Optimization and Engineering
https://doi.org/10.1007/s11081-020-09559-y

RESEARCH ARTICLE

Controlling transient gas flow in real-world pipeline intersection areas

Felix Hennings[1,2] · Lovis Anderson[1] · Kai Hoppmann-Baum[1,2] · Mark Turner[1,2] · Thorsten Koch[2,1]

Received: 1 November 2019 / Revised: 27 August 2020 / Accepted: 4 September 2020

Abstract

Compressor stations are the heart of every high-pressure gas transport network. Located at intersection areas of the network, they are contained in huge complex plants, where they are in combination with valves and regulators responsible for routing and pushing the gas through the network. Due to their complexity and lack of data compressor stations are usually dealt with in the scientific literature in a highly simplified and idealized manner. As part of an ongoing project with one of Germany's largest transmission system operators to develop a decision support system for their dispatching center, we investigated how to automatize the control of compressor stations. Each station has to be in a particular configuration, leading in combination with the other nearby elements to a discrete set of up to 2000 possible feasible operation modes in the intersection area. Since the desired performance of the station changes over time, the configuration of the station has to adapt. Our goal is to minimize the necessary changes in the overall operation modes and related elements over time while fulfilling a preset performance envelope or demand scenario. This article describes the chosen model and the implemented mixed-integer programming based algorithms to tackle this challenge. By presenting extensive computational results on real-world data, we demonstrate the performance of our approach.

Keywords Transient gas network optimization · Mixed-integer programming · Realistic modeling of compressor stations

✉ Felix Hennings
hennings@zib.de

Extended author information available on the last page of the article

Published online: 03 October 2020

1 Introduction

Throughout the past years, the mathematics of gas transport has been an intensively studied topic. Natural gas was, is, and will be one of the primary energy sources in Germany, making efficient and safe transport a field of high economic and political relevance according to the German Federal Ministry for Economic Affairs and Energy (2019). Furthermore, the task is also challenging from a mathematical point of view. One such challenge is in the compressors, which push the gas through the network by increasing its pressure. Compressors are typically set up as a compressor station consisting of multiple compressor units, each representing the union of one single compressor machine and its associated drive. These compressor units are dynamically switched on or off and used in different sequences to meet the current needs in terms of compression ratios and flow rates. Out of the theoretically possible arrangements of compressor units, the set of technically feasible arrangements are known as the configurations of a compressor station. The intersections of major transportation pipelines are usually composed of multiple compressor stations as well as other elements like valves or pressure regulators. Such a structure makes it possible to adjust the connections of the intersecting pipelines and operate the system for various pressure levels. To optimize control of these areas, which includes all the discrete decisions that need to be made for the single elements while taking into account the corresponding technical restrictions, is itself already a combinatorial challenge. The physics of gas flow further increases the complexity of the problem. In pipes, the Euler Equations describe the flow of gas. This set of nonlinear hyperbolic partial differential equations (PDEs) yields even in simplified versions computationally challenging constraints, see Osiadacz (1996).

Research in this field first focused on the simulation of gas flow, i.e., dealing with the partial differential equations given all the discrete decisions. This topic has been studied for many decades already, see for example Brouwer et al. (2011) and the references therein. In recent years however, the optimization of gas transport, including the consideration of the combinatorial aspects, has gained increased attention. In the article of Ríos-Mercado and Borraz-Sánchez (2015), a general overview of optimization problems related to natural gas is given, including the transport of gas. The majority of the referenced literature considered the stationary (or steady-state) gas transport problem. There, they deal with one stable network state which allows for an algebraic description of the gas flow. Koch et al. (2015) and Pfetsch et al. (2014) give an overview of state-of-the-art approaches tackling the stationary problem on large real-world instances with a considerable amount of detail regarding the different network elements.

The more challenging variant is the transient gas transportation problem, which we consider in this article. Here, the goal is to find a set of control decisions for the elements over a future time horizon discretized into individual time steps. For this problem, research is still in the early stages. One of the first publications on transient gas transport optimization was the thesis of Moritz (2007) in which they presented a mixed-integer programming (MIP) model for the problem. In contrast to the possible configurations described above, they only used a model consisting of single compressor units, whose compression capabilities are limited by a given minimum and maximum value for the power used for the compression. The nonlinear constraints resulting from these power bounds, as well as the nonlinear pipe equations, are approximated by piecewise linear functions. To solve the model for the objective of minimizing a combination of the compressors' fuel consumption and the cost to switch the elements, they used a special branching scheme for the piecewise linear functions as well as a simulated annealing heuristic described in the article of Mahlke et al. (2007). A little later, Domschke et al. (2011) also presented a solution to the problem of minimizing the fuel consumption of compressors. They considered a similar problem formulation as Moritz (2007), using the same model for compressors and approximating the nonlinear constraints by piecewise linear functions. However, they combined solving this MIP with solving a nonlinear problem (NLP) formulation of the problem in an alternating way. From the solution of that nonlinear problem, they deduce a refinement of the piecewise linear approximations and repeat this procedure until finally ending up with a solution to the overall mixed-integer nonlinear problem (MINLP) within a chosen approximation error. In other articles dealing with transient transport optimization, the discrete nature of the problem is neglected, which leads to NLP formulations. As an example, we mention the work of Zlotnik et al. (2015) and Mak et al. (2016), who decide on the compression ratios of compressors that are assumed to be always active, while again minimizing their fuel consumption. Very recently, a few more studies on transient gas network optimization have been published. Hahn et al. (2017) use a specialized branching rule to solve a MINLP formulation of the problem with the objective of minimizing fuel consumption. For the compressors, they introduced the theoretical concept of different modes to switch between configurations, which each have separate feasible regions. However, in the end, they restrict to exactly one mode with nearly unlimited compression capabilities for their experiments. Another approach combining different specialized solving techniques is presented by Gugat et al. (2018), where a MIP model and an NLP model are iteratively solved for each single time step to determine an instantaneous control of the problem, i.e., a control that only looks one time step ahead when making a decision. The two used models are based on a specific discretization of the Euler equations. For compression, both models use single compressor units featuring a linear feasible region. In contrast to the other previously mentioned publications, Gugat et al. (2018) do not minimize fuel consumption. Rather, the objective is to deviate as little as possible from a set of future pressure and flow values given at the boundary nodes. To close, we mention the work of Burlacu et al. (2019), who consider maximizing the amount of temporarily stored gas in the network while maintaining a feasible transient control of the elements. They also introduce a new discretization of the Euler Equations resulting in a formulation close to the algebraic form of the stationary model. However, for the compressors, they model only single units restricted by upper and lower bounds in the compression ratio as well as the absolute pressure difference. This problem is solved by alternating between solving a MIP model, which they obtain by replacing the nonlinear constraints by piecewise linear functions, and an NLP model, in which the discrete decisions from the MIP are already fixed. Using this approach they are able to obtain globally optimal solutions for their model.

All approaches in the publications on transient gas network optimization mentioned above use a somewhat idealized model for the compressor stations. In contrast to this, we present in this article a transient gas network optimization problem featuring a compressor model with a so-far unmatched amount of detail. The model is based on the modeling presented by Koch et al. (2015) for the stationary case and takes the above described substructures for compressors into account. We consider the problem not on the complete network, but on those individual network areas that contain the compressor stations as well as additional active elements. We call these subgraphs *network stations*, an example station is presented in Fig. 1. The set of all network stations contains the majority of active elements in the whole network. Regarding the number of contained elements, a single network station is comparable to or even larger than the networks considered in the literature on transient optimization. Hence, they constitute challenging instances even though only representing a subgraph of the original gas network. Furthermore, precise control recommendations for network stations can help gas network operators to manage critical situations that arise during day-to-day operation.

One difference to general gas network problems is the shortness of pipes due to the proximity of the elements in a network station, see Table 1 in Sect. 5. Because of their shortness, the pipes' ability to store gas is negligible owing to their small volumes. Furthermore, the pressure loss induced by friction in the pipe depends on its length and has therefore reduced impact. This allows us to use a linear pipe model as introduced by Hennings (2018) without losing much accuracy and still producing realistic results.

As inputs to the problem, we are given the topology of the network station under consideration, an initial state for each of the contained network elements, and future demands in terms of both inflow and pressure levels at the station's boundaries. The goal is to then find a feasible control of all the network elements over time, which can be interpreted as a recommendation for network operators on how to best operate the network station in the future. The overall objective is split into two parts and will be realized as the minimization of one weighted sum over the different terms. On the one hand, we try to meet the future demands as best as possible, i.e., allowing and penalizing the deviation from these. On the other hand, we minimize the total number of control changes. Having such a stable control with a small number of changes is preferable since it reduces strain on the technical elements and enables the gas network operators to understand and actually perform the desired control recommendations.

To solve this problem, we formulate it as a MIP model. However, the model turns out to be quite challenging, according to our experiments. On larger network stations, state-of-the-art MIP solvers cannot solve it reliably, even for a small number of future time steps. For this reason, we present a heuristic algorithm that can efficiently and reliably solve the problem also on harder instances. Its main idea is to determine a majority of the discrete control decisions by solving independent stationary variants of the original MIP model for each of the future time steps. Based on these we create a solution for the overall transient problem by again solving a variant of the original MIP. Apart from the complexity reduction gained from splitting the MIP model, our algorithm also allows to

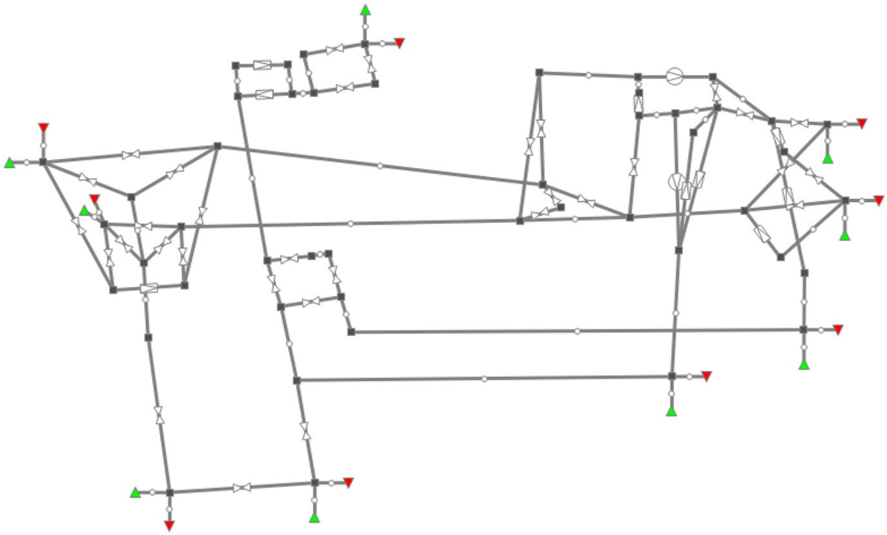


Fig. 1 Example of the medium-size network station E, see Table 1 for more details to its properties. The colored triangles represent the entry ▲ and exit ▼ nodes of the station. Furthermore, we have denoted the single network elements by —∘— (pipe), ⋈ (valve), ▭ (regulator), and ⊘ (compressor station). (Color figure online)

enforce certain technical restrictions outside of its MIP subroutines. Note that the decisions determined by the stationary model yield good solutions for the original transient problem too, since network stations only have a limited amount of pipeline volume due to the pipes' shortness. Therefore, only small amounts of gas can be stored in the network station, which makes the corresponding transient optimization problem a "stationary-flavored" one.

The rest of the article is structured as follows: In Sect. 2, we will describe the mathematical constraints for all used elements and formulate the original MIP model. We highlight that we do not present a MIP formulation for the restrictions introduced in Sect. 2.8 since these will be satisfied by our heuristic algorithm without including them in the solved model variants. In Sect. 3, we describe the preprocessing needed to convert the given compressor data into a linear description of the feasible operating range. Our heuristic algorithm is the topic of Sect. 4, which also introduces additional variants of the MIP model. Sect. 5 then presents the results of our algorithm when applied to a large number of real-world network situations. We finish by concluding this work in Sect. 6.

2 Mathematical model

In this section, we give a detailed description of the considered transient gas network optimization problem on a network station. At the same time, we introduce the corresponding variables and constraints to create a MIP formulation for the problem.

The exception to this is the restrictions presented in Sect. 2.8 for which the formulation is not given. Our algorithm developed in Sect. 4 produces control decisions that satisfy these restrictions without their explicit inclusion in any MIP subroutine. Therefore, a corresponding MIP formulation is not required for these restrictions.

We model the topology of a network station as a directed graph $G = (\mathcal{V}, \mathcal{A})$ in which the arcs $\mathcal{A}$ represent the different network elements and nodes $\mathcal{V}$ represent the junctions of the arcs. We split $\mathcal{A}$ into individual sets $\mathcal{A} = \mathcal{A}^{\mathrm{pi}} \cup \mathcal{A}^{\mathrm{va}} \cup \mathcal{A}^{\mathrm{rs}} \cup \mathcal{A}^{\mathrm{rg}} \cup \mathcal{A}^{\mathrm{cs}}$ for the network elements contained in a network station, i.e., pipes, valves, resistors, regulators, and compressor stations. Note that regulators are also often referred to as control valves in literature, e.g., in Ehrhardt and Steinbach (2003) or Koch et al. (2015). Likewise, we split the set of nodes $\mathcal{V} = \mathcal{V}^{\flat} \cup \mathcal{V}^{0}$ into boundary nodes and inner nodes of the network station, respectively. Here, boundary nodes $\mathcal{V}^{\flat}$ represent those having inflow and pressure level demand values for future time steps. We define the set of considered time steps as $\mathcal{T}_0 := \{0, \dots, k\}$ where $\mathcal{T} := \mathcal{T}_0 \setminus \{0\}$ is the set of future time steps having demand conditions at boundary nodes. Each time step t has an associated value $\tau(t)$ representing the time difference in seconds from t to the initial state time step 0. However, some restrictions may not be aligned with this discrete set of time points but consider intermediate time points instead. For those cases, we define that all variable values at some $t \in \mathcal{T}_0$ are present for the open interval between t and the subsequent discrete time point $t + 1$ at which a new value might be set.

The essential quantities we use to describe the gas flow are pressure and mass flow. We have pressure variables $p_{v,t}$ at each node $v \in \mathcal{V}$ and time $t \in \mathcal{T}_0$ as well as variables $q_{a,t}$ for the flow from ℓ to r on each non-pipe arc $(\ell, r) = a \in \mathcal{A} \setminus \mathcal{A}^{\mathrm{pi}}$ and time $t \in \mathcal{T}_0$. For a pipe $(\ell, r) = a \in \mathcal{A}^{\mathrm{pi}}$, we have two flow variables $q_{\ell,a,t}$ representing the inflow into the pipe at end node ℓ and $q_{r,a,t}$ representing the outflow out of the pipe at end node r, similar to the model of Ehrhardt and Steinbach (2003) on pipes or the one of Domschke et al. (2011) on all arcs. The last flow quantity we introduce is the inflow $d_{v,t}$ entering the network station at each boundary node $v \in \mathcal{V}^{\flat}$ and $t \in \mathcal{T}_0$. Although we are given flow demands for the future, we allow deviations from them, and hence have to have a variable capturing the actual inflow value.

We assume that there are upper and lower bounds on all stated quantities for each point in time $t \in \mathcal{T}_0$, represented by bars above and below the variable, respectively. Note that while pressure is always positive, flow can be negative, representing flow against its orientation in the case of arcs and flow out of the network in the case of boundary nodes.

Gas flowing through a network always induces a loss of pressure in flow direction due to friction. However, we only consider friction-induced pressure loss in pipes and resistors. For valves, regulators, and compressor stations, we either assume the friction to be small enough to be negligible or to be captured by resistors surrounding the element in the network station.

