ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Lemont, IL 60439

Mixed-Integer PDE-Constrained Optimal Control of Gas Networks

Mirko Hahn, Sven Leyffer, and Victor M. Zavala

Mathematics and Computer Science Division

Preprint ANL/MCS-P7095-0817

August 10, 2017

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

Contents

Mixed-Integer PDE-Constrained Optimal Control of Gas Networks*

Mirko Hahn†, Sven Leyffer‡, and Victor M. Zavala§,

August 10, 2017

Abstract

We develop a mixed-integer optimal control model with partial differential equation constraints for gas transport networks, designed for planning extreme state transitions, such as flow reversals. We use binary control variables to model different configurations of compressor stations. We model the flow of gas using a simplified variant of the Euler equations, which we discretize using a finite volume scheme, which obeys conservation of mass and average impulse independent of the direction of flow. The resulting large-scale mixed-integer nonlinear optimization is difficult to solve by using standard branch-and-bound solvers, and we propose several heuristics that allow us to solve realistic instances. We compare the performance of several solution schemes on ten test instances.

Keywords: Gas networks, flow reversal, PDE constraints, mixed-integer nonlinear optimization.

AMS-MSC2000: 35L60, 35L65, 90C06, 90C11, 90C30

1 Introduction: Background and Motivation

Since the early 2000s, the world has seen a dramatic increase in natural gas as an energy carrier. In the United States, where production of natural gas has increased by more than a third since the year 2000 according to the U.S. Energy Information Administration, natural gas is now the foremost energy carrier in electricity production; and consumption for other industrial uses is steadily increasing [18]. Meanwhile, a movement to renewable energy sources in Europe and parts of Asia has led to increased use of gas as an interim energy carrier, emergency energy reserve, and potential means of energy storage [6].

*Preprint ANL/MCS-P7095-0817

†Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, hahnm@anl.gov.

‡Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, leyffer@anl.gov.

§Chemical and Biological Engineering, University of Wisconsin Madison, Madison, WI 53706, victor.zavala@wisc.edu.

This development is not without risks, since greater demand for natural gas is accompanied by correspondingly greater stresses on natural gas distribution networks. While supply has increased in recent years, the consumption of consumer-grade natural gas [14, tbl. 1] still regularly exceeds production based on weather and other external factors. Since natural gas transport is significantly slower than long-distance electricity transport, supply network operators depend on an ability to anticipate demands and act accordingly. Failure to do so in situations where demands fluctuate severely can lead to forced electrical outages and production losses [1].

In order to minimize such losses, various efforts are currently under way to develop mathematical descriptions of natural gas transport networks and derive insights into their optimal operation from these descriptions. The range of prior work by other researchers is too broad to properly address fully here and ranges from detailed descriptions of the operation of a single compressor station [9,15] to more ambitious work attempting to optimize controls for entire networks with integer switches that inhabit a steady state [13], approximate optimal transient controls on larger networks with integer switches [8], and even find optimal transient controls on simplified networks without integer switches with [21] and without [22] uncertainty.

In this paper, we develop a mixed-integer optimal control model for gas transport networks that do not inhabit a stationary state, using partial differential equation (PDE) constraints to model the flow of gas. Our goal in designing this model is to find control configurations that satisfy given, deterministically fluctuating demand patterns at minimal network operation cost. In doing so, we attempt to minimize the number of restrictions we place on control inputs. For instance, we avoid prior assumptions on fixed direction of flow such as those made in [21] and simplifications made concerning the operational constraints of compressor stations such as those found in [22]. In contrast to approximate solution methods as proposed by [8], we attempt to solve the resulting optimization problems to integer optimality and propose methods of accelerating this process.

Section 2 introduces our model formulation, which is based on [21] and makes simplifying assumptions similar to those made by [8, 13, 22]. We avoid assumptions on fixed flow directions made in [21] by using a finite volume discretization scheme. In designing our model, we follow a modular approach with an eye to future efforts to increase model fidelity, scale the method to larger networks, and use parallel solvers. Section 3 proposes several modifications designed to lessen the performance impact of large numbers of integers in branch-and-bound solvers. Possible extensions to increase the potential for future parallelization are discussed. Section 4 presents two test problems to which we apply our modified solvers to estimate the impact of our modifications. We find that performance can be increased substantially by modifying tree-search methods and that approximate solvers may benefit from branching rules that take into account the structure of the problem. Section 5 summarizes our findings and proposes future avenues of research.

Notation

We adopt a number of useful notational conventions to improve the presentation of our optimization model.

- Uppercase calligraphic letters (e.g. $\mathcal{A}$) denote sets.
- i, j, k, l denote indices.
- K, L (with appropriate subscripts) denote break points in index sequences.
- m, n, N denote integer sizes (e.g., of vectors and matrices).
- v, w denote vertices of the network graph.
- Parenthesized superscripts (e.g., $p^{(v)}$) denote an associated network element.
- Non cursive roman subscripts (e.g., p_{in}) are symbolic.
- A horizontal bar (e.g., $\bar{p}$) denotes an averaged quantity.

When discussing individual model elements, we attempt to introduce index variables as late as possible. For instance, when discussing the dynamics of a single pipe (v, w), we omit the superscript $\cdot^{(v,w)}$ denoting the particular pipe in question unless a distinction between variables and constraints associated with different pipes becomes material to the discussion.

2 Optimization and Control Model

In this section, we describe our model of gas flow through a gas transport network. Section 2.1 describes the structure of a network and its various components, which we will later model as individual units with interactions through shared optimization variables. Section 2.2 briefly introduces the time grids we will use when discretizing the partial differential equations, which are introduced in Section 2.3. We use a simplified version of the one-dimensional Euler equations, which is widely used for large-scale gas network models. While we briefly sketch the derivation of these equations from the original Euler equations and the assumptions made to justify simplifications, our primary focus will be the PDE discretization described in Section 2.3.3. Section 2.4 focuses on the modeling of junctions and compressor stations, which constitute our primary means of controlling flow within the network. We adopt a realistic compressor model that can be expressed solely in terms of linear constraints on three continuous variables and an auxiliary binary indicator variable. Discretized versions of the appropriate constraints are described in Section 2.4.2. Section 2.5 aggregates the constraints and objective functions derived in the preceding sections into a complete discretized MINLP (2.29). The issues we will face in solving this problem and our approach to solving them are summarized in Section 2.6.

2.1 Network Structure

Gas transport networks are complex systems. However, they can be thought of as combinations of simpler components that can be described in isolation from one another and interact through shared sets of variables. The most commonly modeled network components are *pipes*, *junctions*, *compressor machines*, and various types of *valves*. For our discussion, we will omit the modeling of valves and assume that valves and compressor machines occur only in larger assemblies called *compressor stations*. We assume such compressor stations to occur only at junctions and will discuss them as part of our discussion on junction modeling in Section 2.4.

Our primary network components are therefore pipes and junctions. The structure of a network can be described by using a directed graph $G = (\mathcal{V}, \mathcal{A})$ where arcs $a \in \mathcal{A} \subset \{(v, w) \in \mathcal{V}^2 \mid v \neq w\}$ represent pipes and vertices $v \in \mathcal{V}$ represent junctions at which these pipes meet and interact. While we do not assume a fixed direction of flow, the direction of an arc indicates the nominal direction of positive flow in the corresponding pipe. A given pipe $(v, w) \in \mathcal{A}$ leads from its nominal *inlet junction* $v \in \mathcal{V}$ to its nominal *outlet junction* $w \in \mathcal{V}$. We denote the neighborhood of $v \in \mathcal{V}$ by

$$\mathcal{N}(v) := \{w \in \mathcal{V} \mid (v, w) \in \mathcal{A} \vee (w, v) \in \mathcal{A}\}.$$

Junctions also represent locations where the network may exchange gas with the environment. To this end, some junctions are marked as *supply junctions* $v \in \mathcal{S}$ that inject gas into the network or *demand junctions* $v \in \mathcal{D}$ that withdraw gas from it. For reasons that we will elaborate on in Section 2.4, we assume that, for supply junctions $v \in \mathcal{S}$,

$$\left|\mathcal{N}(v)\right| = 1.$$

To simplify notation when introducing compressor stations in Section 2.4, we also prohibit antiparallel arcs, that is,

$$(v, w) \in \mathcal{A} \;\Rightarrow\; (w, v) \notin \mathcal{A} \quad \forall v, w \in \mathcal{V}.$$

2.2 Time Discretization

To keep the number of integer variables low, we will distinguish between two time grids when discretizing our PDE system: the *integer control grid* and the *sampling grid*. The sampling grid is the finer of the two and serves as a time grid for both the discretization of continuous controls and the discretization of state variables in the optimization problem. Its name is derived from the fact that, depending on the choice of simulation method and the presence of adaptive error control, the *simulation grid* may actually be finer than the sampling grid. In this case, we enforce the constraints of our optimization problem on average over each interval of the sampling grid. Thus, state variables can be said to be *sampled* over intervals of the sampling grid.

For this discussion, we assume that a finite time horizon $[0, T] \neq \emptyset$ and a number of sampling intervals $N_\mathrm{t} \in \mathbb{N}$ are given. For $i \in \{0, \ldots, N_\mathrm{t}\}$, the points of the sampling grid are given by $t_i := \frac{iT}{N_\mathrm{t}}$. The resolution of the sampling grid is determined by its time step $\Delta t := \frac{T}{N_\mathrm{t}}$.

The sampling grid is assumed to be a refinement of the integer control grid, which we also assume to be equidistant; that is, there is a step $\delta \in \{1, \ldots N_\mathrm{t}\}$ with $\delta \mid N_\mathrm{t}$ such that the grid points of the integer control grid are given by t_{K_i} with

$$K_i := i\delta \quad \forall i \in \left\{0, \ldots, \frac{N_\mathrm{t}}{\delta}\right\}.$$

For ease of notation, we define $N_\mathrm{w} := \frac{N_\mathrm{t}}{\delta}$. Further, for every sampling interval $j \in \{1, \ldots, N_\mathrm{t}\}$ we denote the index of the associated integer control interval by

$$J_j := \left\lfloor \frac{i-1}{\delta} \right\rfloor + 1.$$

2.3 Pipes and Flow Modeling

Single-phase subsonic gas flow is generally modeled by using the Euler equations [8, 11, 13, 16, 21, 22]. We adopt the following simplified version of the one-dimensional Euler equations, suggested in [16] and adopted for transient stochastic optimal control of gas networks in [21]:

$$\partial_t \begin{pmatrix} p \\ q \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & \frac{c^2}{A} \\ A & 0 \end{pmatrix}}_{=:B} \partial_x \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} 0 \\ -\alpha \cdot \frac{|q|q}{p} \end{pmatrix}, \tag{2.1}$$

where p denotes the average pressure across the pipe's cross section, q denotes the mass of the gas passing through the pipe's cross section in a given unit of time, A denotes the cross-sectional area of the pipe, and c^2 denotes the square of the speed of sound inside the gas. The coefficient α weighs the friction term, which we will discuss in Section 2.3.1.

The PDE system (2.1) is derived under the assumption that the gas is isothermal, which eliminates the necessity of modeling the distribution of heat. In addition, we assume that gas flows at sufficiently low speed and follows the following variant of the ideal gas law:

$$p = z\frac{R}{M}T\rho, \tag{2.2}$$

where z is the gas' *compressibility factor*, M is its *molar mass*, T is its *temperature*, and

$$R = 8{,}314.4598\,\frac{\mathrm{J}}{\mathrm{kmol\,K}}$$

is the *universal gas constant*.

We also assume that global bounds exist on both pressure and flow rate (see Section 2.4), given by $p \in [\check{p}, \hat{p}] \subset \mathbb{R}_{>0}$ and $q \in [-\hat{q}, \hat{q}] \subset \mathbb{R}$, respectively.

2.3.1 Source Term

The source term appearing in (2.1) represents an external force acting on the gas due to friction at the inner wall of the pipe. The coefficient α is given by

$$\alpha := \frac{\lambda c^2}{2AD},$$

where D is the pipe's *inner diameter*, $A = \frac{\pi}{4}D^2$ is its cross-sectional area, and λ is the *Darcy friction coefficient*, which can be derived from the ratio between D and the pipe wall's *effective roughness height* ϵ by solving an approximation of the Colebrook-White equation [4] originally proposed by the Swanee and Jain [19], under the assumption that the gas's Reynolds number is high in regimes where friction is relevant:

$$\lambda = \left(2\log\left(\frac{3.7D}{\epsilon}\right)\right)^{-2}.$$

We assume that all pipes are located on an isosurface of the gravitational potential, meaning that gravity does not affect the movement of gas, although this assumption can be avoided by adding a linear source term depending on the local incline and the pressure of the gas.

To make the solutions of the discretized system twice differentiable with respect to the boundary controls, we make the following approximation for sufficiently large $M > 0$:

$$-\alpha \cdot \frac{|q|q}{p} \approx -\frac{2\alpha}{\pi} \arctan(Mq)\frac{q^2}{p}.$$

Note that the scaled arctangent deviates from the signum function only for small q, where q^2 can be assumed to make the error negligible unless p approaches zero. To simplify notation in subsequent sections, we write

$$\varphi_{\alpha,M}(p,q) := -\frac{2\alpha}{\pi} \arctan(Mq)\frac{q^2}{p}.$$

2.3.2 Boundary Conditions and Existence of Solutions

The existence and uniqueness of continuously differentiable solutions over a limited time horizon (which need not span the entire time horizon of the problem) in the presence of nonlinear source terms are guaranteed by the results of [12] assuming an appropriate set of boundary conditions.

According to [12], the number and location of the boundary conditions are dictated by the eigenvalues and left eigenvectors of the coefficient matrix B, which has two distinct real eigenvalues $\lambda_{1/2} = \mp c$. The corresponding left eigenvectors define the Riemann invariants of the homogeneous problem:

$$\xi_1 = \sqrt{\frac{A^2}{c^2+A^2}}p - \sqrt{\frac{c^2}{c^2+A^2}}q \text{ and } \xi_2 = \sqrt{\frac{A^2}{c^2+A^2}}p + \sqrt{\frac{c^2}{c^2+A^2}}q. \tag{2.3}$$

Note that this linear transformation of (p,q)-space is bijective with

$$p = \frac{1}{2}\sqrt{\frac{c^2+A^2}{A^2}}(\xi_2+\xi_1) \text{ and } q = \frac{1}{2}\sqrt{\frac{c^2+A^2}{c^2}}(\xi_2-\xi_1). \tag{2.4}$$

In transformed space, the simplified Euler equations (2.1) with the approximations described in Section 2.3.1 take the following form:

$$\partial_t\xi_1 - c\cdot\partial_x\xi_1 = -\sqrt{\frac{c^2}{c^2+A^2}}\varphi_{\alpha,M}\left(\frac{1}{2}\sqrt{\frac{c^2+A^2}{A^2}}(\xi_2+\xi_1), \frac{1}{2}\sqrt{\frac{c^2+A^2}{c^2}}(\xi_2-\xi_1)\right), \tag{2.5a}$$

$$\partial_t\xi_2 + c\cdot\partial_x\xi_2 = \sqrt{\frac{c^2}{c^2+A^2}}\varphi_{\alpha,M}\left(\frac{1}{2}\sqrt{\frac{c^2+A^2}{A^2}}(\xi_2+\xi_1), \frac{1}{2}\sqrt{\frac{c^2+A^2}{c^2}}(\xi_2-\xi_1)\right), \tag{2.5b}$$

meaning that the state components ξ_1 and ξ_2 travel through the pipe in opposite directions at constant speed c while being converted into one another due to the source term.

