_i = p_j$. Finally, Constraints (8.2o)–(8.2s) model compressors where uncompressed reverse flow is allowed and $\underline{\alpha}_{ij} \neq 1$. In this case, the behavior of each compressor is disjunctive in its flow direction. To model this disjunction, the discrete variables $y_{ij} \in \{0,1\}$ are introduced in Constraints (8.2o) to model the direction of flow through each compressor. Here, $y_{ij} = 1$ indicates mass flow from i to j, and $y_{ij} = 0$ indicates flow from j to i. Constraints (8.2p)–(8.2s) model the pressures and pressure differences between junctions as per the specified flow direction and compression ratio bounds.

The remaining Constraints (8.2t)–(8.2v) are variable bounds. Constraints (8.2t) are mass flow bounds, Constraints (8.2u) are pressure bounds, and Constraints (8.2v) are receipt and delivery bounds. Most importantly, note that Constraints (8.2v) differ from the typical assumption of fixed supply and demand. These modifications ensure mass flow conservation constraints will remain satisfied even for damaged gas networks.

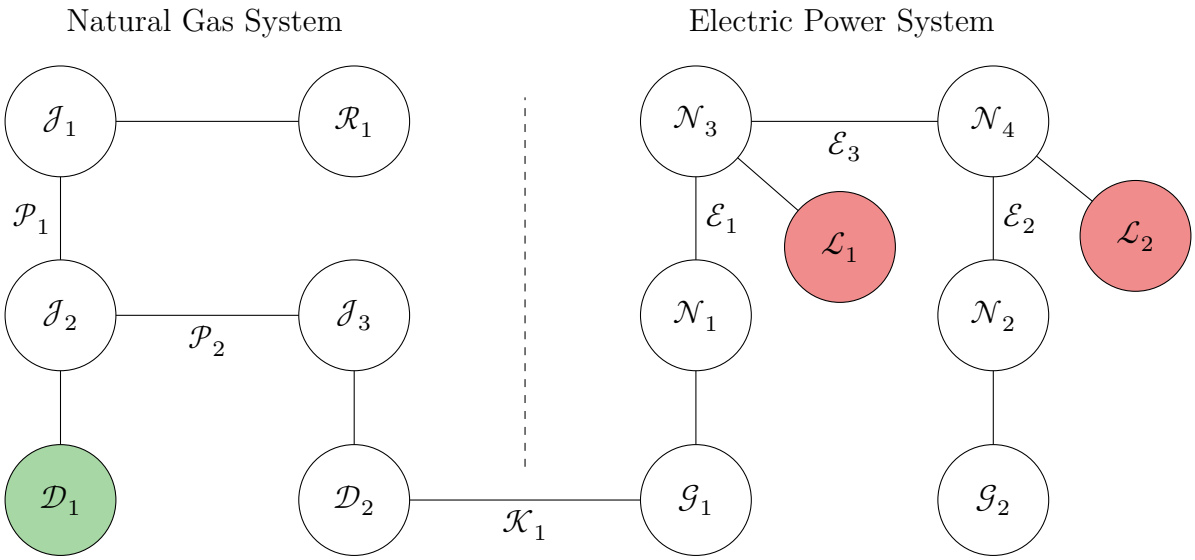


Figure 8.2: Diagrammatic representation of a small joint gas-power network. Here, the demand of gas at delivery $\mathcal{D}_1$ contributes to the objective term $\eta_G(\cdot)$ and power loads $\mathcal{L}_1$, $\mathcal{L}_2$ contribute to $\eta_P(\cdot)$. Finally, the linkage between gas and power systems occurs at $\mathcal{K}_1 = (\mathcal{D}_2, \mathcal{G}_1)$.

8.1.3. Interdependency Modeling

As in [7] and [23], gas and power systems are connected via heat rate curve models for gas-fired power generators. These relate the amount of gas consumed to power, i.e.,

$$\sum_{i:(i,j)\in\mathcal{K}} h_i^1 \Re(S_i^g)^2 + h_i^2 \Re(S_i^g) + h_i^3 z_i^g = d_j, \ \forall j \in \mathcal{D}_G. \tag{8.3}$$

Each constraint links the active power generated at possibly multiple generators with a single gas delivery. Here, $h_i^{1,2,3}$ are coefficients of the heat rate curve for $i \in \mathcal{G}$, and $\mathcal{K}$ is the set of linkages between gas-fired generators in $\mathcal{G}$ and their corresponding gas delivery points in $\mathcal{D}_G \subset \mathcal{D}$. Furthermore, $h_i^1 \geq 0$ for all $(i,j) \in \mathcal{K}$, and thus the left-hand side is always a convex function. However, note that Constraint (8.3) is nonlinear nonconvex when $h_i^1 \neq 0$. Finally, the presence of the term $h_i^3 z_i^g$ ensures that when $z_i = 0$, the intercept of the heat rate curve, and thus both active power generation and gas required, will be zero when a generator is uncommitted from the dispatch scenario.

A diagramatic illustration of the joint network model is illustrated in Figure 8.2. Here, gas and power systems are linked by the interdependency $\mathcal{K}_1$, which relates the delivery

$\mathcal{D}_2$ to the generator $\mathcal{G}_2$. Contributions to gas and power delivery objectives, which are later described in Section 8.2.1, are depicted by green and red nodes, respectively.

8.1.4. Challenges

Although independent gas and power MLD models were explored in Chapter 7 and [34], respectively, the joint MLD problem that includes Constraints (8.1)-(8.3) is more challenging for several reasons. Most importantly, the nonlinear nonconvexities that appear in Models 8.1 and 8.2 arise primarily from different sources: Model 8.1 includes many nonlinear equations with bilinear variable products, whereas Model 8.2 includes more manageable quadratic nonlinear equations. To model them exactly, Model 8.1 must be formulated as a challenging MINCP, but Model 8.2 can be written as a more tractable MINQP. These formulation differences suggest potentially incompatible numerical methods and solution technologies when solving the joint MLD problem.

Although omitted here for brevity, this work also employed a number of important preprocessing steps used to ensure the construction of feasible damaged joint networks that satisfy Constraints (8.1)-(8.3). Finally, Models 8.1 and 8.2 constrain each system using steady-state physical assumptions. In practice, modeling the transient dynamics of the gas system could be crucial. However, as will be shown in subsequent sections, even the steady-state variant considered in this chapter is computationally difficult. This work is an important step toward building MLD techniques that also consider transients.

8.2. Maximal Load Delivery Formulations

This section derives the joint gas-power MLD formulations used throughout the remainder of this chapter. Section 8.2.1 defines the competing objectives of the joint MLD problem. Section 8.2.2 poses lexicographic and weighted MLD formulations that prioritize the gas-power delivery tradeoff in different ways. Section 8.2.3 derives MICP relaxations of the MLD formulations. Finally, Section 8.2.4 summarizes naming conventions used for these various MLD formulations, which are compared in Section 8.3.

8.2.1. Objectives of the Maximal Load Delivery Problem

The objective of the MLD problem is to maximize the amount of *nongeneration* gas and *active* power load delivered simultaneously under a multi-contingency scenario. Note that the maximization of *nongeneration* gas, specifically, allows the model to decouple the practical objectives of the gas system (e.g., delivery of fuel for residential heating) from practical objectives of the power system. However, because the delivery of non-generation gas load can inhibit the amount of active power generation, and thus active power delivered, there exists an important tradeoff between these two competing objectives. For notational ease, we first write the gas and power objective functions as

$$\eta_G(d) := \left(\sum_{i\in\mathcal{D}'} \beta_i d_i\right)\left(\sum_{i\in\mathcal{D}'} \beta_i \bar{d}_i\right)^{-1} \tag{8.4a}$$

$$\eta_P(z^d) := \left(\sum_{i\in\mathcal{L}} \beta_i z_i^d |\Re(S_i^d)|\right)\left(\sum_{i\in\mathcal{L}} \beta_i |\Re(S_i^d)|\right)^{-1}. \tag{8.4b}$$

Here, Equation (8.4a) denotes the normalized sum of all prioritized nongeneration gas demand, where $\mathcal{D}' := \mathcal{D} \setminus \{j : (i,j) \in \mathcal{K}\}$ (i.e., the set of all nongeneration gas deliveries), and $\beta_i \in \mathbb{R}_+$ is a predefined restoration priority for delivery $i \in \mathcal{D}'$. Similarly, Equation (8.4b) denotes the normalized sum of all prioritized active power loads. Note that for all of the experiments considered in this study, $\beta_i = 1$ for all $i \in \mathcal{D}' \cup \mathcal{L}$.

The tradeoff between nongeneration gas and active power load lends the MLD problem to the broader category of multi-objective optimization. A survey of multi-objective optimization methods in engineering is presented by [89] and describes a number of techniques for specifying preferences among multiple objectives. These include weighted sum, weighted product, lexicographic, and bounded objective optimization methods. In Section 8.2.2, we define lexicographic and weighted sum variants of the MLD problem.

8.2.2. Lexicographic and Weighted MLD Formulations

To explore the gas-power tradeoff, we introduce three MLD models that prioritize gas and power delivery in different ways. The first is a lexicographic formulation that prioritizes delivery of nongeneration gas load. This situation is representative of common

contractual requirements for gas grid operators. Here, the MLD problem is written as

$$\begin{aligned} \text{maximize} \quad & \eta_P(z^d) \\ \text{subject to} \quad & \eta_G(d) \geq \eta_G(d^*) \\ & \text{Constraints (8.1)}-\text{(8.3)}, \end{aligned} \tag{MLD-G}$$

where $\eta_G(d^*)$ is the optimal objective when maximizing gas delivery alone. The second MLD is a formulation that prioritizes the amount of active power load delivered, i.e.,

$$\begin{aligned} \text{maximize} \quad & \eta_G(d) \\ \text{subject to} \quad & \eta_P(z^d) \geq \eta_P(z^{d*}) \\ & \text{Constraints (8.1)}-\text{(8.3)}. \end{aligned} \tag{MLD-P}$$

The last is a single-level formulation that weights normalized sums of nongeneration gas and active power delivery. It considers the direct tradeoffs between objectives, i.e.,

$$\begin{aligned} \text{maximize} \quad & \lambda\eta_G(d) + (1-\lambda)\eta_P(z^d) \\ \text{subject to} \quad & \text{Constraints (8.1)}-\text{(8.3)}, \end{aligned} \tag{MLD-W}$$

where $0 < \lambda < 1$ is a weighting parameter that shifts prioritization of the objective.

Note that (MLD-G), (MLD-P), and (MLD-W) are MINCPs. The nonconvexities arise from three sources: (i) discrete operations of controllable components (e.g., z_i^g for generator commitment); (ii) bilinear products that appear in both gas and power network physics (e.g., $V_i V_j^*$ in Ohm's law); and (iii) nonlinear equations used for satisfying important physical relationships (e.g., pressure-flow relationships for pipes). In the following, we leverage a number of relaxations to render these problems more tractable.

8.2.3. Relaxation of Products and Nonlinear Equations

Convexification of Power Physics The primary sources of nonconvexity in Model 8.1 are the bilinear products that appear in Constraints (8.1a)–(8.1c) (e.g., $V_i V_j^*$). A large body of literature has developed relaxations of similar terms, and for a comprehensive review, we refer the reader to the recent survey of [94]. In this chapter, we utilize a model based on an SOCP relaxation of the AC power flow equations, described by [74]

and used for AC power MLD analysis in [34]. The primary insight of the SOCP formulation is that variable products ($|V_i|^2$ and $V_i V_j^*$) can be lifted into a higher-dimensional variable space (W_{ii} and W_{ij}, respectively). This renders any terms involving these products linear, and the relaxation is ultimately strengthened via the constraints

$$|W_{ij}|^2 \leq W_{ii} W_{jj}, \ \forall (i,j) \in \mathcal{E}. \tag{8.5}$$

Constraint (8.5) is a rotated SOCP constraint, lending the formulation its name.

Convexification of Gas Physics Many nonconvexities in Model 8.2 appear in the form of nonlinear equations (e.g., Constraints (8.2b)) and bilinear variable products (e.g., Constraints (8.2i)). Here, we briefly review the convexification techniques previously described in Chapter 7. To resolve both sources of nonlinear nonconvexity, direction variables $y_{ij} \in \{0,1\}$ are first introduced for each node-connecting component $(i,j) \in \mathcal{A}$. We also introduce variables $\pi_i \in \mathbb{R}_+$ to denote squared pressures p_i^2 for $i \in \mathcal{J}$. This first change allows for a partial linearization of the pressure-flow equations for pipelines, i.e.,

$$p_i^2 - p_j^2 = \pi_i - \pi_j = w_{ij} f_{ij} |f_{ij}|, \ \forall (i,j) \in \mathcal{P}. \tag{8.6}$$

Then, variables ℓ_{ij} for $(i,j) \in \mathcal{P}$ are introduced to model the difference in squared pressures across each pipe. The introduction of y, π, and ℓ, as well as convexly relaxing the equality requirements of Constraints (8.6), give rise to the convex relaxation

$$\pi_j - \pi_i \leq \ell_{ij} \leq \pi_i - \pi_j, \ \forall (i,j) \in \mathcal{P} \tag{8.7a}$$
$$\ell_{ij} \leq \pi_j - \pi_i + (2y_{ij})(\overline{\pi}_i - \underline{\pi}_j), \ \forall (i,j) \in \mathcal{P} \tag{8.7b}$$
$$\ell_{ij} \leq \pi_i - \pi_j + (2y_{ij} - 2)(\underline{\pi}_i - \overline{\pi}_j), \ \forall (i,j) \in \mathcal{P} \tag{8.7c}$$
$$w_{ij} f_{ij}^2 \leq \ell_{ij}, \ \forall (i,j) \in \mathcal{P}. \tag{8.7d}$$

Note that Constraints (8.7d) comprise the primary physical relaxations. These constraints imply that the pressure-flow equations need not be satisfied with equality. Convexification of the remaining nonlinear nonconvex terms in Model 8.2 is accomplished in a similar manner to the above. However, in this chapter, to avoid repetition, we omit the derivation of the full MICQP relaxation, which can be found in Chapter 7.

Convexification of Gas-fired Generation Constraints (8.3) are linear when $h_i^1 = 0$ but nonconvex when $h_i^1 > 0$. In the latter case, Constraints (8.3) are relaxed as

$$\sum_{i:(i,j)\in\mathcal{K}'} h_i^1 \mathfrak{R}(S_i^g)^2 + h_i^2 \mathfrak{R}(S_i^g) + h_i^3 z_i^g \leq d_j, \ \forall j \in \mathcal{D}_G, \tag{8.8}$$

where $\mathcal{K}' := \{(i,j) \in \mathcal{K} : h_i^1 \neq 0\}$. However, in our experiments, all h_i^1 are zero.

8.2.4. Summary of Formulations

The rest of this chapter compares formulations of (MLD-G), (MLD-P), and (MLD-W):

- (MLD-*): Exact MINCP formulations of power and gas constraints.
- (MLD-*-R): Formulations with SOCP and MICQP power and gas relaxations.

These formulations provide different tradeoffs between MLD model accuracy and computational performance. An empirical evaluation of both allows us to quantify the effects of the relaxations, as well as to guide our subsequent MLD analytical proofs of concept.

8.3. Computational Experiments

In the following, Section 8.3.1 describes the networks, computational resources, and parameters used throughout the computational experiments; Section 8.3.2 compares the efficacy of exact and relaxed MLD formulations on randomized $N-k$ multi-contingency scenarios; Section 8.3.3 evaluates the runtime performance of formulations over the same experimental sets; Section 8.3.4 provides a proof-of-concept MLD analysis across the same experimental sets, illustrating the tradeoffs when lexicographically maximizing gas and power load delivery; and Section 8.3.5 provides a proof-of-concept Pareto analysis of gas and power load delivery on a single joint network, considering many prioritizations.