For the formulation of the MIP model, we often use *indicator constraints*. Here, a constraint $a^T x \leq a_0$ for a set of variables $x = \{x_1, \dots, x_i\}$ with $a \in \mathbb{R}^i$, $a_0 \in \mathbb{R}$ is only active when a logical condition holds, i.e., a binary variable y attains the value $b \in \{0, 1\}$. We denote such a constraint by

$$y = b \quad \rightarrow \quad a^T x \leq a_0.$$

Since we have bounds on all the variables, constraints of this kind can be reformulated as linear constraints in a *bigM approach*, see Bonami et al. (2015) for more details. A list of all used variables can be found in the appendix in Table 3.

2.1 Compressor stations

Compressor stations are the most complex elements in the network. They are responsible for increasing the pressure and thereby are essential in controlling the gas flow. We based our model on the description of Fügenschuh et al. (2015), where the elements are called *compressor groups* instead.

2.1.1 Structure of a compressor station

A compressor station $(\ell, r) = a \in \mathcal{A}^{\mathrm{cs}}$ has three different modes, referred to as bypass, closed, and active mode. In bypass mode, the element is bypassed and therefore allows unrestricted gas flow without changing the pressure level. For the closed mode, the element blocks the gas flow, which disconnects the network between both end nodes. Finally, in active mode, gas is compressed and the pressure is increased along the direction of the flow.

When compressing, the station uses its set of associated compressor units $\mathcal{U}_a$. These represent the union of a single turbo compressor machine, the technical element performing the actual compression of the gas, and a corresponding drive delivering the power needed for this process. In the compressor station, these units are combined in series and/or parallel to allow proper reactions to different compression requirements. The set of all allowed serial-parallel compressor unit combinations is called the set of configurations $\mathcal{C}_a$ for a compressor station $a \in \mathcal{A}^{\mathrm{cs}}$, from which exactly one has to be chosen if the compressor station is in active mode. For each of these configurations $c \in \mathcal{C}_a$, we create a polytope describing the feasible operating range of the compressor station using configuration c. The polytope is embedded in the three dimensional space of incoming pressure p_ℓ, outgoing pressure p_r and flow q. It is described as the intersection of a set of half-spaces $\mathcal{H}_c = \{(\alpha_0, \alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^4\}$ encoding inequalities of the form $\alpha_0 \cdot p_\ell + \alpha_1 \cdot p_r + \alpha_2 \cdot q + \alpha_3 \leq 0$. In Sect. 3 we describe how we create these feasible operating range polytopes for the single configurations of the compressor stations based on the operating ranges of the corresponding compressor units used in the configuration.

2.1.2 Compressor station model

To model the above-described constraints, we use the disjunctive formulation of Balas (1985, 2018). Thus, we introduce binary "selection" variables $m^{\mathrm{cf}}_{c,a,t}$ for each configuration $c \in \mathcal{C}_a$ of a compressor station $(\ell, r) = a \in \mathcal{A}^{\mathrm{cs}}$, as well as $m^{\mathrm{by}}_{a,t}$ and $m^{\mathrm{cl}}_{a,t}$ for bypass and closed mode, respectively, where a value of 1 indicates the selected configuration or mode. Furthermore, we introduce corresponding sets of pressure and flow variables:

$p_{a,t}^{\text{by}} \quad q_{a,t}^{\text{by}}$	bypass mode variables
$p_{a,t}^{\text{l-cl}} \quad p_{a,t}^{\text{r-cl}}$	closed mode variables
$p_{c,a,t}^{\text{l-cf}} \quad p_{c,a,t}^{\text{r-cf}} \quad q_{c,a,t}^{\text{cf}} \quad \forall c \in \mathcal{C}_a$	configuration variables

Note that we do not need all variables for each mode since $p_\ell = p_r$ holds true for bypass, and $q = 0$ holds true for closed. Using the binary variables, we force the pressure and flow variables of all non-selected configurations or modes to be zero and only enforce the constraints of the selected configuration or mode. The actual pressure and flow variables of the compressor station can hence be represented as the sum of the corresponding variables for the configurations and modes. The bounds of these variables specific to the configurations and modes are equal to those for the compressor station. The bound interval may not include zero however. In these cases we expand the interval by replacing its endpoint with the smallest absolute value by zero to make the non-selected variable values feasible.

The formal constraints for all $(\ell, r) = a \in \mathcal{A}^{\text{cs}}$ and $t \in \mathcal{T}$ are then given as:

$$1 = \sum_{c \in \mathcal{C}_a} m_{c,a,t}^{\text{cf}} + m_{a,t}^{\text{by}} + m_{a,t}^{\text{cl}} \tag{1}$$

$$p_{\ell,t} = p_{a,t}^{\text{by}} + p_{a,t}^{\text{l-cl}} + \sum_{c \in \mathcal{C}_a} p_{c,a,t}^{\text{l-cf}} \tag{2}$$

$$p_{r,t} = p_{a,t}^{\text{by}} + p_{a,t}^{\text{r-cl}} + \sum_{c \in \mathcal{C}_a} p_{c,a,t}^{\text{r-cf}} \tag{3}$$

$$q_{a,t} = q_{a,t}^{\text{by}} + \sum_{c \in \mathcal{C}_a} q_{c,a,t}^{\text{cf}} \tag{4}$$

$$\underline{p}_{c,a,t}^{\text{l-cf}} m_{c,a,t}^{\text{cf}} \leq p_{c,a,t}^{\text{l-cf}} \leq \bar{p}_{c,a,t}^{\text{l-cf}} m_{c,a,t}^{\text{cf}} \quad \forall c \in \mathcal{C}_a \tag{5}$$

$$\underline{p}_{c,a,t}^{\text{r-cf}} m_{c,a,t}^{\text{cf}} \leq p_{c,a,t}^{\text{r-cf}} \leq \bar{p}_{c,a,t}^{\text{r-cf}} m_{c,a,t}^{\text{cf}} \quad \forall c \in \mathcal{C}_a \tag{6}$$

$$\underline{q}_{c,a,t}^{\text{cf}} m_{c,a,t}^{\text{cf}} \leq q_{c,a,t}^{\text{cf}} \leq \bar{q}_{c,a,t}^{\text{cf}} m_{c,a,t}^{\text{cf}} \quad \forall c \in \mathcal{C}_a \tag{7}$$

$$\underline{p}_{a,t}^{\text{by}} m_{a,t}^{\text{by}} \leq p_{a,t}^{\text{by}} \leq \bar{p}_{a,t}^{\text{by}} m_{a,t}^{\text{by}} \tag{8}$$

$$\underline{q}^{\text{by}}_{a,t} m^{\text{by}}_{a,t} \leq q^{\text{by}}_{a,t} \leq \bar{q}^{\text{by}}_{a,t} m^{\text{by}}_{a,t} \tag{9}$$

$$\underline{p}^{\text{l-cl}}_{a,t} m^{\text{cl}}_{a,t} \leq p^{\text{l-cl}}_{a,t} \leq \bar{p}^{\text{l-cl}}_{a,t} m^{\text{cl}}_{a,t} \tag{10}$$

$$\underline{p}^{\text{r-cl}}_{a,t} m^{\text{cl}}_{a,t} \leq p^{\text{r-cl}}_{a,t} \leq \bar{p}^{\text{r-cl}}_{a,t} m^{\text{cl}}_{a,t} \tag{11}$$

$$\begin{aligned} \alpha_0 p^{\text{l-cf}}_{c,a,t} + \alpha_1 p^{\text{r-cf}}_{c,a,t} + \alpha_2 q^{\text{cf}}_{c,a,t} + \alpha_3 m^{\text{cf}}_{c,a,t} &\leq 0 \\ \forall(\alpha_0, \alpha_1, \alpha_2, \alpha_3) \in \mathcal{H}_c \quad \forall c &\in \mathcal{C}_a \\ m^{\text{by}}_{a,t}, m^{\text{cl}}_{a,t} &\in \{0,1\} \\ m^{\text{cf}}_{c,a,t} &\in \{0,1\} \quad \forall c \in \mathcal{C}_a. \end{aligned} \tag{12}$$

2.2 Pipes

Gas flow in pipelines is for operational purposes modeled as one-dimensional flow through a straight cylindrical pipe. We use the *friction dominated* model derived from the isothermal Euler Equations and the equation of state for real gases, see the model (FD1) of Brouwer et al. (2011) and the model (ISO3) of Domschke et al. (2017).

For the discretization, we use the implicit box scheme introduced by Domschke et al. (2011); Kolb et al. (2010). Here, the spatial domain is the length L_a of pipe $a = (\ell, r)$, and the time domain is the set of time steps $\mathcal{T}_0$. For two adjacent time points t_0 and t_1, the discretized model reads as

$$p_{\ell,t_1} + p_{r,t_1} - p_{\ell,t_0} - p_{r,t_0} + \frac{2R_s T z_a(\tau(t_1) - \tau(t_0))}{L_a A_a}\left(q_{r,a,t_1} - q_{\ell,a,t_1}\right) = 0 \tag{13}$$

$$\begin{aligned} p_{r,t_1} - p_{\ell,t_1} + \frac{\lambda_a R_s T z_a L_a}{4 D_a A_a^2}\left(\frac{|q_{\ell,a,t_1}| q_{\ell,a,t_1}}{p_{\ell,t_1}} + \frac{|q_{r,a,t_1}| q_{r,a,t_1}}{p_{r,t_1}}\right) \\ + \frac{g s_a L_a}{2 R_s T z_a}\left(p_{\ell,t_1} + p_{r,t_1}\right) = 0. \end{aligned} \tag{14}$$

Here R_s denotes the specific gas constant, T the gas temperature, z_a the compressibility factor, A_a the cross-sectional area of the pipe, λ_a the friction factor of the pipe, D_a the diameter of the pipe, and g the gravitational acceleration. With $s_a \in [-1, 1]$ we denote the slope of the pipe, i.e., the quotient of the elevation increase between the pipes endpoints and the length L_a of the pipe. We assume that all these values are predetermined parameters.

For the friction factor, we use the formula of Nikuradse (1950) for hydraulic rough pipelines assuming full turbulence:

$$\lambda_a = \left(2\log_{10}\left(\frac{D_a}{k_a}\right) + 1.138\right)^{-2}.$$

Apart from the pipe diameter, it depends only on the integral roughness parameter k_a of the pipe and is commonly used in gas network optimization. As an example, we refer to the article of Pfetsch et al. (2014), which furthermore provides additional background information on friction factor formulas. Regarding the compressibility factor, the approximation formula developed by Papay (1968) is used, see Saleh (2002) for an English reference. It is valid for natural gas of up to 150 bar and therefore fits our use-case of pressure values usually being between 1 bar and 100 bar.

$$z(p) = 1 - 3.52\frac{p}{p^{\mathrm{c}}}e^{-2.26\frac{T}{T^{\mathrm{c}}}} + 0.247\left(\frac{p}{p^{\mathrm{c}}}\right)^2 e^{-1.878\frac{T}{T^{\mathrm{c}}}}$$

The formula is based on the pseudo-critical pressure p^{c} and temperature T^{c}. Both values depend on the mixture of the gas, which we assume to be constant all over the network. The constant compressibility factor z_a per pipe $a = (\ell, r)$, which we use in the pipe equations, is determined as the average of the corresponding values derived from the initial state pressure values of the pipes' end nodes, i.e., by $z_a = (z(p_{\ell,0}) + z(p_{r,0}))/2$.

In our model, we use a linearized variant of Eq. (14) as proposed by Hennings (2018). Analogous to them, we fix the absolute velocity $|v| = \frac{R_s T z_a}{A_a}\frac{|q|}{p}$ to a predefined constant in the friction term. As mentioned before, we found that pipes in network stations are relatively short in comparison to those in the rest of the network. For example, the network stations considered in our computational experiments in Sect. 5 have an average length per station of at most 500 m, according to Table 1. On the other hand, the transport pipelines that are incident to the stations are routinely 100km or longer. Therefore, the pipes have much less impact inside stations, which allows us to assume a linear pipe model without losing much in terms of overall accuracy.

The final equations for each pipe $(\ell, r) = a \in \mathcal{A}^{\mathrm{pi}}$ and all adjacent time points $t_0, t_1 \in \mathcal{T}_0$ reduce to:

$$p_{\ell,t_1} + p_{r,t_1} - p_{\ell,t_0} - p_{r,t_0} + \frac{2R_s T z_a(\tau(t_1) - \tau(t_0))}{L_a A_a}\left(q_{r,a,t_1} - q_{\ell,a,t_1}\right) = 0 \quad (13)$$

$$\begin{aligned} p_{r,t_1} - p_{\ell,t_1} + \frac{\lambda_a L_a}{4D_a A_a}\left(|v_{\ell,a}|q_{\ell,a,t_1} + |v_{r,a}|q_{r,a,t_1}\right) & \\ + \frac{g s_a L_a}{2R_s T z_a}\left(p_{\ell,t_1} + p_{r,t_1}\right) = 0. & \end{aligned} \quad (15)$$

Note that (13) stays the same as above. The constant velocity $|v_{x,a}|$ for one of the end nodes $x \in \{\ell, r\}$ of pipe $a = (\ell, r)$ is determined based on the flow and pressure values of the given initial state, i.e.,

$$|v_{x,a}| = \frac{R_s T z_a}{A_a} \frac{|q_{x,a,0}|}{p_{x,0}}.$$

2.3 Resistors

Resistors are artificial elements created to model places of high friction in the network, e.g., caused by non-standard network elements without further relevance for the operation of the network. The pressure drop over a resistor arc $(\ell, r) = a \in \mathcal{A}^{\mathrm{rs}}$ at time $t \in \mathcal{T}$ is, according to Fügenschuh et al. (2015), defined by the *Darcy-Weisbach equation*:

$$p_{\ell,t} - p_{r,t} = \frac{\zeta_a R_s T z_a}{2A_a^2}\left(\frac{|q_{a,t}| q_{a,t}}{p_{\mathrm{in},t}}\right).$$

Here, the friction factor of the resistor is called the *drag factor* and is represented by the parameter ζ_a. We determine the constant compressibility factor z_a in the same way as for pipes. Note that the formula depends on the flow's direction due to $p_{\mathrm{in},t}$ representing the pressure of the incoming gas. The similar looking friction term in the Eqs. (14) and (15) of the pipe model does not contain this dependency thanks to the chosen discretization scheme.