According to [12], it is sufficient to impose a Dirichlet boundary condition on ξ_1 at $x = L$ and a Dirichlet boundary condition on ξ_2 at $x = 0$, where either boundary condition may involve the current boundary value of the other Riemann invariant. Specifically, we may choose either pressure or flow boundary conditions for either side, but not both. Here, we specify boundary

pressure on both sides of the pipe, since this is convenient for modeling compressor stations on the inlet and outlet of each pipe:

$$\begin{aligned} p(0,x) = p_0(x) \text{ and } q(0,x) = q_0(x) \quad & \forall x \in [0,L], \\ p(t,0) = p_{\text{in}}(t) \text{ and } p(t,L) = p_{\text{out}}(t) \quad & \forall t \in [0,T], \end{aligned}$$

where it is assumed that $p_0(0) = p_{\text{in}}(0)$ and $p_0(L) = p_{\text{out}}(0)$. Under these assumptions, it follows from [12] that the system (2.1) has a unique solution within a neighborhood of $t = 0$.

2.3.3 Discretization

Next, we describe our discretization of the PDE system (2.1). We discretize the system by first transforming it into a system of ordinary differential equations (ODEs) by means of a finite volume method derived from Godunov's scheme [7] and then discretizing that ODE system using a numerical integration method. This approach allows us to easily extend the model to include adaptive timestepping methods later.

Finite volume methods in general are based on the observation that, for continuously differentiable p and q, the following equality holds due to the divergence theorem:

$$\int_{x_1}^{x_2} \left(\partial_t \begin{pmatrix} p \\ q \end{pmatrix} + B \cdot \partial_x \begin{pmatrix} p \\ q \end{pmatrix} \right) \mathrm{d}x = \partial_t \left(\int_{x_1}^{x_2} \begin{pmatrix} p \\ q \end{pmatrix} \mathrm{d}x \right) + B \cdot \begin{pmatrix} p|_{x=x_2} - p|_{x=x_1} \\ q|_{x=x_2} - q|_{x=x_1} \end{pmatrix}.$$

We divide each pipe by an equidistant grid

$$x_i := i \underbrace{\frac{L}{N_{\text{x}}}}_{=:\Delta x} \quad \forall i \in \{0, \dots, N_{\text{x}}\}$$

and denote by $\bar{p}_i$ and $\bar{q}_i$ for $i \in \{1, \dots, N_{\text{x}}\}$, the averages of p and q over the ith spatial subinterval $[x_{i-1}, x_i]$:

$$\bar{p}_i(t) := \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} p(t,x) \, \mathrm{d}x, \quad \bar{q}_i(t) := \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} q(t,x) \, \mathrm{d}x.$$

Accepting an error incurred by replacing the values of pressure and flow rate in the source term with their averages, we can rewrite (2.1) as

$$\frac{\mathrm{d}\bar{p}_i}{\mathrm{d}t} = \frac{c^2}{A} \cdot \frac{q|_{x=x_{i-1}} - q|_{x=x_i}}{\Delta x} \qquad \forall i \in \{1, \dots, N_{\text{x}}\}, \tag{2.7a}$$

$$\frac{\mathrm{d}\bar{q}_i}{\mathrm{d}t} = A \cdot \frac{p|_{x=x_{i-1}} - p|_{x=x_i}}{\Delta x} + \varphi_{\alpha,M}(\bar{p}_i, \bar{q}_i) \quad \forall i \in \{1, \dots, N_{\text{x}}\}. \tag{2.7b}$$

To complete the discretized ODE system, we must approximate the state function values at the grid points x_i. This approximation is made by solving the homogeneous problem locally for small

time horizons. The homogeneous problem can be solved exactly by mixing the Riemann invariants from the subintervals immediately incident to the grid point:

$$\begin{aligned} p|_{x=x_i} &= \frac{1}{2}\sqrt{\frac{c^2+A^2}{A^2}}\left(\bar{\xi}_{2,i} + \bar{\xi}_{1,i+1}\right) \\ &= \frac{1}{2}\left(\bar{p}_i + \bar{p}_{i+1}\right) + \frac{c}{2A}\left(\bar{q}_i - \bar{q}_{i+1}\right), \\ q|_{x=x_i} &= \frac{1}{2}\sqrt{\frac{c^2+A^2}{c^2}}\left(\bar{\xi}_{2,i} - \bar{\xi}_{1,i+1}\right) \\ &= \frac{A}{2c}\left(\bar{p}_i - \bar{p}_{i+1}\right) + \frac{1}{2}\left(\bar{q}_i + \bar{q}_{i+1}\right). \end{aligned}$$

Note that, in order for these terms to be independent of the states in other subintervals, we have to impose an upper bound on the time step that is equal to the time it takes a pressure wave to cross an entire subinterval. This leads to the following Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t \leq \frac{\Delta x}{c}. \tag{2.8}$$

For the boundary grid points $i = 0$ and $i = N_\mathrm{x}$, we must substitute $\bar{\xi}_{1,N_\mathrm{x}+1}$ and $\bar{\xi}_{2,0}$ with appropriate expressions derived from the boundary conditions. For pressure, these boundary values are self-evident. For flow rates, however, the calculation is slightly more complicated. Since $p|_{x=x_0} = p_\mathrm{in}$ and $p|_{x=x_{N_\mathrm{x}}} = p_\mathrm{out}$, we find that

$$\begin{aligned} \bar{\xi}_{2,0} &:= 2\sqrt{\frac{A^2}{c^2+A^2}}p_\mathrm{in} - \bar{\xi}_{1,1}, \\ \bar{\xi}_{1,N_\mathrm{x}+1} &:= 2\sqrt{\frac{A^2}{c^2+A^2}}p_\mathrm{out} - \bar{\xi}_{2,N_\mathrm{x}} \end{aligned}$$

are the only choices that satisfy the boundary conditionseq:boundary-conditions, meaning that

$$\begin{aligned} q|_{x=x_0} &= \frac{A}{c}\left(p_\mathrm{in} - \bar{p}_1\right) + \bar{q}_1, \\ q|_{x=x_{N_\mathrm{x}}} &= \frac{A}{c}\left(\bar{p}_{N_\mathrm{x}} - p_\mathrm{out}\right) + \bar{q}_{N_\mathrm{x}}. \end{aligned}$$

By entering these terms into the spatially discretized Euler equationseq:discrete-euler and applying a numerical time integration scheme, the explicit Euler method in our case, to solve the resulting ODE system, we can derive a suitable system of equality constraints for a given pipe $a \in \mathcal{A}$.

To simplify notation, we split the right-hand side of the ODE into a linear vector function $F^{(a)} \colon \mathbb{R}^{N_\mathrm{x}^{(a)}} \times \mathbb{R}^{N_\mathrm{x}^{(a)}} \times \mathbb{R}^2 \to \mathbb{R}^{N_\mathrm{x}^{(a)}} \times \mathbb{R}^{N_\mathrm{x}^{(a)}}$ given by

$$F_i^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_\mathrm{in}^{(a)}, p_\mathrm{out}^{(a)}\right) := \frac{c}{2\Delta x^{(a)}}\left(\bar{p}_{i-1}^{(a)} - 2\bar{p}_i^{(a)} + \bar{p}_{i+1}^{(a)}\right) + \frac{c^2}{2A^{(a)}\Delta x^{(a)}}\left(\bar{q}_{i-1}^{(a)} - \bar{q}_{i+1}^{(a)}\right), \tag{2.9a}$$

$$F_{N_\mathrm{x}^{(a)}+i}^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_\mathrm{in}^{(a)}, p_\mathrm{out}^{(a)}\right) := \frac{A^{(a)}}{2\Delta x^{(a)}}\left(\bar{p}_{i-1,j}^{(a)} - \bar{p}_{i+1,j}^{(a)}\right) + \frac{c}{2\Delta x^{(a)}}\left(\bar{q}_{i-1,j}^{(a)} - 2\bar{q}_{i,j}^{(a)} + \bar{q}_{i+1,j}^{(a)}\right), \tag{2.9b}$$

for all $i \in \{2, \dots, N_{\mathrm{x}}^{(a)} - 1\}$ and the following terms for the boundary cells:

$$F_1^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := \frac{c}{2\Delta x^{(a)}}\left(2p_{\mathrm{in}}^{(a)} - 3\bar{p}_1^{(a)} + \bar{p}_2^{(a)}\right) + \frac{c^2}{2A^{(a)}\Delta x^{(a)}}\left(\bar{q}_1^{(a)} - \bar{q}_2^{(a)}\right), \tag{2.9c}$$

$$F_{N_{\mathrm{x}}^{(a)}+1}^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := \frac{A^{(a)}}{2\Delta x^{(a)}}\left(2p_{\mathrm{in}}^{(a)} - \bar{p}_1^{(a)} - \bar{p}_2^{(a)}\right) + \frac{c}{2\Delta x^{(a)}}\left(-\bar{q}_1^{(a)} + \bar{q}_2^{(a)}\right), \tag{2.9d}$$

$$F_{N_{\mathrm{x}}^{(a)}}^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := \frac{c}{2\Delta x^{(a)}}\left(\bar{p}_{N_{\mathrm{x}}^{(a)}-1}^{(a)} - 3\bar{p}_{N_{\mathrm{x}}^{(a)}}^{(a)} + 2p_{\mathrm{out}}^{(a)}\right) + \frac{c^2}{2A^{(a)}\Delta x^{(a)}}\left(\bar{q}_{N_{\mathrm{x}}^{(a)}-1}^{(a)} - \bar{q}_{N_{\mathrm{x}}^{(a)}}^{(a)}\right), \tag{2.9e}$$

$$F_{2N_{\mathrm{x}}^{(a)}}^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := \frac{A^{(a)}}{2\Delta x^{(a)}}\left(\bar{p}_{N_{\mathrm{x}}^{(a)}-1}^{(a)} + \bar{p}_{N_{\mathrm{x}}^{(a)}}^{(a)} - 2p_{\mathrm{out}}^{(a)}\right) + \frac{c}{2\Delta x^{(a)}}\left(\bar{q}_{N_{\mathrm{x}}^{(a)}-1}^{(a)} - \bar{q}_{N_{\mathrm{x}}^{(a)}}^{(a)}\right). \tag{2.9f}$$

The nonlinear portion of the right-hand side of the ODE system is given by the vector function $G^{(a)} \colon \mathbb{R}^{N_{\mathrm{x}}^{(a)}} \times \mathbb{R}^{N_{\mathrm{x}}^{(a)}} \times \mathbb{R}^2 \to \mathbb{R}^{N_{\mathrm{x}}^{(a)}} \times \mathbb{R}^{N_{\mathrm{x}}^{(a)}}$ with

$$G_i^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := 0 \qquad \forall i \notin \{N_{\mathrm{x}}^{(a)} + 1, \dots, 2N_{\mathrm{x}}^{(a)}\}, \tag{2.10a}$$

$$G_{N_{\mathrm{x}}^{(a)}+i}^{(a)}\left(\bar{p}^{(a)}, \bar{q}^{(a)}, p_{\mathrm{in}}^{(a)}, p_{\mathrm{out}}^{(a)}\right) := \varphi_{\alpha^{(a)},M}\left(\bar{p}_i^{(a)}, \bar{q}_i^{(a)}\right) \ \forall i \in \{1, \dots, N_{\mathrm{x}}^{(a)}\}. \tag{2.10b}$$

With these shorthands, the discretized equation system describing the behavior of the gas within the pipe takes the following form:

$$\frac{1}{\Delta t}\begin{pmatrix} \bar{p}_{*,j}^{(a)} - \bar{p}_{*,j-1}^{(a)} \\ \bar{q}_{*,j}^{(a)} - \bar{q}_{*,j-1}^{(a)} \end{pmatrix} = F^{(a)}\left(\bar{p}_{*,j-1}^{(a)}, \bar{q}_{*,j-1}^{(a)}, p_{\mathrm{in},j}^{(a)}, p_{\mathrm{out},j}^{(a)}\right) + G^{(a)}\left(\bar{p}_{*,j-1}^{(a)}, \bar{q}_{*,j-1}^{(a)}, p_{\mathrm{in},j}^{(a)}, p_{\mathrm{out},j}^{(a)}\right) \quad \forall j \in \{1, \dots, N_{\mathrm{t}}\}, \tag{2.11}$$

where j is an index identifying the interval of the time grid with which the equation is associated. Furthermore, since the time discretization implied by the explicit Euler method is piecewise linear, the average boundary flow rate over a time interval specified by the index $j \in \{1, \dots, N_{\mathrm{t}}\}$ is given by

$$\bar{q}_{\mathrm{in},j}^{(a)} = \frac{A}{c}\left(p_{\mathrm{in},j}^{(a)} - \frac{1}{2}\left(\bar{p}_{1,j-1}^{(a)} + \bar{p}_{1,j}^{(a)}\right)\right) + \frac{1}{2}\left(\bar{q}_{1,j-1}^{(a)} + \bar{q}_{1,j}^{(a)}\right), \tag{2.12a}$$

$$\bar{q}_{\mathrm{out},j}^{(a)} = \frac{A}{c}\left(\frac{1}{2}\left(\bar{p}_{N_{\mathrm{x}},j-1}^{(a)} + \bar{p}_{N_{\mathrm{x}},j}^{(a)}\right) - p_{\mathrm{out},j}^{(a)}\right) + \frac{1}{2}\left(\bar{q}_{N_{\mathrm{x}},j-1}^{(a)} + \bar{q}_{N_{\mathrm{x}},j}^{(a)}\right). \tag{2.12b}$$

Note that equations in (2.11 – 2.12) are completely symmetric with respect to the orientation of the pipe and the designation of inlet and outlet. Indeed, the direction of nominally positive flow does not factor into the derivation of finite volume methods at all. In addition, since the approximate pressures and flow rates at the grid points x_i are shared between adjacent cells, finite volume methods guarantee that, in total, no mass or impulse is lost between cells.

This indifference to direction of flow presents a major advantage of the finite volume method over simple upwinding schemes such as the one used in [21], which can guarantee proper conservation of mass and impulse only if the direction of flow is known ahead of time and may require artificial flow splitting schemes and the introduction of large numbers of integers to be used in scenarios where the direction of flow is not known ahead of time and may vary across the length of the pipe.

2.3.4 Initial Conditions and Steady States

For the PDE solution to be well defined, an initial state must be specified. For instance, this state may simply be assumed or derived by interpolating between sparse sensor data. For the purposes of a purely hypothetical scenario, such data are generally not available. We may therefore wish to derive an initial state using a different method.