8.3.1. Benchmark Data Sets and Experimental Setup

The computational experiments in this chapter consider gas and power networks of various sizes that appear in the literature or have been derived by subject matter experts. These networks are summarized in Table 8.1. The networks in this table are named according to the number of junctions in the natural gas network (e.g., `NG11`) and the

Network	Refs.	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{W}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{K}\rvert$
NG11-EP14	[5], [124]	11	8	0	0	1	0	2	14	20	1
NG25-EP14	[5], [46], [120]	25	24	0	0	0	0	6	14	20	2
NG25-EP30	[5], [124]	25	19	1	1	0	2	3	30	41	1
NG40-EP39	[5], [124]	40	39	0	0	0	0	6	39	46	4
NG146-EP36	[7]	146	93	0	0	0	42	29	36	121	34
NG134-EP162	[5], [124]	134	86	45	0	0	1	1	162	284	5
NG135-EP179	[5], [124]	135	141	0	0	0	0	29	179	263	12
NG247-EP240	[105]	247	254	0	0	0	0	12	240	448	6
NG603-EP588	[5], [124]	603	278	269	8	26	44	5	588	686	12

Table 8.1: Summary of benchmark joint gas-power network data set properties.

number of buses in the electric power network (e.g., EP14). The references from which the gas, power, and/or joint network properties are derived appear in the second column of this table. The numbers of natural gas and electric power system components of the joint networks vary substantially and are specified in the second and third delineated portions of Table 8.1, respectively. Finally, we remark that, for networks referencing [5], heavily loaded variants of the corresponding electric power network data sets are used.

Joint gas-power network properties are summarized in the last column of Table 8.1. Here, NG25-EP14 uses the linking and heat rate properties of the joint network instance developed by [120], and NG146-EP36 uses the properties of the instance developed by [7]. Linkages within the NG247-EP240 network were derived from open data, and heat rate curves were estimated in a manner similar to [7]. The remaining networks combine instances from GasLib and PGLib-OPF to create new joint networks of various sizes. The purpose of these new instances is twofold: (i) to explore the tractability of joint MLD instances as network sizes grow and (ii) to explore the tradeoffs involved in maximizing gas versus power delivery. In these new instances, the number of gas-fired generators, $|\mathcal{K}|$, was estimated to be near $\min\{0.25|\mathcal{D}|, 0.4|\mathcal{G}|\}$, i.e., $\approx 25\%$ of all gas deliveries or $\approx 40\%$ of all generators. After determining the total number of gas-fired generators, the largest-capacity generators in each power network were then assumed to be linked to the smallest-withdrawal delivery points in the gas network. The heat rate at each gas-fired generator was then assumed to be equal to the proportion between the maximum withdrawal at the delivery point and the maximum power at the generator. Note that

these networks thus use synthetically generated linkages between GasLib and PGLib-OPF instances, and these linkages are not necessarily reflective of real-world data sets. They are, however, instances where gas and power interdependencies are consequential, which in turn allows for a meaningful computational exploration of the MLD method.

All MLD formulations were implemented in the Julia programming language using the mathematical modeling layer JuMP, version 0.21 [50]; version 0.9 of GasModels, a package for steady-state and transient natural gas network optimization [9]; version 0.18 of PowerModels, a package for steady-state power network optimization [33]; and version 0.4 of GasPowerModels, a package for joint steady-state gas-power network optimization [8]. Furthermore, for the exact nonconvex representation of Model 8.1 in (MLD-*), the polar form of the AC power flow equations, introduced by [25] and implemented by [33], was used. Similarly, for the exact representation of Model 8.2, the MINQP formulation of Chapter 7, which is implemented in GasModels, was used.

Each optimization experiment was prescribed a wall-clock time limit of one hour on a node containing two Intel Xeon E5-2695 v4 processors, each with 18 cores at 2.10 GHz, and 125 GB of memory. For solutions of (MLD-W), version 0.7 of the open source Juniper MINCP solver was used [82]. Within Juniper, Ipopt 3.12 was leveraged as the NLP solver, using a feasibility tolerance of 10^{-6} and the underlying linear system solver MA57, as recommended by [134] for nonlinear network problems. Note that Juniper does not provide global optimality guarantees for (MLD-W), and feasible solutions obtained from the solver serve only as *lower bounds* on the true amount of maximum deliverable load. For solutions of (MLD-*-R), Gurobi 9.1 was used with its default parameterization. Here, since (MLD-*-R) is an MICP, globally optimal solutions are obtained via Gurobi. However, since (MLD-*-R) is an MLD problem *relaxation*, a globally optimal solution corresponds only to an *upper bound* on (MLD-*)'s objective.

8.3.2. Multi-contingency Damage Scenarios

This section examines the robustness and accuracy of the exact and relaxed weighted MLD formulations, (MLD-W) and (MLD-W-R), respectively, with $\lambda = 0.5$. Specifically, it studies these properties on large sets of randomized multi-contingency or $N-k$ scenarios, where k indicates the number of components simultaneously removed from the joint gas-power network. These scenarios are intended to capture the effects of severe

	(MLD-W) % Cases			(MLD-W-R) % Cases		
Network	Conv.	Lim.	Inf.	Conv.	Lim.	Inf.
NG11-EP14	100.00	0.00	0.00	100.00	0.00	0.00
NG25-EP14	99.90	0.00	0.10	100.00	0.00	0.00
NG25-EP30	98.80	0.10	1.10	99.70	0.00	0.30
NG40-EP39	99.00	0.50	0.50	100.00	0.00	0.00
NG146-EP36	1.00	83.70	15.30	100.00	0.00	0.00
NG134-EP162	26.70	25.70	47.60	100.00	0.00	0.00
NG135-EP179	0.10	95.20	4.70	100.00	0.00	0.00
NG247-EP240	0.00	97.90	2.10	100.00	0.00	0.00
NG603-EP588	0.00	70.30	29.70	100.00	0.00	0.00

Table 8.2: Comparison of solver statuses over weighted random $N-k$ scenarios.

multimodal network outages across joint systems. In each scenario, a random selection of 15% node-connecting components were assumed to be damaged (i.e., $k \approx 0.15N$). Through a parameter sensitivity study, we observed that this proportion of outages appeared to generate challenging MLD scenarios while providing interesting gas and power delivery tradeoffs. For each network, one thousand such scenarios were generated.

Table 8.2 compares statistics of solver termination statuses across all $N-k$ scenarios for each network and formulation. Here, "Conv." corresponds to the percentage of cases where the solver converged, "Lim." to cases where the solver time or other solver limit was reached, and "Inf." to cases that were classified as infeasible by the solver. Although both formulations are typically capable of converging on cases containing tens of nodes, for larger networks, (MLD-W-R) clearly outperforms (MLD-W), solving nearly all $N-k$ instances. The results are especially dramatic for the three largest networks, where only one of three thousand (MLD-W) cases converges but all (MLD-W-R) cases converge. Note that three (MLD-W-R) cases are classified as infeasible due to numerical difficulties, but many more (MLD-W) cases are classified as infeasible due to the MINCP formulation and solver's greater tendencies to converge to locally infeasible points.

Whereas Table 8.2 measures the numerical reliability of exact and relaxed MLD formulations, Table 8.3 compares the solution quality of relaxed formulations with feasible lower bounds obtained from (MLD-W). Here, "# Compared" corresponds to the number of cases used in each comparison, "Mean Obj." is the mean objective value obtained by (MLD-W) over all compared instances, "Mean" is the mean relative gap between

	(MLD-W) Solutions		(MLD-W-R) Gap (%)	
Network	# Compared	Mean Objective	Mean	Median
NG11-EP14	1000	0.61	0.61	0.03
NG25-EP14	999	0.68	1.45	0.17
NG25-EP30	983	0.45	27.22	0.09
NG40-EP39	990	0.65	0.45	0.01
NG146-EP36	10	0.75	4.00	1.01
NG134-EP162	267	0.53	52.63	1.35
NG135-EP179	1	0.59	0.33	0.33
NG247-EP240	0	–	–	–
NG603-EP588	0	–	–	–

Table 8.3: Comparison of solution quality of exact and relaxed MLD formulations.

(MLD-W) and (MLD-W-R) objective values, and "Median" is the median relative gap between objective values. In each such measurement, the relative gap is computed as

$$\text{Relative Gap} := \left(\frac{\tilde{\eta} - \eta}{\eta}\right) 100\%, \tag{8.9}$$

where $\tilde{\eta}$ is the objective value of (MLD-W-R) and η is the objective value of (MLD-W).

We note that, for NG25-EP30, five instances were excluded in the comparison: the three infeasible (MLD-W-R) instances and two instances that implied a negative relative gap. Proceeding with the analysis, the mean objective values for all sets of feasible solutions indicate that between around 50% and 75% of gas and power loads are being delivered across all multi-contingency scenarios. Second, the mean relative gap between feasible solutions obtained by (MLD-W) and the upper bounds obtained by (MLD-W-R) are sometimes large, with the largest being 52.63% across all NG134-EP162 scenarios.

These extreme gaps have only two sources from which they can arise. First, a feasible solution obtained by Juniper for an (MLD-W) instance is not guaranteed to be near the globally optimal solution. That is, the globally optimal (MLD-W) objective value is potentially much larger than what Juniper reports at solver termination. Second, since (MLD-W-R) is a relaxation, it *upper-bounds* the globally optimal objective value of (MLD-W). The median column in Table 8.3 reports measures of centrality without the outliers that are likely arising from the first source of discrepancy (i.e., median relative gaps). Through these measurements, (MLD-W-R) is observed to often provide

reliable and tight bounds on the optimal objective of (MLD-W), with relative gaps most often ranging from nearly zero to less than 1.35%. This indicates that the relaxation is capable of providing good upper bounds on maximum capacities of damaged networks.

8.3.3. Computational Performance

This section compares the performance of (MLD-W) and (MLD-W-R) using the instances described in Section 8.3.2. The performance profiles for these cases are depicted in Figure 8.3 and divided into three categories: (S) networks containing tens of nodes; (M) networks containing hundreds of nodes; and (L) networks containing more than a thousand nodes (i.e., `NG603-EP588`). In all such categories, it is shown that the (MLD-W-R) formulation is able to solve substantially more problems than (MLD-W) in significantly shorter amounts of time. For joint networks with tens of nodes, both formulations are able to solve many instances within the one hour time limit. For networks with hundreds of nodes, (MLD-W-R) is capable of solving most instances within ten seconds, while (MLD-W) requires hundreds or thousands of seconds to solve only a small proportion. For networks with thousands of nodes, (MLD-W-R) solves all instances within ten seconds, whereas (MLD-W) does not solve any. The efficiency of (MLD-W-R) compared to (MLD-W) highlights its applicability to (i) real-time multi-contingency analysis and (ii) analyses that require distributions of many multi-contingency scenarios.

8.3.4. Proof-of-concept Maximum Load Delivery Analysis

Whereas Sections 8.3.2 and 8.3.3 study the computational and accuracy tradeoffs between (MLD-W) and (MLD-W-R), this section provides a proof-of-concept MLD analysis using the (MLD-G-R) and (MLD-P-R) formulations on the same set of $N-k$ damage scenarios. Figures 8.4 and 8.5 display nine histograms each that evaluate the proportions of gas and power loads delivered across *solved* damage scenarios for the nine joint networks while using the two problem specifications. Here, green bars correspond to histogram frequencies obtained from analyzing results of (MLD-G-R) solutions (i.e., gas prioritization) and red bars correspond to (MLD-P-R) solutions (i.e., power prioritization). Brown, overlapping bars correspond to frequencies that appear in both (MLD-P-R) and (MLD-G-R) histograms. These results indicate qualitative differences in the hypothetical robustness of each joint network. They also display the extremal tradeoffs

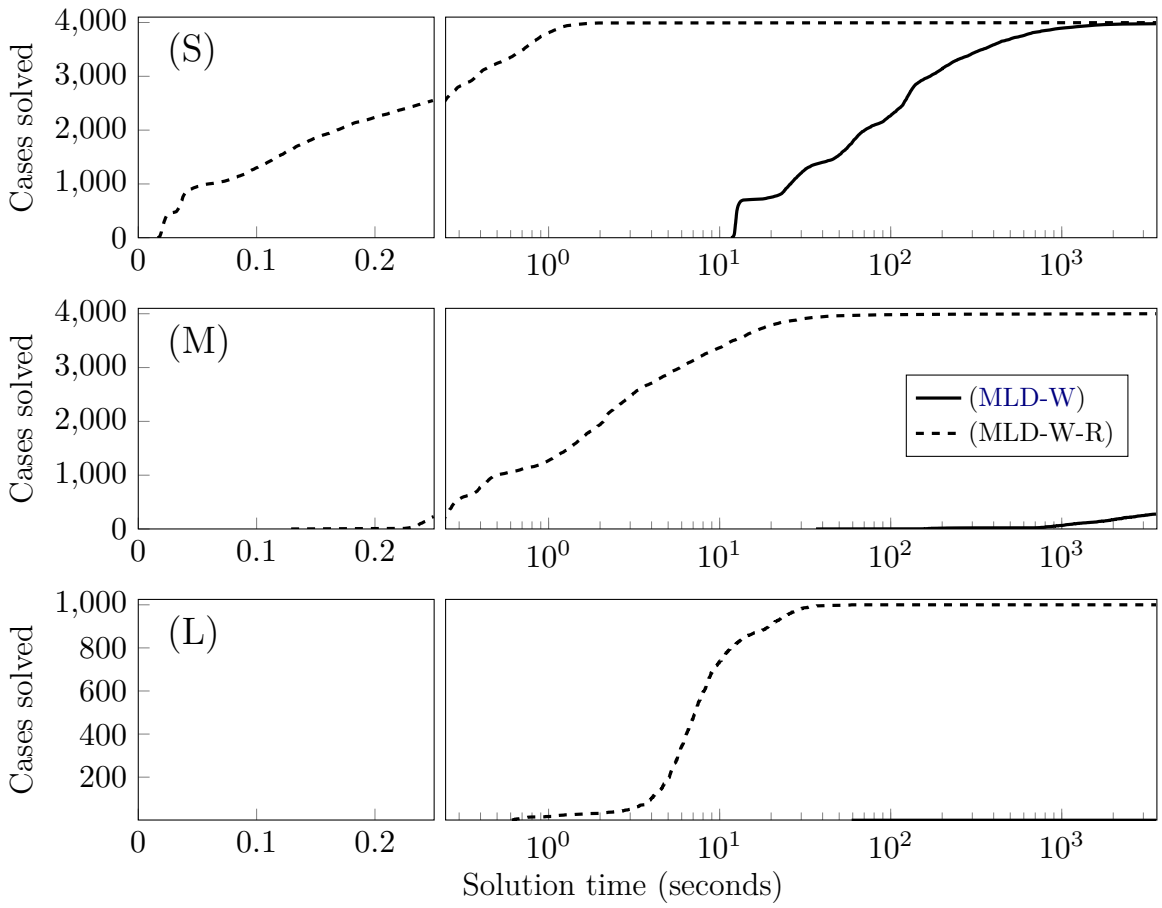


Figure 8.3: Performance profiles of exact and relaxed gas-power MLD formulations. Specifically, the performance profiles measure the solve times of the $N-k$ instances described in Section 8.3.2. Here, the performance profiles are partitioned into three categories for (S) networks containing tens of nodes; (M) networks containing hundreds of nodes; and (L) networks containing more than a thousand nodes (i.e., `NG603-EP588`).

between prioritizing gas versus power delivery in the presence of extreme outages. Finally, they indicate the sensitivity of each gas or power network to the interdependencies that link them. These histograms serve as proofs of concept for real-world analyses.