Analogous to the approach for pipes in Eq. (15), we linearize the model by assuming a constant velocity $|v| = \frac{R_s T z_a}{A_a} \frac{|q|}{p}$, which also includes the flow direction dependent pressure value. The equation for each arc $(\ell, r) = a \in \mathcal{A}^{\mathrm{rs}}$ and time $t \in \mathcal{T}$ then reads as

$$p_{\ell,t} - p_{r,t} = \frac{\zeta_a |v_a|}{2A_a} q_{a,t}. \tag{16}$$

The constant velocity value is again calculated based on the initial element state and is defined as an average of the two velocities using the pressure from the corresponding resistor's end nodes as

$$|v_a| := \frac{1}{2}\left(\frac{R_s T z_a}{A_a} \frac{|q_{a,0}|}{p_{\ell,0}} + \frac{R_s T z_a}{A_a} \frac{|q_{a,0}|}{p_{r,0}}\right).$$

2.4 Valves

By being open or closed, valves connect or disconnect two nodes of the network. Closed valves work exactly as described by the closed mode of a compressor station, while open valves imply the same behavior as the corresponding compressor station's bypass mode. The valve's mode is captured by the binary variable $m_{a,t}^{\mathrm{op}}$, which attains the value 1 for the open mode and 0 for closed. For a valve arc $(\ell, r) = a \in \mathcal{A}^{\mathrm{va}}$ and time $t \in \mathcal{T}$, we can capture the modes' behavior using indicator constraints as:

$$m^{\mathrm{op}}_{a,t} = 1 \quad \to \quad p_{\ell,t} \le p_{r,t} \tag{17}$$

$$m^{\mathrm{op}}_{a,t} = 1 \quad \to \quad p_{\ell,t} \ge p_{r,t} \tag{18}$$

$$m^{\mathrm{op}}_{a,t} = 0 \quad \to \quad q_{a,t} \le 0 \tag{19}$$

$$\begin{gathered} m^{\mathrm{op}}_{a,t} = 0 \quad \to \quad q_{a,t} \ge 0 \\ m^{\mathrm{op}}_{a,t} \in \{0,1\}. \end{gathered} \tag{20}$$

2.5 Regulators

A regulator or control valve is a valve with a variable opening degree, used to reduce the pressure along the direction of the flow. Regulators can reduce the pressure in active mode and also be bypassed or closed like a compressor station. For each mode, there is a binary selection variable, where a value of 1 again indicates the uniquely selected mode per time. For each arc $a \in \mathcal{A}^{\mathrm{rg}}$ and all times $t \in \mathcal{T}$, the behavior is captured by the following constraints:

$$m^{\mathrm{cl}}_{a,t} + m^{\mathrm{by}}_{a,t} + m^{\mathrm{ac}}_{a,t} = 1 \tag{21}$$

$$m^{\mathrm{by}}_{a,t} = 1 \quad \to \quad p_{\ell,t} \le p_{r,t} \tag{22}$$

$$m^{\mathrm{by}}_{a,t} + m^{\mathrm{ac}}_{a,t} = 1 \quad \to \quad p_{\ell,t} \ge p_{r,t} \tag{23}$$

$$m^{\mathrm{cl}}_{a,t} = 1 \quad \to \quad q_a \le 0 \tag{24}$$

$$\begin{gathered} 0 \le q_a \\ m^{\mathrm{cl}}_{a,t}, m^{\mathrm{by}}_{a,t}, m^{\mathrm{ac}}_{a,t} \in \{0,1\}. \end{gathered} \tag{25}$$

Note that for the logical condition in (23), the sum $m^{\mathrm{by}}_{a,t} + m^{\mathrm{ac}}_{a,t}$ can only be $\in \{0,1\}$ due to (21) and therefore acts as an implicit binary variable. Furthermore, all regulators have a flap trap preventing gas flowing against the topological orientation, which is modeled by (25).

2.6 Nodes

The nodes do not represent technical elements but rather establish the connections between them. The pressure coupling is realized by using the node's pressure variables in the constraints of all its incident arcs. To connect the mass flow variables of the arcs, we have the following flow conservation constraints for all $t \in \mathcal{T}$:

$$\begin{aligned} &\sum_{(\ell,v)=a\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} - \sum_{(v,r)=a\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} \\ &+ \sum_{(\ell,v)=a\in\mathcal{A}\setminus\mathcal{A}^{\mathrm{pi}}} q_{a,t} - \sum_{(v,r)=a\in\mathcal{A}\setminus\mathcal{A}^{\mathrm{pi}}} q_{a,t} + d_{v,t} = 0 \qquad \forall v \in \mathcal{V}^{\mathrm{b}} \end{aligned} \tag{26}$$

$$\begin{aligned} &\sum_{(\ell,v)=a\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} - \sum_{(v,r)=a\in\mathcal{A}^{\mathrm{pi}}} q_{v,a,t} \\ &+ \sum_{(\ell,v)=a\in\mathcal{A}\setminus\mathcal{A}^{\mathrm{pi}}} q_{a,t} - \sum_{(v,r)=a\in\mathcal{A}\setminus\mathcal{A}^{\mathrm{pi}}} q_{a,t} = 0 \qquad \forall v \in \mathcal{V}^{0}. \end{aligned} \tag{27}$$

Note that pipes have separated inflow and outflow variables, while all other arcs only have a single flow variable, as explained in the beginning of Sect. 2.

2.7 Network station

In addition to the constraints imposed by the single elements contained in the network station, we also consider restrictions associated with the network station itself. The model described below is an extension of the one described in Fügenschuh et al. (2015) for similar structures. There, network stations are called *compressor stations*, and the elements we call compressor stations are called *compressor groups*. Our project partner suggested adjusting the naming scheme here since the previous one can be misleading, especially so as a network station does not need to contain an element with compression capabilities.

2.7.1 Network station structure

For a network station, two main decisions have to be made for each point in time. We have to choose exactly one *flow direction* from the set $\mathcal{F}$ of available flow directions and exactly one *operation mode* from the set $\mathcal{O}$ of available operation modes.

A flow direction represents the general orientation of the gas flow through the station, for example, flow from the northern and eastern boundary nodes to those at the southern border of the station. Hence, a flow direction is defined as a partition of the boundary nodes into entry nodes, exits nodes, and no-flow nodes, at which respectively, gas enters the station, gas leaves the station, or the gas inflow and outflow is zero. Only a subset of the theoretically possible partitions is feasible for the station, forming the set $\mathcal{F}$.

Similarly, the modes and configurations of the network station's elements cannot be arbitrarily combined. For example, separating an active compressor station from the network by closing all surrounding valves would not be feasible in reality. Therefore we are given the operation modes, which represent the available mode and configuration combinations of all valves and compressor stations of the network station.

The set of feasible mode combinations for valves and compressor stations is, most of the time, very small in comparison to the set of all possibilities. Often, a feasible combination prescribes only one or a few specific paths for the gas flow through the station using non-closed elements. However, imagine a situation in which a compressor station is in active mode in a particular operation mode of the network station. In this case, all available configurations of the compressor station generally yield feasible operation modes when combined with the same settings for the other elements.

Note that operation modes do not prescribe the mode of regulators. For these, the network operator cannot set the desired mode directly but chooses *target values* for the incoming and outgoing pressure of the element as well as its flow. The regulator tries to meet these values by adjusting its opening degree according to a given set of rules, which also determines the mode. Since target value modeling is out of scope for this paper, we simply do not include regulator modes in the operation mode restrictions.

Finally, not all combinations of flow directions and operation modes are valid. One could, for example, think of a compressor station that is only active when the network station operates in a south to north flow direction. Hence, we are given the set $\mathcal{Q} \subset \mathcal{F} \times \mathcal{O}$ of valid combinations of flow directions and operation modes.

2.7.2 Operation modes model

We introduce binary variables $m^{\text{om}}_{o,t}$ for each $o \in \mathcal{O}$ and $t \in \mathcal{T}_0$, where a value of 1 represents the selection of o at time t. Furthermore, we define the function $M(o, a)$ that maps operation modes to the individual modes and configurations of valves and compressor stations:

$$\begin{aligned} M(o,a) &:= x \text{ where } x \text{ is the mode or configuration of arc } a \\ &\quad\ \text{in operation mode } o \quad \forall o \in \mathcal{O} \quad \forall a \in \mathcal{A}^{\text{va}} \cup \mathcal{A}^{\text{cs}} \\ \text{with} &\quad x \in \{\text{op}, \text{cl}\} \text{ if } a \in \mathcal{A}^{\text{va}} \\ &\quad x \in \{\text{by}, \text{cl}\} \cup \mathcal{C}_a \text{ if } a \in \mathcal{A}^{\text{cs}}. \end{aligned}$$

Using $M(o, a)$, we can state the constraints modeling the relations related to operation modes for all $t \in \mathcal{T}$ as:

$$\sum_{o \in \mathcal{O}} m^{\text{om}}_{o,t} = 1 \tag{28}$$

$$m^{\text{op}}_{a,t} = \sum_{o \in \mathcal{O} : M(o,a) = \text{op}} m^{\text{om}}_{o,t} \quad \forall a \in \mathcal{A}^{\text{va}} \tag{29}$$

$$m^{\text{by}}_{a,t} = \sum_{o \in \mathcal{O} : M(o,a) = \text{by}} m^{\text{om}}_{o,t} \quad \forall a \in \mathcal{A}^{\text{cs}} \tag{30}$$

$$m^{\mathrm{cl}}_{a,t} = \sum_{o \in \mathcal{O}: M(o,a)=\mathrm{cl}} m^{\mathrm{om}}_{o,t} \quad \forall a \in \mathcal{A}^{\mathrm{cs}} \tag{31}$$

$$\begin{aligned} m^{\mathrm{cf}}_{c,a,t} &= \sum_{o \in \mathcal{O}: M(o,a)=c} m^{\mathrm{om}}_{o,t} \quad \forall c \in \mathcal{C}_a \quad \forall a \in \mathcal{A}^{\mathrm{cs}} \\ m^{\mathrm{om}}_{o,t} &\in \{0,1\} \quad \forall o \in \mathcal{O}. \end{aligned} \tag{32}$$

Note that either Eqs. (30), (31) or one of the constraints of Eq. (32) for one of the configurations $c \in \mathcal{C}_a$ can be omitted, since it follows from the remaining constraints combined with Eq. (1).

2.7.3 Operation mode unavailability

Certain operation modes are not available at specific points in time, and we set the corresponding variables $m^{\mathrm{om}}_{o,t}$ for each associated operation mode o and time t to zero. The basis for this is the non-availability of compressor units over time, which is part of the model input data. As explained in Sect. 2.1, a configuration $c \in \mathcal{C}_a$ of some compressor station $a \in \mathcal{A}^{\mathrm{cs}}$ represents the serial and/or parallel combination of a subset of the compressor station's compressor units. Hence, the unavailability of a certain compressor unit at time t results in the unavailability of all configurations using this unit, which causes all operation modes using these configurations to be unavailable as well.

The unavailability period of a compressor unit may not be aligned with the set of discrete time points $\mathcal{T}_0$. According to the definition given at the start of Sect. 2, the network station operation mode of some time $t \in \mathcal{T}_0$ is also used in the open interval $(t, t+1)$. Thus, the unavailability of an operation mode $k \in (t, t+1)$ with $t, t+1 \in \mathcal{T}_0$ results in the unavailability of that operation mode for time t. At the same time, it stays potentially available for time $t+1$.

2.7.4 Flow directions model

Similar to the way we handle operations modes, we introduce binary variables $m^{\mathrm{fd}}_{f,t}$ representing the selection of a flow direction $f \in \mathcal{F}$ at time $t \in \mathcal{T}_0$ if the variable attains the value 1. As explained above, a flow direction f is defined by the signs of the flows on boundary nodes. We write f^+ for entry nodes with positive inflow and f^- for exits with negative inflow, i.e. outflow. Hence, using the power set $\mathcal{P}$, a flow direction f is defined as

$$(f^+, f^-) = f \in \mathcal{F} \subseteq \mathcal{P}(\mathcal{V}^{\mathrm{b}}) \times \mathcal{P}(\mathcal{V}^{\mathrm{b}}) \text{ where } f^+ \cap f^- = \emptyset.$$

Note that if $v \notin f^+$ and $v \notin f^-$ for flow direction (f^+, f^-), then node v is a no-flow node, and its inflow and outflow are zero.

Using the inflow variable $d_{v,t}$ for boundary node v and time t, we can define the flow direction constraints for each $t \in \mathcal{T}$ as:

$$\sum_{f \in \mathcal{F}} m^{\mathrm{fd}}_{f,t} = 1 \tag{33}$$

$$\sum_{f=(f^+,f^-) \in \mathcal{F}:\, v \notin f^-} m^{\mathrm{fd}}_{f,t} = 1 \quad \rightarrow \quad 0 \le d_{v,t} \quad \forall v \in \mathcal{V}^{\mathrm{b}} \tag{34}$$

$$\begin{gathered} \sum_{f=(f^+,f^-) \in \mathcal{F}:\, v \notin f^+} m^{\mathrm{fd}}_{f,t} = 1 \quad \rightarrow \quad 0 \ge d_{v,t} \quad \forall v \in \mathcal{V}^{\mathrm{b}} \\ m^{\mathrm{fd}}_{f,t} \in \{0, 1\} \quad \forall f \in \mathcal{F}. \end{gathered} \tag{35}$$

Note that in the constraints (34) and (35), the sum of the $m^{\mathrm{fd}}_{f,t}$ variables act as an implicit binary variable since its value can only be in $\{0, 1\}$ due to (33).

The restriction to feasible combinations of operation modes and flow directions is implemented as:

$$m^{\mathrm{om}}_{o,t} \le \sum_{(f,o) \in \mathcal{Q}} m^{\mathrm{fd}}_{f,t} \quad \forall o \in \mathcal{O} \quad t \in \mathcal{T}. \tag{36}$$

2.7.5 Flow direction exit pressures

Apart from the consequences that the flow direction choice has on the corresponding boundary node inflows, it may add restrictions to the upper pressure bounds on some boundary nodes. Each node v in the set $\mathcal{V}^{\text{b-ex}} \subseteq \mathcal{V}^{\mathrm{b}}$ is given a limiting upper pressure value $\bar{p}^{\mathrm{exit}}_v$ that is only considered if the node is in the outflow set of the currently active flow direction, i.e., serving as exit node of this flow direction. These exit pressure restrictions originate from the historical development of the network infrastructure and may, for example, occur when a transport pipeline has been upgraded to allow for higher maximum pressure values. If then the adjacent network station contains equipment, for example a metering station, that is a) only used when gas leaves the station in the direction of this pipeline and b) has an upper pressure bound in line with the old pipeline, we have to model this bound explicitly when gas is flowing in this direction. Before the pipeline's upgrade, this upper pressure bound was implied by the corresponding pressure bounds of the nodes in the pipeline. The corresponding constraint is the following for each time $t \in \mathcal{T}$

$$\sum_{f=(f^+,f^-) \in \mathcal{F}:\, v \in f^-} m^{\mathrm{fd}}_{f,t} = 1 \quad \rightarrow \quad p_{v,t} \le \bar{p}^{\mathrm{exit}}_v \quad \forall v \in \mathcal{V}^{\text{b-ex}}. \tag{37}$$

Again, the sum of the $m^{\mathrm{fd}}_{f,t}$ variables acts as an implicit binary variable due to (33).

2.7.6 Flow direction conditions

Denoting the final constraints concerning flow directions, there can exist a special set of conditions $\mathcal{W}$ that concern the amount of flow over sets of boundary nodes. Each condition $w = (f, \mathcal{V}^{w_1}, \mathcal{V}^{w_2}) \in \mathcal{W}$ states that the flow over the set of boundary nodes $\mathcal{V}^{w_1}$ has to be smaller than or equal to the flow over the second set of boundary nodes $\mathcal{V}^{w_2}$. These conditions must be met if an associated flow direction $f = (f^+,f^-)$ is active. By definition the node sets $\mathcal{V}^{w_1}$ and $\mathcal{V}^{w_2}$ have to be a subset of either f^+ or f^-, while they do not need to belong to the same set.

An example of a relation requiring the constraints is as follows: Consider a transport pipeline intersecting a network station. These constraints can then be used to model a flow direction that not only restricts the flow at the boundary nodes, but also enforces that gas is directed into the intersecting transport pipeline. We do this by stating that the outflow of the transport pipeline at the network station's boundary is larger than its inflow into the network station.