Since our objective is to find the optimal response to a fluctuation in demand in a system that was previously in a stable state, one conceivable method is to choose a stable initial condition. For this discussion, we take the word "stable" to mean that the network is in a steady state, in other words, that all initial demands are being satisfied in such a way that the state of the network does not change over time. For a single pipe, this can easily be expressed by demanding that the right-hand side of the ODE system be zero at the beginning of the time horizon:

$$F_i^{(a)}\left(\bar{p}_{*,0}^{(a)}, \bar{q}_{*,0}^{(a)}, p_{\text{in},1}^{(a)}, p_{\text{out},1}^{(a)}\right) + G_i^{(a)}\left(\bar{p}_{*,0}^{(a)}, \bar{q}_{*,0}^{(a)}, p_{\text{in},1}^{(a)}, p_{\text{out},1}^{(a)}\right) = 0 \quad \forall i \in \{1, \dots, 2N_{\text{x}}^{(a)}\}. \tag{2.13}$$

2.3.5 Objective Function Terms

Our goal is that the network bridge the demand fluctuation being modeled without deviating from the stable state that it starts in, as discussed in Section 2.3.4. Otherwise, the network could avoid drawing in gas from external sources by running a deficit over the time horizon of the simulation. One way to avoid this situation would be to penalize a difference in the total amount of gas contained within the network:

$$\kappa \Delta x^{(v,w)} \Big| \sum_{(v,w)\in\mathcal{A}} \sum_{i=1}^{N_{\text{x}}^{(v,w)}} \left(\bar{p}_{i,N_{\text{t}}}^{(v,w)} - \bar{p}_{i,0}^{(v,w)}\right) \Big|,$$

where $\kappa \geq 0$ is a global penalty parameter.

However, this runs a risk of the network ending up in an unsustainable state due to the location and impulse of the gas. Therefore, we adopt a stricter L_1 penalty for both average pressure and average mass flow rate. For each pipe $(v, w) \in \mathcal{A}$, we introduce additional variables $\Delta\bar{p}^{(v,w)}, \Delta\bar{q}^{(v,w)} \in \mathbb{R}^{N_{\text{x}}^{(v,w)}}$ and inequality constraints

$$\Delta\bar{p}_*^{(v,w)} \geq \bar{p}_{*,0}^{(v,w)} - \bar{p}_{*,N_{\text{t}}}^{(v,w)}, \tag{2.14a}$$

$$\Delta\bar{p}_*^{(v,w)} \geq \bar{p}_{*,N_{\text{t}}}^{(v,w)} - \bar{p}_{*,0}^{(v,w)}, \tag{2.14b}$$

$$\Delta\bar{q}_*^{(v,w)} \geq \bar{q}_{*,0}^{(v,w)} - \bar{q}_{*,N_{\text{t}}}^{(v,w)}, \tag{2.14c}$$

$$\Delta\bar{q}_*^{(v,w)} \geq \bar{q}_{*,N_{\text{t}}}^{(v,w)} - \bar{q}_{*,0}^{(v,w)}. \tag{2.14d}$$

The penalty term for (v, w) then takes the form

$$\kappa \Delta x^{(v,w)} \sum_{i=1}^{N_{\text{x}}^{(v,w)}} \left(\Delta\bar{p}_i^{(v,w)} + \Delta\bar{q}_i^{(v,w)}\right). \tag{2.15}$$

Note that by fixing the initial state and penalizing the difference to a fixed terminal state, one can use the model to plan any transition between two known network states.

2.4 Junctions and Compressor Stations

Now that we have defined the constraints modeling the behavior of a single pipe, we turn to those points in the network where pipes interact with one another and with the environment. These junctions correspond to vertices of the network graph.

We assume that junctions themselves do not possess any capacity to store gas. In other words, while gas may be injected into the network at supply junctions $v \in \mathcal{S}$ or withdrawn from it at demand junctions $v \in \mathcal{D}$, we are interested primarily in solutions in which gas is not otherwise removed or added at any junction. This condition can be modeled by using the following balance equation:

$$\sum_{(w,v)\in\mathcal{A}} q^{(v,w)}(t, L_a) - \sum_{(w,v)\in\mathcal{A}} q^{(v,w)}(t, 0) + q^{(v)}_{\mathrm{src}}(t) - q^{(v)}_{\mathrm{snk}}(t) = 0 \quad \forall v \in \mathcal{V}, t \in [0, T], \tag{2.16}$$

where $q^{(v)}_{\mathrm{src}} \colon [0, T] \to [0, \infty)$ denotes the amount of gas injected at junction $v \in \mathcal{V}$ over time and $q^{(v)}_{\mathrm{snk}} \colon [0, T] \to [0, \infty)$ denotes the amount of gas withdrawn at the same junction over time. Naturally, we assume that source and sink flows are zero at junctions that do not act as sources or sinks:

$$\begin{aligned} q^{(v)}_{\mathrm{src}}(t) = 0 \quad & \forall v \in \mathcal{V} \setminus \mathcal{S}, t \in [0, T], \\ q^{(v)}_{\mathrm{snk}}(t) = 0 \quad & \forall v \in \mathcal{V} \setminus \mathcal{D}, t \in [0, T]. \end{aligned}$$

Aside from linking the inlet and outlet flows of connected pipes via equation (2.16), junctions also define their boundary pressures. If no compression is performed, all pipes connected to a given junction must share the same boundary pressure, which is stored in a node pressure variable $p^{(v)} \colon 0 \to \mathbb{R}_+$:

$$\begin{aligned} p^{(w,v)}(t, L_a) = p^{(v)}(t) \quad & \forall (w, v) \in \mathcal{A}, t \in [0, T], \\ p^{(v,w)}(t, 0) = p^{(v)}(t) \quad & \forall (v, w) \in \mathcal{A}, t \in [0, T]. \end{aligned}$$

However, we assume that each junction $v \in \mathcal{V}$ also functions as a compressor station that may apply a multiplicative boost to the boundary pressure of each connected pipe:

$$\begin{aligned} p^{(w,v)}(t, L_a) = \mu^{(v,w)}(t) p^{(v)}(t) \quad & \forall (w, v) \in \mathcal{A}, t \in [0, T], \\ p^{(v,w)}(t, 0) = \mu^{(v,w)}(t) p^{(v)}(t) \quad & \forall (v, w) \in \mathcal{A}, t \in [0, T], \end{aligned}$$

where $\mu^{(v,w)}(t) \geq 1$. We index the compression ratio at the outlet side of a pipe $(w, v) \in \mathcal{A}$ with the reversed tuple (v, w). This notation is well defined because of the prohibition of antiparallel pipes stated in Section 2.1.

Accurate modeling of different types compressor machines exceeds the scope of this paper. See, for instance, [11, ch. 2], for a discussion of the underlying physical principles governing gas compressor operation. The compression ratio, while acting as a control variable in our model, is subject to certain constraints, the precise form of which depends on the type and model of the

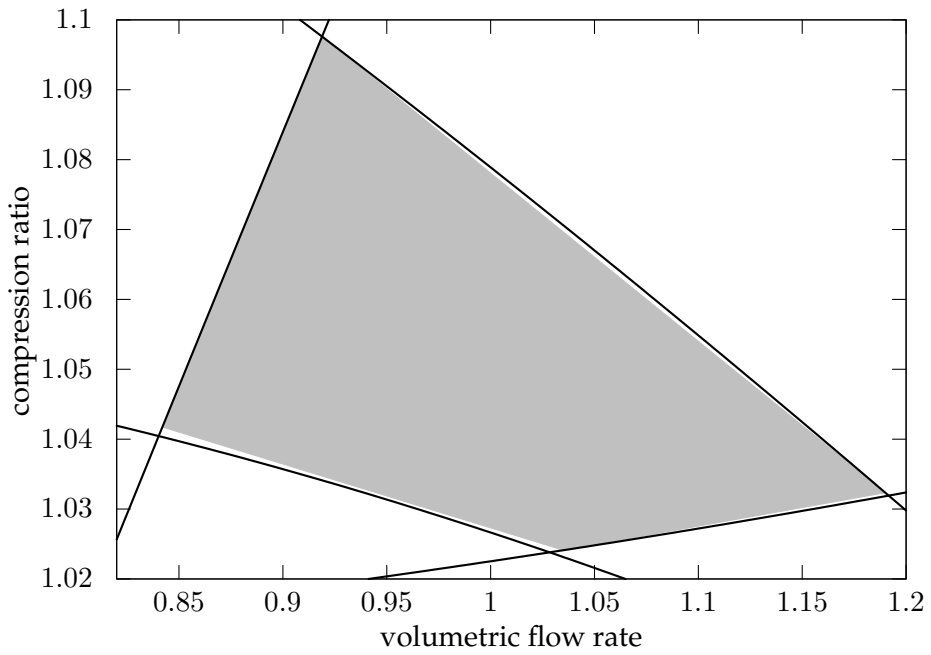


Figure 1: Example of a centrifugal compressor approximate feasible region (adapted from the GasLib-40 test network [10]). Nonlinear bounds are indicated by thick lines.

compressor machine as well as various environmental factors. In [11, ch.2] the feasible region of a compressor machine is expressed in terms of *adiabatic change in specific enthalpy* and *volumetric flow rate*.

The adiabatic change in specific enthalpy is a nonlinear function of the compression ratio, while the volumetric flow rate is proportional to the ratio between the mass flow rate and the compressor suction pressure. By transforming back from adiabatic change in specific enthalpy to compression ratio and then scaling by the compressor suction pressure $p^{(v)}(t) > 0$, we may assume that the feasible region of each compressor machine is given in terms of the pipe boundary mass flow rate and the compressor discharge pressure, which is equal to the pipe boundary pressure.

The precise shape of the feasible region depends on the type of compressor in question. For instance, [20] indicates that a centrifugal compressor is limited by simple bounds on the speed of the rotor rotation and the ratio between the volumetric flow rate and that speed. After several nonlinear transformations, these simple bounds generally become nonlinear and even nonconvex. To avoid complications, we will work with a suitably large polyhedral subset of the feasible region such as the one depicted in Figure 1.

Assume that the polyhedral *approximate feasible region* depicted in Figure 1 is given in terms of compression ratio μ and volumetric flow rate V by the following linear system of equations:

$$A \begin{pmatrix} \mu \\ \frac{M}{zRT} V \end{pmatrix} \leq b.$$

Then we can obtain an equivalent system in terms of discharge pressure p_{d} and mass flow rate q

by scaling both sides of the equation with the suction pressure p_s:

$$A \begin{pmatrix} p_\mathrm{d} \\ q \end{pmatrix} \leq b p_\mathrm{s}, \text{ or equivalently } \begin{pmatrix} A & -b \end{pmatrix} \begin{pmatrix} p_\mathrm{d} \\ q \\ p_\mathrm{s} \end{pmatrix} \leq 0.$$

We will therefore assume that the feasible region of any compressor is given as a polyhedral cone in terms of discharge pressure (i.e., pipe boundary pressure), mass flow rate, and suction pressure (i.e., node pressure).

In larger compressor stations, different types of compressor machines may be used either in parallel or in series, with different arrangements often being used to cover various subsets of the (μ, Q)-space. To account for this type of complex behavior, we allow compressors to operate in different *modes*. For each compressor installed at an endpoint (v, w) of a pipe (v, w) or $(w, v) \in \mathcal{A}$, let $N_\mathrm{c}^{(v,w)} \in \mathbb{N}_0$ denote the number of such modes with $N_\mathrm{c}^{(v,w)} = 0$ indicating that no compressor is installed. To switch between these modes, we introduce integer controls.

For each $k \in \{1, \dots, N_\mathrm{c}^{(v,w)}\}$, we introduce a binary indicator variable $\chi_k^{(v,w)} \colon [0,T] \to \{0,1\}$ that indicates that the kth mode is active at a given point in time. At any given time, at most one node may be active:

$$\sum_{k=1}^{N_\mathrm{c}^{(v,w)}} \chi_k^{(v,w)}(t) \leq 1 \quad \forall (v,w) \in \mathcal{A}, t \in [0,T], \tag{2.18a}$$

$$\sum_{k=1}^{N_\mathrm{c}^{(w,v)}} \chi_k^{(w,v)}(t) \leq 1 \quad \forall (v,w) \in \mathcal{A}, t \in [0,T]. \tag{2.18b}$$

Further, we allow for the specification of a matrix $A^{(v,w,k)} \in \mathbb{R}^{m^{(v,w,k)} \times 3}$ and such that the feasible region of the kth mode of the compressor is enforced by the indicator constraint:

$$\chi_k^{(v,w)}(t) = 1 \;\Rightarrow\; A^{(v,w,k)} \begin{pmatrix} p^{(v,w)}(t,0) \\ q^{(v,w)}(t,0) \\ p^{(v)}(t) \end{pmatrix} \leq 0 \qquad \forall (v,w) \in \mathcal{A}, k \in \{1, \dots, N_\mathrm{c}^{(v,w)}\}, \tag{2.19a}$$

$$\chi_k^{(w,v)}(t) = 1 \;\Rightarrow\; A^{(w,v,k)} \begin{pmatrix} p^{(v,w)}(t,L_{v,w}) \\ -q^{(v,w)}(t,L_{v,w}) \\ p^{(w)}(t) \end{pmatrix} \leq 0 \; \forall (v,w) \in \mathcal{A}, k \in \{1, \dots, N_\mathrm{c}^{(w,v)}\}. \tag{2.19b}$$

When no mode is active, we assume that compressors are being bypassed. Therefore, the compression ratio must be 1:

$$\sum_{k=1}^{N_\mathrm{c}^{(v,w)}} \chi_k^{(v,w)}(t) = 0 \;\Rightarrow\; \mu^{(v,w)}(t) = 1 \quad \forall (v,w) \in \mathcal{V}^2 \colon (v,w) \in \mathcal{A} \vee (w,v) \in \mathcal{A}, t \in [0,T]. \tag{2.20}$$

2.4.1 Indicator Constraint Relaxation

Since indicator constraints, such as those used to encode a compressor feasible region in equations (2.19) and (2.20), are not supported by most MINLP solvers, we relax them manually using a

big-M formulation. In Section 2.3, we had assumed pressure and flow rate to be globally bounded throughout the network. Note that the pressure bounds imply bounds on the compression ratio:

$$1 \leq \mu \leq \hat{\mu},$$

where $\hat{\mu} := \frac{\hat{p}}{\check{p}}$.