Figure 8.4 displays histograms of maximum gas load delivered in the presence of severe $N-k$ outages. First, note that these histograms display a variety of load distributions across the cases and networks considered. Some networks, e.g., `NG25-EP30`, `NG247-EP240`, and `NG603-EP588` suggest gas grids that are highly sensitive to the out-

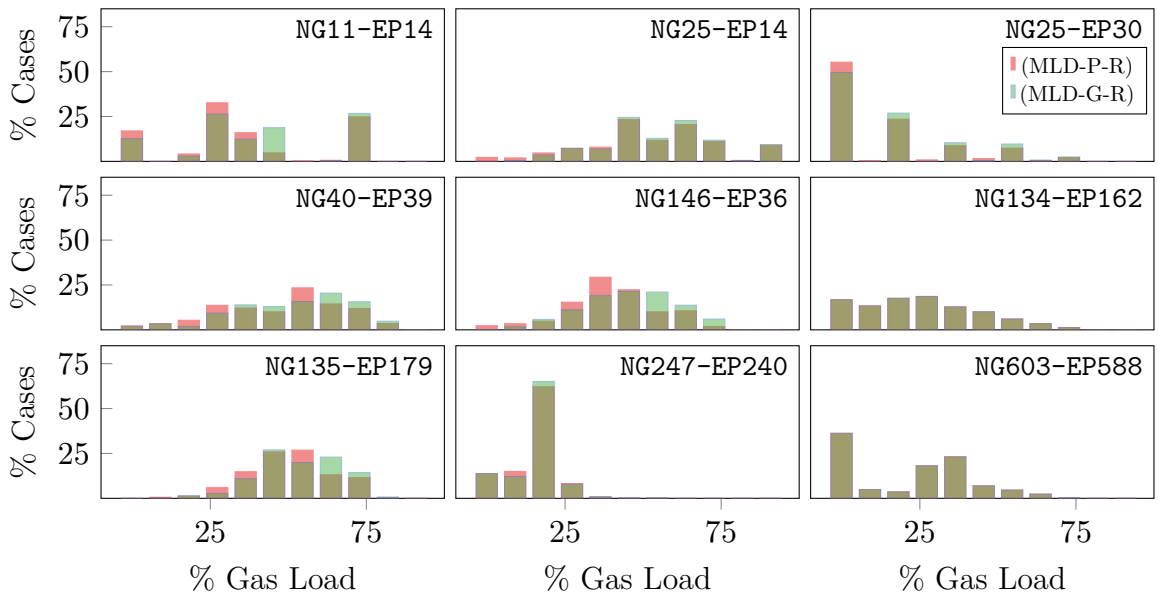


Figure 8.4: Histograms of gas load delivered over random $N-k$ scenarios. The x-axis indicates the proportion of load delivered, and the y-axis indicates the proportion of solved cases that deliver load within an interval.

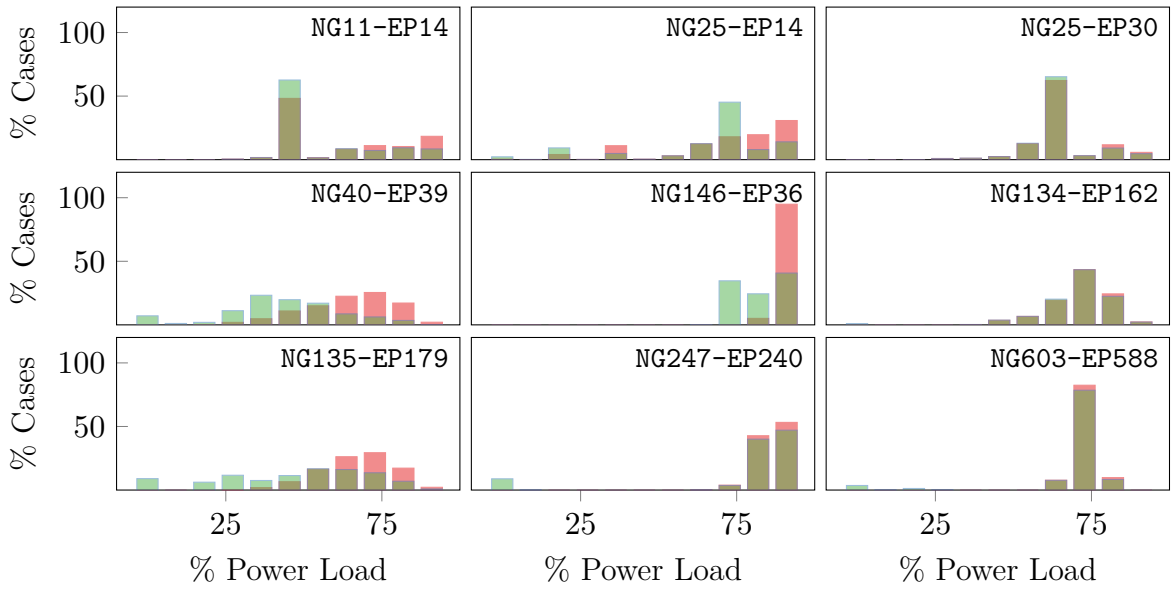


Figure 8.5: Histograms of power load delivered over random $N-k$ scenarios. Note that these histograms assume the same settings as Figure 8.4.

ages considered, with large proportions of damaged networks often incapable of delivering more than 50% of gas load. Other networks, e.g., NG40-EP39, NG146-EP36, and NG135-EP179 show less severe but still substantial sensitivities to outages. The remaining networks display gas network sensitivities somewhere between these two extremes.

The overlapping histograms also display the tradeoffs encountered when prioritizing gas versus power delivery. For the three joint networks NG25-EP14, NG134-EP162, and NG603-EP588 shown in Figure 8.4, gas and power interdependencies appear mostly inconsequential, and prioritizing either gas or power barely affects the maximum capacity of the gas network. This is likely a result of excess generation capacity in the corresponding power networks. Other networks, e.g., NG40-EP39, NG146-EP36, and NG135-EP179 show more interesting tradeoffs, where prioritizing either gas or power results in substantial changes in the overall maximum load distributions. The remaining networks show less interesting tradeoffs, although NG11-EP14 displays large tradeoffs, likely due to the drastic effects that even minor outages can have on the relatively small network.

Figure 8.5 displays histograms of maximum active power delivered in the presence of the $N-k$ outages. First, the four networks NG25-EP30, NG134-EP162, NG247-EP240, and NG603-EP588 appear robust to outages in the joint network and are often capable of delivering more than 75% of the original power load. The remaining networks see a greater variety in their maximum load distributions. Whereas some networks, e.g., NG25-EP30, NG134-EP162, and NG603-EP588, appear less reliant on gas-fired power generators, the remaining networks exhibit more drastic changes when prioritizing gas versus power delivery. The most extreme example appears to be NG146-EP36, which is often capable of delivering a large amount of power across all $N-k$ cases when power is prioritized but also often loses more than 25% capacity when gas delivery is prioritized.

We finally remark that, to solve (MLD-G-R) and (MLD-P-R), inner- and outer-level problems of the lexicographic maximization are solved sequentially. For example, to solve (MLD-G-R), (i) the inner level problem maximizing $\eta_G(d)$ is solved, yielding a solution d^*, then (ii) $\eta_P(z^d)$ is maximized, subject to Constraints (8.1)-(8.3) and $\eta_G(d) \geq \eta_G(d^*) - \epsilon$. The latter ensures that nongeneration gas load delivered in the outer level is at least that of the inner level, minus some feasibility tolerance ϵ, taken in this study to be 10^{-7}. A similar algorithm is used for (MLD-P-R). We note that the general algorithm is not as numerically reliable as (MLD-W-R) and does not solve 469 of the 18,000 $N-k$ cases considered in this subsection. In future work, this could be alleviated with a larger

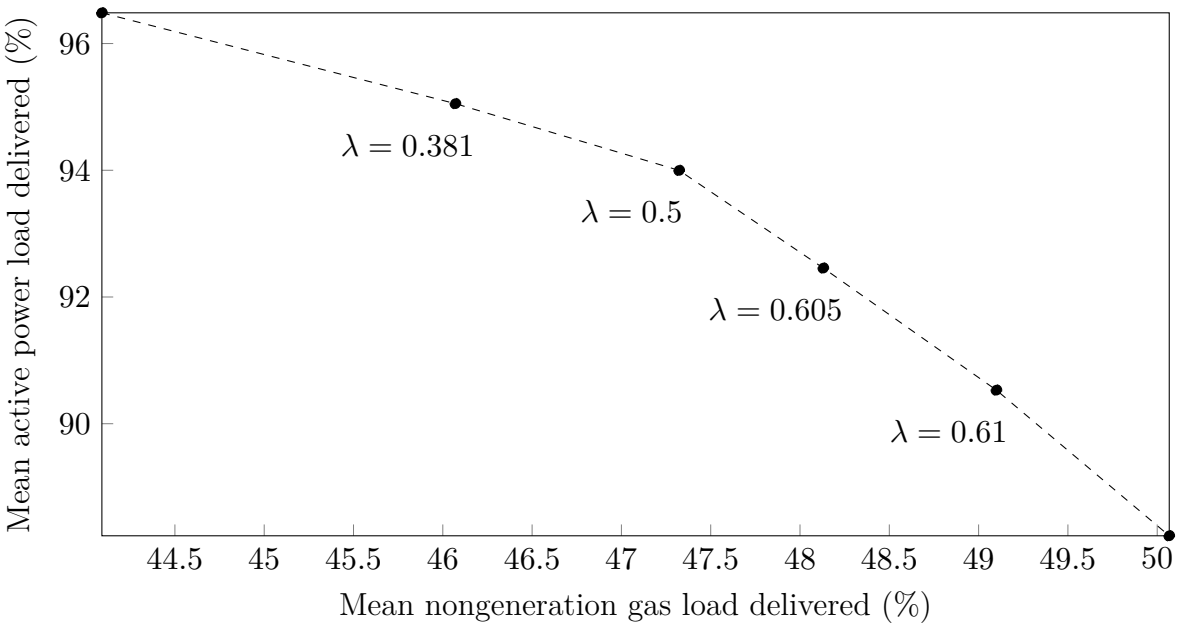


Figure 8.6: Pareto front of power versus gas load over random $N-k$ scenarios.

ϵ or via direct use of lexicographic features available in some solvers (e.g., GUROBI).

8.3.5. Proof-of-concept Pareto Analysis

Together, (MLD-G-R), (MLD-P-R), and (MLD-W-R) allow for a variety of prioritizations of gas versus power load. As such, they serve as powerful modeling tools for exploring the wide range of possible MLD solutions based on the relative importance of gas versus power delivery. This can provide gas and power grid managers with best-case capacity estimates depending on the type of coordination between the two systems. In turn, this enables a better understanding of the extremely complex yet practically important tradeoffs encountered during the operation of a damaged joint network. Whereas Sections 8.3.2 through 8.3.4 focused on analyzing performance and qualitative aspects of MLD analyses across a large number of joint networks, this section focuses on providing a proof-of-concept Pareto analysis on a single joint gas-power network, `NG146-EP36`.

Figure 8.6 shows a linearly-interpolated approximation of the Pareto front for mean active power versus gas delivery across the same set of $N-k$ scenarios considered in previous sections. Here, the upper-left and lower-right endpoints correspond to means

obtained from the (MLD-P-R) and (MLD-G-R) problem formulations, respectively. Interior points correspond to means obtained from (MLD-W-R), where the tradeoff parameter λ was varied to determine interesting and distinct points on the Pareto front.

First, note that when prioritizing power delivery, on average, 96% of active power is delivered but less than 45% of nongeneration gas is delivered. When gas delivery is prioritized, 88% of power is delivered, while 50% of gas is delivered. Between these two extremes, the amount of gas and power increases and decreases, respectively, with increases in λ. For $\lambda \lessapprox 0.5$, active power decreases more slowly as a function of λ, and for $\lambda \gtrapprox 0.5$, the rate of decrease appears larger. In this case, $\lambda \approx 0.5$ happens to represent a value where (MLD-W-R) begins to prefer maximization of gas delivery over power delivery. In practice, a point near this value of λ could be one which maximizes simultaneous delivery of the two quantities while having practically equal prioritizations.

8.4. Conclusion

Recent increases in gas-fired power generation have amplified interdependencies between natural gas and power transmission systems. These interdependencies have engendered greater vulnerabilities to gas and power grids, where natural or man-made disruptions can require the curtailment of load in one or both systems. To address the challenge of estimating maximum joint network capacities under disruptions, this chapter considered the task of determining feasible steady-state operating points for severely damaged joint networks while ensuring the maximal delivery of gas and power loads simultaneously. Mathematically, this task was represented as the nonconvex joint MLD problem.

Three variants of the MLD problem were formulated: one that prioritizes gas delivery, one that prioritizes power delivery, and one that assumes a linear tradeoff between the two objectives. To increase the tractability of these problems, an MICP relaxation of the joint network's physical constraints was proposed. To demonstrate the relaxation's effectiveness, exact and relaxed MLD formulations were computationally compared across a variety of $N-k$ scenarios. The MICP relaxation was found to be a fast and reliable means for determining bounds on maximum capacities of damaged joint networks.

Two proofs of concept were then provided to showcase the analytical power of the relaxed MLD problems. The first presented comparisons between prioritizing gas versus power delivery in an MLD analysis. These examples showcased the sometimes substan-

tial tradeoffs that should be considered when planning for extreme outage scenarios. The second proof of concept provided a Pareto front approximation of gas versus power delivery across $N-k$ scenarios using a single joint network. These proofs of concept highlight that the accuracy and efficiency of the relaxation-based MLD method makes it a potentially valuable tool for complex real-world decision support applications.

Future work should focus on extending the MLD approaches developed in this chapter. First, additional gas and power relaxations should be considered to more accurately and efficiently scale to joint networks containing many thousands of nodes. Preprocessing routines, such as OBBT, may also aid in improving existing relaxations. Second, the current problem assumes the full coordination between gas and power systems when deciding operations that maximize load delivery. The modeling of bidding mechanisms that drive both systems in practice could provide more accurate joint capacity estimates. Finally, capturing transient dynamics in gas networks is sometimes crucial for understanding the effects of network disruptions, which may only be realized long after the disruption occurs due to the relatively slow speed of gas transport. Future work should consider these transient effects when modeling load delivery in the gas network.

Chapter 9

Summary and Outlook

This dissertation considered the design, operation, and restoration of critical infrastructures that control systems of fluids. Although these tasks have interested civilization for millenia, even today, they are often carried out suboptimally for a variety of reasons. One of these reasons is the intractability of optimization problems that involve fluids, which are sometimes constrained by PDEs, nonlinear equations, and discrete decisions. To address these challenges, this dissertation focused on developing new algorithmic techniques for select infrastructure optimization problems. In many cases, application of these techniques brings such optimization problems within the realm of tractability. This suggests there is much potential for modern optimization approaches to assist in real-world decision-making contexts for infrastructures that control systems of fluids.

The first application involved the optimization of structural flood mitigation strategies. In Chapter 3, the OFMP was introduced, and various linearized approximations of the 2D shallow water equations were developed. These physical approximations were then embedded in a MILP representation of the OFMP. Although the approach was successfully applied to small, contrived problems, it could not reliably scale to large instances due to the computational complexity of the resultant MILP. This lack of scalability motivated the development of a metaheuristic-based OFMP optimization approach in Chapter 4. Since metaheuristics alone were not sufficient for finding quality solutions to large-scale OFMPs, novel search space reduction and sequential optimization techniques were designed to increase convergence. These chapters suggest that naive mathematical programming approaches to problems constrained by transient dynamics can be unsuccessful due to the complexity associated with the added temporal

dimension. In these cases, heuristics are important for finding quality initial solutions.