Note that the flow over a set of boundary nodes $\mathcal{V}^w$ for time t is defined as $\sum_{v\in\mathcal{V}^w} |d_{v,t}|$, which is potentially a nonlinear expression due to the absolute value. However, by the above definition, the boundary node sets $\mathcal{V}^{w_1}$ and $\mathcal{V}^{w_2}$ of some $\big((f^+,f^-), \mathcal{V}^{w_1}, \mathcal{V}^{w_2}\big) \in \mathcal{W}$ are subsets of either f^+ or f^-. For this reason, we always know the sign of the flow over $\mathcal{V}^w$ in advance, and hence using the following function definition for easier notation

$$\mathrm{sgn} : \mathcal{F}\times\mathcal{V}^{\mathrm{b}} \to \{-1, 1\}\,,\ \big((f^+,f^-), v\big) \to \begin{cases} 1 & \text{if } v \in f^+ \\ -1 & \text{if } v \in f^-, \end{cases}$$

we can write the constraints for each $t \in \mathcal{T}$ as:

$$m^{\mathrm{fd}}_{f,t} = 1 \quad \to \quad \sum_{v\in\mathcal{V}^{w_1}} \mathrm{sgn}(f, v)d_{v,t} \le \sum_{v\in\mathcal{V}^{w_2}} \mathrm{sgn}(f, v)d_{v,t} \qquad \forall (f, \mathcal{V}^{w_1}, \mathcal{V}^{w_2}) \in \mathcal{W}. \tag{38}$$

2.8 Operation mode transition times

In addition to the characterization of operation modes given previously, there exists additional restrictions, namely those concerning transition times between any two operation modes. In contrast to the rest of Sect. 2, we only give a verbal description and do not present a MIP model formulation here, since none of the MIP variants solved in our heuristic algorithm uses these constraints, see Sect. 4. Instead, it explicitly checks potential solutions for complying with this restriction.

Changing the operation mode of a network station can be a complex procedure if the two involved operation modes differ significantly in terms of the actively used elements. The changes in terms of modes and configurations that are needed for the transition are often required to happen in a particular order. Furthermore, it may be necessary to switch into a third, intermediate operation mode during the change to ensure safe operation at all times. However, for the scope of this paper, we are only interested in the implications of the transition process. Therefore, we define a feasible transition from operation mode A to mode B as follows: We allow the transition to happen instantly at some time point $t \in \mathcal{T}_0$. However, we consider the time $\theta(A, B)$ of the transition process from A to B by demanding that the network station is in operation mode A in the time interval $[t - \frac{\theta(A,B)}{2}, t)$ preceding t and in mode B for

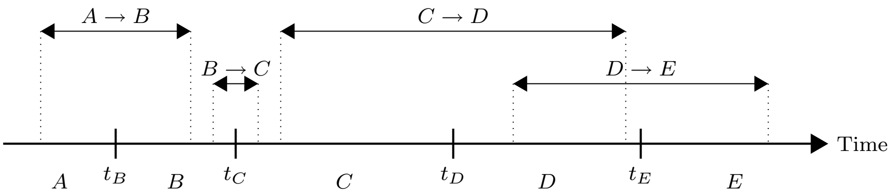


Fig. 2 Transition time example with five operation modes and four transitions, from which two conflict. The time point t_X represents the switching time of the corresponding transition into the operation mode X, i.e., the first $t \in \mathcal{T}_0$ in which operation mode X is active

the subsequent time interval $[t, t + \frac{\theta(A,B)}{2}]$. In addition, this whole transition period $[t - \frac{\theta(A,B)}{2}, t + \frac{\theta(A,B)}{2}]$ is reserved for this transition to happen, i.e., two transition periods cannot overlap. Note that we allow the last transition period to exceed our time horizon, which makes operation mode changes up to the very end of the time horizon possible.

As an example, we denote in Fig. 2 an operation mode sequence over time that includes the involved transition times. Here, each time point t_X represents the switching time of the corresponding transition into the operation mode X, i.e., the first $t \in \mathcal{T}_0$ in which operation mode X is active. Note that the displayed example includes a conflict in the transitions from mode C to mode D and from mode D to mode E since the two corresponding transition periods overlap. Hence this sequence of operation modes does not represent a feasible solution.

2.9 Scenario and initial state

For the future, we are given scenario values for the station's boundary nodes in terms of pressure and inflow (where outflow is represented as negative inflow, see the beginning of Sect. 2). While we have one pressure value $\hat{p}_{v,t}$ per boundary node $v \in \mathcal{V}^b$ for each future time point $t \in \mathcal{T}$, the flow demands are only given for sets of boundary nodes. These sets are called the *fence groups* of the network station and form the set $\mathcal{G}$. For each $g \in \mathcal{G}$ and time point $t \in \mathcal{T}$, the sum of the inflow values of all boundary nodes in g should be equal to the demand value $\hat{d}_{g,t}$.

We do not require strict obedience of the given demand values $\hat{p}_{v,t}$ and $\hat{d}_{g,t}$, but instead allow deviations from them, which we punish in the objective function. These deviations are captured in the *slack variables* σ. For pressure, we have $\sigma^{p+}_{v,t}$ and $\sigma^{p-}_{v,t}$ capturing the positive and negative difference between the pressure values of boundary node v at time t and the given demand $\hat{p}_{v,t}$. The slack variables $\sigma^{d+}_{v,t}$ and $\sigma^{d-}_{v,t}$ are associated with the inflow. They capture the positive and negative contribution to the difference between the inflow demand $\hat{d}_{g,t}$ of fence group $g \in \mathcal{G}$ at time step t and the sum of the corresponding inflow variables of each boundary node v of the fence group g.

We model the described relations for each future time step $t \in \mathcal{T}$ as:

$$\hat{p}_{v,t} = p_{v,t} - \sigma^{p+}_{v,t} + \sigma^{p-}_{v,t} \quad \forall v \in \mathcal{V}^{\flat} \tag{39}$$

$$\hat{d}_{g,t} = \sum_{v \in g} \left(d_{v,t} - \sigma^{d+}_{v,t} + \sigma^{d-}_{v,t} \right) \quad \forall g \in \mathcal{G}. \tag{40}$$

Note that by using inflow slacks, it is possible to choose a flow direction that does not fit the inflow demand values $\hat{d}_{g,t}$ in terms of the Eqs. (34–35).

The second set of prescribed values are those of the initial state represented by time 0. All variables for this time are actually parameters of the model and fixed to the corresponding value.

2.10 Objective

The objective function covers two different aspects. On the one hand, it should minimize the deviation from the given future scenario in terms of inflow and pressure at the boundary nodes. On the other hand, we favor those solutions with stable control of the single elements and hence try to reduce the number of needed control changes. The latter is preferable for a couple of reasons. First, a limited number of changes enables the gas network operators to understand, evaluate, and actually perform the desired control recommendations. Due to the inertia of the gas, they are used to operating the network using a small set of essential adjustments rather than dynamically performing small changes all the time. Finally, a stable control also reduces strain on the technical elements.

To define a measure for the stability, we first quantify the discrete changes of the control in binary variables. The variable δ^{om}_{t} captures the change of the network station into a new operation mode at time t, while the variable $\delta^{\mathrm{rg}}_{a,t}$ represents the change into a new mode of regulator a at time t. We have to track the regulator mode changes explicitly, since their modes are not prescribed by the operation modes, see the description of the network station structure in Sect. 2.7 for an explanation. Furthermore, we capture the start-up of compressor unit u at time t in the variable $\delta^{\mathrm{us}}_{u,t}$. Since starting a compressor is a very time and energy intensive action, it should be avoided if possible. The following set of constraints guarantees for each future time step $t \in \mathcal{T}$ that the variables attain the value 1 if a corresponding change occurs and 0 otherwise.

$$\delta^{\mathrm{om}}_{t} \geq m^{\mathrm{om}}_{o,t} - m^{\mathrm{om}}_{o,t-1} \quad \forall o \in \mathcal{O} \tag{41}$$

$$\delta^{\mathrm{om}}_{t} \leq 2 - m^{\mathrm{om}}_{o,t} - m^{\mathrm{om}}_{o,t-1} \quad \forall o \in \mathcal{O} \tag{42}$$

$$\delta^{\mathrm{rg}}_{a,t} \geq m^{\mathrm{cl}}_{a,t} - m^{\mathrm{cl}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\mathrm{rg}} \tag{43}$$

$$\delta^{\mathrm{rg}}_{a,t} \leq 2 - m^{\mathrm{cl}}_{a,t} - m^{\mathrm{cl}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\mathrm{rg}} \tag{44}$$

$$\delta^{\text{rg}}_{a,t} \geq m^{\text{by}}_{a,t} - m^{\text{by}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\text{rg}} \tag{45}$$

$$\delta^{\text{rg}}_{a,t} \leq 2 - m^{\text{by}}_{a,t} - m^{\text{by}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\text{rg}} \tag{46}$$

$$\delta^{\text{rg}}_{a,t} \geq m^{\text{ac}}_{a,t} - m^{\text{ac}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\text{rg}} \tag{47}$$

$$\delta^{\text{rg}}_{a,t} \leq 2 - m^{\text{ac}}_{a,t} - m^{\text{ac}}_{a,t-1} \quad \forall a \in \mathcal{A}^{\text{rg}} \tag{48}$$

$$\begin{aligned}
&\delta^{\text{us}}_{u,t} \geq \sum_{c \in \mathcal{C}_{u,a}} m^{\text{cf}}_{c,a,t} - \sum_{c \in \mathcal{C}_{u,a}} m^{\text{cf}}_{c,a,t-1} \quad \forall u \in \mathcal{U}_a \forall a \in \mathcal{A}^{\text{cs}} \\
&\delta^{\text{om}}_{t} \in \{0,1\} \\
&\delta^{\text{rg}}_{a,t} \in \{0,1\} \quad \forall a \in \mathcal{A}^{\text{rg}} \\
&\delta^{\text{us}}_{u,t} \in \{0,1\} \quad \forall u \in \mathcal{U}_a \forall a \in \mathcal{A}^{\text{cs}}
\end{aligned} \tag{49}$$

In Eq. (49), the set $\mathcal{C}_{u,a}$ denotes the subset of configurations $\mathcal{C}_a$ of the compressor station $a \in \mathcal{A}^{\text{cs}}$, for which compressor unit $u \in \mathcal{U}_a$ is turned on.

In order to obtain a smooth operation for network situations without discrete mode switching, we add variables tracking the change of the operation point of single elements, i.e., their corresponding changes in flow, incoming pressure and outgoing pressure. We do this for all elements that can precisely control the point of operation, i.e., regulators and compressor stations in active mode, and for all times without discrete mode switching, i.e. a stable network station operation mode or regulator mode. The variables $\delta^{\text{rg-pl}}_{a,t}$, $\delta^{\text{rg-pr}}_{a,t}$, and $\delta^{\text{rg-q}}_{a,t}$ represent changes of the incoming pressure, outgoing pressure, and flow of an active regulator a, while $\delta^{\text{cs-pl}}_{a,t}$, $\delta^{\text{cs-pr}}_{a,t}$, and $\delta^{\text{cs-q}}_{a,t}$ represent the corresponding value changes of an active compressor station a. We establish the behavior of these variables using the following constraints for each $(\ell, r) = a \in \mathcal{A}^{\text{rg}} \cup \mathcal{A}^{\text{cs}}$ and each $t \in \mathcal{T}$

$$p_{\ell,t} - p_{\ell,t-1} \leq \delta^{\text{X-pl}}_{a,t} + (m^{\text{by}}_{a,t} + m^{\text{cl}}_{a,t} + \delta)(\bar{p}_{\ell,t} - \underline{p}_{\ell,t-1}) \tag{50}$$

$$p_{\ell,t-1} - p_{\ell,t} \leq \delta^{\text{X-pl}}_{a,t} + (m^{\text{by}}_{a,t} + m^{\text{cl}}_{a,t} + \delta)(\bar{p}_{\ell,t-1} - \underline{p}_{\ell,t}) \tag{51}$$

$$p_{r,t} - p_{r,t-1} \leq \delta^{\text{X-pr}}_{a,t} + (m^{\text{by}}_{a,t} + m^{\text{cl}}_{a,t} + \delta)(\bar{p}_{r,t} - \underline{p}_{r,t-1}) \tag{52}$$

$$p_{r,t-1} - p_{r,t} \leq \delta^{\text{X-pr}}_{a,t} + (m^{\text{by}}_{a,t} + m^{\text{cl}}_{a,t} + \delta)(\bar{p}_{r,t-1} - \underline{p}_{r,t}) \tag{53}$$

$$q_{a,t} - q_{a,t-1} \leq \delta_{a,t}^{\text{X-q}} + (m_{a,t}^{\text{by}} + m_{a,t}^{\text{cl}} + \delta)(\bar{q}_{a,t} - \underline{q}_{a,t-1}) \tag{54}$$

$$q_{a,t-1} - q_{a,t} \leq \delta_{a,t}^{\text{X-q}} + (m_{a,t}^{\text{by}} + m_{a,t}^{\text{cl}} + \delta)(\bar{q}_{a,t-1} - \underline{q}_{a,t}). \tag{55}$$

In these constraints, we define X=rg as well as $\delta = \delta_{a,t}^{\text{rg}}$ for $a \in \mathcal{A}^{\text{rg}}$, and X=cs as well as $\delta = \delta_t^{\text{om}}$ for $a \in \mathcal{A}^{\text{cs}}$.

Note that we needed to add the upper bound constraints (42), (44), (46), and (48) for the discrete change variables δ_t^{om} and $\delta_{a,t}^{\text{rg}}$ to allow them to be equal to 1 if and only if there is a discrete change. Otherwise, it would be possible to set them to 1 independent of the corresponding mode variable values, thereby deactivating the associated constraints (50)–(55) and avoiding potentially higher costs imposed by the continuous change variables.

Finally, we can state our objective function, which minimizes the weighted sum of the change variables as well as the slack variables defined in Sect. 2.9 as

$$\begin{aligned}
\min \quad & \sum_{t\in\mathcal{T}} \Bigg(\big(\tau(t) - \tau(t-1)\big) \sum_{v\in\mathcal{V}^{\text{b}}} \big(w^{\sigma\text{-p}} \cdot (\sigma_{v,t}^{p+} + \sigma_{v,t}^{p-}) + w^{\sigma\text{-d}} \cdot (\sigma_{v,t}^{d+} + \sigma_{v,t}^{d-})\big) \\
& + w^{\text{om}} \cdot \delta_t^{\text{om}} + \sum_{a\in\mathcal{A}^{\text{rg}}} w^{\text{rg}} \cdot \delta_{a,t}^{\text{rg}} + \sum_{u\in\mathcal{U}_a, a\in\mathcal{A}^{\text{cs}}} w^{\text{us}} \cdot \delta_{u,t}^{\text{us}} \\
& + \sum_{a\in\mathcal{A}^{\text{rg}}} \big(w^{\text{rg-pl}} \cdot \delta_{a,t}^{\text{rg-pl}} + w^{\text{rg-pr}} \cdot \delta_{a,t}^{\text{rg-pr}} + w^{\text{rg-q}} \cdot \delta_{a,t}^{\text{rg-q}}\big) \\
& + \sum_{a\in\mathcal{A}^{\text{cs}}} \big(w^{\text{cs-pl}} \cdot \delta_{a,t}^{\text{cs-pl}} + w^{\text{cs-pr}} \cdot \delta_{a,t}^{\text{cs-pr}} + w^{\text{cs-q}} \cdot \delta_{a,t}^{\text{cs-q}}\big) \Bigg).
\end{aligned} \tag{56}$$

Here, the w^* parameters denote the corresponding positive weights given to the single quantities. Note that the weights for pressure and flow slack variables are additionally multiplied by the length of the corresponding time interval to represent the amount of deviation over time correctly.