Consider any pipe endpoint (v, w) and associated compressor mode $k \in \{1, \ldots, N_{\mathrm{c}}^{(v,w)}\}$. Without restriction of generality, we assume that (v, w) is the inlet endpoint of the pipe $(v, w) \in \mathcal{A}$. For every row $i \in \{1, \ldots, m^{(v,w,k)}\}$ of $A^{(v,w,k)}$, we define

$$M_i^{(v,w,k)} := \max \left\{ A_{i,*}^{(v,w,k)} \begin{pmatrix} p_{\mathrm{d}} \\ q \\ p_{\mathrm{s}} \end{pmatrix} \;\middle|\; p_{\mathrm{d}} \in [\check{p}, \hat{p}],\ p_{\mathrm{s}} \in [\check{p}, \hat{p}],\ q \in [-\hat{q}, \hat{q}],\ p_{\mathrm{d}} \geq p_{\mathrm{s}} \right\} < \infty.$$

Given the vector $M^{(v,w,k)} \in \mathbb{R}^{m^{(v,w,k)}}$, the indicator constraints (2.19) are equivalent to

$$\begin{pmatrix} A^{(v,w,k)} & M^{(v,w,k)} \end{pmatrix} \begin{pmatrix} p^{(v,w)}(t,0) \\ q^{(v,w)}(t,0) \\ p^{(v)}(t) \\ \chi_k^{(v,w)}(t) \end{pmatrix} \leq M^{(v,w,k)} \ \forall\ t \in [0,T],\ (v,w) \in \mathcal{A},\ k \in \{1, \ldots, N_{\mathrm{c}}^{(v,w)}\}, \tag{2.21a}$$

$$\begin{pmatrix} A^{(w,v,k)} & M^{(w,v,k)} \end{pmatrix} \begin{pmatrix} p^{(v,w)}(t,L_{v,w}) \\ -q^{(v,w)}(t,L_{v,w}) \\ p^{(w)}(t) \\ \chi_k^{(w,v)}(t) \end{pmatrix} \leq M^{(w,v,k)} \ \forall\ t \in [0,T],\ (v,w) \in \mathcal{A},\ k \in \{1, \ldots, N_{\mathrm{c}}^{(w,v)}\}. \tag{2.21b}$$

To simplify the reformulation of (2.20) and the formulation of our objective function in Section 2.5, we introduce separate compression ratio variables $\mu_k^{(v,w)} \colon [0,T] \to [1,\infty)$ for each compressor mode k and accumulate them using a sum of logarithms to obtain the following reformulation of (2.17):

$$\log\big(p^{(v,w)}(t,0)\big) = \log\big(p^{(v)}(t)\big) + \sum_{k=1}^{N_{\mathrm{c}}^{(v,w)}} \log\big(\mu_k^{(v,w)}(t)\big) \quad \forall (v,w) \in \mathcal{A},\ t \in [0,T], \tag{2.22a}$$

$$\log\big(p^{(v,w)}(t,L_{v,w})\big) = \log\big(p^{(w)}(t)\big) + \sum_{k=1}^{N_{\mathrm{c}}^{(w,v)}} \log\big(\mu_k^{(w,v)}(t)\big) \quad \forall (v,w) \in \mathcal{A},\ t \in [0,T], \tag{2.22b}$$

which, given that all pressures and compression ratios are strictly positive, is equivalent to multiplying them. The indicator constraint (2.20) can then equivalently be enforced by using

$$\mu_k^{(v,w)}(t) \leq 1 + (\hat{\mu} - 1)\chi_k^{(v,w)}(t) \quad \forall t \in [0,T],\ v \in \mathcal{V},\ w \in \mathcal{N}(v),\ k \in \{1, \ldots, N_{\mathrm{c}}^{(v,w)}\}. \tag{2.23}$$

2.4.2 Time Discretization

At each junction $v \in \mathcal{V}$, we introduce discretized flow balance constraints for each interval of the sampling grid:

$$\sum_{(w,v)\in\mathcal{A}} \bar{q}_{\mathrm{out},j}^{(w,v)} - \sum_{(w,v)\in\mathcal{A}} \bar{q}_{\mathrm{out},j}^{(w,v)} + q_{\mathrm{src},j}^{(v)} = \bar{q}_{\mathrm{snk},j}^{(v)} \quad \forall j \in \{1, \ldots, N_{\mathrm{t}}\}.$$

We relax these constraints by introducing nonnegative slack variables $\bar{q}^{(v)}_{\mathrm{slk},j} \geq 0$, which represent a certain amount of gas to being ejected from the network at junction v. In Section 2.4.4, we will introduce an L^1 penalty term for these slack variables:

$$\sum_{(w,v)\in\mathcal{A}} \bar{q}^{(w,v)}_{\mathrm{out},j} - \sum_{(w,v)\in\mathcal{A}} \bar{q}^{(w,v)}_{\mathrm{out},j} + q^{(v)}_{\mathrm{src},j} - \bar{q}^{(v)}_{\mathrm{slk},j} = \bar{q}^{(v)}_{\mathrm{snk},j} \quad \forall j \in \{1,\dots,N_{\mathrm{t}}\},\ v \in \mathcal{V}. \tag{2.24}$$

Discretization of compressor constraints presents several concerns. Since flow rates are averaged over intervals of the sampling grid, actual flow rates encountered during the simulation may exceed the feasible region of the compressor station. We will not address these concerns here, assuming that sufficiently fine sampling grids will prevent these occurrences from being sufficiently severe to present real risk if sufficient safety margins are maintained when defining the approximate feasible region.

We discretize node pressure $p^{(v)}$ and compression ratio $\mu^{(v,w)}$ as piecewise constant functions that are constant in the interior of each interval of the sampling grid. Similarly, we discretize $\chi^{(v,w)}$ as a piecewise constant vector function that is constant in the interior of each interval of the integer control interval, meaning that for each node $v \in \mathcal{V}$, we obtain a node pressure vector $\bar{p}^{(v)} \in \mathbb{R}^{N_{\mathrm{t}}}$, and for each pipe endpoint (v,w), we obtain a compression ratio matrix $\mu^{(v,w)} \in \mathbb{R}^{N_{\mathrm{c}}^{(v,w)}\times N_{\mathrm{t}}}$ and a compressor mode matrix $\chi^{(v,w)} \in \{0,1\}^{N_{\mathrm{c}}^{(v,w)}\times N_{\mathrm{w}}}$.

The discretized compressor constraints are given by the discretized form of the boundary pressure constraints (2.22)

$$\log\bigl(p^{(v,w)}_{\mathrm{in},j}\bigr) - \log\bigl(p^{(v)}_{j}\bigr) - \sum_{k=1}^{N_{\mathrm{c}}^{(v,w)}} \log\bigl(\mu^{(v,w)}_{k,j}\bigr) = 0 \quad \forall j \in \{1,\dots,N_{\mathrm{t}}\},\ (v,w) \in \mathcal{A}, \tag{2.25a}$$

$$\log\bigl(p^{(v,w)}_{\mathrm{out},j}\bigr) - \log\bigl(p^{(w)}_{j}\bigr) - \sum_{k=1}^{N_{\mathrm{c}}^{(w,v)}} \log\bigl(\mu^{(w,v)}_{k,j}\bigr) = 0 \quad \forall j \in \{1,\dots,N_{\mathrm{t}}\},\ (v,w) \in \mathcal{A}, \tag{2.25b}$$

the constraint (2.23) restricting the compression ratio of inactive compressors,

$$\mu^{(v,w)}_{k,j} - (\hat{\mu}-1)\chi^{(v,w)}_{k,j} \leq 1 \quad \forall j \in \{1,\dots,N_{\mathrm{t}}\},\ v \in \mathcal{V},\ w \in \mathcal{N}(v),\ k \in \{1,\dots,N_{\mathrm{c}}^{(v,w)}\}, \tag{2.25c}$$

the discretization of the constraints (2.21), which define the compressor feasible region,

$$A^{(v,w,k)} \begin{pmatrix} p^{(v,w)}_{\mathrm{in},j} \\ \bar{q}^{(v,w)}_{\mathrm{in},j} \\ p^{(v)}_{j} \end{pmatrix} \leq \left(1 - \chi^{(v,w)}_{k,J_j}\right) M^{(v,w,k)} \quad \begin{array}{l} \forall\ t \in \{1,\dots,N_{\mathrm{t}}\},\ (v,w) \in \mathcal{A}, \\ \qquad k \in \{1,\dots,N_{\mathrm{c}}^{(v,w)}\}, \end{array} \tag{2.25d}$$

$$A^{(w,v,k)} \begin{pmatrix} p^{(v,w)}_{\mathrm{out},j} \\ -\bar{q}^{(v,w)}_{\mathrm{out},j} \\ p^{(w)}_{j} \end{pmatrix} \leq \left(1 - \chi^{(w,v)}_{k,J_j}\right) M^{(w,v,k)} \quad \begin{array}{l} \forall\ t \in \{1,\dots,N_{\mathrm{t}}\},\ (v,w) \in \mathcal{A}, \\ \qquad k \in \{1,\dots,N_{\mathrm{c}}^{(w,v)}\}, \end{array} \tag{2.25e}$$

and the special ordered set (SOS) constraint (2.18),

$$\sum_{k=1}^{N_{\mathrm{c}}^{(v,w)}} \chi^{(v,w)}_{k,j} \leq 1 \quad \forall j \in \{1,\dots,N_{\mathrm{w}}\},\ v \in \mathcal{V},\ w \in \mathcal{N}(v). \tag{2.25f}$$

2.4.3 Assumptions and Simplifications

To simplify the definition of feasible regions for our test problems in Section 4, we assume that all compressors defined there have exactly one mode (i.e., $N_{\mathrm{c}}^{(v,w)} = 1$) and that the only constraint on the feasible region is that compressors may not push against the flow:

$$A^{(v,w,1)} := \begin{pmatrix} 0 & 0 & -1 \end{pmatrix}.$$

Under these assumptions, we simply have

$$M^{(v,w,1)} := \begin{pmatrix} \hat{q} \end{pmatrix}.$$

More generally, we assume that compressors cannot push against the flow. Under this assumption, we can guarantee that every non-source junction $v \in \mathcal{V} \setminus \mathcal{S}$ that is transporting any amount of gas has at least one pipe with an inactive compressor. The boundary pressure of that pipe then defines the internal node pressure, which would otherwise act as a free variable that could be manipulated to compress gas without using compressors.

For source junctions $v \in \mathcal{S}$, we had already assumed that there is exactly one incident pipe. We may therefore assume that the boundary flow in that pipe is outgoing. We assume the internal pressure of these nodes to be fixed, and we introduce a single control variable $p_{\mathrm{fix}}^{(v)} \in \mathbb{R}$ that defines the internal node pressure over the entire time horizon.

2.4.4 Objective Function Terms

We introduce three objective terms for junctions. First, gas supplied by a supply junction $v \in \mathcal{S}$ is associated with a monetary cost proportional to the mass of the gas supplied:

$$\Delta t \sum_{j=1}^{N_{\mathrm{t}}} C_{\mathrm{src},j}^{(v)} \bar{q}_{\mathrm{src},j}^{(v)}, \tag{2.26}$$

where $C_{\mathrm{src}}^{(v)} \in \mathbb{R}_{+}^{N_{\mathrm{t}}}$. Note that in the undiscretized problem, this can be interpreted as a linear functional in $L^2([0,T])$.

Second, for every $v \in \mathcal{V}$ and $w \in \mathcal{N}(v)$, the compressor located at the pipe endpoint (v,w), which we will assume to be the inlet endpoint of the pipe (v,w), incurs costs associated with the energy used to compress the gas flowing through it. The precise amount of energy used by a compressor depends on the specific change in adiabatic enthalpy, the mass flow rate, the compressor's adiabatic efficiency (which we have omitted from our model), and various properties of its drive.

We assume that all compressors are electrically driven and operate at a mode-dependent constant adiabatic efficiency. We further assume that electrical power costs are the same for all compressors. In reality, most compressors are driven by gas turbines fed with gas passing through the junction, while adiabatic efficiency varies nonlinearly depending on the specific change in adiabatic enthalpy and the volumetric flow rate [11, ch. 2]. Under these assumptions, we obtain a

nonconvex objective term of the following form [16, 21] for the endpoint (v, w):

$$\Delta t \sum_{j=1}^{N_\mathrm{t}} \sum_{k=1}^{N_\mathrm{c}^{(v,w)}} C_{\mathrm{cmpr},k,j}^{(v,w)} \bar{q}_{\mathrm{in},j}^{(v,w)} \left(\left(\mu_{k,j}^{(v,w)} \right)^{\omega} - 1 \right).$$

Note that this term is zero for $\mu = 1$, which is necessary to make sure that inactive compressor modes do not contribute to the objective function. To avoid the additional runtime penalty incurred by including this objective term, we assume that the exponent ω, which is always between 0 and 1, is equal to 1. Further, we assume that most boundary flow rates will be roughly equal in magnitude, that the adiabatic efficiency for every compressor mode is equal, and that the cost of electrical power is the same at every pipe endpoint. Under these assumptions, we arrive at a simpler, linear objective term:

$$C_\mathrm{cmpr} \Delta t \sum_{j=1}^{N_\mathrm{t}} \sum_{k=1}^{N_\mathrm{c}^{(v,w)}} \left(\mu_{k,j}^{(v,w)} - 1 \right), \tag{2.27}$$

where $C_\mathrm{cmpr} \geq 0$ is a global parameter.

Third, we penalize the flow balance slack variable for a given junction $v \in \mathcal{V}$ with the following L^1 penalty term:

$$\kappa \Delta t \sum_{j=1}^{N_\mathrm{t}} q_{\mathrm{slk},j}^{(v)}. \tag{2.28}$$

2.5 Optimization Problem

The complete optimization problem is formulated by aggregating the objective terms and constraints listed in Tables 1 and 2:

$$\begin{aligned}
&\underset{\substack{p,\bar{p},\bar{q},p_\mathrm{in},p_\mathrm{out},p_\mathrm{fix},\\ \bar{q}_\mathrm{in},\bar{q}_\mathrm{out},q_\mathrm{src},q_\mathrm{slk},\\ \mu,\chi}}{\text{minimize}} && \sum_{v \in \mathcal{S}} (2.26) + \sum_{v \in \mathcal{V}} \Big((2.28) + \sum_{w \in \mathcal{N}(v)} (2.27) \Big) + \sum_{(v,w) \in \mathcal{A}} (2.15) \\
&\text{subject to} && \text{Discretized PDE constraints (2.11 – 2.12)} && \forall (v,w) \in \mathcal{A}, \\
& && \text{Steady state constraints (2.13)} && \forall (v,w) \in \mathcal{A}, \\
& && \text{Absolute value constraints (2.14)} && \forall (v,w) \in \mathcal{A}, \\
& && \text{Flow balance constraints (2.24)} && \forall v \in \mathcal{V}, \\
& && \text{Compressor constraints (2.25)} && \forall v \in \mathcal{V}, \\
& && \text{Local pressure bounds from Table 2} && \forall v \in \mathcal{V}, \\
& && \text{Local supply flow bounds from Table 2} && \forall v \in \mathcal{S}, \\
& && \text{Local compression ratio bounds from Table 2} && \forall v \in \mathcal{V}, \\
& && \text{Local slack flow bounds from Table 2} && \forall v \in \mathcal{V}, \\
& && \text{Shared boundary pressure bounds} && \forall v \in \mathcal{V}, w \in \mathcal{N}(v)
\end{aligned} \tag{2.29}$$

Table 1: Summary of pipe blocks.