Chapters 5 and 6 considered the development of new approaches to popular problems in water distribution network optimization. Chapter 5 studied the water network design problem, which involves nonconvex equations that model energy losses along pipes and discrete variables that model design decisions. To alleviate the difficulties associated with nonlinear nonconvexity, an *exact* MICP reformulation of the design problem was derived using energy- and duality-based arguments. Using the new formulation as a foundation for a MILP-based algorithm, convergence benefits were observed on moderately-sized design instances. Chapter 6 then studied the related OWF problem, which aims at determining a cost-optimal schedule of discrete pump activations. Compared to the design problem, the OWF is complicated by important temporal dynamics and discrete decisions associated with controllable network elements. A number of new preprocessing, formulation, and algorithmic techniques were developed to improve convergence. However, we observed that (i) finding quality feasible OWF solutions is very difficult and (ii) lower bounds improve slowly, indicating (as in Chapter 3) that temporal dynamics can complicate mathematical programming formulations substantially.

Finally, Chapters 7 and 8 considered infrastructure restoration problems related to the maximal delivery of load in damaged natural gas and joint gas-power networks, respectively. These chapters aimed at applying recent convex relaxation techniques from natural gas and power system optimization to the practical and timely MLD problem. Unlike Chapters 3–6, Chapters 7 and 8 focused on the *application* of relaxation-based mathematical programming techniques in realistic large-scale problem settings and *less* on the development of novel formulation and algorithmic approaches. In both chapters, we observed that relaxation-based methods are the only viable candidates for bounding network properties within the timescales required by decision support applications. This suggests that mathematical programming is especially attractive when the desire is to bound *best*- or *worst*-case estimates of infrastructure properties in a real-time setting.

Despite the disparate applications considered in Chapters 3–8, there are a number of properties that unite them. First, all optimization problems are ultimately constrained by nonconvex equations that model the physical behavior of each system. Second, most problems include discrete elements that model either operational decisions or physical disjunctions. These properties render the optimization problems as difficult MINCPs that are not easily solved using direct methods. As exemplified in this dissertation, due

to the lack of efficient direct methods, tailored approaches must be developed based on properties that are specific either to the problem structure or the physical dynamics. For example, Chapters 3 and 4 highlighted the need to decouple decisions and physical simulation. Chapters 5 and 6 underscored the importance of specialized formulations, relaxations, valid inequalities, and feasibility-checking routines. Finally, Chapters 7 and 8 emphasized that in some applications, modern relaxation approaches are sufficient, although they still require much specialized effort to achieve a robust implementation.

With these observations in mind, in the following sections, we describe our outlook on future solution methods for infrastructure optimization problems involving fluid dynamical constraints. First, in Section 9.1, we discuss the role that physical approximations and relaxations have played in our work and how these could be improved for optimization applications in the future. Then, in Section 9.2, we discuss the importance of deriving problem-specific valid inequalities and how such inequalities could be automatically derived in the future. Finally, noting that directly embedding physical constraints in optimization problems can substantially increase computational complexity, in Section 9.3, we discuss the possibility of developing specialized (i.e., non-solver-based) optimization algorithms that combine fast physical simulation and analysis with traditional paradigms from mathematical programming to ensure both feasibility and optimality.

9.1. Reliable Approximations and Relaxations

The use of physical approximations and relaxations proved to be an important theme of this dissertation. Chapter 3 developed linear approximations of flood dynamics; Chapters 5 and 6 stressed the importance of tight MILP relaxations; and Chapters 7 and 8 leveraged nonlinear convex relaxations. The notion of convexifying difficult physical constraints appears to be key for developing tractable optimization formulations. However, we contend that the application of these techniques is still in its infancy. Several improvements can likely be made to increase tractability without losses in accuracy.

Chapters 5 and 6 stressed that relaxations often lead to infeasible solutions for general problem structures. Chapters 7 and 8 illustrated how relaxations are useful for *bounding* important physical properties. In summary, the required accuracy of a relaxation often depends strongly on the problem context. In a practical sense, it is thus important to understand when relaxation solutions suffice to guide real-world decisions. It is also

important to understand *where* accuracy is required in a relaxed model to maintain near-feasibility. For example, if the head at a water network node is directly influenced by the head at a connected node, accurately modeling loss relationships for other adjacent pipes may be unnecessary. Using domain-based preprocessing to identify *where* modeling accuracy is important in a relaxation will likely be key for increasing problem tractability.

A separate but related research direction resides in the derivation of reduced-order model approximations. Recent efforts in machine learning have enabled the development of fast surrogate methods for flood prediction [96], leak detection in water systems [60], and forecasting demands in natural gas systems [10]. Other work has focused on developing similar techniques to estimate solutions to large-scale infrastructure optimization problems, e.g., the AC optimal power flow problem [68], [147]. Finally, projection-based reduced-order modeling methods, e.g., proper orthogonal decomposition [26] and dynamic mode decomposition [123], could be used to derive quality lower-order approximations of system dynamics. Embedding lower-order models as infrastructure problem constraints is a potentially promising avenue for increasing problem tractability.

9.2. Determining Problem-specific Valid Inequalities

A separate but key component of developing tighter model relaxations is the derivation and application of valid inequalities (i.e., cuts) that strengthen model formulations. The utility of valid inequalities was emphasized in Chapters 5 and 6 and, to a lesser extent, Chapters 7 and 8. Other valid inequalities have shown promise for gas network optimization [73], and general cutting plane algorithms have been successfully applied to more difficult infrastructure optimization problems (e.g., [22]). Successful cutting plane techniques often rely on the problem's structure or some set of physical assumptions.

Ideally, automated methods that derive valid inequalities from model relaxations or physical surrogates should be developed. For example, the OBCG technique of Chapter 6 was used to automatically and efficiently derive large numbers of cuts for the OWF problem. An interesting opportunity could thus reside in general methods that inspect optimization formulation and fluid dynamical structures to determine tighter variable bounds and nonintuitive cuts. However, as emphasized in many portions of the dissertation, creative tailored cuts based on network structure or physical intuition can also be highly beneficial. Novel intertemporal cuts, especially, could prove to be very useful

for increasing the tractability of the problems discussed in Chapters 3, 4, and 6.

9.3. Improving Convergence via Tailored Algorithms

Finally, in every application except Chapter 4, solution methods depended heavily on the use of existing mathematical programming software libraries (e.g., GUROBI and IPOPT). Indeed, the process of (i) creating a suitable problem formulation, (ii) solving the problem using an existing software library, and (iii) appending relevant cuts in situ appears to be the current state of the art for solving difficult infrastructure optimization problems. However, although these implementations can be expedient, it is questionable if this current state of practice is truly the most efficient method for solving MINCPs constrained by *specific* physical relationships. Tailored solution methods that employ the same BB and cutting plane techniques for integer decisions while also leveraging fast oracles for (often unique) solutions that satisfy physical constraints may hold promise.

Examples of this potential can vaguely be seen in the fast shallow water equation simulator used in Chapter 4 and the fast convex oracles for network analysis in Chapters 5 and 6. In a MINCP context, these oracles could potentially act as exact surrogates for discerning constraint satisfaction within custom optimization algorithms that bypass the need for relaxations. Furthermore, the algorithms used to solve oracle subproblems could also be specialized to exploit problem structure and high-performance computing. One example of this paradigm is provided by [79], who develop a specialized structure-exploiting interior point method and parallel, distributed framework for solving the security constrained optimal power flow problem. Here, they observe orders of magnitude speedups over conventional serial solution techniques. Although this philosophy implies greater implementation effort, computational benefits could be manyfold.

Appendix A

Appendix to Chapter 3

A.1. ComputePathline$(\mathbf{U}, x_0, y_0)$

In Algorithm A.1, Line 2, the pathline and current pathline segment lengths, L and ℓ, are initialized to zero, and the pathline-describing point set $\mathcal{L}$ is initialized. In Line 3, the integration loop is defined. Integration halts once the total pathline length is greater than some predefined threshold, $L_{\max}$, or the time falls outside the interval of interest, $[t_0, t_{\text{wet}}]$, where t_{wet} is calculated as per Equation (4.17). In Line 4, the discrete solution indices are obtained. Here, GetIndex(x, y) is a function that maps the spatial coordinates (x, y) to the corresponding spatial index on the rectangular solution grid G, (i, j). Similarly, the time index k is obtained by computing the index of the ordered timestamp set $\mathcal{T}$ corresponding to the least absolute difference with the current integration time t. In Lines 5 through 7, the loop is terminated if the current speed or water depth is smaller than some arbitrarily small positive constant value ϵ_m.

In Line 8, the time step is computed to (approximately) ensure the integrated distance will not be greater than one third the length of a grid cell. In Line 9, the first step of second-order Runge-Kutta integration is performed. In Lines 10 through 12, the loop is terminated if the point suggested by the previous integration step falls outside the flood scenario's spatial domain, denoted as $D(\mathbf{U})$. In Line 13, the discrete indices of the proposed solution are obtained. In Lines 14 through 16, the loop is terminated if the depth at the proposed index is too small. In Line 17, the second Runge-Kutta integration step is performed. In Lines 19 through 21, the loop is terminated if the integrated point falls outside $D(\mathbf{U})$, if the change was small, or if the change was very

Algorithm A.1 ComputePathline: Approximates a pathline emanating to a point.

1: **function** ComputePathline($\mathbf{U}, x_0, y_0$)
2: $L \leftarrow 0,\ \ell \leftarrow 0,\ x \leftarrow x_0,\ y \leftarrow y_0,\ t \leftarrow t_{\text{wet}}(x_0, y_0),\ \mathcal{L} \leftarrow \{(x_0, y_0)\}$
3: **while** $L < L_{\max}$ **and** $t \in [t_0, t_{\text{wet}}(x_0, y_0)]$ **do**
4: $(i, j) \leftarrow \text{GetIndex}(x, y),\ k \leftarrow \text{argmin}\,\{\tau \in \mathcal{T}(\mathbf{U}) : |t - \tau|\}$
5: **if** $\sqrt{u_{ijk}^2 + v_{ijk}^2} \leq \epsilon_m$ **or** $h_{ijk} \leq \epsilon_m$ **then**
6: **break**
7: **end if**
8: $\Delta t \leftarrow -\frac{1}{3} \min\left(\frac{\Delta x}{|u_{ijk}|},\ \frac{\Delta y}{|v_{ijk}|}\right)$
9: $x_* \leftarrow x + u_{ijk}\Delta t,\ y_* \leftarrow y + v_{ijk}\Delta t,\ t_* = t + \Delta t$
10: **if** $(x_*, y_*) \notin D(\mathbf{U})$ **then**
11: **break**
12: **end if**
13: $(i_*, j_*) \leftarrow \text{GetIndex}(x_*, y_*),\ k_* \leftarrow \text{argmin}\,\{\tau \in \mathcal{T}(\mathbf{U}) : |t_* - \tau|\}$
14: **if** $h_{i_*,j_*,k_*} \leq \epsilon_m$ **then**
15: **break**
16: **end if**
17: $x_n \leftarrow x + \frac{1}{2}\Delta t\left(u_{ijk} + u_{i_*,j_*,k_*}\right),\ y_n \leftarrow y + \frac{1}{2}\Delta t\left(u_{ijk} + u_{i_*,j_*,k_*}\right)$
18: $\Delta s \leftarrow \sqrt{(x_n - x)^2 + (y_n - y)^2}$
19: **if** $(x_n, y_n) \notin D(\mathbf{U})$ **or** $\Delta s \leq \epsilon_m$ **or** $\Delta s > 2\alpha$ **then**
20: **break**
21: **end if**
22: $L \leftarrow L + \Delta s,\ \ell \leftarrow \ell + \Delta s$
23: $x \leftarrow x_n,\ y \leftarrow y_n,\ t \leftarrow t_*$
24: **if** $\ell \geq \frac{1}{2}(\Delta x + \Delta y)$ **then**
25: $\ell \leftarrow 0,\ \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_n, y_n)\}$
26: **end if**
27: **end while**
28: **return** $\mathcal{L}$
29: **end function**

large (where α is some predefined fixed distance). In Line 22, the total pathline and temporary segment lengths are updated using the most recent integration distance. In Line 23, the relevant variables are integrated. In Lines 24 through 26, the temporary segment length is reset to zero and the pathline approximation is updated if the total segment length is greater than or equal to the mean grid cell spacing, $\frac{1}{2}(\Delta x + \Delta y)$.

A.2. AlphaShape(Q, α)

In Algorithm A.2, DELAUNAY(Q, α) is a function that computes the *Delaunay triangulation* for a set Q of discrete points. A Delaunay triangulation is a set of triangles such that no point in Q is contained within the circumscribed circle of any triangle. A number of algorithms exist to compute this triangulation. Herein, that of [6] is used.

Algorithm A.2 ALPHASHAPE: Computes an alpha shape from a set of points Q.

```
1:  function AlphaShape(Q, α)
2:      D ← Delaunay(Q),  B ← ∅
3:      for Δ ∈ D do
4:          (a, b, c) ← GetVertices(Δ)
5:          d_a ← ‖a − b‖ ,  d_b ← ‖b − c‖ ,  d_c ← ‖c − a‖
6:          s ← (1/2)(d_a + d_b + d_c)
7:          A ← sqrt(s (s − d_a) (s − d_b) (s − d_c))
8:          if A = 0 then
9:              continue
10:         else if (d_a d_b d_c)/(4A) < α then
11:             B ← B ∪ Δ
12:         end if
13:     end for
14:     return B
15: end function
```

In Line 2 of Algorithm A.2, the set of Delaunay triangles $\mathcal{D}$ is computed for the point set Q, and the set $\mathcal{B}$ comprising the triangular regions of the alpha shape is initialized as the empty set. In Line 4, the function GETVERTICES(Δ) is used to obtain the vertex positions of the triangle Δ. In Line 5, the Euclidean edge distances are computed for the triangle Δ. In Line 6, the semiperimeter s of the triangle Δ is computed. In Line 7, the area of the triangle Δ is computed via Heron's formula. In Line 11, if the circumscribed radius of the triangle is less than the constant α, the triangle is unioned with the set $\mathcal{B}$ describing the alpha shape. In Chapter 4, α is always taken to be $5(\Delta x + \Delta y)/2$, where Δx and Δy are the spacings used to discretize the x- and y- dimensions, respectively.