2.11 Final model

Putting everything together, we can formulate our problem in the following transient gas flow model $\mathscr{P}$:

$$
\begin{array}{lll}
 & \min & (56) \\
\text{s.t.} \quad \forall t \in \mathcal{T} & & (1)-(4),(8)-(11),(30)-(31),(50)-(55) \quad \forall a \in \mathcal{A}^{\text{cs}} \\
 & & (5)-(7),(32) \quad \forall a \in \mathcal{A}^{\text{cs}} \quad \forall c \in \mathcal{C}_a \\
 & & (12) \quad \forall a \in \mathcal{A}^{\text{cs}} \quad \forall c \in \mathcal{C}_a \quad \forall (\alpha_0, \alpha_1, \alpha_2, \alpha_3) \in \mathcal{H}_c \\
 & & (13)-(15) \quad \forall a \in \mathcal{A}^{\text{pi}} \\
 & & (16) \quad \forall a \in \mathcal{A}^{\text{rs}} \\
 & & (17)-(20),(29) \quad \forall a \in \mathcal{A}^{\text{va}} \\
 & & (21)-(25),(43)-(48),(50)-(55) \quad \forall a \in \mathcal{A}^{\text{rg}} \\
 & & (26),(34)-(35),(39) \quad \forall v \in \mathcal{V}^{\text{b}} \\
 & & (27) \quad \forall v \in \mathcal{V}^{0} \\
 & & (28),(33) \\
 & & (36),(41)-(42) \quad \forall o \in \mathcal{O} \\
 & & (37) \quad \forall v \in \mathcal{V}^{\text{b-ex}} \\
 & & (38) \quad \forall w \in \mathcal{W} \\
 & & (40) \quad \forall g \in \mathcal{G} \\
 & & (49) \quad \forall a \in \mathcal{A}^{\text{cs}} \quad u \in \mathcal{U}_a \\
 & \{0,1\} \ni & m^{\text{by}}_{a,t}, m^{\text{cl}}_{a,t} \quad \forall a \in \mathcal{A}^{\text{cs}}, \\
 & & m^{\text{cf}}_{c,a,t} \quad \forall a \in \mathcal{A}^{\text{cs}} \quad \forall c \in \mathcal{C}_a, \\
 & & m^{\text{op}}_{a,t} \quad \forall a \in \mathcal{A}^{\text{va}}, \\
 & & m^{\text{cl}}_{a,t}, m^{\text{by}}_{a,t}, m^{\text{ac}}_{a,t}, \delta^{\text{rg}}_{a,t} \quad \forall a \in \mathcal{A}^{\text{rg}}, \\
 & & m^{\text{om}}_{o,t} \quad \forall o \in \mathcal{O}, \\
 & & m^{\text{fd}}_{f,t} \quad \forall f \in \mathcal{F}, \\
 & & \delta^{\text{om}}_{t}, \\
 & & \delta^{\text{us}}_{u,t} \quad \forall u \in \mathcal{U}_a \quad \forall a \in \mathcal{A}^{\text{cs}}.
\end{array}
$$

Note that we apply the constraints only starting from time step 1, explicitly excluding the initial time step 0. We do this since the initial pressure values, flow values, and modes described in Sect. 2.9 are not guaranteed to fit our model, and only serve as a starting point for the calculations.

3 Feasible operating range of compressor station configurations

In the following section, we describe the creation of the feasible operating range polytope for each configuration $c \in \mathcal{C}_a$ of a compressor station $a = (\ell, r)$. This polytope is represented as the intersection of half-spaces $\mathcal{H}_c$ and was already used in the model description for compressor stations in Sect. 2.1 in Eq. (12).

We shortly recall the structure of a compressor station arc from Sect. 2.1. Each compressor station $a \in \mathcal{A}^{\mathrm{cs}}$ is associated with a set of compressor units $\mathcal{U}_a$, which each represent the union of a compressor machine and the power providing drive. These compressor units can be combined in series and/or parallel to either allow for a higher compression ratio, a higher flow rate, or a mixture of both. The set of all available serial-parallel combinations is called the set of configurations $\mathcal{C}_a$ of a compressor station.

To create the set $\mathcal{H}_c$ for $c \in \mathcal{C}_a$, we first obtain a feasible operating range polytope for each of the compressor station's compressor units. This polytope includes a linear approximation of the nonlinear constraint resulting from the maximum power bound of the unit's drive. Afterwards, we sketch the procedure of Walther and Hiller (2017) to combine these polytopes and obtain $\mathcal{H}_c$.

Note that we drop the index a for the compressor station as well as the time index in this section for ease of notation.

3.1 Feasible operating range for a single compressor unit

For each compressor unit, we are given a feasible operating range of the corresponding compressor machine as a polytope in the space $(Q, \frac{p_r}{p_\ell})$, where the volumetric flow rate Q is defined as

$$Q = q/\rho_\ell, \quad \text{with } \rho_\ell = \frac{p_\ell}{R_s T z_\ell}.$$

Note that we consider the density ρ and the pressure p at the incoming node ℓ of the compressor. For this reason, we use for compressor units the incoming node's compressibility factor $z_\ell := z(p_{\ell,0})$ instead of the average value z_a used for pipes and resistors as defined in Sect. 2.2.

Usually, the feasible operating range, sometimes also referred to as "characteristic diagram" or "performance curve", is given as area in the dimensions (Q, H_{ad}) restricted by a set of possibly concave quadratic curves, see ,e.g., the description of Fügenschuh et al. (2015) or Odom and Muster (2009). The quantity H_{ad} denotes the *specific change in adiabatic enthalpy* and is defined as

$$H_{\mathrm{ad}} = R_s T z_\ell \frac{\kappa}{\kappa - 1}\left[\left(\frac{p_r}{p_\ell}\right)^{\frac{\kappa-1}{\kappa}} - 1\right], \tag{57}$$

using for the *isentropic exponent* κ the constant value 1.296, as stated by Fügenschuh et al. (2015). We can transform such an operating range into our format by applying the unique transformation from H_{ad} to $\frac{p_r}{p_\ell}$ obtainable from (57).

In addition to the feasible operating range polytope, each compressor unit is given an upper bound on the absolute pressure increase $\bar{\Delta}_p \geq p_r - p_\ell$ achievable by the compressor machine and an upper bound on the maximum power $\bar{P}$ that may be delivered by the drive. The power needed for compression is defined as

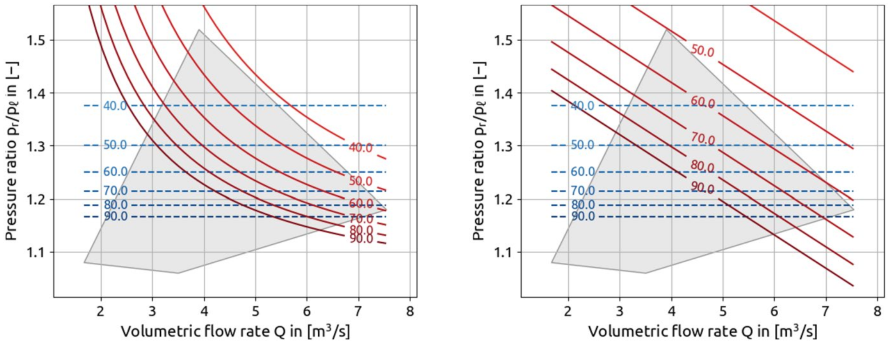
$$P = \frac{qH_{\mathrm{ad}}}{\eta_{\mathrm{ad}}} = \frac{q}{\eta_{\mathrm{ad}}} R_s T z_\ell \frac{\kappa}{\kappa - 1} \left[\left(\frac{p_r}{p_\ell} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right]. \tag{58}$$

Here, η_{ad} denotes the *adiabatic efficiency* of the compression, which we assume to be a known constant per compressor unit. We give an example of a feasible operating range of a compressor unit combined with the two constraints in Fig. 3a, where different levels of the maximum pressure difference bound and the maximum power constraint are denoted based on different values for the incoming pressure p_ℓ.

We now create a first version of the feasible operating range of the compressor unit in the space (p_ℓ, p_r, q) that does not include the maximum power constraint. Therefore, we transform each of the faces $\alpha_0 + \alpha_1 Q + \alpha_2 \frac{p_r}{p_\ell} \leq 0$ of the original polytope in the space $(Q, \frac{p_r}{p_\ell})$ into constraints $\tilde{\alpha}_0 p_\ell + \tilde{\alpha}_1 p_r + \tilde{\alpha}_2 q \leq 0$ of the desired higher-dimensional space:

$$\begin{aligned}
&\alpha_0 + \alpha_1 Q + \alpha_2 \frac{p_r}{p_\ell} && \leq 0 \\
\Leftrightarrow\, &\alpha_0 + \alpha_1 \frac{q R_s T z_\ell}{p_\ell} + \alpha_2 \frac{p_r}{p_\ell} && \leq 0 \\
\Leftrightarrow\, &\alpha_0 p_\ell + \alpha_1 R_s T z_\ell q + \alpha_2 p_r && \leq 0 \\
\Leftrightarrow\, &\tilde{\alpha}_0 p_\ell + \tilde{\alpha}_1 p_r + \tilde{\alpha}_2 q && \leq 0.
\end{aligned}$$

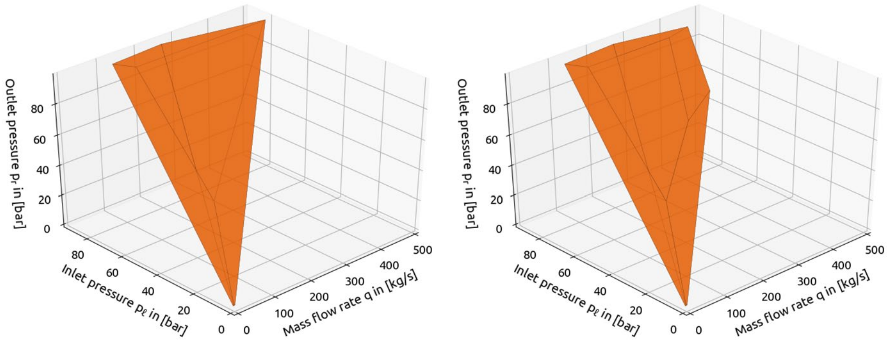
To bound the polyhedron described by the new constraints, we add the restriction of the absolute pressure difference as well as two pressure bounds of the end nodes of the compressor station arc $(\ell, r) = a \in \mathcal{A}^{\mathrm{cs}}$ containing the compressor unit.



(a) Original power bound

(b) Linearized power bound

Fig. 3 The feasible operating range of a compressor unit. The grey region shows the operating range given as a polytope. For different values of the incoming pressure p_ℓ the blue dashed lines represent the upper bound on the absolute pressure increase $\bar{\Delta}_p$, and the red solid lines illustrate the power bound $\bar{P}$. While the left picture shows the original nonlinear nonconvex constraint resulting from the power bound, the right picture shows the linearized version, see Sect. 3.2



(a) Without power bound

(b) With power bound

Fig. 4 The feasible operating range of a compressor unit in the space (p_ℓ, p_r, q), computed from the two-dimensional operating range shown in Fig. 3. While the left picture shows the lifted polytope based on the original feasible operating range, the maximum absolute pressure difference, and the end nodes pressure bounds, the right picture also includes the linearized constraint resulting from the power bound, see Sect. 3.2

$$p_r - p_\ell \leq \bar{\Delta}_p \quad p_\ell \geq \underline{p}_\ell \quad p_r \leq \bar{p}_r$$

A picture of the corresponding three-dimensional polytope created based on the feasible operating range of Fig. 3a is given in Fig. 4a. Note that it does not include the max power constraint yet.

3.2 Power bound linearization

To complete the three-dimensional feasible operation range polytope, we now add the maximum power constraint described in 3.1. Note that this constraint $P \leq \bar{P}$ cuts into the original two-dimensional feasible operating range in a nonlinear and non-convex fashion, see Fig. 3a. Since the same holds for the feasible operating range representation in (p_ℓ, p_r, q), we derive a linear approximation to this constraint.

For the approximation, we generate a set of N random sample points from within the three-dimensional operating range polytope of the compressor unit derived in the previous section. For each point, we determine the corresponding compression power using Eq. (58). Given this, we apply an ordinary least-squares method in order to determine the coefficients of the linear representation of P. This enables us to ultimately add the constraint in the form

$$\beta_0 + \beta_1 p_\ell + \beta_2 p_r + \beta_3 q \approx P \leq \bar{P} \quad \beta_0, \beta_1, \beta_2, \beta_3 \in \mathbb{R}$$

to the three-dimensional polytope of the operating range and thereby complete its definition. An example of the final polytope is illustrated in Fig. 4b, while the linearized power bound projected to the original two-dimensional operating range can be seen in Fig. 3b.

3.3 Feasible operating range for a compressor station configuration

As final step, we derive the polytope description for each configuration by combining the polytopes of the used compressor units. Each configuration is given as a serial sequence of parallel compressor machine arrangements. Therefore, we first combine the compressor unit polytopes of each parallel arrangement and then create the final configuration polytope from these. The set of facets of this configuration polytope ultimately defines the set of half-spaces $\mathcal{H}_c$. The overall procedure was introduced by Humpola et al. (2015), while we exactly followed the steps of the variant presented by Walther and Hiller (2017).

4 Specialized network station algorithm

The MIP model $\mathscr{P}$ presented in Sect. 2.11 turns out to be quite challenging according to our experiments, despite only considering a part of a larger gas network by restricting ourselves to a single network station. For this reason, we create a specialized heuristic algorithm to solve the problem. It should be able to efficiently solve the problem even on harder instances, i.e., those featuring complex network stations and many discrete time steps. Furthermore, we require it to reliably find feasible solutions for real-world instances and achieve a solution quality that is close to the optimum.

To reach these goals, we design the algorithm based on the before mentioned observation that pipes inside a network station are relatively short due to the proximity of the contained elements. This affects the pipes' capability to store gas, which is often referred to as *linepack*. The consideration of linepack is the main improvement of a transient pipe model over a stationary one. However, for short pipes, linepack is insignificant in comparison to the long pipelines outside of network stations, making the presented transient optimization problem a "stationary-flavored" one.

This property inspired us to determine the network station's operation modes and flow directions based on the solution of stationary (or steady-state) variants of $\mathscr{P}$ solved independently for each time step. The above description implies that these values should be similar to the ones of the optimal solution of the overall transient problem. Furthermore, the temporal decoupling results in a significant complexity reduction of the corresponding MIP models. In the algorithm, we ensure that the sequence of operation modes over time respects the transition time conditions introduced in Sect. 2.8, which we explicitly did not include in $\mathscr{P}$.

After determining the operation modes and flow directions, we obtain our final solution by using another variant of $\mathscr{P}$, which covers multiple time steps. By prescribing the values determined in the stationary calculations and solving the model in a rolling horizon fashion, we make this variant tractable as well. This procedure is called `transientSmoothing` and is the topic of Sect. 4.3. The whole algorithm is summarized in Algorithm 1, in which we split the creation of the sequence of operation modes and flow directions into the two-step procedure that we will present in Sect. 4.2.

```
opModes, flowDirs ← initialSolutionCreation()          // see Section 4.2.1
opModes, flowDirs ← improvementHeuristic(opModes, flowDirs) // see Section 4.2.2
return transientSmoothing(opModes, flowDirs)           // see Section 4.3
```

Algorithm 1: Specialized network station algorithm

Note that our algorithm has no guarantee of global optimality. We step-wise determine the decision variables without considering all potential consequences for the subsequent routines, which may lead to sub-optimal solutions. However, the order in which these decisions are made matters. Algorithm 1 first determines the operation modes and flow directions in the subroutines `initialSolutionCreation` and `improvementHeuristic`, while taking into account the deviation from the given future demands. The objective function weights associated with these decisions are w^{om} for changing operation modes, w^{us} for starting new compressor units, as well as $w^{\sigma\text{-p}}$ and $w^{\sigma\text{-d}}$ for not fulfilling the demand requirements. Usually, these weights are the most dominant ones, see the used objective function values in our computational experiments stated in Sect. 5.1. The remaining parts of the solution are obtained in the subsequent step `transientSmoothing`. Since here the operation modes and flow directions are already fixed, the remaining decisions may lead to higher values in the penalty variables, like for example those tracking the continuous change in the points of operation. However, due to the smaller objective weights these variables also have a smaller influence on the overall objective value. Hence we can summarize that Algorithm 1 is likely to obtain good overall solutions for the given problem, which is confirmed by the computational results we present in Sect. 5.