Object:	Pipe $(v, w) \in \mathcal{A}$
Parameters:	• Penalty parameter $\kappa \geq 0$ • Length $L_{v,w} > 0$ • Diameter $D_{v,w} > 0$ • Effective roughness height $\epsilon_{v,w} \geq 0$ • Number of spatial intervals $N_{\mathrm{x}}^{(v,w)} \in \mathbb{N}$
Variables:	• Boundary pressure controls $\left(p_{\mathrm{in}}^{(v,w)},\ p_{\mathrm{out}}^{(v,w)}\right) \in [\check{p}, \hat{p}]^{2N_{\mathrm{t}}}$ (*independent*) • State deviation $\left(\Delta\bar{p}^{(v,w)},\ \Delta\bar{q}^{(v,w)}\right) \in \mathbb{R}^{2N_{\mathrm{x}}^{(v,w)}}$ (*independent*) • State over time $\left(\bar{p}^{(v,w)},\ \bar{q}^{(v,w)}\right) \in \mathbb{R}^{2N_{\mathrm{x}}^{(v,w)} \times (N_{\mathrm{t}}+1)}$ (*dependent*) • Average boundary flow rates $\left(\bar{q}_{\mathrm{in}}^{(v,w)},\ \bar{q}_{\mathrm{out}}^{(v,w)}\right) \in \mathbb{R}^{2N_{\mathrm{t}}}$ (*dependent*)
Constraints:	• Discretized PDE constraints (2.11) – (2.12): $\left(2N_{\mathrm{x}}^{(v,w)} + 2\right) N_{\mathrm{t}}$ equations • Steady state constraints (2.13): $2N_{\mathrm{x}}^{(v,w)}$ equations • Absolute value constraints (2.14): $4N_{\mathrm{x}}^{(v,w)}$ inequalities
Objective:	• Periodicity penalty (2.15)
Total:	• $2N_{\mathrm{x}}^{(v,w)}(N_{\mathrm{t}} + 2) + 4N_{\mathrm{t}}$ continuous variables • $2N_{\mathrm{x}}^{(v,w)}(N_{\mathrm{t}} + 1) + 2N_{\mathrm{t}}$ equality constraints • $4N_{\mathrm{x}}^{(v,w)}$ inequality constraints • $4N_{\mathrm{t}}$ bound constraints

Table 2: Summary of junction blocks.

Object:	Junction $v \in \mathcal{V}$
Parameters:	• Supply bounds $\left(\check{q}_{\mathrm{src}}^{(v)}, \hat{q}_{\mathrm{src}}^{(v)}\right) \in \mathbb{R}^{2N_\mathrm{t}}$ (*if* $v \in \mathcal{S}$) • Specific supply cost $C_{\mathrm{src}}^{(v)} \in \mathbb{R}_+^{N_\mathrm{t}}$ (*if* $v \in \mathcal{S}$) • Demand profile $\bar{q}_{\mathrm{snk}}^{(v)} \in \mathbb{R}_+^{N_\mathrm{t}}$ (*if* $v \in \mathcal{D}$) • Specific compressor cost $C_{\mathrm{cmpr}} \in \mathbb{R}_+$ • Compressor modes $N_\mathrm{c}^{(v,w)} \in \mathbb{N}_0$ for all $w \in \mathcal{N}(v)$ • Matrices $A^{(v,w,k)} \in \mathbb{R}^{m^{(v,w,k)} \times 3}$ for all $w \in \mathcal{N}(v)$ and $k \in \{1, \dots, N_\mathrm{c}^{(v,w)}\}$ • Penalty parameter $\kappa \geq 0$
Variables:	• Supply flow rates $\bar{q}_{\mathrm{src}}^{(v)} \in \left[\check{q}_{\mathrm{src}}^{(v)}, \hat{q}_{\mathrm{src}}^{(v)}\right]$ (*if* $v \in \mathcal{S}$, *independent*) • Fixed internal pressure $p_{\mathrm{fix}}^{(v)} \in [\check{p}, \hat{p}]$ (*if* $v \in \mathcal{S}$, *independent*) • Flow slack variables $q_{\mathrm{slk}}^{(v)} \in \mathbb{R}_+^{N_\mathrm{t}}$ (*independent*) • Compression ratios $\mu^{(v,w)} \in \mathbb{R}_{\geq 1}^{N_\mathrm{c}^{(v,w)} \times N_\mathrm{t}}$ for all $w \in \mathcal{N}(v)$ (*independent*) • Indicators $\chi^{(v,w)} \in \{0,1\}^{N_\mathrm{c}^{(v,w)} \times N_\mathrm{w}}$ for all $w \in \mathcal{N}(v)$ (*independent*) • Boundary flow $\bar{q}_{\mathrm{in}}^{(v,w)}$ or $\bar{q}_{\mathrm{out}}^{(w,v)} \in \mathbb{R}^{N_\mathrm{t}}$ for all $w \in \mathcal{N}(v)$ (*independent*) • Internal pressure $p^{(v)} \in [\check{p}, \hat{p}]^{N_\mathrm{t}}$ (*dependent*) • Boundary pressure $p_{\mathrm{in}}^{(v,w)}$ or $p_{\mathrm{out}}^{(w,v)} \in [\check{p}, \hat{p}]^{N_\mathrm{t}}$ for $w \in \mathcal{N}(v)$ (*dependent*)
Constraints:	• Flow balance (2.24): N_t equations • Boundary pressure (2.25): $N_\mathrm{t}\lvert\mathcal{N}(v)\rvert$ equations • Compressor modes (2.25): $\sum_{w \in \mathcal{N}(v)} \left(N_\mathrm{w} + \sum_{k=1}^{N_\mathrm{c}^{(v,w)}} (1 + m^{(v,w,k)}) N_\mathrm{t} \right)$ inequalities
Objective:	• Supply costs (2.26) (*if* $v \in \mathcal{S}$) • Compressor operating costs (2.27) • Flow balance slack penalty (2.28)
Total:	• $(2+2\lvert\mathcal{N}(v)\rvert)N_\mathrm{t} + \sum_{w \in \mathcal{N}(v)} N_\mathrm{c}^{(v,w)} N_\mathrm{t}$ continuous variables ($+(N_\mathrm{t}+1)$ *if* $v \in \mathcal{S}$) • $\sum_{w \in \mathcal{N}(v)} N_\mathrm{c}^{(v,w)} N_\mathrm{t}$ binary variables • $(1 + \lvert\mathcal{N}(v)\rvert)N_\mathrm{t}$ equality constraints • $\sum_{w \in \mathcal{N}(v)} \left(N_\mathrm{w} + \sum_{k=1}^{N_\mathrm{c}^{(v,w)}} (1 + m^{(v,w,k)}) N_\mathrm{t} \right)$ inequality constraints • $\left(3 + 2\lvert\mathcal{N}(v)\rvert + \sum_{v \in \mathcal{N}(v)} N_\mathrm{c}^{(v,w)}\right) N_\mathrm{t}$ bound constraints ($+(2N_\mathrm{t}+2)$ *if* $v \in \mathcal{S}$)

2.6 Computational Challenges

The optimization problem (2.29) is a discretized mixed-integer PDE-constrained optimization (MIPDECO) problem, where the integer controls arise as degrees of freedom in the discretization of the time-dependent binary-valued control function χ. Therefore, the problem has features of a mixed-integer optimal control (MIOC) problem. However, efficient approximate solution methods for MIOC problems, such as combinatorial integral approximation [17], generally do not account for constraints such as (2.25) that restrict controls based on the solution of the differential equation. We are therefore confined to using general-purpose mixed-integer solvers. For the remainder of this paper, we will describe several methods by which the performance of a general-purpose branch-and-bound solver may be improved to achieve adequate performance in problems of this kind.

Note also that the governing PDE constraints and several of the compressor constraints are nonlinear equality constraints, which are inevitably nonconvex. We do not address this nonconvexity here, which would require the use of global optimization methods that, given the large number of variables in a discretized PDE-constrained problem, would be prohibitive.

3 Branch and Bound

As part of this paper, we provide an implementation of the model described in Section 2, written in the modeling language AMPL [5]. We make several modifications to the underlying branch-and-bound MINLP solver Minotaur[1].

A basic branch-and-bound algorithm to solve the convex mixed-integer optimization problem

$$\min_{x,y} f(x,y) \quad \text{s.t.} (x,y) \in \mathcal{F},\ x \in \mathbb{R}^n,\ y \in \mathbb{Z}^m \tag{3.30}$$

is given as Algorithm 1. Variants of this basic algorithm differ in the precise design of the data structure of Q and the implementation of insert, select, bound, and $\mathrm{branchvar}$.

Generally, for convex problems, the operation bound is used to identify a tight lower bound for the optimization problem associated with a given node and, if $f_{\mathrm{lb}}^{(n)} < \infty$, a solution $\hat{x}, \hat{y}$ such that $f_{\mathrm{lb}}^{(n)} = f(\hat{x}, \hat{y})$. It is implemented by solving a restricted variant of the relaxed problem

$$\min_{x,y} f(x,y) \quad \text{s.t.} (x,y) \in \mathcal{F},\ x \in \mathbb{R}^n,\ y \in \mathbb{R}^m \tag{3.31}$$

where further restrictions are introduced by branching.

The structure of Q and the operations insert and select are closely related. Together, they form the algorithm's *node selection method*, which defines the order in which the nodes of the branch-and-bound tree are explored. We use two different node selection methods: a well-established method referred to as *best-then-dive* and a custom method we will refer to as *gradual backtracking*.

[1] Version 0.2.0, available from https://wiki.mcs.anl.gov/minotaur/ (as of August 10, 2017)

Algorithm 1: Branch-and-bound algorithm

Data: Q is an (initially empty) data structure suitable for storing unprocessed nodes
Result: $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{Z}^n$ is an optimal solution of (3.30) or (3.30) is infeasible

Construct root node r, set $f_{\mathrm{lb}}^{(p_r)} \leftarrow -\infty$, set $Q \leftarrow \mathrm{insert}(r, Q)$, and set $f_{\mathrm{ub}} \leftarrow \infty$;
while Q *not empty* **do**
  $(n, Q) \leftarrow \mathrm{select}(Q)$; `// Pick next subproblem to be solved.`
  if $f_{\mathrm{lb}}^{(p_n)} < f_{\mathrm{ub}}$ **then**
    $(f_{\mathrm{lb}}^{(n)}, \hat{x}, \hat{y}) \leftarrow \mathrm{bound}(n)$; `// Find a lower bound for the subproblem`
    if $f_{\mathrm{lb}}^{(n)} < f_{\mathrm{ub}}$ **then**
      if $\hat{y} \in \mathbb{Z}^m$ **then**
        `// Register new best integer-feasible solution`
        $(x^*, y^*, f_{\mathrm{ub}}) \leftarrow (\hat{x}, \hat{y}, f(\hat{x}, \hat{y}))$;
      else
        `// Branch on a variable that violates integrality`
        $i \leftarrow \mathrm{branchvar}(n, \hat{x}, \hat{y})$;
        Create node u from n by adding constraint $y_i \geq \lceil \hat{y}_i \rceil$;
        Create node d from n by adding constraint $y_i \leq \lfloor \hat{y}_i \rfloor$;
        $(p_u, p_d) \leftarrow (n, n)$;
        $Q \leftarrow \mathrm{insert}(d, \mathrm{insert}(u, Q))$;

In best-then-dive, the structure Q is a tuple of a special node d and a set of all remaining nodes H. When children are inserted into Q, as shown in Algorithm 2, d is always the child with the best-estimated objective. All other children are inserted into H. When selecting nodes, as shown in Algorithm 3, the solver first attempts to select the best immediate child d and then, if $d = \mathrm{NULL}$, defaults to the node with the best rigorous lower bound. Best-then-dive has disadvantages if the number m of integer variables is high. Since the best node is not determined by its estimated objective but rather by the lower bound of its parent, the best node is likely to be located at a shallow depth, and subsequent dives are liable to explore drastically different solutions rather than first attempting to correct erroneous decisions made during the current dive.

Gradual backtracking attempts to address this issue while simultaneously avoiding the pitfalls of a full depth-first search, which can easily degenerate into an exhaustive search if bad decisions are made near the root of the branch-and-bound tree. It can be interpreted as a depth-first search in which all nodes above a cutoff value $f_{\mathrm{co}} \in [f_{\mathrm{lb}}, f_{\mathrm{ub}}]$ that is dynamically placed between the global lower and upper bounds are disregarded and deferred for later exploration. As the tree is explored, global lower and upper bounds change, meaning that f_{co} also changes over time. Once a depth-first search is exhausted, gradual backtracking chooses the deferred node with the best lower bound and starts a new depth-first search rooted in that node. Note that at that point the lower bound of the best deferred node is equal f_{lb} and therefore guaranteed to satisfy the cutoff criterion. To this end, both insert (Algorithm 4) and select (Algorithm 5) maintain a stack of

Algorithm 2: $Q' \leftarrow \text{insert}(n, Q)$ for best-then-dive

Input: n is a node, $Q = (d, H)$ where d is a node or NULL, and H is a set of nodes.

Output: $Q' = (d', H')$ contains all nodes in Q and n.

Data: $f_{\text{est}}^{(n)}$ and $f_{\text{est}}^{(d)}$ are estimated objective values for n and d or ∞ if none are available.

if $d = \text{NULL}$ **then**
  $(d', H') \leftarrow (n, H)$; `// Mark first child for diving.`
else if $f_{\text{est}}^{(n)} < f_{\text{est}}^{(d)}$ **then**
  $(d', H') \leftarrow (n, H \cup \{d\})$; `// Mark best child for diving.`
else
  $(d', H') \leftarrow (d, H \cup \{n\})$; `// Delay dealing with other children.`

Algorithm 3: $(n, Q') \leftarrow \text{select}(Q)$ for best-then-dive

Input: $Q = (d, H)$ where d is a node or NULL, and H is a set of nodes.

Output: n is a node, $Q' = (d', H')$ contains all nodes in Q except n.

Data: For every node m in Q, p_m denotes m's parent and $f_{\text{lb}}^{(p_m)}$ is that parent's lower bound.

if $d \neq \text{NULL}$ **then**
  $(n, d', H') \leftarrow (d, \text{NULL}, H)$; `// Select child marked for diving.`
else
  $(n, d') \leftarrow \left(\text{argmin}_{m \in H} f_{\text{lb}}^{(p_m)}, \text{NULL}\right)$; `// Select best remaining node.`
  $H' \leftarrow H \setminus \{n\}$;

nodes for the depth-first search and a set of deferred nodes. Nodes are moved from the depth-first stack to the deferred set only if they violate the cutoff at the point where they would normally be explored during the depth-first search.

While the choice of cutoff point can, in principle, be adapted based on ancillary data, we restrict ourselves to choosing a fixed convex combination of f_{lb} and f_{ub}. That is, we introduce a parameter $\beta \in [0, 1]$ and a cutoff threshold

$$f_{\text{co}}^{(1)} := f_{\text{lb}} + \beta(f_{\text{ub}} - f_{\text{lb}}).$$

To allow the user to introduce an initial cutoff $f_{\text{co}} < \infty$ when $f_{\text{ub}} = \infty$, we also introduce an additional parameter $\gamma \geq 0$ and a second cutoff threshold

$$f_{\text{co}}^{(2)} := f_{\text{lb}} + \gamma|f_{\text{lb}}|.$$

The actual cutoff threshold is then chosen to be the minimum of both thresholds:

$$f_{\text{co}} := \min\{f_{\text{co}}^{(1)}, f_{\text{co}}^{(2)}\}.$$

We note that for $\alpha = \beta = 0$, gradual backtracking is equivalent to a simple best-first search. For $\alpha = 1$ and $\beta = \infty$, on the other hand, gradual backtracking becomes a depth-first search.