Appendix B

Appendix to Chapter 5

B.1. Derivation of (P(r))'s Dual

A straightforward method to derive the dual of (P(r)) is via Lagrangian duality, i.e.,

$$\max_{h} \min_{q^{\pm} \geq 0} \mathcal{L}(q^+, q^-, h) = \max_{h} g(h), \tag{B.1}$$

where $\mathcal{L}$ is the Lagrangian of (P(r)) with dual variables h (corresponding to flow conservation constraints), and $g(h)$ is the Lagrangian dual function (to later be maximized). Following the notation of (P(r)), described in Section 5.3.3, its Lagrangian is written as

$$\begin{aligned}
\mathcal{L}(q^+, q^-, h) := &- \sum_{i \in \mathcal{J}} h_i d_i + \sum_{a:=(i,j) \in \mathcal{A}: i \in \mathcal{S}} \left[\frac{L_a r_a}{1+\alpha} (q_a^+)^{1+\alpha} - (h_i^s - h_j) q_a^+ \right] \\
&+ \sum_{a:=(i,j) \in \mathcal{A}: i \in \mathcal{S}} \left[\frac{L_a r_a}{1+\alpha} (q_a^-)^{1+\alpha} + (h_i^s - h_j) q_a^- \right] + \sum_{a:=(i,j) \in \mathcal{A}: i \in \mathcal{J}} \left[\frac{L_a r_a}{1+\alpha} (q_a^+)^{1+\alpha} - (h_i - h_j) q_a^+ \right] \\
&+ \sum_{a:=(i,j) \in \mathcal{A}: i \in \mathcal{J}} \left[\frac{L_a r_a}{1+\alpha} (q_a^-)^{1+\alpha} + (h_i - h_j) q_a^- \right].
\end{aligned} \tag{B.2}$$

Observe that Equation (B.2) is highly separable in $q_a^{\pm}$. As such, minimization over $q^{\pm}$ in Equation (B.1) is straightforward. To derive $g(h)$, it suffices to minimize each component of the second through fifth sums over their corresponding $q_a^{\pm}$ while imposing nonnegativity on $q_a^{\pm}$. Note that all terms are of the form $\frac{b}{1+\alpha} \left(q_a^{\pm}\right)^{1+\alpha} + t q_a^{\pm}$, where $b > 0$

and the sign of t is unknown. There are two possibilities: if $t \geq 0$, the component is nondecreasing in $q_a^{\pm}$ over $q_a^{\pm} \geq 0$, which implies its minimum is attained at $q_a^{\pm} = 0$. Otherwise, if $t < 0$, the function is decreasing at $q_a^{\pm} = 0$, attains its minimum, then starts increasing. This minimum is attained at the stationary point $\hat{q}_a^{\pm} = \sqrt[\alpha]{-\frac{t}{b}}$. In this case, after simplification, the minimum value of the corresponding component is thus

$$\left(\frac{b}{1+\alpha} - b\right)\left(-\frac{t}{b}\right)^{1+\frac{1}{\alpha}} = \frac{-\alpha}{1+\alpha}\frac{(-t)^{1+\frac{1}{\alpha}}}{\sqrt[\alpha]{b}}. \tag{B.3}$$

Next, note that the second and third, as well as the fourth and fifth terms of the sums in Equation (B.2) can be paired such that the b coefficients of each term are the same, while the t coefficients are opposite. That is, one term (with nonnegative t) has a minimum at zero, while the other has a minimum equal to the right-hand side of Equation (B.3). Since the sign of t is unknown, $|t|$ is thus used instead to write $g(h)$ as

$$g(h) = -\sum_{i\in\mathcal{J}} h_i d_i - \sum_{a:=(i,j)\in\mathcal{A}:i\in\mathcal{S}} \frac{\alpha}{1+\alpha}\frac{|h_i^s - h_j|^{1+\frac{1}{\alpha}}}{\sqrt[\alpha]{L_a r_a}} - \sum_{a:=(i,j)\in\mathcal{A}:i\in\mathcal{J}} \frac{\alpha}{1+\alpha}\frac{|h_i - h_j|^{1+\frac{1}{\alpha}}}{\sqrt[\alpha]{L_a r_a}}. \tag{B.4}$$

Rewriting the absolute value terms, the dual problem becomes equivalent to (D(r)).

B.2. Physical Interpretation of Strong Duality

Four sums appear in the objectives of (P(r)) and (D(r)). Each of these sums can be thought of as having a unique physical connotation. This implies some overall physical meaning of the strong duality constraint. To begin, let this constraint be expanded as

$$f_P(q) - f_D(h) = f_1(q) - f_2(q) + f_3(\Delta h) + f_4(h) \leq 0. \tag{B.5}$$

Consider the first summation,

$$f_1(q) = \sum_{a\in\mathcal{A}} \frac{L_a r_a}{1+\alpha}\left[(q_a^+)^{1+\alpha} + (q_a^-)^{1+\alpha}\right]. \tag{B.6}$$

The terms involved are similar to those appearing in the head loss relationships, where $L_a r_a (q_a^{\pm})^{\alpha}$ (conventionally in units of length) can be interpreted as the energy per unit

weight of fluid lost to friction between the moving water and the interior wall of the pipe. Thus, up to a multiplicative constant, $L_a r_a (q_a^{\pm})^{\alpha} q_a^{\pm}$ can be interpreted as the rate of heat transference (i.e., power) between the volume of water and the interior wall of the pipe. This sum can then be interpreted as all power losses from friction of the pipe.

Next, consider the second sum appearing in the primal problem, (P(r)). That is,

$$f_2(q) = \sum_{i \in \mathcal{S}} h_i^s \sum_{a \in \delta_i^+} q_a. \tag{B.7}$$

Here, each head h_i^s is static and can be viewed as the amount of energy per unit weight of water available for extraction from the reservoir. Thus, its product with the reservoir's (outgoing) flow can be interpreted, again up to a multiplicative constant, as the power generated by the reservoir. The sum of all contributions in Equation (B.7) is thus proportional to the power supplied to the network in terms of elevated reservoir flow.

Next, consider the first sum appearing in the objective function of (D(r)). That is,

$$f_3(\Delta h) = \frac{\alpha}{1+\alpha} \sum_{a \in \mathcal{A}} \frac{1}{\sqrt[\alpha]{L_a r_a}} \left[(\Delta h_a^+)^{1+\frac{1}{\alpha}} + (\Delta h_a^-)^{1+\frac{1}{\alpha}} \right]. \tag{B.8}$$

Here, each $\Delta h_a^{\pm}$ denotes the difference in energy per unit weight between adjacent nodes. Each term thus represents, up to a multiplicative constant, the power loss along the pipe in the form of a head differential (i.e., *not* losses to heat from friction). The sum denotes the total loss in *usable* power across the network, represented in terms of head differences.

Finally, consider the second sum appearing in the objective function of (D(r)), i.e.,

$$f_4(h) = \sum_{i \in \mathcal{J}} h_i d_i. \tag{B.9}$$

Here, each demand is fixed, while the energy per unit weight h_i at each junction can vary. Using an argument similar to that of Equation (B.7), this sum thus denotes, up to a constant, the power demanded across all junctions, in terms of their flow demands.

Combining interpretations, the strong duality constraint can be thought of as encoding

$$\text{(frictional loss)} + \text{(realized loss)} + \text{(demand)} \leq \text{(generation)}, \tag{B.10}$$

which implies the conservation of power, with an inequality replacing the usual equality.

B.3. Application of an MICP Solver to (MICP-E)

As discussed in Section 5.4.2, (MICP-E) is a valid reformulation of the design problem, but for the reason of *efficiency*, we do not recommend this problem be solved directly. As [110] discovered in the development of their design algorithm, a MILP outer approximation-based approach seems to be the most efficient technique for solving problems of this type. For this reason, the inner problem of the algorithm we ultimately propose is a MILP *relaxation* of (MICP-E). Furthermore, for checking the feasibility of solutions, we use (P(r)) (not (CP(r))), similar to [110]. Thus, in practice, our study never suffers from the loss of guarantees associated with empty interior subproblems.

That being said, anecdotally, we have found no numerical implications of the empty interior property when applying direct solvers to (CP(r)) and (MICP-E). To show this, we provide a numerical example, which solves the (MICP-E) formulation of the small `shamir` instance directly. The convergence profile of this example is shown in Figure B.1. The example solves this instance using the MICP solver Juniper [82]. Within Juniper, Ipopt version 3.13 is employed as the underlying NLP solver. First and foremost, the solver converges to the known optimal objective value of 419,000. Second, we note that optimality is proven after over seven hours, supporting our claims that (i) the direct solution of (MICP-E) is challenging and (ii) a MILP relaxation is beneficial.

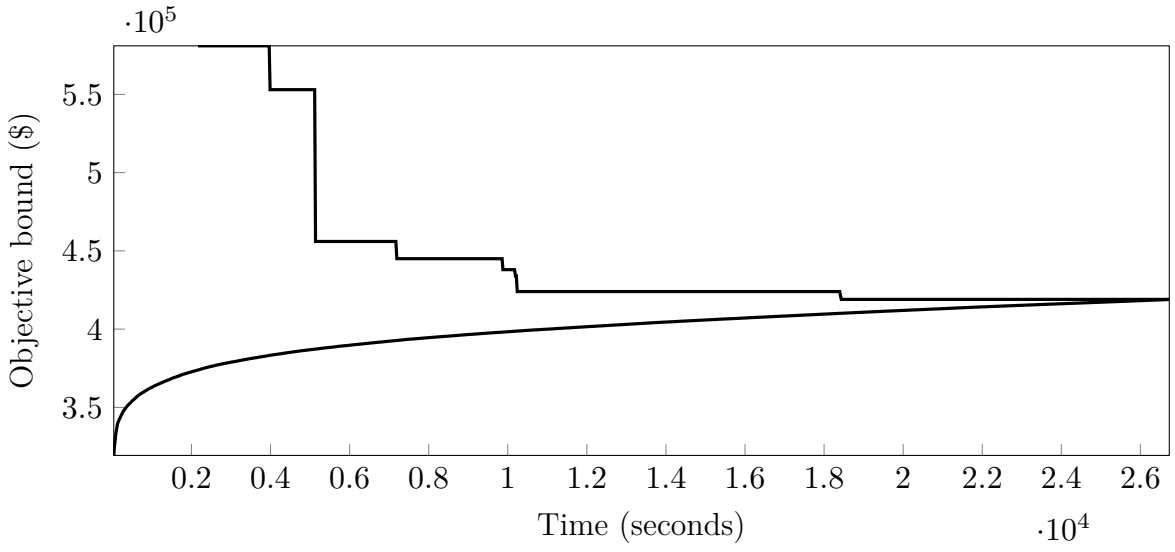


Figure B.1: Direct solution of (MICP-E) on the `shamir` instance using Juniper.

We have not examined the scaling of direct methods on instances larger than `shamir` due to the extremely large solve times that would be required. We thus cannot conclusively state that direct solution of (MICP-E) on larger instances will perform in a numerically stable manner. We expect that either (i) direct methods work "out of the box," as JUNIPER does for `shamir`, or (ii) an epsilon-relaxed reformulation is required.

B.4. Modified Global Optimization Algorithm

Algorithm B.1 LP/NLP-BB algorithm for solving (MINLP)/(MICP-E).

$r^* \leftarrow$ **InitialSoln** (Algorithm 4 of [110]); $\eta^* \leftarrow \sum_{a\in\mathcal{A}} L_a c_{ar_a^*}$; $\tilde{\mathcal{X}} \leftarrow \emptyset$.
$(\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y}) \leftarrow$ Solve (MICP-E) with $0 \le z \le 1$ using an NLP solver.
$r_a^{\pm} \leftarrow arg\,max_{p\in\mathcal{R}_a} \left\{L_a p(\hat{q}_{ap}^{\pm})^{\alpha}\right\}$, $\forall a \in \mathcal{A}$; $\tilde{\mathcal{Q}}_{ar_a}^{\pm} \leftarrow \{\hat{q}_{ar_a}^{\pm}\}$, $\forall a \in \mathcal{A}$.
$r_a^{\pm} \leftarrow arg\,max_{p\in\mathcal{R}_a} \left\{p(\hat{q}_{ap}^{\pm})^{1+\alpha}\right\}$, $\forall a \in \mathcal{A}$; $\tilde{\mathcal{Q}}_{ar_a}^{\mathrm{NL}^{\pm}} \leftarrow \{\hat{q}_{ar_a}^{\pm}\}$, $\forall a \in \mathcal{A}$.
$r_a^{\pm} \leftarrow arg\,max_{p\in\mathcal{R}_a} \left\{\frac{1}{\sqrt[\alpha]{p}}(\Delta\hat{h}_{ap}^{\pm})^{1+\frac{1}{\alpha}}\right\}$, $\forall a \in \mathcal{A}$; $\tilde{\mathcal{H}}_{ar_a}^{\mathrm{NL}^{\pm}} \leftarrow \{\Delta\hat{h}_{ar_a}^{\pm}\}$, $\forall a \in \mathcal{A}$.
Add $\sum_{a\in\mathcal{A}} L_a \sum_{p\in\mathcal{R}_a} c_{ap} z_{ap} \ge \sum_{a\in\mathcal{A}} L_a \sum_{p\in\mathcal{R}_a} c_{ap} \hat{z}_{ap}$ to (MIP-E).
while (MIP-ER) termination criteria is not satisfied **do**
    $(\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y}) \leftarrow$ Solve the current nodal linear subproblem of (MIP-ER).
    $\hat{\eta} \leftarrow \sum_{a\in\mathcal{A}} L_a \sum_{p\in\mathcal{R}_a} c_{ap}\hat{z}_{ap}$.
    if $\hat{z}_{ap} \in \mathbb{B},\ \forall a \in \mathcal{A},\ \forall p \in \mathcal{R}_a$ **then**
        $r_a \in \{p \in \mathcal{R}_a : \hat{z}_{ap} = 1\},\ \forall a \in \mathcal{A}$.
        $(\hat{q}, \hat{h}) \leftarrow$ Solve (P(r)).
        if $\underline{q} \le \hat{q} \le \overline{q}$ and $\underline{h} \le \hat{h} \le \overline{h}$ **then**
            $r^* \leftarrow r$; $\eta^* \leftarrow \sum_{a\in\mathcal{A}} L_a c_{ar_a^*}$.
            $\mathcal{Q}_{ar_a}^{\pm} \leftarrow \tilde{\mathcal{Q}}_{ar_a}^{\pm} \cup \{\pm\hat{q}_a\}$, $\forall a \in \mathcal{A} : \pm\hat{q}_a > 0$.
            $\mathcal{Q}_{ar_a}^{\mathrm{NL}\pm} \leftarrow \tilde{\mathcal{Q}}_{ar_a}^{\mathrm{NL}\pm} \cup \{\pm\hat{q}_a\}$, $\forall a \in \mathcal{A} : \pm\hat{q}_a > 0$.
            $\tilde{\mathcal{H}}_{ar_a}^{\mathrm{NL}\pm} \leftarrow \tilde{\mathcal{H}}_{ar_a}^{\mathrm{NL}\pm} \cup \{\pm\Delta\hat{h}_a\}$, $\forall a \in \mathcal{A} : \pm\Delta\hat{h}_a > 0$.
        else
            $\tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}} \cup \{\hat{z}\}$ (i.e., add a feasibility cut that removes $\hat{z}$).
            (repaired, r) $\leftarrow$ **Repair**(r, n, η^*) (Algorithm 3 of [110]).
            if repaired **then**
                $r^* \leftarrow r$; $\eta^* \leftarrow \sum_{a\in\mathcal{A}} L_a c_{ar_a^*}$.
                $(\hat{q}, \hat{h}) \leftarrow$ Solve (P(r)).
                $\tilde{\mathcal{Q}}_{ar_a}^{\pm} \leftarrow \tilde{\mathcal{Q}}_{ar_a}^{\pm} \cup \{\pm\hat{q}_a\}$, $\forall a \in \mathcal{A} : \pm\hat{q}_a > 0$.

... Continuation of Algorithm B.1.