4.1 Model variants

As mentioned above, we do not solve the transient model $\mathscr{P}$ directly, but rather use the following variants in our overall solution approach.

4.1.1 $\mathscr{P}^{\text{s}}$—Stationary model

For the first variant, we solve a stationary version of the model to determine the best operation mode and flow direction for one independent time step t. The necessary changes in $\mathscr{P}$ mainly affect the pipes. Here, Eq. (13) is no longer part of the model. Furthermore, a pipe can no longer store gas, which balances the incoming and outgoing pipe flows, i.e., $q_{\ell,a,t} = q_{r,a,t} = q_{a,t}$ holds for all pipes $(\ell, r) = a \in \mathcal{A}^{\text{pi}}$. Thus, we can state the stationary version of the Momentum Eq. (15) for pipe $(\ell, r) = a \in \mathcal{A}^{\text{pi}}$ and stationary time step t as

$$p_{r,t} - p_{\ell,t} + \frac{\lambda_a L_a}{4D_a A_a}\left(|v_{\ell,a}| + |v_{r,a}|\right) q_{a,t} + \frac{g s_a L_a}{2R_s T z_a}\left(p_{\ell,t} + p_{r,t}\right) = 0.$$

For all other elements, the constraints only consider variables of the same time step. Therefore we simply apply these for the time step t of the stationary case. As a final step, we adjust the objective function. We keep the slack penalties for each boundary node, which are defined based on individual time steps. In addition, we still track the change to a new network station operation mode in variable δ^{om}_{t} as well as the start of new compressor units using $\delta^{\mathrm{us}}_{u,t}$ for all $u \in \mathcal{U}_a$ $a \in \mathcal{A}^{\mathrm{cs}}$. This is done by calling the model with the parameter `prevMode`, which represents the operation mode of the previous time step, see Algorithm 2. All the other variables that track changes are not considered. Therefore, we have the following stationary objective function for the time step t:

$$\begin{aligned} \min \quad & \big(\tau(t) - \tau(t-1)\big) \cdot \sum_{v \in \mathcal{V}^{\mathrm{b}}} w^{\sigma\text{-p}} \cdot (\sigma^{p+}_{v,t} + \sigma^{p-}_{v,t}) + w^{\sigma\text{-d}} \cdot (\sigma^{d+}_{v,t} + \sigma^{d-}_{v,t}) \\ & + w^{\mathrm{om}} \cdot \delta^{\mathrm{om}}_{t} + \sum_{u \in \mathcal{U}_a, a \in \mathcal{A}^{\mathrm{cs}}} w^{\mathrm{us}} \cdot \delta^{\mathrm{us}}_{u,t}. \end{aligned}$$

4.1.2 $\mathscr{P}^{\mathrm{f}}$—Transient with fixed operation modes and flow directions

In this variant, we are given for all future time steps $t \in \mathcal{T}$ a fixed operation mode o_t and flow direction f_t of the network station. This removes the corresponding variables $m^{\mathrm{om}}_{o_t,t}$ and $m^{\mathrm{fd}}_{f_t,t}$ from the model by setting them to 1 or 0, depending on whether they are chosen or not. Since the modes of valves and compressor stations, as well as the configurations of active compressor stations, depend on the selected operation mode of the network station, their corresponding variables are eliminated as well. Furthermore, all indicator constraints that use the binary variables mentioned above in their logical condition are resolved, i.e., we either remove them from the model if the condition is not fulfilled or add the implied constraint otherwise. Finally, we no longer need to use a disjunctive model for compressor stations. Instead, we apply the corresponding constraints (2)–(12) for the chosen mode and configuration of time step t directly to the variables $p_{\ell,t}$, $p_{r,t}$, and $q_{a,t}$ of each compressor station $(\ell, r) = a \in \mathcal{A}^{\mathrm{cs}}$.

4.1.3 $\mathscr{P}^{\mathrm{sf}}$—Stationary with fixed operation mode

For this last version of model $\mathscr{P}$, we combine the two variants above and use the stationary version of the model for one independent time step t with an already fixed operation mode. Note that in contrast to model $\mathscr{P}^{\mathrm{f}}$, the flow direction of the network station is not determined yet, which results in more binary variables and more indicator constraints that are not yet resolved.

4.2 Determining operation modes and flow directions

In this section, we present the procedure that finds a series of operation modes of the network station over time by solving the stationary variants of $\mathscr{P}$. We split the approach into two steps. First, we create an initial solution by a greedy, forward-oriented procedure presented in Sect. 4.2.1. It represents the function called `initialSolutionCreation` in Algorithm 1. In a second step, we try to improve this sequence of operation modes. Here, we replace certain operation modes of the sequence by similar ones in order to obtain a better solution. This second step is referred to as `improvementHeuristic` in Algorithm 1 and is described in Sect. 4.2.2. For the sake of efficiency, we try to reduce the number of times we need to solve model $\mathscr{P}^{\mathrm{s}}$ and try to use model $\mathscr{P}^{\mathrm{sf}}$ instead. In $\mathscr{P}^{\mathrm{sf}}$, we prescribe an operation mode for the network station, which makes the remaining problem very easy and extremely fast to solve. In contrast to this, the stationary model $\mathscr{P}^{\mathrm{s}}$ considering all available operation modes is solved significantly slower, especially when used on larger network stations. Therefore, we only use it if necessary.

Besides the operation mode, we return for each time step t a flow direction. The returned flow direction for t is the one determined by the stationary model that computed the optimal operation mode for t. For simplicity of notation, we omit the flow directions in the returned entities in the remainder of this section.

4.2.1 Initial solution creation

To find a first feasible sequence of operation modes over time, we follow a rather simple idea: We determine an operation mode for time step t by first testing the used operation mode of the previous time step $t-1$ using $\mathscr{P}^{\mathrm{sf}}$. The general stationary model $\mathscr{P}^{\mathrm{s}}$ to determine the best operation mode for t is only called when the previously used operation mode is infeasible or yields a costly solution in terms of the objective function value. Using this approach, we keep the number of needed operation mode changes small and speed up the algorithm by reducing the number of calls to $\mathscr{P}^{\mathrm{s}}$. A detailed description is given as Algorithm 2.

```
   Data: Operation mode $o_0$ of the network station in the initial state
   Result: A list of operation modes for each time $t \in \mathcal{T}_0$
 1 operationModes ← list()
 2 operationModes.add($o_0$)
 3 for $t \in \mathcal{T}$ do
 4     oldMode ← operationModes.last()
       // call $\mathscr{P}^{sf}$ with fixed operation mode oldMode for time t
 5     oldModeFeasible, oldModeCost
         ← $\mathscr{P}^{sf}$(oldMode, t, $prevMode$ = oldMode)
 6     if oldModeFeasible
         and modeAvailable(oldMode, t)
         and oldModeCost < $w^{om}$ then
           // operation mode of t-1 is also good for t
 7         operationModes.add(oldMode)
 8     else
           // operation mode of t-1 is NOT good for t
           // ⇒ search best possible valid operation mode for t
 9         validModes ← list()
10         for $o \in \mathcal{O}$ do
11             if modeAvailable(o, t)
                 and transitionsWork(concat(operationModes, list(o)))
                 and notSoonInfeasible(t, $newMode$ = o, $oldMode$ = oldMode) then
12                 validModes.add(o)
           // find best from validModes by calling $\mathscr{P}^{s}$ for time t
13         bestMode, bestModeFeasible, bestModeCost
             ← $\mathscr{P}^{s}$(t, $\mathcal{O}$ = validModes, $prevMode$ = oldMode)
14         if not bestModeFeasible then abort without solution
           // bestMode is best choice for t
15         operationModes.add(bestMode)
16 return operationModes
```

Algorithm 2: Initial solution creation

In Algorithm 2, we call $\mathscr{P}^{s}$ with the parameter validModes, which replaces the set of valid network station operation modes $\mathcal{O}$. Furthermore, we call the functions modeAvailable, transitionsWork, and notSoonInfeasible in Algorithm 2. The function modeAvailable deals with the operation mode unavailability introduced in Sect. 2.7, and checks if a given operation mode o is available at time t. In transitionsWork a given operation mode sequence is tested for validity regarding the transition time restriction presented in Sect. 2.8. Finally, we check in notSoonInfeasible, if choosing a new operation mode o for time t would result in an infeasibility at one of the subsequent time steps that is caused by a combination of operation mode unavailability and too long transition times. More specifically, we check if the new operation mode o for time t will become unavailable in one of the future time steps. If this is the case, we test if there is enough time left to transition into another operation mode until then. This also take into account the time needed by the transition from the old operation mode at time $t - 1$ to this new operation mode o at time t.

Note that Algorithm 2 may abort without a feasible solution at line 14. However, this is extremely unlikely to happen in real-world instances. A possible cause for failing at this line is that the algorithm may, due to a restrictive combination of operation mode availability and transition times, not find a valid operation mode to add to the set `validModes` before the call of the model $\mathscr{P}^{\mathrm{S}}$. There are multiple reasons why we do not expect this situation to happen. Firstly, the compressor units are rarely unavailable in practice, and even when all units of a compressor station are unavailable, the compressor station is still operational in bypass or closed modes. Secondly, the restrictions imposed by the transition times are usually rather mild in the sense that long transition times only occur for select pairs of operation modes. Finally, the function `notSoonInfeasible` prevents choosing an operation mode that may lead to a conflict in upcoming time steps. Hence it is likely that `validModes` contains at least one operation mode. A further possible cause for aborting the algorithm occurs when $\mathscr{P}^{\mathrm{S}}$ is infeasible for all given operation modes. This again is not expected for real-world instances, namely because the demand scenario in terms of pressure and flow is adjustable without limitation, even if these adjustments would result in a massive usage of the corresponding slack variables. Hence, the model can only be infeasible if all available operation modes have a conflict in every possible demand scenario. Therefore, infeasibility is a sign of a structural conflict in the network elements' modes and configurations imposed by the corresponding operation modes of the network station and is not assumed to happen for well-formed, realistic operation modes. The fact that Algorithm 2 has never aborted in one of our real-world test cases presented in Sect. 5.3 supports our argumentation. However, since we cannot guarantee feasibility here, we leave the removal of this theoretical weakness of Algorithm 2 open for future research.

```
Data: A valid sequence $S$ of operation modes over time
Result: A valid sequence $S^*$ of operation modes with $\mathsf{obj}(S^*) \leq \mathsf{obj}(S)$
backward ← $True$
while not having two iterations without improvements do
    changeTimes ← list of times $t$ with $S[t-1] \neq S[t]$
    if backward then reverse(changeTimes)
    for $t \in$ changeTimes do
        if $S[t-1] = S[t]$ then continue
        if backward then
            phaseToReplace ← phase ending at $t-1$
        else
            phaseToReplace ← phase starting from $t$
        $S^{\text{best}}$, bestImprovement ← list(), 0.0
        for newMode ∈ convexCombination($S[t-1], S[t]$) do
            $S^{\text{new}}$ ← $S$.replace(phaseToReplace, newMode)
            improvement ← $\mathsf{obj}(S) - \mathsf{obj}(S^{\text{new}})$
            if allModesAvailable($S^{new}$)
              and transitionsWork($S^{new}$)
              and improvement > bestImprovement then
                $S^{\text{best}}$, bestImprovement ← $S^{\text{new}}$, improvement
        if bestImprovement > 0 then
            // Found improvement in this iteration!
            $S \leftarrow S^{\text{best}}$
    backward ← not backward
return $S$
```

Algorithm 3: Improvement heuristic

4.2.2 An improvement heuristic

After having found an initial feasible solution, we look for further improvements using Algorithm 3. Therefore, we identify all sequences of identical operation modes over time in the solution. We call these sequences stable phases, or just *phases*, of a feasible solution. Obviously, the switch from one phase to the subsequent one happens when changing the operation mode. For each of these phases, we check if its operation mode can be replaced by a similar one that is more beneficial in terms of the objective function value. The general idea is to compare different network station operation modes over longer periods of time, which is in contrast to the focus on individual time steps implemented in Algorithm 2. Note that we only use the fast solving MIP model variant $\mathscr{P}^{\mathrm{sf}}$ here.

To generate new operation modes to check for improvements, we call the function `convexCombination`, which is the main feature of Algorithm 3. To define it, we use the function $M(o, a)$ that returns the mode or active configuration of a valve or compressor station a prescribed by operation mode o, see Sect. 2.7. Furthermore, we denote by $\mathcal{U}(x)$ the compressor units used in mode or configuration $x \in \{\mathrm{by}, \mathrm{cl}\} \cup \mathcal{C}_a$ for some compressor station $a \in \mathcal{A}^{\mathrm{cs}}$, where $\mathcal{U}(\mathrm{by}) = \mathcal{U}(\mathrm{cl}) = \emptyset$. We first define the function $\texttt{convexCombinationCS}_a$ on a tuple (x, y) with $x, y \in \{\mathrm{by}, \mathrm{cl}\} \cup \mathcal{C}_a$ as

$$\texttt{convexCombinationCS}_a(x,y) := \{x,y\} \quad \cup \left\{ c \in \mathcal{C}_a \quad \middle| \quad \left(\forall u \in \mathcal{U}(x) \cap \mathcal{U}(y) : u \in \mathcal{U}(c)\right) \wedge \left(\forall u \in \mathcal{U}(c) : u \in \mathcal{U}(x) \cup \mathcal{U}(y)\right) \right\}.$$

Note that $\texttt{convexCombinationCS}_a(x,y) \subseteq \{\mathrm{by}, \mathrm{cl}\} \cup \mathcal{C}_a$ holds. Now we can define `convexCombination` on a tuple (o_1, o_2) of operation modes as

$$\begin{aligned}\texttt{convexCombination}(o_1,o_2) := \Big\{ o \in \mathcal{O} \mid & \\ & \big(\forall a \in \mathcal{A}^{\mathrm{va}} : M(o,a) = M(o_1,a) \quad \vee \quad \forall a \in \mathcal{A}^{\mathrm{va}} : M(o,a) = M(o_2,a)\big) \\ & \wedge \quad \forall a \in \mathcal{A}^{\mathrm{cs}} : M(o,a) \in \texttt{convexCombinationCS}_a\big(M(o_1,a), M(o_2,a)\big) \Big\}.\end{aligned}$$

For a compressor station a, we permit it to be in a configuration using a compressor unit set "in-between" the used compressor unit set of the configurations used in o_1 and o_2 for a. However, we only allow the exact valve mode combination used in o_1 or the one used in o_2. This choice is based on the structure of operation modes described in Sect. 2.7. While changing the configuration of an active compressor station in an existing operation mode usually yields another existing operation mode, the set of valid combinations of valve modes is rather limited. Therefore, finding a valid valve setting from the combination of valve modes of two operation modes is unlikely and hence not included in the algorithm.

Algorithm 3 uses the function `transitionsWork`, which works in the same way as described above for Algorithm 2. Furthermore, it calls `allModesAvailable`, which is similar to `modeAvailable` from Algorithm 2. Here, we check the availability of a whole sequence of operation modes at those times corresponding to their positions in the sequence. Furthermore, we evaluate the objective function value of a sequence of operation modes using the function `obj` by successively calling $\mathscr{P}^{\mathrm{sf}}$ for each operation mode and time corresponding to its position in the sequence. If one of the models turns out to be infeasible, the returned objective function value is infinity.

Finally, we start the algorithm in the backward oriented mode since the initial solution is obtained by Algorithm 2, which works in the forward direction. Furthermore, we highlight that Algorithm 3 has the potential to reduce the total number of needed operation mode changes since the two original operation modes defining the change from one phase to the other are always part of the result of `convexCombination`. Since it is not possible to introduce new operation mode changes, we can even guarantee the new number of operation mode changes to be less than or equal to that in the initial solution.