Algorithm 4: $Q' \leftarrow \mathrm{insert}(n, Q)$ for gradual backtracking

Input: n is a node, $Q = (d, (s_1, \ldots, s_k), H)$ with $k \in \mathbb{N}_0$, where d is a node or NULL, s_i is a node for $i \in [k]$, and H is a set of nodes.

Output: $Q' = (d', (s'_1, \ldots, s'_{k'}), H')$ contains all nodes in Q and n.

Data: $\beta \in [0, 1], \gamma \geq 0$. For $m \in \{n, d, s_1, \ldots, s_k\} \cup H$, p_m is m's parent and $f_{\mathrm{lb}}^{(p_m)}$ is its lower bound. $f_{\mathrm{est}}^{(n)}$ and $f_{\mathrm{est}}^{(d)}$ are estimated objective values for n and d or ∞ if none are available. f_{ub} is the current global upper bound.

$f_{\mathrm{lb}} \leftarrow \min_{m \in \{d, s_1, \ldots, s_m\} \cup H} f_{\mathrm{lb}}^{(p_m)}$;

$f_{\mathrm{co}} \leftarrow \min\{f_{\mathrm{lb}} + \beta(f_{\mathrm{ub}} - f_{\mathrm{lb}}),\ f_{\mathrm{lb}} + \gamma|f_{\mathrm{lb}}|\}$;

if $f_{\mathrm{lb}}^{(p_n)} > f_{\mathrm{co}}$ **then**

  $(d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (d, k, s_1, \ldots, s_k, H \cup \{n\})$; `// Defer nodes above cutoff.`

else if $d = \mathrm{NULL}$ **then**

  $(d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (n, k, s_1, \ldots, s_k, H)$; `// Mark first child.`

else if $f_{\mathrm{est}}^{(n)} < f_{\mathrm{est}}^{(d)}$ **then**

  $(d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (n, k+1, d, s_1, \ldots, s_k, H)$; `// Mark best child.`

else

  $(d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (d, k+1, n, s_1, \ldots, s_k, H)$; `// Bypass other children.`

Therefore, gradual backtracking can be seen as an interpolation between these two tree search methods.

To understand the potential benefit of gradual backtracking, assume that the objective function f is convex. Under this assumption, the sublevel set $\{(x, y) \mid f(x, y) \leq f_{\mathrm{co}}\}$ is a convex neighborhood of the solution of the root node relaxation that decreases in size as the global upper bound f_{ub} decreases. This is illustrated in Figure 2. All integer-feasible solutions outside this neighborhood are then implicitly excluded from the search by being located in deferred subtrees. While increasing the lower bound f_{lb} also increases the size of the neighborhood, such increases occur only once it has been established that the sublevel set $\{(x, y) \mid f(x, y) \leq f_{\mathrm{lb}}\}$ does not contain any integer-feasible solutions.

We now come to the subroutine branchvar, which selects the index of a variable to branch on. This routine implements the algorithm's *branching rule*. We use reliability branching [2, 3] on the variables with highest branching priority. Branching priorities for integer control variables are chosen to be equal to the index of the time interval they are associated with. Thus, variables will always be branched on in a reverse time order. While this choice of priorities is based primarily on the observation that regular reliability branching tends to yield a reverse time branching order, prioritized reliability branching is not equivalent in these cases. For comparison, we also test a forward-in-time branching order and the standard method, which assigns equal priority to all variables.

Algorithm 5: $(n, Q') \leftarrow \text{select}(Q)$ for gradual backtracking

Input: $Q = (d, (s_1, \ldots, s_k), H)$ with $k \in \mathbb{N}_0$, where d is a node or NULL, s_i is a node for $i \in [k]$, and H is a set of nodes.

Output: n is a node, $Q' = (d', (s'_1, \ldots, s'_{k'}), H')$ contains all nodes in Q except n.

Data: $\beta \in [0, 1], \gamma \geq 0$. For every node m in Q, p_m denotes m's parent and $f_{\text{lb}}^{(p_m)}$ is that parent's lower bound. f_{ub} denotes the current global upper bound.

$f_{\text{lb}} \leftarrow \min_{m \in \{d, s_1, \ldots, s_m\} \cup H} f_{\text{lb}}^{(p_m)}$;

$f_{\text{co}} \leftarrow \min\{f_{\text{lb}} + \beta(f_{\text{ub}} - f_{\text{lb}}),\ f_{\text{lb}} + \gamma|f_{\text{lb}}|\}$;

if $d = \text{NULL}$ **then**

  `// Backtrack using the DFS stack.`

  $(n, d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (\text{NULL}, d, k, s_1, \ldots, s_k, H)$;

  while $n = \text{NULL}$ *and* $k' > 0$ **do**

    if $f_{\text{lb}}^{(p_{s'_1})} \leq f_{\text{co}}$ **then**

      $(n, k', s'_1, \ldots, s'_{k'}) \leftarrow (s'_1, k' - 1, s'_2, \ldots, s'_{k'})$;

    else

      $(k', s'_1, \ldots, s'_{k'}, H') \leftarrow (k' - 1, s'_2, \ldots, s'_{k'}, H' \cup \{s'_1\})$; `// Defer node.`

  if $n = \text{NULL}$ **then**

    `// Rebase to the best deferred node.`

    $n \leftarrow \text{argmin}_{m \in H} f_{\text{lb}}^{(p_m)}$;

    $(d', k', H') \leftarrow (\text{NULL}, 0, H \setminus \{n\})$;

else

  `// Dive into the marked direct decendant.`

  $(n, d', k', s'_1, \ldots, s'_{k'}, H') \leftarrow (d, \text{NULL}, k, s_1, \ldots, s_k, H)$;

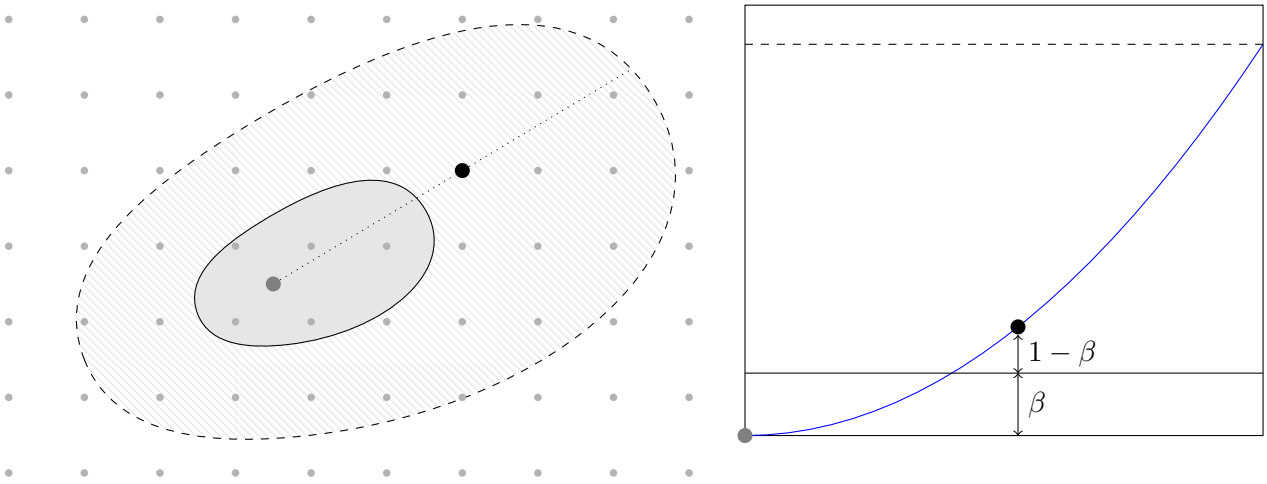
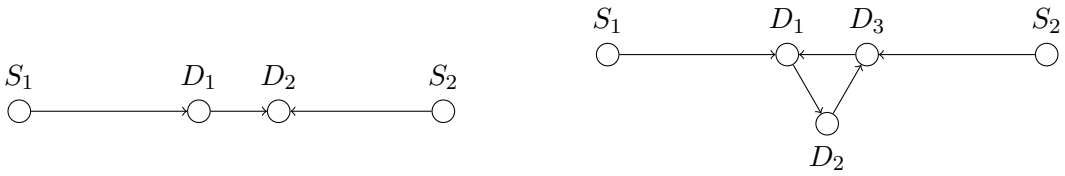


Figure 2: Example of convex search regions induced by the cutoff threshold (dashed region is restricted to solid region because of an integer solution found at the black point; the graph shows objective along dotted line).



(a) Test network "Inversion".

(b) Test network "Triangle".

Figure 3: Network graphs for both test networks.

4 Experiments

We apply our model to two test networks, described in Sections 4.1 and 4.2. For each network, we describe a method by which instances can be randomized. Section 4.3 gives parameters that are common to both networks. The methods described in Section 3 are then applied to six instances of either network. Section 4.5 discusses the results of these experiments and their implications for the solution methods proposed in Section 3.

4.1 Test Network "Inversion"

Our first test network is designed to test the model's capability to model the reversal of flow in a pipeline. The network is deliberately kept simple to ensure that a flow reversal is necessary to meet the specified demand. The network is tree like and consists of three pipelines and four junctions. It is depicted in Figure 3a and its pipe-specific parameters are given in Table 3a. The time horizon of the problem is fixed to $1\,\mathrm{h}$.

Table 3: Pipe parameters for both test networks.

(a) Pipe parameters for "Inversion" network.

Pipe	L_a [km]	D_a [mm]	ϵ_a [mm]	$N_x^{(a)}$
S_1D_1	300	920	0.025	10
D_1D_2	100	920	0.025	6
S_2D_2	300	920	0.025	10

(b) Pipe parameters for "Triangle" network.

Pipe	L_a [km]	D_a [mm]	ϵ_a [mm]	$N_x^{(a)}$
S_1D_1	300	920	0.025	10
D_1D_2	100	920	0.025	6
D_2D_3	100	920	0.025	6
D_3D_1	100	920	0.025	6
S_2D_3	300	920	0.025	10

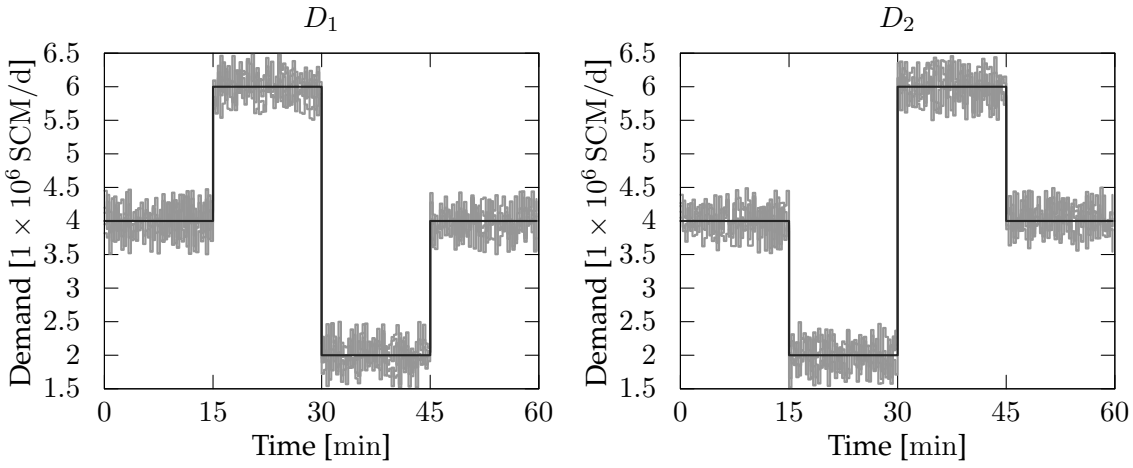


Figure 4: Demand patterns for "Inversion" instances. Black line indicates mean.

Nodes S_1 and S_2 act as supply nodes. Both can supply between $\check{q}_{\text{src}}^{(S_1)}(t) = \check{q}_{\text{src}}^{(S_2)}(t) = 0.0\,\text{SCM/d}$ and $\hat{q}_{\text{src}}^{(S_1)}(t) = \hat{q}_{\text{src}}^{(S_2)}(t) = 4.5 \times 10^6\,\text{SCM/d}$ of natural gas to the network. Nodes D_1 and D_2 act as demand nodes. Their individual demands are randomized according to a normal distribution with a standard deviation $\sigma = 0.25 \times 10^6\,\text{SCM/d}$ and a mean that changes over time as follows.

- For the first $15\,\text{min}$, the mean demand for both D_1 and D_2 is $4.0 \times 10^6\,\text{SCM/d}$.
- From minute 16 to minute 30, the mean demand for D_1 is $6.0 \times 10^6\,\text{SCM/d}$, while the mean demand for D_2 is $2.0 \times 10^6\,\text{SCM/d}$.
- From minute 31 to minute 45, the mean demand for D_1 is $2.0 \times 10^6\,\text{SCM/d}$, while the mean demand for D_2 is $6.0 \times 10^6\,\text{SCM/d}$.
- From minute 46 to minute 60, the mean demand for both D_1 and D_2 is $4.0 \times 10^6\,\text{SCM/d}$.

Any sample drawn from these distributions that deviates from the mean by more than 2σ is re-drawn. This guarantees that the sum of both demands can always be met by the maximal total supply. The demand pattern is shown in Figure 4.

Since a single demand of between $5.5 \times 10^6\,\text{SCM/d}$ and $6.5 \times 10^6\,\text{SCM/d}$ cannot be met by a single supply node, part of the higher of the two demands between minute 16 and minute 45 must

be met by gas traveling through the pipe connecting D_1 and D_2. By switching the roles of D_1 and D_2, we ensure that a flow reversal in that pipe is required to meet demand.

Problem instances using the "Inversion" network are designated `inversion-base`, for an instance without any random fluctuation, or `inversion-random`i for $i \in \{01, 02, 03, 04, 05\}$.

4.2 Test Network "Triangle"

The second test network is designed to test the behavior of our model for networks that contain cycles. The network consists of seven pipelines and five junctions and is therefore necessarily not tree like. Figure 3b depicts the structure of the network and the parameters for individual pipes are given in Table 3b.

Again, nodes designated S_i for $i \in \{1, 2\}$ act as supply nodes, while nodes designated D_i for $i \in \{1, 2, 3\}$ act as demand nodes. Both demand flow rates and supply cost are subject to normally distributed random fluctuation where the mean value is subject to a sinusoidal fluctuation with a period or $1\,\mathrm{h}$, which is equal to the length of the time horizon.