25: $\tilde{\mathcal{Q}}^{\mathrm{NL}\pm}_{ar_a} \leftarrow \tilde{\mathcal{Q}}^{\mathrm{NL}\pm}_{ar_a} \cup \{\pm\hat{q}_a\},\ \forall a \in \mathcal{A} : \pm\hat{q}_a > 0.$
26: $\tilde{\mathcal{H}}^{\mathrm{NL}\pm}_{ar_a} \leftarrow \tilde{\mathcal{H}}^{\mathrm{NL}\pm}_{ar_a} \cup \{\pm\Delta\hat{h}_a\},\ \forall a \in \mathcal{A} : \pm\Delta\hat{h}_a > 0.$
27: **end if**
28: **end if**
29: **else if** (node index) mod $J = 0$ **then**
30: **NodeCutsNew**$(\tilde{\eta}, \hat{\eta}, m, (\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y}), \mathrm{True})$.
31: $r_a \leftarrow arg\,min\left\{\hat{z}_{ap} \geq \frac{1}{|\mathcal{R}_a|} : p \in \mathcal{R}_a\right\},\ \forall a \in \mathcal{A}.$
32: (repaired, r) $\leftarrow$ **Repair**$(r, \mathrm{iter}^{\max}, \eta^*)$ (Algorithm 3 of [110]).
33: **if** repaired **then**
34: $r^* \leftarrow r$; $\eta^* \leftarrow \sum_{a\in\mathcal{A}} L_a c_{ar^*_a}$.
35: **end if**
36: **else**
37: **NodeCutsNew**$(\tilde{\eta}, \hat{\eta}, m, (\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y}), \mathrm{False})$.
38: **end if**
39: $\tilde{\eta} \leftarrow \hat{\eta}$.
40: **end while**

Algorithm B.1 modifies the algorithm of [110] to define the global optimization algorithm of Chapter 5. Portions that substantially modify the original algorithm are denoted by blue colored font. The algorithm begins in Line 1 by heuristically generating a feasible solution via Algorithm 4 of [110] and storing the corresponding resistances r^* and initial objective η^*. It also initializes the set of encountered infeasible designs $\tilde{\mathcal{X}}$ to the empty set. Line 2 solves the root relaxation of (MICP-E) via a nonlinear programming algorithm (e.g., IPOPT of [142]). Lines 3 through 5 use the root relaxation's solution to generate initial linear outer approximations. In each cut, outer approximation points are chosen to coincide with maximal values of the corresponding nonlinear terms. Finally, since the root relaxation of (MICP-E) provides a lower bound on the optimal objective, Line 6 imposes this bound explicitly on the objective of (MIP-ER).

Line 7 begins the search of a MILP solver, where termination criteria comprises a minimal optimality gap or a time limit. Line 8 obtains the relaxation solution for the current node in the search tree. Line 9 computes the objective corresponding to this solution. Line 10 checks if the current BB node's resistance choice solution is integer. If so, Line 11 obtains the corresponding active resistance parameters. Using these resistances, Line 12 solves (P(r)) to obtain its primal and dual solutions, $\hat{q}$ and $\hat{h}$,

respectively. This solution is then compared against variable bounds in Line 13.

If bounds are satisfied, $\hat{z}$ is feasible for (MICP-E), and in Line 14, the incumbent is updated. If the solution is physically feasible, Lines 15 through 17 add outer approximations using the corresponding solution of (P(r)). Otherwise, if the design is *not* feasible, Line 19 adds a traditional combinatorial no-good feasibility cut. In Line 20, Algorithm 3 of [110] is called in the attempt to heuristically recover a new feasible incumbent solution by "repairing" infeasibilities of the design $\hat{z}$, where n denotes the (fixed) maximum number of repair iterations to be used by the procedure. If the solution *was* repaired, as indicated in Line 21, then a new incumbent was found, which is updated in Line 22. Using the resistances from this design, Line 23 solves (P(r)) to obtain the exact physical solution of the network. Then, in Lines 24 through 26, additional outer approximations are appended to (MIP-ER) based on the physical solution corresponding to the design.

If the design solution is not integral, Line 29 serves as a possible entry point for adding outer approximations to (MIP-ER) and heuristically discovering new solutions. In this case, if the integer index of the BB node is divisible by some integer J, these routines are called. In Line 30, Algorithm B.2 is called, which adds outer approximations based on the relaxation solution at the current BB node. This algorithm is described in Appendix B.5. Then, in Line 31, a heuristic resistance solution is prepared, which selects from active resistances in the current relaxation solution. The repair algorithm is then invoked on this (inexpensive but presumably infeasible) network design in Lines 32 through 34. For greater detail, this heuristic procedure is developed and elaborated upon by [110]. Otherwise, if both the design solution is fractional and the above heuristic is not activated, Algorithm B.2 is called on Line 37, which conditionally refines (MIP-ER)'s outer approximations using the solution of the relaxation at the current BB node.

Finally, on Line 39, the objective at the current BB node, $\hat{\eta}$, is stored as $\tilde{\eta}$ for use within algorithmic methods of the next node. Specifically, both the current objective $\hat{\eta}$ and previous objective $\tilde{\eta}$ are used in a conditional step of Algorithm B.2 to determine whether new outer approximations should be added to the master problem, (MIP-ER).

Algorithm B.1 is similar to the algorithm of [110] but differs in a few important respects. First, in Lines 3 through 5, initial outer approximations are based on the root relaxation of (MICP-E), whereas the algorithm of [110] uses the root relaxation of (MICP-R). Second, the algorithm of [110] only applies outer approximations similar to Constraints (5.22), which correspond to head loss relaxations. Algorithm B.1 extends

this by adding outer approximations of terms appearing in the strong duality Constraint (5.27). Lastly, Line 30 ensures outer approximations will occasionally be added to (MIP-ER). The conditions in [110] (and their **NodeCuts**) are more restrictive.

In Chapter 5, implementations of the new and previous algorithms required some modifications, as necessitated by feature limitations of JuMP. First, the depth of a node in the BB tree (m in Algorithm B.1) is used as a parameter in both algorithms but is not easily accessible via JuMP. Thus, the number of unity-valued components of $\hat{z}$ at a node in the BB tree is used instead. Additionally, a number of modifications are made to the algorithm of [110] to ensure fairer comparison with Algorithm B.1. The most substantial of these are as follows: (i) initial outer approximation points are taken from the root relaxation of (MICP-E); and (ii) similar to Lines 29 through 35 in Algorithm B.1, the heuristic *and* **NodeCuts** methods are called every J nodes. Aside from these small modifications, the algorithm of [110] is nearly reproduced in its entirety.

B.5. Modified NodeCuts Algorithm

Algorithm B.2 modifies the **NodeCuts** algorithm of [110] to define Chapter 5's implementation. Portions that substantially modify the original are denoted by blue font. The method requires five parameters: $\tilde{\eta}$, the objective value of the previous BB node's relaxation solution; $\hat{\eta}$, the objective value at the current node; m, the depth of the current node; $(\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y})$, the relaxation solution at the current node; and Force, a Boolean variable describing whether or not to force the addition of cuts. Line 1 begins by checking three conditions. First, a random number uniformly generated between zero and one is compared against the term $\beta_{oa}2^{-m}$, where β_{oa} is a positive parameter. This encourages cuts to be added near the top of the search tree. The second condition, $\left|\frac{\hat{\eta}-\tilde{\eta}}{\tilde{\eta}}\right| \geq K_{oa}$, computes the relative change in the objective value between successive nodes and compares it with the positive constant K_{oa}. This encourages cuts to be added only when the solution has significantly changed. Finally, Force allows the algorithm to override the previous conditions, as used in Line 30 of Algorithm B.1.

If conditions have been satisfied, the algorithm proceeds in Line 2 by looping over all arcs. In Line 3, if the relaxed flow solution along an arc is overall positive, cuts for positive flow are potentially added in Lines 5, 7, and 9. The reference resistances for each of these cuts, r_a, are independently computed in Lines 4, 6, and 8, where each

Algorithm B.2 NodeCutsNew$(\tilde{\eta}, \hat{\eta}, m, (\hat{q}, \hat{h}, \Delta\hat{h}, \hat{z}, \hat{y}),$ Force$)$

1: **if** $\left[(\text{rand}([0,1]) \leq \beta_{oa} 2^{-m}) \wedge \left(\left|\frac{\hat{\eta}-\tilde{\eta}}{\tilde{\eta}}\right| \geq K_{oa}\right)\right] \vee \text{Force} = \text{True}$ **then**
2: **for** $a \in \mathcal{A}$ **do**
3: **if** $\hat{y}_a \geq 0.5$ **then**
4: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{L_a p(\hat{q}^+_{ap})^\alpha\right\}$
5: **if** $\left|L_a r_a \left(\hat{q}^+_{ar_a}\right)^\alpha - \Delta\hat{h}^+_{ar_a}\right| > \epsilon$ **then** $\tilde{\mathcal{Q}}^+_{ar_a} \leftarrow \tilde{\mathcal{Q}}^+_{ar_a} \cup \{\hat{q}^+_{ar_a}\}$ **end if**
6: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{p(\hat{q}^+_{ap})^{1+\alpha}\right\}$
7: **if** $\left|\frac{r_a\left(\hat{q}^+_{ar_a}\right)^{1+\alpha}}{1+\alpha} - \hat{q}^{\text{NL}}_a\right| > \epsilon$ **then** $\tilde{\mathcal{Q}}^{\text{NL}+}_{ar_a} \leftarrow \tilde{\mathcal{Q}}^{\text{NL}+}_{ar_a} \cup \{\hat{q}^+_{ar_a}\}$ **end if**
8: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{\frac{1}{\sqrt[\alpha]{p}} (\Delta\hat{h}^+_{ap})^{1+\frac{1}{\alpha}}\right\}$
9: **if** $\left|\frac{(\Delta\hat{h}^+_{ar_a})^{1+\frac{1}{\alpha}}}{(1+\frac{1}{\alpha})\sqrt[\alpha]{r_a}} - \Delta\hat{h}^{\text{NL}}_a\right| > \epsilon$ **then** $\tilde{\mathcal{H}}^{\text{NL}+}_{ar_a} \leftarrow \tilde{\mathcal{H}}^{\text{NL}+}_{ar_a} \cup \{\Delta\hat{h}^+_{ar_a}\}$ **end if**
10: **else**
11: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{L_a p(\hat{q}^-_{ap})^\alpha\right\}$
12: **if** $\left|L_a r_a \left(\hat{q}^-_{ar_a}\right)^\alpha - \Delta\hat{h}^-_{ar_a}\right| > \epsilon$ **then** $\tilde{\mathcal{Q}}^-_{ar_a} \leftarrow \tilde{\mathcal{Q}}^-_{ar_a} \cup \{\hat{q}^-_{ar_a}\}$ **end if**
13: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{p(\hat{q}^-_{ap})^{1+\alpha}\right\}$
14: **if** $\left|\frac{r_a\left(\hat{q}^-_{ar_a}\right)^{1+\alpha}}{1+\alpha} - \hat{q}^{\text{NL}}_a\right| > \epsilon$ **then** $\tilde{\mathcal{Q}}^{\text{NL}-}_{ar_a} \leftarrow \tilde{\mathcal{Q}}^{\text{NL}-}_{ar_a} \cup \{\hat{q}^-_{ar_a}\}$ **end if**
15: $r_a \leftarrow arg\,max_{p \in \mathcal{R}_a} \left\{\frac{1}{\sqrt[\alpha]{p}} (\Delta\hat{h}^-_{ap})^{1+\frac{1}{\alpha}}\right\}$
16: **if** $\left|\frac{(\Delta\hat{h}^-_{ar_a})^{1+\frac{1}{\alpha}}}{(1+\frac{1}{\alpha})\sqrt[\alpha]{r_a}} - \Delta\hat{h}^{\text{NL}}_a\right| > \epsilon$ **then** $\tilde{\mathcal{H}}^{\text{NL}-}_{ar_a} \leftarrow \tilde{\mathcal{H}}^{\text{NL}-}_{ar_a} \cup \{\Delta\hat{h}^-_{ar_a}\}$ **end if**
17: **end if**
18: **end for**
19: **end if**

corresponds to the maximum nonlinear term given the current relaxation solution. The conditionals on Lines 5, 7, and 9 imply these cuts will only be added if the relaxation solution deviates substantially (greater than a violation of some small constant, ϵ) compared to the solution of a corresponding nonlinear formulation. This limits the number of cuts being added by ensuring that only constraints with significant violations will be linearized. In Lines 11 through 16, the process is analogously completed for arcs where flow along that arc is overall negative. Parameters of Algorithms B.1 and B.2 coincided with those used by [110], namely, $\beta_{oa} = 5$, $J = 500$, $K_{oa} = 10^{-3}$, $n = 50$, and $\epsilon = 10^{-6}$.

Bibliography

[1] A. Abeliuk, H. Aziz, G. Berbeglia, S. Gaspers, P. Kalina, N. Mattei, D. Peters, P. Stursberg, P. Van Hentenryck, and T. Walsh, "Interdependent scheduling games," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16, New York, New York, USA: AAAI Press, 2016, pp. 2–9, ISBN: 9781577357704.

[2] M. Ahumada-Paras, K. Sundar, R. Bent, and A. Zlotnik, "N-k interdiction modeling for natural gas networks," *Electric Power Systems Research*, vol. 190, p. 106 725, 2021, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2020.106725.

[3] S. Artina, C. Bragalli, G. Erbacci, A. Marchi, and M. Rivi, "Contribution of parallel NSGA-II in optimal design of water distribution networks," *Journal of Hydroinformatics*, vol. 14, no. 2, pp. 310–323, Jun. 2011, ISSN: 1464-7141. DOI: 10.2166/hydro.2011.014.

[4] Atmos International, *Atmos Pipe*, https://www.atmosi.com/us/products-services/atmos-pipe, 2021.

[5] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, C. Josz, R. Korab, B. Lesieutre, J. Maeght, T. W. K. Mak, D. K. Molzahn, T. J. Overbye, P. Panciatici, B. Park, J. Snodgrass, A. Tbaileh, P. V. Hentenryck, and R. Zimmerman, *The Power Grid Library for benchmarking AC optimal power flow algorithms*, 2021. arXiv: 1908.02788 [math.OC].

[6] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, Dec. 1996, ISSN: 0098-3500. DOI: 10.1145/235815.235821.

[7] R. Bent, S. Blumsack, P. Van Hentenryck, C. Borraz-Sánchez, and M. Shahriari, "Joint electricity and natural gas transmission planning with endogenous market feedbacks," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6397–6409, 2018. DOI: 10.1109/TPWRS.2018.2849958.

[8] R. Bent, C. Coffrin, K. Sundar, and B. Tasseff, *GasPowerModels.jl*, https://github.com/lanl-ansi/GasPowerModels.jl, 2021.

[9] R. Bent, K. Sundar, and D. Fobes, *GasModels.jl*, https://github.com/lanl-ansi/GasModels.jl, 2020.

[10] O. F. Beyca, B. C. Ervural, E. Tatoglu, P. G. Ozuyar, and S. Zaim, "Using machine learning tools for forecasting natural gas consumption in the province of Istanbul," *Energy Economics*, vol. 80, pp. 937–949, 2019, ISSN: 0140-9883. DOI: 10.1016/j.eneco.2019.03.006.

[11] R. G. Bland, D. Goldfarb, and M. J. Todd, "The ellipsoid method: A survey," *Operations Research*, vol. 29, no. 6, pp. 1039–1091, 1981. DOI: 10.1287/opre.29.6.1039.

[12] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008, ISSN: 1572-5286. DOI: 10.1016/j.disopt.2006.10.011.

[13] G. Bonvin and S. Demassey, "Extended linear formulation of the pump scheduling problem in water distribution networks," in *International Network Optimization Conference*, ser. Open Proceedings, Jun. 2019, pp. 13–18. DOI: 10.5441/002/inoc.2019.04.

[14] G. Bonvin, S. Demassey, C. Le Pape, N. Maïzi, V. Mazauric, and A. Samperio, "A convex mathematical program for pump scheduling in a class of branched water networks," *Applied Energy*, vol. 185, pp. 1702–1711, 2017, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2015.12.090.

[15] G. Bonvin, S. Demassey, and A. Lodi, "Pump scheduling in drinking water distribution networks with an LP/NLP-based branch and bound," *Optimization and Engineering*, Jan. 2021, ISSN: 1573-2924. DOI: 10.1007/s11081-020-09575-y.

[16] C. Borraz-Sánchez, R. Bent, S. Backhaus, H. Hijazi, and P. V. Hentenryck, "Convex relaxations for gas expansion planning," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 645–656, 2016. DOI: 10.1287/ijoc.2016.0697.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004, ISBN: 0521833787.