4.3 Transient solution smoothing

As a final step of our specialized network station approach presented in Algorithm 1, we solve the transient model variant $\mathscr{P}^{\mathrm{f}}$ with fixed network station operation modes and flow directions. We obtain the operations modes as well as the flow directions for each time step from the stationary model solutions created in the previous steps of the algorithm.

```
Data: A sequence $S$ of tuples of operation modes and flow directions over time
Parameter: A time horizon size h
Result: A set of transient solution states for all $t \in \mathcal{T}_0$
if $\mathrm{h} \geq |\mathcal{T}_0|$ then
    // overall time horizon covered by smoothing time horizon
    return $\mathscr{P}^{\mathrm{f}}(S, \mathit{initialState} = \text{initialState})$
solutionStates ← list()
solutionStates.add(initialState)
currTime ← 0
while $\text{currTime} + h \leq |\mathcal{T}_0|$ do
    $S^{\mathrm{h}} \leftarrow S$.slice(currTime, currTime + h)
    thisTimeStates ← $\mathscr{P}^{\mathrm{f}}(S^{\mathrm{h}}, \mathit{initialState} = \text{solutionStates.last}())$
    if $\text{currTime} + h = |\mathcal{T}_0|$ then
        solutionStates.addAll(thisTimeStates)
    else
        solutionStates.add(thisTimeStates.first())
    currTime ← currTime + 1
return solutionStates;
```

Algorithm 4: Transient smoothing

In our computational experiments, we observed that, even though most of the binary decision variables of $\mathscr{P}$ are fixed in $\mathscr{P}^{\mathrm{f}}$, only a limited number of time steps can be solved for large network stations. Therefore, we use a *rolling horizon* approach to solve $\mathscr{P}^{\mathrm{f}}$, which we describe in Algorithm 4. The idea is to solve a series of smaller $\mathscr{P}^{\mathrm{f}}$ models with a fixed time horizon size h representing the number of time steps for which we solve, which include the time step for the given initial state. After each iteration, we save the solution regarding the earliest time step for the final result and shift the time horizon by 1 to prepare for the next call of $\mathscr{P}^{\mathrm{f}}$. In the function call to solve $\mathscr{P}^{\mathrm{f}}$ in Algorithm 4, we pass the operation modes and flow directions that correspond to the current time horizon as well as the fixed initial state.

The main benefit of this method is that increasing the size $|\mathcal{T}_0 := \{0, \dots, k\}|$ of the overall time horizon only increases the number of equally sized MIP models to solve, which hence have similar complexity. Therefore, the increase of $|\mathcal{T}_0 := \{0, \dots, k\}|$ is expected to result in a proportional run time increase. If we would have solved $\mathscr{P}^{\mathrm{f}}$ over the entire time horizon, increasing the time horizon would result in a larger complexity of the model, which may lead to an exponential increase in runtime instead.

5 Computational experiments

When designing Algorithm 1, we wanted it to be competitive in terms of the reliability to find a solution, the quality of that solution, and the overall execution time. In order to verify that we indeed achieved these objectives, we evaluated the algorithm on a large number of test instances. These instances represent network stations in the network of our project partner Open Grid Europe (OGE), for which we generated scenario values based on historic real-world situations.

5.1 Instances

We considered 7 different network stations from the network of OGE with different sizes and properties. An overview can be found in Table 1.

For each of these stations, we have a set of 159 instances. To create these instances, we solved a macroscopic gas flow model on an aggregated version of

Table 1 Overview of different properties of the 7 network stations A to G

Name	$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	$\frac{\sum_{a\in\mathcal{A}^{\mathrm{pi}}} L_a}{\|\mathcal{A}^{\mathrm{pi}}\|}$ (km)	$\|\mathcal{C}_a\| \forall a \in \mathcal{A}^{\mathrm{cs}}$	$\|\mathcal{O}\|$	$\|\mathcal{F}\|$	$\|\mathcal{V}^{\mathrm{b\text{-}ex}}\|$	$\|\mathcal{W}\|$
A	14	11	0.001	16	34	3	1	0
B	11	12	0.001	18	23	2	1	0
C	27	34	0.012	2	13	4	2	0
D	25	31	0.404	2, 6	92	6	2	0
E	48	67	0.308	3, 5	82	12	1	4
F	51	66	0.024	2, 3, 6, 12	2836	3	3	0
$\mathrm{G}_{\min}$	118	148	0.079	1, 1, 2, 7, 20	1267	15	2	0
$\mathrm{G}_{\max}$	120	150	0.079	1, 1, 2, 7, 20	1285	20	2	0

For station G, the topology changed during the considered time period. Hence, we denoted by $\mathrm{G}_{\min}$ the minimal values of all the quantities for station G and by $\mathrm{G}_{\max}$ the corresponding maximum values the complete network using simplified versions for all the network stations. A full description of this model is given by Hoppmann et al. (2019).

Each of the 159 instances corresponds to one point in time, for which we are given the actually occurred state of the network of our project partner. These points in time are distributed over 3 months and have a time difference of at least 12 hours. In addition to one of these states, we have measured pressure and flow values for the following 12 hours at the boundary nodes of the entire network. The macroscopic gas flow model can then be formulated using the state of the network as initial state and the pressure and flow values for the following 12 hours as demand values at the boundaries of the network. The solutions of the macroscopic gas flow model for each of these instances yield pressure and inflow values at the boundaries of each network station. The input data of one instance in the test set of this paper hence consists of a) the topology of one of the 7 network stations, b) the initial state values for the elements contained in this station for one of the 159 given points in time, and c) the pressure and inflow values at the network station's boundaries for the following 12 hours taken from the corresponding macroscopic gas flow model solution.

For the macroscopic gas flow model, the time horizon of 12 hours consists of 4 periods of 15 minutes followed by 11 periods of 60 minutes, resulting in 15 future time steps in total. For our analysis, we created the following four different partitions of the 12 hour time horizon built from left to right:

12 time steps:	$4 \times 15\,\text{min}, 5 \times 60\,\text{min}, 3 \times 120\,\text{min}$
24 time steps:	$4 \times 15\,\text{min}, 18 \times 30\,\text{min}, 2 \times 60\,\text{min}$
48 time steps:	$48 \times 15\,\text{min}$
96 time steps:	$96 \times 7.5\,\text{min}.$

Our partitioning of the time horizon can have additional or missing time points compared to the 15 time steps used in the macroscopic model. In order to create scenarios values that fit the time horizon, we interpolated using the original values.

To conclude, we used the following set of weights for the objective function for all instances for X$\in$ {rg, cs}:

$$w^{\sigma\text{-p}} = 1000.0/(\text{bar}\,\text{h}) \quad w^{\sigma\text{-d}} = 100.0/(1000\,\text{m}^3) \quad w^{\text{om}} = 1000.0 \quad w^{\text{us}} = 1200.0$$
$$w^{\text{rg}} = 50.0 \quad w^{\text{X-pl}} = w^{\text{X-pr}} = 10.0/(\text{bar}) \quad w^{\text{X-q}} = 1.0/(1000\,\text{m}^3/\text{h}).$$

The values regarding flow are based on the unit $1000\text{m}^3/\text{h}$ for volumetric flow under normal conditions. Using the linear transformation $[1000\text{m}^3/\text{h}] = \frac{3600}{1000\rho^0}[\text{kg/s}]$ based on the given normal density ρ^0 parameter, we can transform the corresponding values into mass flow. The unit of $w^{\sigma\text{-d}}$ is deduced by $\frac{1}{1000\,\text{m}^3/\text{h}\cdot\text{h}} = \frac{1}{1000\,\text{m}^3}$.

To give an impression of the instances' complexity, we list in Table 2 the size of the original model $\mathscr{P}$ for each network station and each of the above-defined time horizon partitions. The given numbers state the number of continuous variables, binary variables, and constraints, each rounded to the nearest hundred. As one can see, the values increase proportionally to the number of time steps. Note that we did not solve $\mathscr{P}$ directly but rather used our Algorithm 1 presented in Sect. 4 to obtain solutions for the problem. This algorithm also takes care of the restrictions regarding the operation mode transition times from Sect. 2.8, which were not included in $\mathscr{P}$.

5.2 Computational setup

We performed our computations on a cluster using 4 cores and 16 GB of RAM of a machine composed of two *Intel Xeon CPU E5-2670 v2* running at 2.50 GHz. As a solver for the underlying MIP problems, we used *Gurobi* in version 8.1.0, see Gurobi Optimization (2018), which we accessed via the *Pyomo* modeling language introduced by Hart et al. (2011, (2017). Since the corresponding MIP models were numerically challenging, we used the solver with the maximal *NumericFocus* parameter. Furthermore, we specified the optimality conditions and maximum run times for each solved model variant as

Variant	Rel. Gap	Abs. Gap	Time limit
$\mathscr{P}^{\mathrm{s}}, \mathscr{P}^{\mathrm{sf}}$	1E-4	1E-2	$10h$
$\mathscr{P}^{\mathrm{f}}$	5E-3	1E-2	$60s$.

Finally, we chose the rolling horizon parameter h to be 4, i.e., we solved the transient smoothing for 4 future time steps. We found in our experiments that this number represents a good trade-off between efficient model solving speed and foresightedness of the obtained solution.

5.3 Results

As described in Sect. 5.1 above, we tested Algorithm 1 on $159\text{ start times} \times 7\text{ network stations} \times 4\text{ time horizons } = 4452$ instances. For all of these instances, it found a feasible solution. In particular, we always found a feasible solution via the initial solution creation in Algorithm 2 described in Sect. 4.2.1, although the algorithm itself does not guarantee this.

In Fig. 5, one can find an overview of the average run times of Algorithm 1 sorted by network station and number of solved time steps. As one can see, the ranking of the single stations regarding run time as well as the ratio between the single stations run times is stable over the different time horizon partitions. Furthermore, the run time per station corresponds roughly to the complexity deduced from the station statistics in the sense that more nodes, arcs, and operation modes make the overall problem more complex to solve and therefore increase the run time. The exception to this reasoning is station C, which has a rather high average run time, although its node and arc counts are comparable to station D. Its number of operation modes is even the smallest of all seven stations. We assume that this is due to the station's topology, in which the regulators may be arranged in a way that is hard to solve for the final smoothing step of the algorithm. The higher portion of runtime spent in the smoothing of station C in comparison to station D is illustrated in Fig. 7. We have been unable to prove this assumption so far, however. Regarding model complexity, we did not observe that the existence of a non-empty set $\mathcal{W}$ of flow direction conditions adds any notable difficulty to solving station E, which is the only station

Table 2 The statistics of model $\mathscr{P}$ for each network station and each number of time steps. The triples state the number of continuous variables, binary variables, and constraints, each rounded to the nearest hundred. Note that $\mathscr{P}$ does not include the restrictions regarding the operation mode transition times from Sect. 2.8

Station	Number of time steps											
	12			24			48			96		
	Cont	Binary	Constr	Cont	Binary	Constr	Cont	Binary	Constr	Cont	Binary	Constr
A	2200	900	7300	4300	1700	14400	8500	3400	28,600	17,000	6800	57,100
B	2000	700	6900	4000	1300	13700	8000	2600	27,300	15,900	5200	54,400
C	3100	700	5000	6100	1500	10000	12,100	2900	19900	24,200	5900	39,800
D	2500	1700	8100	4900	3400	16000	9700	6800	31,800	19,300	13,600	63,500
E	5100	2000	11,800	10,100	3900	23,500	20,200	7800	46,800	40,300	15,700	93,500
F	6000	19700	117,300	11,900	36,600	231,600	23,700	70,400	460,100	47,300	137,900	917,500
G	10,200	18600	67,400	20,300	35,900	132,600	40,300	70,500	265,200	80,500	139,700	528,800

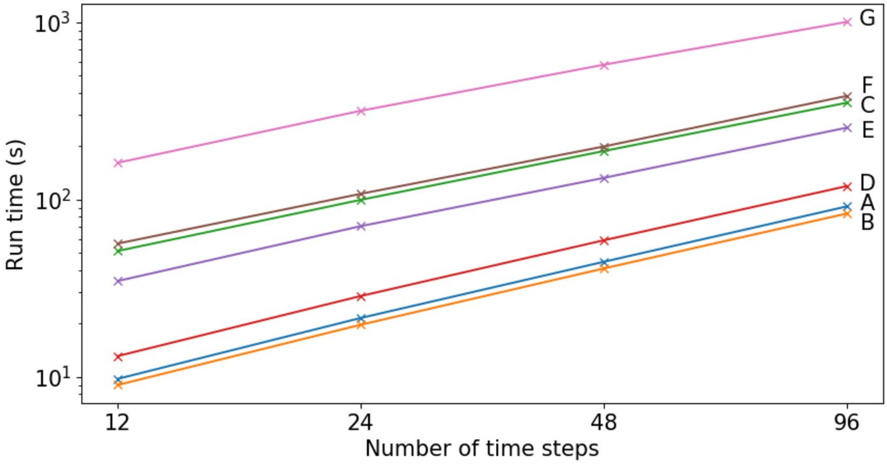


Fig. 5 Average run times for Algorithm 1 sorted by number of time steps and network station

incorporating these constraints. The last fact we want to highlight here is the almost perfectly proportional average run time increase with increasing number of time steps. As mentioned in Sect. 4.3, this property was an explicit goal when designing the algorithm.

In addition to the average run times, we also depict the distribution of run times sorted by station in Fig. 6 for instances solved for 12 time steps. The distribution of the other time steps are quite similar in appearance and can be found in the Appendix as Figs. 9, 10, and 11.

For the fast solving stations A, B, and D, the run times of the instances are all very similar in that the lines representing the 25th percentile and the 75th percentile are extremely close. Station C is again outstanding by having a rather large spread of run times only matched by the biggest and most complex station G. However, in general the maximum run time is less than an order of magnitude larger than the median run time for all of the stations. This means that for our instance set, even the extreme cases are still in reach and have a somewhat similar run time to the rest of the instances, making Algorithm 1 well suited for strict time limit situations, which we would face in a production environment.