The cost of gas at supply nodes S_1 and S_2 is given by

$$C_{\mathrm{src}}^{(S_i)}(t) := 0.265\,\frac{\mathrm{USD}}{\mathrm{SCM}} + 0.053\,\frac{\mathrm{USD}}{\mathrm{SCM}} \cdot \sin\left(\phi_i + t \cdot 2\pi\,\mathrm{h}^{-1}\right) + X_i,$$

where X_i denotes a random variable that is drawn from a normal distribution with a mean of $0.0\,\mathrm{USD/SCM}$ and a standard deviation of $0.018\,\mathrm{USD/SCM}$. If the resulting cost is below $0.071\,\mathrm{USD/SCM}$, the random variable is redrawn. The phase angles are $\phi_1 = 0$ and $\phi_2 = \pi/2$.

The demand at a given demand node D_i with $i \in \{1, 2, 3\}$ is given by

$$q_{\mathrm{snk}}^{(D_i)}(t) := 10 \times 10^6\,\frac{\mathrm{SCM}}{\mathrm{d}} + 5 \times 10^6\,\frac{\mathrm{SCM}}{\mathrm{d}} \cdot \sin\left(\psi_i + t \cdot 2\pi\,\mathrm{h}^{-1}\right) + Y_i,$$

where Y_i denotes a random variable that is drawn from a normal distribution with a mean of $0.0\,\mathrm{SCM/d}$ and a standard deviation of $0.5 \times 10^6\,\mathrm{SCM/d}$, which is redrawn if it deviates from the mean by more than two standard deviations. The phase angles are $\psi_i = \frac{2\pi}{3}(i-1)$, ensuring that the sum of the mean demands in all three demand nodes is always $30 \times 10^6\,\mathrm{SCM/d}$. The maximum supply flow from either supply node is $24 \times 10^6\,\mathrm{SCM/d}$, ensuring that the maximum momentary demand of $48 \times 10^6\,\mathrm{SCM/d}$ can be met. This demand pattern is shown in Figure 5.

Problem instances using the "Triangle" network are designated `triangle-base`, for an instance without any random fluctuation, or `triangle-random`i for $i \in \{01, 02, 03, 04, 05\}$.

4.3 Common Parameters

Both test networks are optimized on a time horizon of one hour, which is partitioned into $n_t = 180$ equally sized intervals for the purposes of both simulation and control. Therefore, every simulation and continuous control interval corresponds to a timespan of $20\,\mathrm{s}$. In order to keep the number of integer control variables small, only ten integer control intervals are included, so that integer controls remain constant over timespans of $6\,\mathrm{min}$.

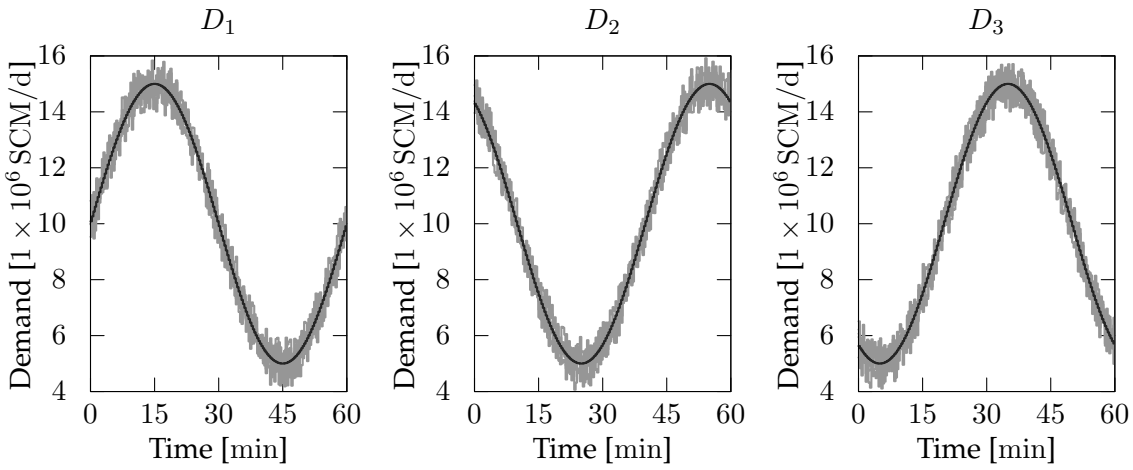


Figure 5: Demand patterns for "Triangle" instances. Black line indicates mean.

Table 4: Numbers of variables (total and integer), constraints (equality, inequality, linear, nonlinear), and nonzeros (constraint Jacobian, objective gradient) for the two test problems.

Problem	Var	Int	Eq	Ineq	Lin	Nonlin	NZ in Constr	NZ in Obj
Inversion	13,486	60	11,572	2,264	8,050	5,786	79,432	2,212
Triangle	19,694	100	16,816	3,752	11,890	8,678	118,764	3,136

The natural gas flowing in the network is assumed to be pure methane, which has the following physical parameters, which are largely adopted from [21]:

$$z = 0.80,\ T = 293.15\,\mathrm{K},\ M = 18.0\,\frac{\mathrm{kg}}{\mathrm{kmol}},\ \rho_0 = 0.72\,\frac{\mathrm{kg}}{\mathrm{SCM}},$$

where the conversion factor ρ_0 corresponds to the density of methane under standard conditions. Note that, along with a universal gas constant of

$$R = 8{,}314.4598\,\frac{\mathrm{J}}{\mathrm{kmol\,K}},$$

this yields a speed of sound of

$$c = \sqrt{\frac{zRT}{M}} = 329.13\,\frac{\mathrm{m}}{\mathrm{s}},$$

meaning that the CFL condition (2.8) is equivalent to

$$\Delta x \geq 6.583\,\mathrm{km},$$

which is satisfied by the pipe parameters laid out in Table 3.

4.4 Experimental Setup

All test instances are solved by using a modified version of Minotaur 0.2.0, using IPOPT 3.12 as a relaxation solver. The exact dimensions of the resulting optimization problems are given in Ta-

ble 4. Linear equation systems within IPOPT are solved by using the direct solver MA57 from the Harwell Subroutine Library[2], using a single-threading implementation of BLAS and LAPACK provided by the COIN-OR[3] project. Therefore, all optimization solvers are entirely single-threading with no attempt of parallelization.

Test instances are solved on single compute nodes with Intel E5430 Xeon CPUs and $32\,\mathrm{GB}$ of DDR2 $667\,\mathrm{MHz}$ (4 modules of $8\,\mathrm{GB}$) memory each. Our primary performance metric is wall time, which may be unreliable because the compute nodes are shared with other applications.

We employ six methods of solving the problem:

- `bthend-fwdbr`: Best-then-dive tree search with forward-in-time branching order;
- `bthend-revbr`: Best-then-dive tree search with reverse-in-time branching order;
- `bthend-stdbr`: Best-then-dive tree search with normal reliability branching;
- `rlad-fwdbr`: Gradual backtracking tree search with forward-in-time branching order;
- `rlad-revbr`: Gradual backtracking tree search with reverse-in-time branching order;
- `rlad-stdbr`: Gradual backtracking tree search with normal reliability branching.

For gradual backtracking, we use an initial target gap $\alpha = 1.0$ and a target gap decay rate of $\beta = 0.1$. The branch-and-bound algorithm is prematurely terminated if the relative integer gap falls below $0.1\,\%$ or the tree search lasts for more than $48\,\mathrm{h}$.

4.5 Experimental Results

Figure 6 shows the total runtime, total number of branch-and-bound nodes processed, and average runtime per node for each instance and solution method. We omit the results for best-then-dive with forward-in-time branching on instance `triangle-random02` because IPOPT exceeds the maximum number of iterations on most subproblems for this method, which leads to a misleadingly low number of processed nodes and a very high average processing time per node. Most notable in the remaining results is the fact that except for `inversion-random05`, no branching rule other than conventional reliability branching terminates in under $48\,\mathrm{h}$ for any instance or solution method, indicating that both forward- and reverse-in-time branching do not significantly improve the overall performance of the solver. The fact that this is the case for both best-then-dive and gradual backtracking appears to indicate that this is not due to a negative interaction between gradual backtracking and time-based branching orders.

In cases where standard reliability branching is used, gradual backtracking appears to provide a significant advantage in both total runtime and the number of nodes processed to reach a solution. This appears most pronounced in instances using the more complex "Triangle" network. While the average time needed to process a single branch-and-bound node in a "Triangle" instance appears to decrease with the total number of branch-and-bound nodes processed, this is likely due to a higher concentration of strong branching solves.

[2] Version 2014.01.10, available at http://www.hsl.rl.ac.uk/

[3] See https://www.coin-or.org/

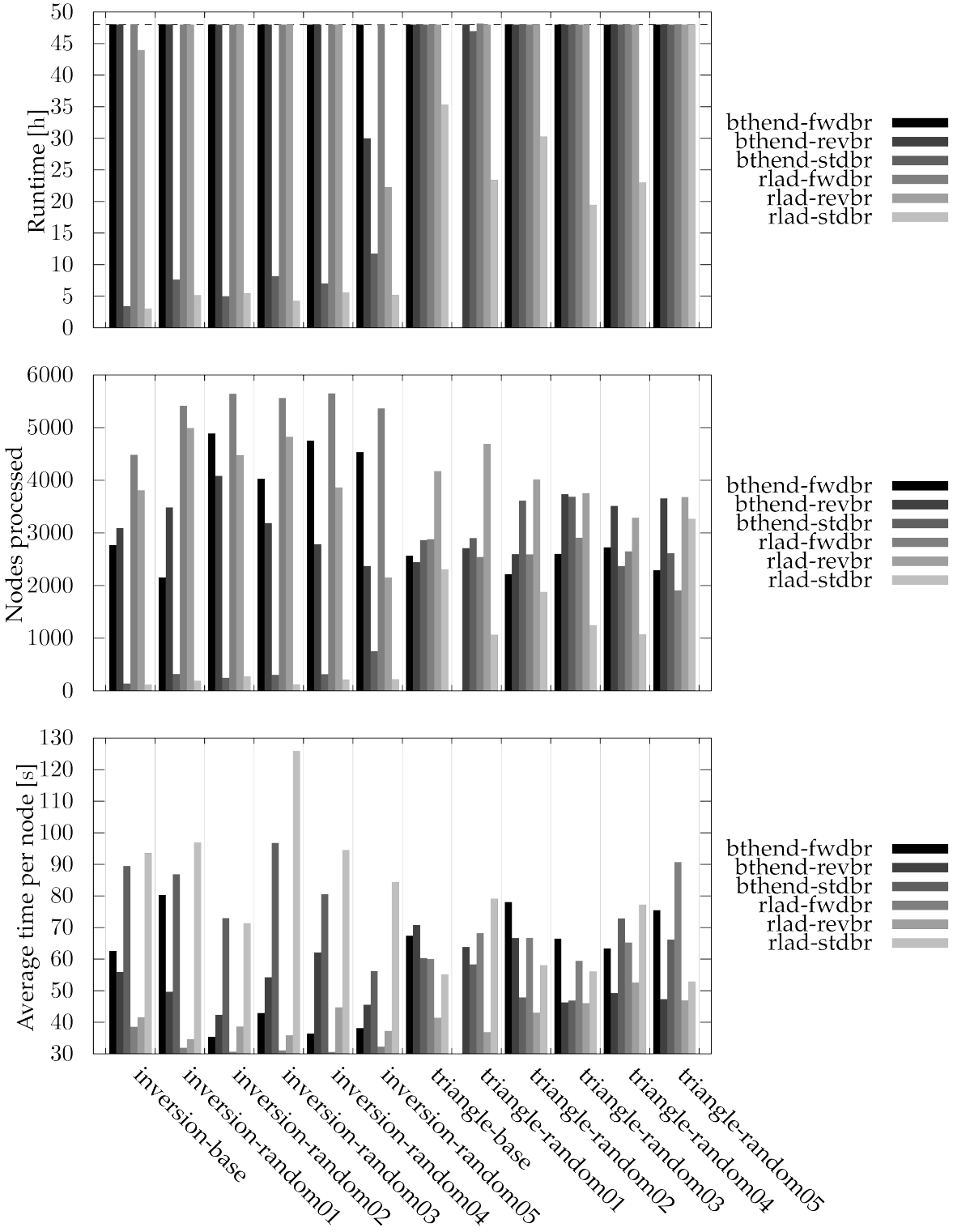


Figure 6: Runtime, nodes processed, and average node processing time for different instances (dashed line indicates runtime limit).

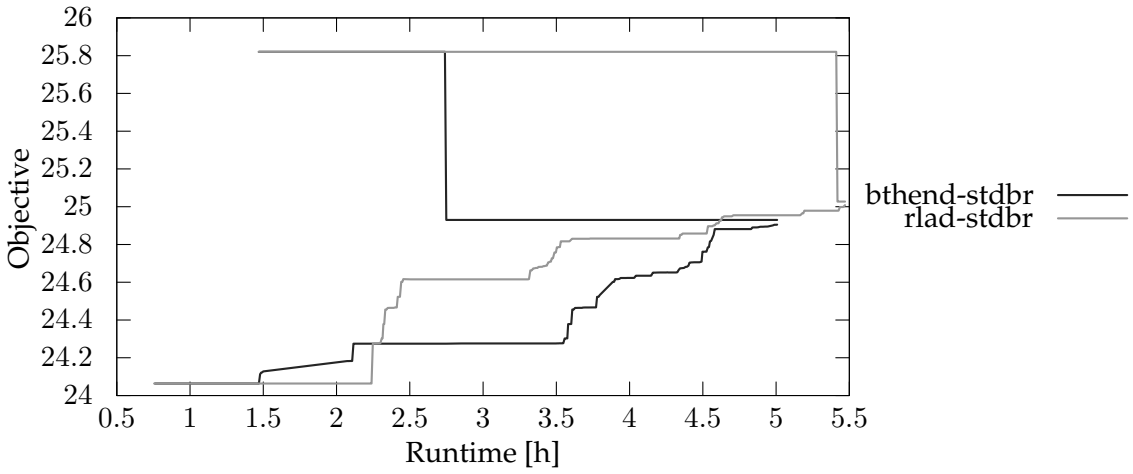


Figure 7: Progression of objective bounds over time for `inversion-random02`.

Note that while gradual backtracking performs better than best-then-dive in almost all instances, this is not the case for `triangle-random02`, which is the only instance for which best-then-dive outperforms gradual backtracking. Figure 7 shows the progression of global objective bounds over time for this particular instance. It appears that, likely due to nonconvexity, best-then-dive and gradual backtracking find different solutions in this instance. This does not occur in any other test instance.

Figure 8 shows that, in instances using the "Triangle" test network, both forward- and reverse-in-time branching orders often improve the quality of the first integer-feasible solution found. The lower initial integer optimality gap may go some way to explaining the poor performance of gradual backtracking with these branching orders. With a lower initial gap, the initial cutoff threshold is also tighter, making gradual backtracking behave more like best-first node selection. However, it is not immediately apparent how one might avoid this without introducing more complex methods of choosing the cutoff point.

Flow Reversals

We now turn our attention to the solution generated for `inversion-base` using `rlad-stdbr`, which serves as an example of the "Inversion" problem. As noted in Section 4.1, this network was specifically designed to test our model's ability to depict flow inversions.