[18] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth, "On the optimal design of water distribution networks: A practical MINLP approach," *Optimization and Engineering*, vol. 13, no. 2, pp. 219–246, Jun. 2012, ISSN: 1573-2924. DOI: 10.1007/s11081-011-9141-7.

[19] R. Brekelmans, D. d. Hertog, K. Roos, and C. Eijgenraam, "Safe dike heights at minimal costs: The nonhomogeneous case," *Operations Research*, vol. 60, no. 6, pp. 1342–1355, 2012. DOI: 10.1287/opre.1110.1028.

[20] A. R. Brodtkorb, M. L. Sætra, and M. Altinakar, "Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation," *Computers & Fluids*, vol. 55, pp. 1–12, 2012, ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2011.10.012.

[21] G. O. Brown, "The history of the Darcy-Weisbach equation for pipe flow resistance," in *Environmental and Water Resources History*, ch. 4, pp. 34–43. DOI: 10.1061/40650(2003)4.

[22] G. Byeon and P. V. Hentenryck, *Benders subproblem decomposition for bilevel problems with convex follower*, 2021. arXiv: 1902.04375 [math.OC].

[23] G. Byeon and P. Van Hentenryck, "Unit commitment with gas network awareness," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1327–1339, 2020. DOI: 10.1109/TPWRS.2019.2942298.

[24] K. W. Campbell, "Empirical near-source attenuation relationships for horizontal and vertical components of peak ground acceleration, peak ground velocity, and pseudo-absolute acceleration response spectra," *Seismological Research Letters*, vol. 68, no. 1, pp. 154–179, Jan. 1997, ISSN: 0895-0695. DOI: 10.1785/gssrl.68.1.154.

[25] J. Carpentier, "Contribution to the economic dispatch problem," *Bulletin de la Societe Francoise des Electriciens*, vol. 3, no. 8, pp. 431–447, 1962.

[26] A. Chatterjee, "An introduction to the proper orthogonal decomposition," *Current Science*, vol. 78, no. 7, pp. 808–817, 2000, ISSN: 0011-3891.

[27] D. Che and L. W. Mays, "Development of an optimization/simulation model for real-time flood-control operation of river-reservoirs systems," *Water Resources Management*, vol. 29, no. 11, pp. 3987–4005, Jun. 2015, ISSN: 0920-4741. DOI: 10.1007/s11269-015-1041-8.

[28] S. Chen, Z. Wei, G. Sun, D. Wang, and H. Zang, "Steady state and transient simulation for electricity-gas integrated energy systems by using convex optimisation," *IET Generation, Transmission & Distribution*, vol. 12, no. 9, pp. 2199–2206, 2018. DOI: 10.1049/iet-gtd.2017.1318.

[29] C. Cherry, "CXVII: Some general theorems for non-linear systems possessing reactance," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 333, pp. 1161–1177, 1951. DOI: 10.1080/14786445108561362.

[30] A. Chertock, S. Cui, A. Kurganov, and T. Wu, "Well-balanced positivity preserving central-upwind scheme for the shallow water system with friction terms," *International Journal for Numerical Methods in Fluids*, vol. 78, no. 6, pp. 355–383, 2015, ISSN: 1097-0363. DOI: 10.1002/fld.4023.

[31] C. M. Chini and A. S. Stillwell, "The state of U.S. urban water: Data and the energy-water nexus," *Water Resources Research*, vol. 54, no. 3, pp. 1796–1811, 2018. DOI: 10.1002/2017WR022265.

[32] A. Christie, *Chinese Mythology*. Hamlyn, 1973.

[33] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference*, 2018, pp. 1–8. DOI: 10.23919/PSCC.2018.8442948.

[34] C. Coffrin, R. Bent, B. Tasseff, K. Sundar, and S. Backhaus, "Relaxations of AC maximal load delivery for severe contingency analysis," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1450–1458, 2019. DOI: 10.1109/TPWRS.2018.2876507.

[35] C. Coffrin, P. Van Hentenryck, and R. Bent, "Last-mile restoration for multiple interdependent infrastructures," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12, Toronto, Ontario, Canada: AAAI Press, 2012, pp. 455–463. DOI: 10.5555/2900728.2900794.

[36] M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc, "Solving the pipe network analysis problem using optimization techniques," *Management Science*, vol. 24, no. 7, pp. 747–760, 1978. DOI: 10.1287/mnsc.24.7.747.

[37] M. Collins, J. Kennington, and L. Cooper, "Multiple operating points in complex pump networks," *Journal of the Hydraulics Division*, vol. 105, no. 3, pp. 229–244, 1979. DOI: 10.1061/JYCEAJ.0005160.

[38] R. M. Colombo, G. Guerra, M. Herty, and V. Schleper, "Optimal control in networks of pipes and canals," *SIAM Journal on Control and Optimization*, vol. 48, no. 3, pp. 2032–2050, Jan. 2009, ISSN: 0363-0129. DOI: 10.1137/080716372.

[39] J. Conti, P. Holtberg, J. Diefenderfer, A. LaRose, J. T. Turnure, and L. Westfall, "International energy outlook 2016 with projections to 2040," U.S. DOE Energy Information Administration (EIA), Washington, D.C., Tech. Rep., 2016.

[40] C. Copeland, "Energy-water nexus: The water sector's energy use," Congressional Research Service, Tech. Rep., 2014.

[41] A. Costa and G. Nannicini, "RBFOpt: An open-source library for black-box optimization with costly function evaluations," *Mathematical Programming Computation*, vol. 10, no. 4, pp. 597–629, Dec. 2018, ISSN: 1867-2957. DOI: 10.1007/s12532-018-0144-7.

[42] C. D'Ambrosio, A. Lodi, S. Wiese, and C. Bragalli, "Mathematical programming techniques in water network optimization," *European Journal of Operational Research*, vol. 243, no. 3, pp. 774–788, 2015, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.12.039.

[43] G. B. Dantzig, "Origins of the simplex method," in *A History of Scientific Computing*. New York, NY, USA: Association for Computing Machinery, 1990, ch. 10, pp. 141–151, ISBN: 0201508141. [Online]. Available: 10.1145/87252.88081.

[44] C. Dawson and C. M. Mirabito, *The shallow water equations*, 2008. [Online]. Available: https://users.oden.utexas.edu/~arbogast/cam397/dawson_v2.pdf.

[45] D. De Wolf and Y. Smeers, "Mathematical properties of formulations of the gas transmission problem," SMG Preprint 94/12, Universit libre de Bruxelles, Tech. Rep., 1994.

[46] ——, "The gas transmission problem solved by an extension of the simplex algorithm," *Management Science*, vol. 46, no. 11, pp. 1454–1465, 2000. DOI: 10.1287/mnsc.46.11.1454.12087.

[47] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000, ISSN: 0045-7825. DOI: 10.1016/S0045-7825(99)00389-8.

[48] J. W. Delleur, "The evolution of urban hydrology: Past, present, and future," *Journal of Hydraulic Engineering*, vol. 129, no. 8, pp. 563–573, 2003. DOI: 10.1061/(ASCE)0733-9429(2003)129:8(563).

[49] M. W. Downton, J. Z. B. Miller, and R. A. Pielke, "Reanalysis of U.S. National Weather Service flood loss database," *Natural Hazards Review*, vol. 6, no. 1, pp. 13–22, 2005. DOI: 10.1061/(ASCE)1527-6988(2005)6:1(13).

[50] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017. DOI: 10.1137/15M1020575.

[51] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983. DOI: 10.1109/TIT.1983.1056714.

[52] Energy Solutions, *Gas Management Software*, http://www.energy-solutions.com/products/esi-gas-management-solutions, 2021.

[53] G. Engineering, *NextGen Simulation Suite*, https://www.greggeng.com/software-solutions/nextgen-simulation-suite/, 2021.

[54] M. Farrokhifar, Y. Nie, and D. Pozo, "Energy systems planning: A survey on models for integrated power and natural gas networks coordination," *Applied Energy*, vol. 262, p. 114 567, 2020, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2020.114567.

[55] I. Fent, M. Putti, C. Gregoretti, and S. Lanzoni, "Modeling shallow water flows on general terrains," *Advances in Water Resources*, vol. 121, pp. 316–332, 2018, ISSN: 0309-1708. DOI: 10.1016/j.advwatres.2017.12.017.

[56] K. Fischer, *Introduction to alpha shapes*, https://graphics.stanford.edu/courses/cs268-11-spring/handouts/AlphaShapes/as_fisher.pdf, Accessed: 2017-09-19, 2000.

[57] D. Fooladivanda and J. A. Taylor, "Energy-optimal pump scheduling and water flow," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1016–1026, 2018. DOI: 10.1109/TCNS.2017.2670501.

[58] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, Oct. 1972, ISSN: 1573-2878. DOI: 10.1007/BF00934810.

[59] B. Ghaddar, J. Naoum-Sawaya, A. Kishimoto, N. Taheri, and B. Eck, "A Lagrangian decomposition approach for the pump scheduling problem in water networks," *European Journal of Operational Research*, vol. 241, no. 2, pp. 490–501, 2015, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.08.033.

[60] M. M. Giraldo-González and J. P. Rodríguez, "Comparison of statistical and machine learning models for pipe failure modeling in water distribution networks," *Water*, vol. 12, no. 4, 2020, ISSN: 2073-4441. DOI: 10.3390/w12041153.

[61] A. M. Gleixner, H. Held, W. Huang, and S. Vigerske, "Towards globally optimal operation of water supply networks," *Numerical Algebra, Control & Optimization*, vol. 2, no. 4, pp. 695–711, 2012.

[62] R. Goldstein and W. Smith, *Water & Sustainability (Volume 4): U.S. Electricity Consumption for Water Supply & Treatment - The Next Half Century*. Electric Power Research Institute, 2002.

[63] J. Gondzio, "Interior point methods 25 years later," *European Journal of Operational Research*, vol. 218, no. 3, pp. 587–601, 2012, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2011.09.017.

[64] M. Gugat, G. Leugering, A. Martin, M. Schmidt, M. Sirvent, and D. Wintergerst, "MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems," *Computational Optimization and Applications*, vol. 70, no. 1, pp. 267–294, May 2018, ISSN: 1573-2894. DOI: 10.1007/s10589-017-9970-1.

[65] M. Hahn, S. Leyffer, and V. M. Zavala, "Mixed-integer PDE-constrained optimal control of gas networks," *Mathematics and Computer Science*, 2017.

[66] F. M. Hante, "Mixed-integer optimal control for PDEs: Relaxation via differential inclusions and applications to gas network optimization," in *Mathematical Modelling, Optimization, Analytic and Numerical Solutions*. Singapore: Springer Singapore, 2020, pp. 157–171. DOI: 10.1007/978-981-15-0928-5_7.

[67] ——, "Relaxation methods for hyperbolic PDE mixed-integer optimal control problems," *Optimal Control Applications and Methods*, vol. 38, no. 6, pp. 1103–1110, 2017. DOI: 10.1002/oca.2315.

[68] F. Hasan, A. Kargarian, and A. Mohammadi, "A survey on applications of machine learning for optimal power flow," in *2020 IEEE Texas Power and Energy Conference (TPEC)*, 2020, pp. 1–6. DOI: 10.1109/TPEC48276.2020.9042547.

[69] B. Hiller, T. Koch, L. Schewe, R. Schwarz, and J. Schweiger, "A system to evaluate gas network capacities: Concepts and implementation," *European Journal of Operational Research*, vol. 270, no. 3, pp. 797–808, 2018, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.02.035.

[70] K. Hoppmann-Baum, F. Hennings, R. Lenz, U. Gotzes, N. Heinecke, K. Spreckelsen, and T. Koch, "Optimal operation of transient gas transport networks," *Optimization and Engineering*, vol. 22, no. 2, pp. 735–781, Jun. 2021, ISSN: 1573-2924. DOI: 10.1007/s11081-020-09584-x.

[71] Z. Horváth, A. Buttinger-Kreuzhuber, A. Konev, D. Cornel, J. Komma, G. Blöschl, S. Noelle, and J. Waser, "Comparison of fast shallow-water schemes on real-world floods," *Journal of Hydraulic Engineering*, vol. 146, no. 1, p. 05 019 005, 2020. DOI: 10.1061/(ASCE)HY.1943-7900.0001657.

[72] J. Humpola and A. Fügenschuh, "A new class of valid inequalities for nonlinear network design problems," Zuse Institute Berlin, Tech. Rep., 2013.

[73] J. Humpola, A. Fügenschuh, and T. Koch, "Valid inequalities for the topology optimization problem in gas network design," *OR Spectrum*, vol. 38, no. 3, pp. 597–631, Jul. 2016, ISSN: 1436-6304. DOI: 10.1007/s00291-015-0390-2.

[74] R. Jabr, "Radial distribution load flow using conic programming," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1458–1459, 2006. DOI: 10.1109/TPWRS.2006.879234.

[75] Y. Jiang, J. Xu, Y. Sun, C. Wei, J. Wang, S. Liao, D. Ke, X. Li, J. Yang, and X. Peng, "Coordinated operation of gas-electricity integrated distribution system with multi-CCHP and distributed renewable energy sources," *Applied Energy*, vol. 211, pp. 237–248, 2018, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2017.10.128.

[76] D. R. Judi, B. A. Tasseff, R. W. Bent, and F. Pan, "Topography-based flood planning and optimization capability development report," Feb. 2014. DOI: 10.2172/1122029.

[77] D. R. Judi, D. Pasqualini, and J. D. Arnold, "Computational challenges in consequence estimation for risk assessment - numerical modelling, uncertainty quantification, and communication of results," Los Alamos National Laboratory, Tech. Rep., 2014.

[78] A. J. Kalyanapu, S. Shankar, E. R. Pardyjak, D. R. Judi, and S. J. Burian, "Assessment of GPU computational enhancement to a 2D flood model," *Environmental Modelling and Software*, vol. 26, no. 8, pp. 1009–1016, 2011. DOI: 10.1016/j.envsoft.2011.02.014.

[79] J. Kardoš, D. Kourounis, and O. Schenk, "Two-level parallel augmented schur complement interior-point algorithms for the solution of security constrained optimal power flow problems," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1340–1350, 2020. DOI: 10.1109/TPWRS.2019.2942964.

[80] S. Khan, "Sanitation and wastewater technologies in Harappa/Indus Valley civilization (ca. 2600–1900 BC)," *Evolution of Sanitation and Wastewater Technologies Through the Centuries*, vol. 25, 2014.

[81] T. Koch, B. Hiller, M. Pfetsch, and L. Schewe, *Evaluating Gas Network Capacities*. SIAM, 2015, ISBN: 978-1-611973-68-6.

[82] O. Kröger, C. Coffrin, H. Hijazi, and H. Nagarajan, "JUNIPER: An open-source nonlinear branch-and-bound solver in Julia," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, W.-J. van Hoeve, Ed., Cham: Springer International Publishing, 2018, pp. 377–386, ISBN: 978-3-319-93031-2.

[83] A. Kurganov and G. Petrova, "A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system," *Communications in Mathematical Sciences*, vol. 5, no. 1, pp. 133–160, 2007. DOI: 10.4310/CMS.2007.v5.n1.a6.

[84] G. Lanzano, E. Salzano, F. Santucci de Magistris, and G. Fabbrocino, "Seismic vulnerability of gas and liquid buried pipelines," *Journal of Loss Prevention in the Process Industries*, vol. 28, pp. 72–78, 2014, European Process Safety Pioneers, ISSN: 0950-4230. DOI: 10.1016/j.jlp.2013.03.010.

[85] Y. Liu and M. S. Mauter, "Assessing the demand response capacity of U.S. drinking water treatment plants," *Applied Energy*, vol. 267, p. 114 899, 2020, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2020.114899.

[86] H. Maier, Z. Kapelan, J. Kasprzyk, and L. Matott, "Thematic issue on evolutionary algorithms in water resources," *Environmental Modelling & Software*, vol. 69, pp. 222–225, 2015, ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2015.05.003.

[87] H. Mala-Jetmarova, N. Sultanova, and D. Savic, "Lost in optimisation of water distribution systems? A literature review of system design," *Water*, vol. 10, no. 3, 2018, ISSN: 2073-4441. DOI: 10.3390/w10030307.

[88] ——, "Lost in optimisation of water distribution systems? A literature review of system operation," *Environmental Modelling & Software*, vol. 93, pp. 209–254, 2017, ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2017.02.009.

[89] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr. 2004. DOI: 10.1007/s00158-003-0368-6.

[90] L. Mays, D. Koutsoyiannis, and A. Angelakis, "A brief history of urban water supply in antiquity," *Water Supply*, vol. 7, no. 1, pp. 1–12, Mar. 2007, ISSN: 1606-9749. DOI: 10.2166/ws.2007.001.

[91] X. Mei, P. Decaudin, and B.-G. Hu, "Fast hydraulic erosion simulation and visualization on GPU," in *15th Pacific Conference on Computer Graphics and Applications*, Maui, Hawaii, 2007, pp. 47–56, ISBN: 0769530095. DOI: 10.1109/PG.2007.15.

[92] M. A. Mirzaei, M. Nazari-Heris, B. Mohammadi-Ivatloo, K. Zare, M. Marzband, and A. Anvari-Moghaddam, "A novel hybrid framework for co-optimization of power and natural gas networks integrated with emerging technologies," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3598–3608, 2020. DOI: 10.1109/JSYST.2020.2975090.

[93] L. Mitridati, J. Kazempour, and P. Pinson, "Heat and electricity market coordination: A scalable complementarity approach," *European Journal of Operational Research*, vol. 283, no. 3, pp. 1107–1123, 2020, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2019.11.072.

[94] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Foundations and Trends in Electric Energy Systems*, vol. 4, no. 1-2, pp. 1–221, 2019, ISSN: 2332-6557. DOI: 10.1561/3100000012.

[95] N. Moosavian and M. R. Jaefarzade, "Particle swarm optimization for hydraulic analysis of water distribution systems," *Civil Engineering Infrastructures Journal*, vol. 48, no. 1, pp. 9–22, 2015, ISSN: 2322-2093. DOI: 10.7508/ceij.2015.01.002.

[96] A. Mosavi, P. Ozturk, and K.-w. Chau, "Flood prediction using machine learning models: Literature review," *Water*, vol. 10, no. 11, 2018, ISSN: 2073-4441. DOI: 10.3390/w10111536.

[97] J. Naoum-Sawaya, B. Ghaddar, E. Arandia, and B. Eck, "Simulation-optimization approaches for water pump scheduling and pipe replacement problems," *European Journal of Operational Research*, vol. 246, no. 1, pp. 293–306, 2015, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.04.028.

[98] J. Nault and F. Papa, "Lifecycle assessment of a water distribution system pump," *Journal of Water Resources Planning and Management*, vol. 141, no. 12, A4015004, 2015. DOI: 10.1061/(ASCE)WR.1943-5452.0000546.

[99] I. Navigant Consulting, "Refining estimates of water-related energy use in California," California Energy Commission, Tech. Rep., 2006.

[100] G. Ntinas, G. Zhang, V. Fragos, D. Bochtis, and C. Nikita-Martzopoulou, "Airflow patterns around obstacles with arched and pitched roofs: Wind tunnel measurements and direct simulation," *European Journal of Mechanics - B/Fluids*, vol. 43, pp. 216–229, 2014, ISSN: 0997-7546. DOI: 10.1016/j.euromechflu.2013.09.004.

[101] L. Ormsbee and T. Walski, "Darcy-Weisbach versus Hazen-Williams: No calm in West Palm," in *World Environmental and Water Resources Congress 2016*, 2016, pp. 455–464. DOI: 10.1061/9780784479865.048.

[102] A. Osiadacz, *Simulation and Analysis of Gas Networks*. Gulf Publishing Company, Houston, TX, 1987.

[103] R. A. Pielke and M. W. Downton, "Precipitation and damaging floods: Trends in the United States, 1932–97," *Journal of Climate*, vol. 13, no. 20, pp. 3625–3637, Oct. 2000, ISSN: 0894-8755. DOI: 10.1175/1520-0442(2000)013<3625:PADFTI>2.0.CO;2.

[104] PJM Interconnection, "Analysis of operational events and market impacts during the January 2014 cold weather events," Tech. Rep., 2014.

[105] J. E. Price and J. Goodin, "Reduced network modeling of WECC as a market design prototype," in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6. DOI: 10.1109/PES.2011.6039476.

[106] V. Priymak and T. Miyazaki, "Accurate Navier–Stokes investigation of transitional and turbulent flows in a circular pipe," *Journal of Computational Physics*, vol. 142, no. 2, pp. 370–411, 1998, ISSN: 0021-9991. DOI: 10.1006/jcph.1998.5931.

[107] D. Proverbs, S. Mambretti, C. Brebbia, and D. Wrachien, *Flood Recovery Innovation and Response III*. Southampton, UK: WIT Press, 2012.

[108] I. Quesada and I. E. Grossmann, "An LP/NLP based branch and bound algorithm for convex MINLP optimization problems," *Computers & Chemical Engineering*, vol. 16, no. 10, pp. 937–947, 1992, An International Journal of Computer Applications in Chemical Engineering, ISSN: 0098-1354. DOI: 10.1016/0098-1354(92)80028-8.

[109] M. Ragheb, *Fluid mechanics, Euler and Bernoulli equations*, 2013. [Online]. Available: http://magdiragheb.com/NPRE%20475%20Wind%20Power%20Systems/Fluid%20Mechanics%20Euler%20and%20Bernoulli%20Equations.pdf.

[110] A. U. Raghunathan, "Global optimization of nonlinear network design," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 268–295, 2013. DOI: 10.1137/110827387.

[111] S. Rajagopalan, "Design and maintenance planning problems in commodity distribution and chemical site networks," Ph.D. dissertation, 2018, p. 187, ISBN: 978-0-355-84163-3.

[112] *Review of February 2021 extreme cold weather event*, Feb. 24, 2021. [Online]. Available: http://www.ercot.com/content/wcm/key_documents_lists/225373/Urgent_Board_of_Directors_Meeting_2-24-2021.pdf (visited on 02/24/2021).

[113] N. Rhodes, D. M. Fobes, C. Coffrin, and L. Roald, "PowerModelsRestoration.jl: An open-source framework for exploring power network restoration algorithms," *Electric Power Systems Research*, vol. 190, p. 106 736, 2021, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2020.106736.

[114] N. Rhodes, L. Ntaimo, and L. Roald, "Balancing wildfire risk and power outages through optimized power shut-offs," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3118–3128, 2021. DOI: 10.1109/TPWRS.2020.3046796.

[115] R. Z. Ríos-Mercado and C. Borraz-Sánchez, "Optimization problems in natural gas transportation systems: A state-of-the-art review," *Applied Energy*, vol. 147, pp. 536–555, 2015. DOI: 10.1016/j.apenergy.2015.03.017.

[116] L. A. Rossman, "EPANET 2: Users manual," Tech. Rep., 2000.

[117] P. H. Rydlund, "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," U.S. Department of the Interior and U.S. Geological Survey, Tech. Rep., 2006.

[118] H. Sahebi, S. Nickel, and J. Ashayeri, "Strategic and tactical mathematical programming models within the crude oil supply chain context – a review," *Computers & Chemical Engineering*, vol. 68, pp. 56–77, 2014, ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2014.05.008.

[119] E. Salomons and M. Housh, "A practical optimization scheme for real-time operation of water distribution systems," *Journal of Water Resources Planning and Management*, vol. 146, no. 4, p. 04 020 016, 2020. DOI: 10.1061/(ASCE)WR.1943-5452.0001188.

[120] C. B. Sánchez, R. Bent, S. Backhaus, S. Blumsack, H. Hijazi, and P. van Hentenryck, "Convex optimization for joint expansion planning of natural gas and power systems," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 2536–2545. DOI: 10.1109/HICSS.2016.317.

[121] K. T. Sanders and M. E. Webber, "Evaluating the energy consumed for water use in the United States," *Environmental Research Letters*, vol. 7, no. 3, p. 034 034, Sep. 2012. DOI: 10.1088/1748-9326/7/3/034034.

[122] P. Sayers, L. Yuanyuan, G. Galloway, E. Penning-Rowsell, S. Fuxin, W. Kang, C. Yiwei, and T. Le Quesne, *Flood Risk Management: A Strategic Approach*. Asian Development Bank, GIWP, UNESCO, and WWF-UK, 2013, ISBN: 978-92-3-001159-8.

[123] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010. DOI: 10.1017/S0022112010001217.

[124] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent, "GasLib - a library of gas network instances," *Data*, vol. 2, no. 4, 2017, ISSN: 2306-5729. DOI: 10.3390/data2040040.

[125] M. Schmidt, M. C. Steinbach, and B. M. Willert, "High detail stationary optimization models for gas networks: Validation and results," *Optimization and Engineering*, vol. 17, no. 2, pp. 437–472, Jun. 2016, ISSN: 1573-2924. DOI: 10.1007/s11081-015-9300-3.

[126] M. K. Singh and V. Kekatos, "Optimal scheduling of water distribution systems," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 711–723, 2020. DOI: 10.1109/TCNS.2019.2939651.

[127] R. Sojka, D. Bjorneberg, and J. Entry, "Irrigation: An historical perspective," in *Encyclopedia of Soil Science*, R. Lal, Ed., New York: Marcel Dekker, Inc., 2002.

[128] R. Storn and K. Price, "Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997, ISSN: 1573-2916. DOI: 10.1023/A:1008202821328.

[129] K. Sundar, C. Coffrin, H. Nagarajan, and R. Bent, "Probabilistic N-k failure-identification for power systems," *Networks*, vol. 71, no. 3, pp. 302–321, 2018. DOI: 10.1002/net.21806.

[130] K. Sundar, M. Vallem, R. Bent, N. Samaan, B. Vyakaranam, and Y. Makarov, "N-k failure analysis algorithm for identification of extreme events for cascading outage pre-screening process," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*, 2019, pp. 1–5. DOI: 10.1109/PESGM40551.2019.8973425.

[131] B. Tasseff, R. Bent, and C. Coffrin, *WaterModels.jl*, https://github.com/lanl-ansi/WaterModels.jl, 2019.

[132] B. Tasseff, R. Bent, and P. Van Hentenryck, "Optimal flood mitigation over flood propagation approximations," in *Integration of AI and OR Techniques in Constraint Programming*, C.-G. Quimper, Ed., Cham: Springer International Publishing, 2016, pp. 358–373, ISBN: 978-3-319-33954-2. DOI: 10.1007/978-3-319-33954-2_26.

[133] ——, "Optimization of structural flood mitigation strategies," *Water Resources Research*, vol. 55, no. 2, pp. 1490–1509, 2019. DOI: 10.1029/2018WR024362.

[134] B. Tasseff, C. Coffrin, A. Wächter, and C. Laird, *Exploring benefits of linear solver parallelism on modern nonlinear optimization applications*, 2019. arXiv: 1909.08104 [math.OC].

[135] B. Tasseff and D. Judi, *NUFLOOD, version 1.x*, Aug. 2016. DOI: 10.11578/dc.20171025.1807.

[136] A. C. Telea, *Data Visualization: Principles and Practice*. CRC Press, 2014. DOI: 10.1201/b17217.

[137] E. Todini and S. Pilati, "A gradient algorithm for the analysis of pipe networks," in *Computer Applications in Water Supply: Vol. 1 – Systems Analysis and Simulation*. GBR: Research Studies Press Ltd., 1988, ch. 1, pp. 1–20, ISBN: 0471917834.

[138] *Evaluating the Energy Intensity of the U.S. Public Water System*, vol. ASME 2011 5th International Conference on Energy Sustainability, Parts A, B, and C, Energy Sustainability, Aug. 2011, pp. 1735–1748. DOI: 10.1115/ES2011-54165.

[139] B. Ulanicki, J. Kahler, and B. Coulbeck, "Modeling the efficiency and power characteristics of a pump group," *Journal of Water Resources Planning and Management*, vol. 134, no. 1, pp. 88–93, 2008. DOI: 10.1061/(ASCE)0733-9496(2008)134:1(88).

[140] D. Verleye and E.-H. Aghezzaf, "Optimising production and distribution operations in large water supply networks: A piecewise linear optimisation approach," *International Journal of Production Research*, vol. 51, no. 23-24, pp. 7170–7189, 2013. DOI: 10.1080/00207543.2013.850550.

[141] B. S. Vieira, S. F. Mayerle, L. M. Campos, and L. C. Coelho, "Optimizing drinking water distribution system operations," *European Journal of Operational Research*, vol. 280, no. 3, pp. 1035–1050, 2020, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2019.07.060.

[142] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006, ISSN: 1436-4646. DOI: 10.1007/s10107-004-0559-y.

[143] C. Wang, W. Wei, J. Wang, L. Bai, Y. Liang, and T. Bi, "Convex optimization based distributed optimal gas-power flow calculation," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 3, pp. 1145–1156, 2018. DOI: 10.1109/TSTE.2017.2771954.

[144] J. L. Weisdorf, "From foraging to farming: Explaining the Neolithic revolution," *Journal of Economic Surveys*, vol. 19, no. 4, pp. 561–586, 2005. DOI: 10.1111/j.0950-0804.2005.00259.x.

[145] F. Wu, H. Nagarajan, A. Zlotnik, R. Sioshansi, and A. M. Rudkevich, "Adaptive convex relaxations for gas pipeline network optimization," in *2017 American Control Conference (ACC)*, 2017, pp. 4710–4716. DOI: 10.23919/ACC.2017.7963683.

[146] Q. Wu, Z. Zhao, L. Liu, D. E. Granger, H. Wang, D. J. Cohen, X. Wu, M. Ye, O. Bar-Yosef, B. Lu, J. Zhang, P. Zhang, D. Yuan, W. Qi, L. Cai, and S. Bai, "Outburst flood at 1920 BCE supports historicity of China's Great Flood and the Xia dynasty," *Science*, vol. 353, no. 6299, pp. 579–582, 2016, ISSN: 0036-8075. DOI: 10.1126/science.aaf0842.

[147] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast ac optimal power flow," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1–6. DOI: 10.1109/SmartGridComm47815.2020.9303008.

[148] L. Zeghadnia, J. L. Robert, and B. Achour, "Explicit solutions for turbulent flow friction factor: A review, assessment and approaches classification," *Ain Shams Engineering Journal*, vol. 10, no. 1, pp. 243–252, 2019, ISSN: 2090-4479. DOI: 10.1016/j.asej.2018.10.007.

[149] J. Zhang and D. Zhu, "A bilevel programming method for pipe network optimization," *SIAM Journal on Optimization*, vol. 6, no. 3, pp. 838–857, 1996. DOI: 10.1137/S1052623493260696.

[150] F. Zheng, A. C. Zecchin, J. P. Newman, H. R. Maier, and G. C. Dandy, "An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 773–791, 2017. DOI: 10.1109/TEVC.2017.2682899.

[151] A. Zlotnik, L. Roald, S. Backhaus, M. Chertkov, and G. Andersson, "Coordinated scheduling for interdependent electric power and natural gas infrastructures," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 600–610, 2017. DOI: 10.1109/TPWRS.2016.2545522.