Our interest in flow reversals lies in the fact that we restrict our compressors in such a way that they cannot push directly against the flow of gas, making a flow reversal an event that requires intervention from other network components. Indeed, the aforementioned solution shows this. Figures 9 and 10 show the pipe boundary pressures at the inlet sides of the supply pipes (S_1, D_1) and (S_2, D_2). Given a speed of sound of $c = 329.13\,\mathrm{m/s}$ and a pipe length of $L = 300\,\mathrm{km}$ for both pipes, the expected travel time for a pressure wave is between $900\,\mathrm{s}$ and $920\,\mathrm{s}$, or slightly over $15\,\mathrm{min}$. An accordingly shifted version of the inlet pressure curve is shown in both figures,

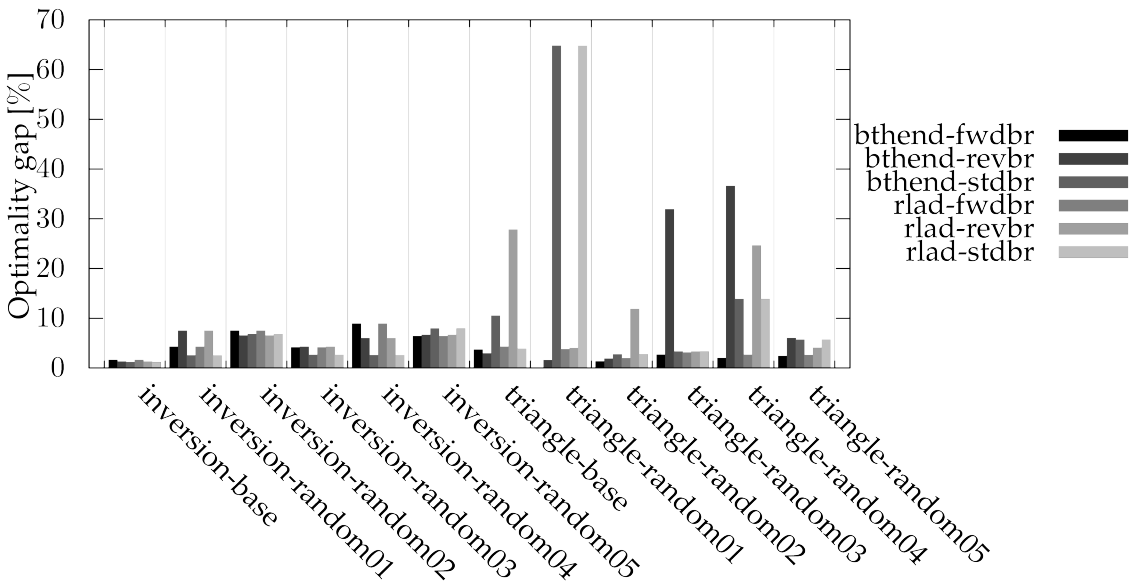


Figure 8: Relative integer optimality gap of first solution for different instances.

indicating the projected outlet pressure assuming constant initial pressure and complete absence of friction.

We see in Figure 10 that pressure at S_2 spikes immediately after the beginning of the simulation and around $30\,\mathrm{min}$ later. The former pressure wave is projected to reach D_2 approximately $15\,\mathrm{min}$ after the beginning of the simulation, causing flow in the exchange pipe (D_1, D_2) to become negative, which can be seen in Figure 11. Accordingly, the second spike reaches D_2 at around $45\,\mathrm{min}$, where it slows the flow in the exchange pipe in order to return the network to an equilibrium. Similarly, Figure 9 shows a pressure spike at S_1 at around $15\,\mathrm{min}$ that would reach D_1 at around $30\,\mathrm{min}$, when the flow in the exchange pipe reverses direction.

These results indicate that the direction of flow in the exchange pipe is controlled mainly by the pressure waves from the supply nodes S_1 and S_2. Models with the ability to depict flow reversals may be of significant benefit in developing control configurations that result in flow reversals since reversals require complex interactions between different network components that may not be intuitive to human operators.

5 Conclusions and Outlook

In this paper, we have introduced an optimization model for transient optimal control of gas transport networks. Our overall goal was to find inexpensive control configurations that satisfy fluctuating demand. We model the flow of gas through these networks using a system of partial differential equations derived from the Euler equations, which we discretize using a finite volume method, making our model independent of prior knowledge of direction of flow, thus broadening the range of allowed inputs beyond what is allowed by comparable models. We also showed that

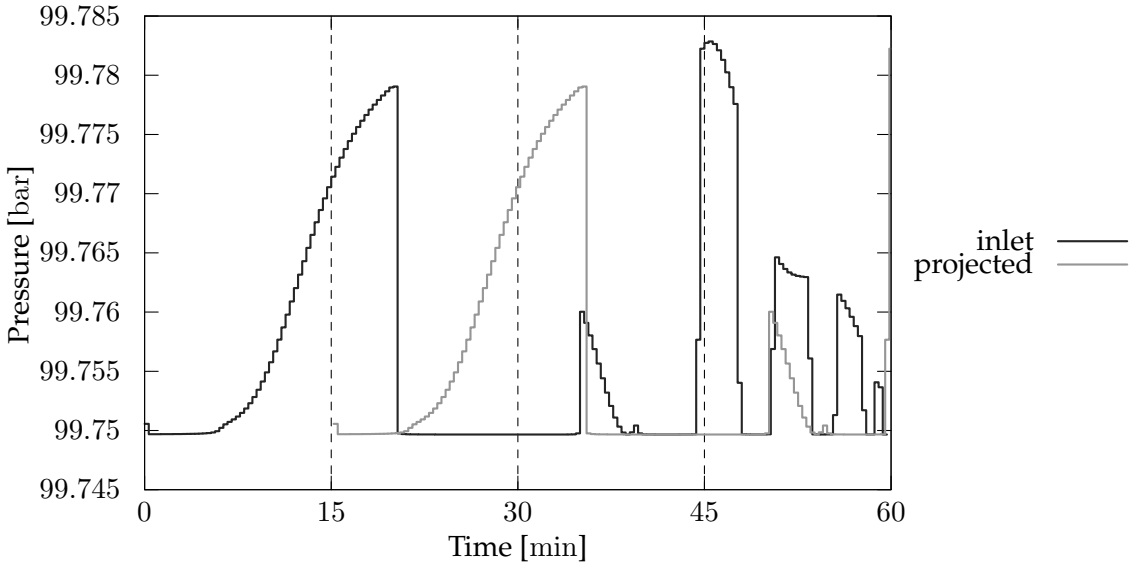


Figure 9: Inlet and projected outlet pressure at (S_1, D_1) in instance `inversion-base`.

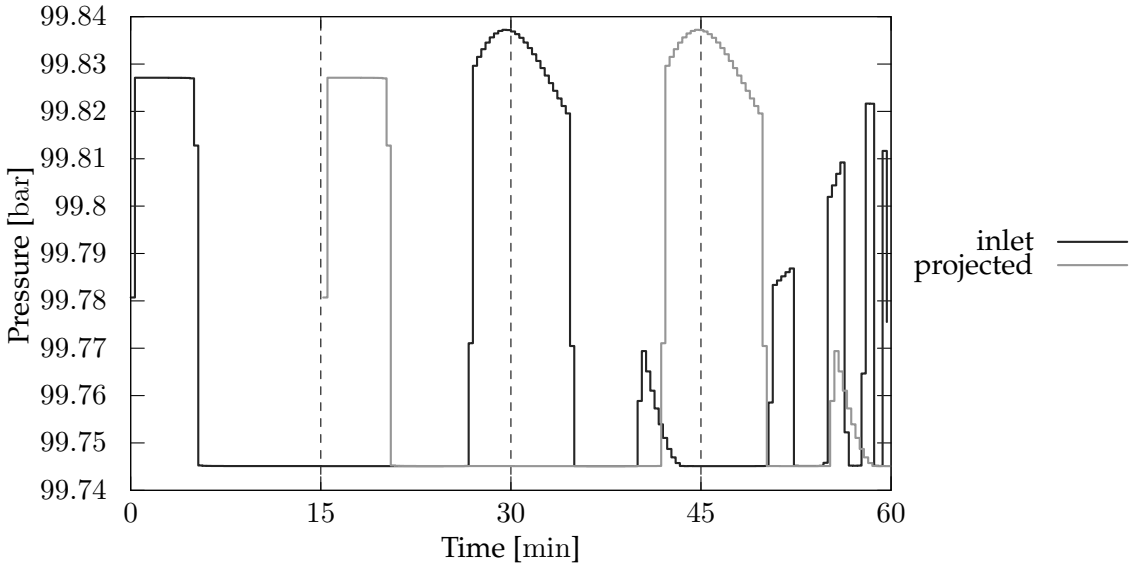


Figure 10: Inlet and projected outlet pressure at (S_2, D_2) in instance `inversion-base`.

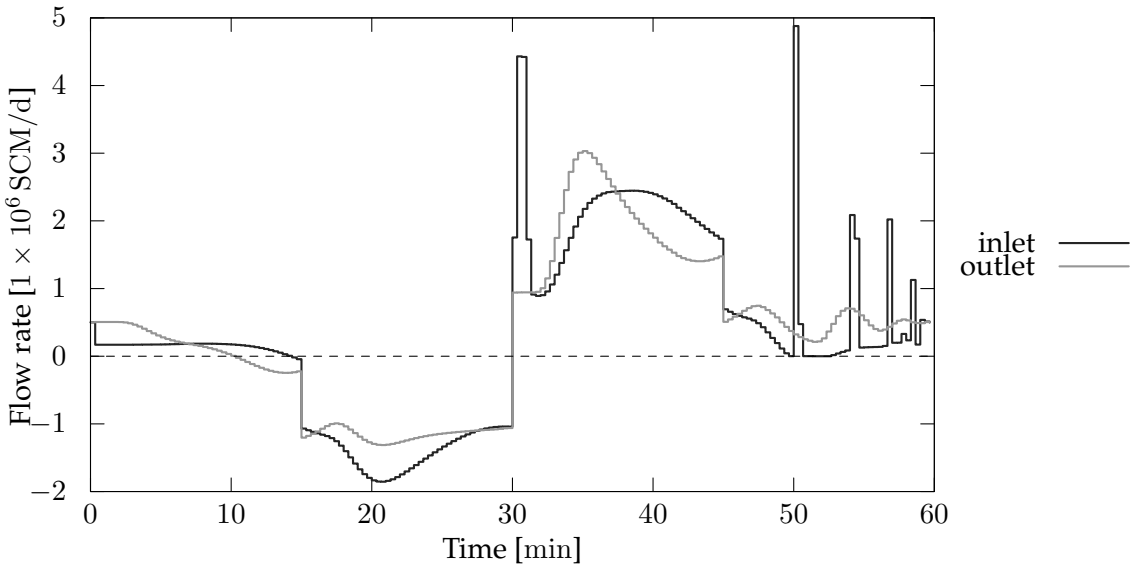


Figure 11: Inlet and outlet flow rates at (D_1, D_2) in instance `inversion-base`.

interesting insights into ways to produce more severe changes in the network state, such as flow reversals, can be gained by using our model.

In modeling the compressor stations used to steer the flow of gas in our network, we introduced time-dependent binary switches to the problem, thus making our problem part of the challenging field of mixed-integer PDE-constrained optimization. While mixed-integer optimal control problems can often be solved by using good approximate solution methods, our problem is inaccessible to these methods because of the complex relationship between controls and PDE solutions. We presented several modifications to branch-and-bound solvers that exploit the same structural aspects as approximate solution methods to accelerate the solution process.

In several randomized instances of two test problems, we showed that, particularly by applying a gradually backtracking tree search method, we can substantially improve the performance of branch-and-bound solvers. However, this tree search method appears to negatively interact with branching orders that attempt to take into account the direction of the arrow of time, which is commonly exploited in approximate solvers for mixed-integer optimal control.

Future avenues of research therefore include ways of dynamically choosing the cutoff point for gradual backtracking to adapt to small integer optimality gaps. To scale the model to larger networks, we plan to develop implementations using dedicated simulators and parallelization using distributed scientific computing libraries, specifically the PETSc toolkit.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. This

work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.

References

[1] Polar vortex review. Technical report, North American Energy Reliability Corporation, September 29, 2014.

[2] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

[3] Pierre Bonami, Jon Lee, Sven Leyffer, and Andreas Wächter. On branching rules for convex mixed-integer nonlinear optimization. *Journal of Experimental Algorithmics (JEA)*, 18:2–6, 2013.

[4] CF Colebrook and CM White. Experiments with fluid friction in roughened pipes. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 367–381, 1937.

[5] Robert Fourer, David M Gay, and Brian W Kernighan. *AMPL: A mathematical programming language*. AT&T Bell Laboratories, Murray Hill, NJ 07974, 1987.

[6] M Gattrell, N Gupta, and A Co. Electrochemical reduction of CO_2 to hydrocarbons to store renewable electrical energy and upgrade biogas. *Energy Conversion and Management*, 48(4):1255–1265, 2007.

[7] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[8] Martin Gugat, Günter Leugering, Alexander Martin, Martin Schmidt, Mathias Sirvent, and David Wintergerst. MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems. *Preprint*, 2017.

[9] Benjamin Hiller, René Saitenmacher, and Tom Walther. Analysis of operating modes of complex compressor stations. pages 251–257, 2018.

[10] Jesco Humpola, Imke Joormann, Djamal Oucherif, Marc E Pfetsch, Lars Schewe, Martin Schmidt, and Robert Schwarz. GasLib – a library of gas network instances, 2015.

[11] Thorsten Koch, Benjamin Hiller, Marc E Pfetsch, and Lars Schewe. *Evaluating gas network capacities*. SIAM, 2015.

[12] Ta-Tsien Li and Yi Jin. Semi-global C1 solution to the mixed initial-boundary value problem for quasilinear hyperbolic systems. *Chinese Annals of Mathematics*, 22(03):325–336, 2001.

[13] Alexander Martin, Markus Möller, and Susanne Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical programming*, 105(2-3):563–582, 2006.

[14] Robert Merriam. Natural gas monthly. Technical report, U.S. Energy Information Administration, May 2017.

[15] A Osiadacz. Nonlinear programming applied to the optimum control of a gas compressor station. *International Journal for Numerical Methods in Engineering*, 15(9):1287–1301, 1980.

[16] Andrzej Osiadacz. Simulation of transient gas flows in networks. *International Journal for Numerical Methods in Fluids*, 4(1):13–24, 1984.

[17] Sebastian Sager, Michael Jung, and Christian Kirches. Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3):363, 2011.

[18] Adam Sieminski. Annual energy outlook 2017. Technical report, U.S. Energy Information Administration, 2017.

[19] PK Swanee and Akalank K Jain. Explicit equations for pipeflow problems. *Journal of the Hydraulics Division*, 102(5), 1976.

[20] Suming Wu, Roger Z Rios-Mercado, E Andrew Boyd, and L Ridgway Scott. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modelling*, 31(2-3):197–220, 2000.

[21] Victor M Zavala. Stochastic optimal control model for natural gas networks. *Computers & Chemical Engineering*, 64:103–113, 2014.

[22] Anatoly Zlotnik, Michael Chertkov, and Scott Backhaus. Optimal control of transient flow in natural gas networks. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4563–4570. IEEE, 2015.