As the last graphic related to the run time, Fig. 7 shows the average portion of run time spent in each individual subroutine of Algorithm 1 for 12 time steps as well as for 96 time steps. It states that the transient smoothing dominates the run time already for 12 time steps. This run time domination increases as the number of time steps do, while the heuristic's run time proportion, and therefore impact, decreases. This can also be verified through the 24 and 48 time step instances displayed in Fig. 12 in the Appendix. The observed time distributions fit the structure of the single subroutines from Sect. 4. While in the initial solve, as well as in the smoothing, more instances of the harder model variants $\mathscr{P}^{\mathrm{s}}$ and $\mathscr{P}^{\mathrm{f}}$ are solved with each additional time step, the improvement heuristic iterates over pairs of subsequent time steps with different operation modes. Since the differently sized time horizon

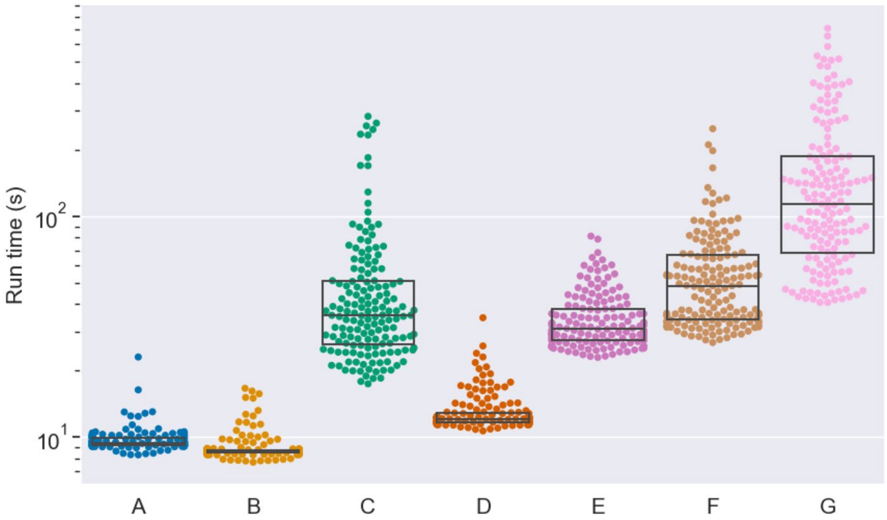
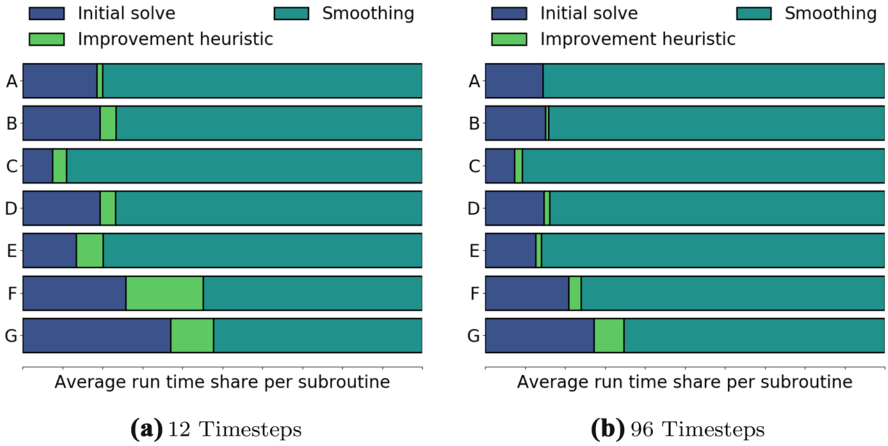


Fig. 6 Run times of instances with 12 time steps, sorted by station. One dot represents one instance, however, dots may overlap when too many have similar run time. In addition, we draw a typical whiskerless box plot. Therefore the lines represent the 25th percentile, the median, and the 75th percentile



(a) 12 Timesteps

(b) 96 Timesteps

Fig. 7 Portion of run time spent in the single subroutines of Algorithm 1 displayed for each of the single stations. Figure 7a represents the results for 12 time steps, Fig. 7b the results for 96 time steps

partitions all cover the same 12 hours, the number of total operation mode changes in the result of the initial solve is very unlikely to increase much with an increasing time granularity. This is also due to the initial solve algorithm itself, which tries to use the same operation mode as long as possible before changing to a different one.

Apart from the run time, we were also interested in the quality of the solution of Algorithm 1. Therefore, we solved the complete model $\mathscr{P}$ directly on the smaller network stations A to E to obtain a valid lower bound. Note that $\mathscr{P}$ itself is already a relaxation of the problem solved by Algorithm 1 since the transition time restrictions for operation modes introduced in Sect. 2.8 are not included in this model. The obtained lower bound can be used to give an upper bound on the relative difference of the solution found by Algorithm 1 to the optimal solution. We call this difference *gap* and define it as $\frac{\text{obj(solution)}-\text{lowerBound}}{\text{obj(solution)}}$. Since the lower bound cannot be negative, the gap is always smaller than or equal to 100%.

We solved model $\mathscr{P}$ on stations A to E for 12 time steps using the same computational setup as described above with a time limit of 10 hours. Furthermore, we gave the solution obtained from Algorithm 1 as a starting solution. The solver finished for all 159 instances of the stations A and B within the time limit. For the stations C, D, and E, we hit the time limit for 47, 8, and 18 instances, respectively, resulting in potentially sub-optimal lower bounds in these cases. For the stations F and G, more than half of the instances did not finish in time, which is why we did not include these in our evaluation.

The results can be found in Fig. 8, where we plotted the gap for all 5 network stations on a logarithmic scale and then plotted the same data for stations C and E

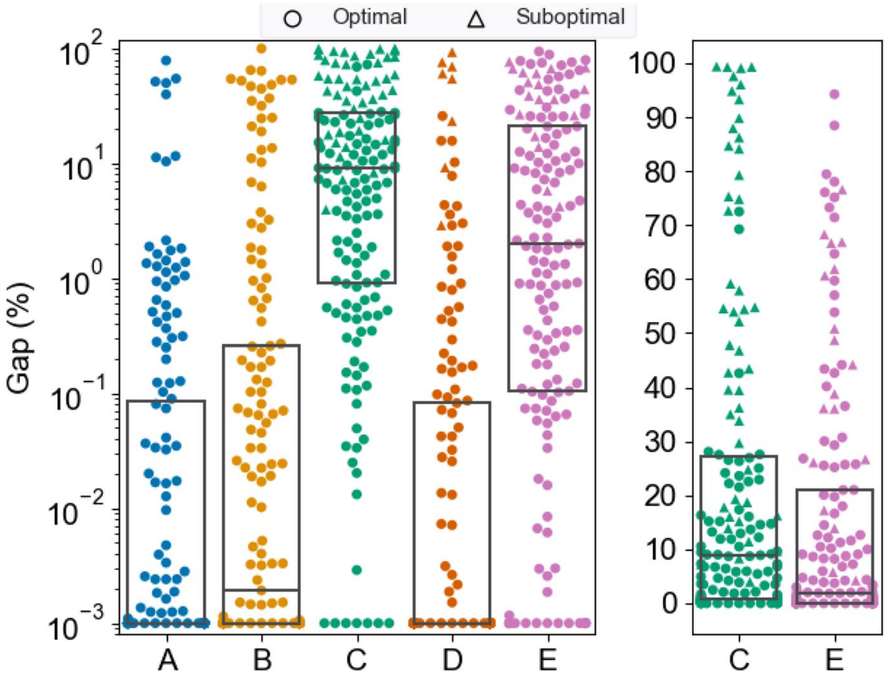


Fig. 8 Gap between the solution of Algorithm 1 and a lower bound obtained from solving $\mathscr{P}$ directly. We considered the stations A to E and solved them using 12 time steps. One dot represents one instance, however, dots may overlap when too many have similar run time. The instances with potentially suboptimal lower bounds due to hitting the time limit are marked by triangles instead of dots. In addition, we draw a typical whiskerless box plot. Therefore the lines represent the 25th percentile, the median, and the 75th percentile. The left graphic uses a logarithmic scale for the gap, and the right one displays the same data for stations C and E on a linear scale. For all instances in which the objective function value and the lower bound are smaller than 0.1, we defined the gap to be zero. Furthermore, we plotted all values below 10^{-3} as 10^{-3} for the logarithmic scale

on a linear scale to better depict their distributions. Note that the instances that have a potentially suboptimal lower bound are marked with triangles instead of dots. As one can observe, we have good results for all of the network stations, with more than half of the instances for each station having a solution with at most 10% difference to the lower bound. For the best three stations A, B, and D, we can even state that at least 75% of the values have less than 1% difference to the optimal solution. The picture is more diverse for the other two stations, in which the 75% percentile is between 20% and 30% gap. However, for those stations, we have the highest number of instances for which $\mathscr{P}$ did not solve the problem in time. Therefore, there is the potential for further improvement of the gap by increasing the lower bound here. Most of the instances with potentially suboptimal lower bounds also have high gap values. The picture is extreme for Station C, where all but 3 instances above the 75% percentile had suboptimal lower bounds, making it very likely that one can further decrease the gap value by increasing the run time limit of $\mathscr{P}$ directly.

Altogether, the results show that Algorithm 1 is able to find good solutions reliably and fast. Hence it can solve even large instances of the presented gas transportation problem, which would be far out of reach when trying to solve the original MIP model formulation $\mathscr{P}$ directly.

A comparison to experiments done in the existing literature on transient gas network optimization confirms our claim of a competitive algorithm, albeit the fact that a detailed evaluation of the different approaches is not possible since every publication considers a different model. However, we are in general able to solve our transient gas transportation problem faster than the algorithms presented in the literature solve their corresponding problem versions. This is true, even though we consider larger instances. All the experiments conducted by Domschke et al. (2011), Hahn et al. (2017), and Burlacu et al. (2019) consider networks of at most 20 nodes and arcs. Furthermore, the observed run times and given time limits are often larger than those of our algorithm. For the largest instance in Domschke et al. (2011) solved using 4 time steps, their algorithm hits the time limit of 10000 seconds with a gap of 34%. In Hahn et al. (2017), the time horizon was split into 180 time steps, while in only 10 of these time steps discrete control decisions were allowed. There, the solving times ranged from 2 hours up to the time limit of 50 hours. Finally, the largest instance considered in Burlacu et al. (2019) used 48 time steps, and their algorithm hits the time limit of 3 hours with a gap of 5.5%. However, as already mentioned above, the problems described in these publications differ from the one presented in this paper. In general, each of them considers a more accurate, nonlinear pipe model. At the same time, they only use simple models for the compressors consisting of single compressor units without configurations or interdependent modes. Furthermore, we included a series of realistic constraint types, like the choice of flow directions together with the corresponding exit pressure constraints and flow direction conditions, or the operation mode transition times. For these, only Burlacu et al. (2019) included a similar type of constraint by introducing switching times for valves and compressor units.

6 Conclusion

We presented the transient gas network transportation problem on network stations, which represent the intersection points of major transportation pipelines and contain the majority of active elements to control the network. For this problem, we introduced a mixed-integer programming model, including a sophisticated model for compressor stations, as well as additional variables and constraints for the network station itself. For the pipes, we found that due to their shortness, they have less overall impact in network stations, which enabled us to use a linear description for them. The MIP model was observed, through the use of state-of-the-art solvers, to be intractable for large network stations or a large number of time steps. Therefore, we developed a specialized algorithm to solve the problem. Its design is influenced by the fact that network stations contain only short pipes and hence a negligible amount of linepack. This causes the transient problem to be a “stationary-flavored” one. For this reason, our algorithm determines the operation modes and gas flow directions of the network station based on solving multiple stationary (or steady-state) variants of the MIP. By using this approach, we were also able to satisfy the transition time restrictions for the operation modes, which we excluded from the original MIP formulation. In order to obtain a feasible solution for the problem, we finally solved another variant of the MIP with fixed operation modes and flow directions for the network station in a rolling horizon fashion.

To verify the competitiveness of our algorithm regarding run time, reliability to find a solution, and quality of that solution, we performed tests on 159 different scenarios based on past flow situations in 7 real-world network stations provided by our project partner. Our algorithm was able to compute feasible solutions for all of the presented instances very fast. Even for the largest of the presented network stations consisting of more than 100 nodes and arcs as well as over 1000 different network station operation modes, our algorithm terminates on average in under 20 minutes for each of the stations. By running experiments using 4 different types of granularity, we found that the run time increases proportionally to the number of time steps used, indicating that our algorithm scales very well. In terms of quality of the solution, we tested the results against a lower bound obtained from solving the original MIP formulation for 12 time steps on all except the largest two network stations. For the worst of the five remaining stations, more than half of the instances had a solution with at most 10% difference to the lower bound. For the best three stations, we found near-optimal solutions with less than 1% difference for more than 75% of the instances. Altogether we were able to show that our algorithm can reliably find good solutions to the problem in a short amount of time.

There are a lot of different possibilities to continue this research. To increase the model accuracy, the approximative linearization of the friction in pipes and resistors, as well as the maximum power bound constraint of compressor units could be replaced by their original nonlinear versions. This would turn the model into a MINLP and thereby increase its complexity. From a theoretical point of view, extending Algorithm 2 such that it has the theoretical guarantee to always find existing feasible solutions would improve its robustness. Finally, we name ramp-up and cool-down times for compressor units, as well as using target value based control of regulators and compressor units, as examples of extensions to our model. These are examples of a never-ending list of unique elements and extra constraints that shapes the complicated business of real-world gas network operation.

Acknowledgements The work for this article has been conducted in the Research Campus MODAL funded by the German Federal Ministry of Education and Research (BMBF) (fund number 05M14ZAM)

Funding Open Access funding enabled and organized by Projekt DEAL.

Appendix

See Table 3; Figs. 9, 10, 11 and 12.

Table 3 List of all used variables, specifying their domain, meaning and unit. All variables are defined for $t \in \mathcal{T}$, not taking into account the given values for the initial state, see Sect. 2.9. Note that 1 bar = 10^5 Pa

Variable	Meaning	Unit
$p_{v,t} \in \mathbb{R}_{\geq 0}$	Pressure at node $v \in \mathcal{V}$	bar
$q_{a,t} \in \mathbb{R}$	Flow over arc $a \in \mathcal{A} \setminus \mathcal{A}^{\text{pi}}$	kg/s
$q_{v,a,t} \in \mathbb{R}$	Flow into or out of pipe $a \in \mathcal{A}^{\text{pi}}$	kg/s
$d_{v,t} \in \mathbb{R}$	Inflow into boundary node $v \in \mathcal{V}^{\text{b}}$	kg/s
$m^{\text{cf}}_{c,a,t} \in \{0, 1\}$	Selection of configuration $c \in \mathcal{C}_a$ for $a \in \mathcal{A}^{\text{cs}}$	1
$m^{\text{by}}_{a,t} \in \{0, 1\}$	Selection of bypass mode for $a \in \mathcal{A}^{\text{rg}} \cup \mathcal{A}^{\text{cs}}$	1
$m^{\text{cl}}_{a,t} \in \{0, 1\}$	Selection of closed mode for $a \in \mathcal{A}^{\text{rg}} \cup \mathcal{A}^{\text{cs}}$	1
$m^{\text{ac}}_{a,t} \in \{0, 1\}$	Selection of active mode for $a \in \mathcal{A}^{\text{rg}}$	1
$m^{\text{op}}_{a,t} \in \{0, 1\}$	Selection of open/closed mode for $a \in \mathcal{A}^{\text{va}}$	1
$p^{\text{by}}_{a,t} \in \mathbb{R}_{\geq 0}$	Pressure in bypass mode for $a \in \mathcal{A}^{\text{cs}}$	bar
$q^{\text{by}}_{a,t} \in \mathbb{R}$	Flow in bypass mode for $a \in \mathcal{A}^{\text{cs}}$	kg/s
$p^{\text{l-cl}}_{a,t} \in \mathbb{R}_{\geq 0}$	Incoming pressure in closed mode for $a \in \mathcal{A}^{\text{cs}}$	bar
$p^{\text{r-cl}}_{a,t} \in \mathbb{R}_{\geq 0}$	Outgoing pressure in closed mode for $a \in \mathcal{A}^{\text{cs}}$	bar
$p^{\text{l-cf}}_{c,a,t} \in \mathbb{R}_{\geq 0}$	Incoming pressure in configuration $c \in \mathcal{C}_a$ of $a \in \mathcal{A}^{\text{cs}}$	bar
$p^{\text{r-cf}}_{c,a,t} \in \mathbb{R}_{\geq 0}$	Outgoing pressure in configuration $c \in \mathcal{C}_a$ of $a \in \mathcal{A}^{\text{cs}}$	bar
$q^{\text{cf}}_{c,a,t} \in \mathbb{R}$	Flow in configuration $c \in \mathcal{C}_a$ of $a \in \mathcal{A}^{\text{cs}}$	kg/s
$m^{\text{om}}_{o,t} \in \{0, 1\}$	Selection of mode $o \in \mathcal{O}$	1
$m^{\text{fd}}_{f,t} \in \{0, 1\}$	Selection of flow direction $f \in \mathcal{F}$	1
$\sigma^{p+}_{v,t} \in \mathbb{R}_{\geq 0}$	Positive pressure slack for boundary node $v \in \mathcal{V}^{\text{b}}$	bar
$\sigma^{p-}_{v,t} \in \mathbb{R}_{\geq 0}$	Negative pressure slack for boundary node $v \in \mathcal{V}^{\text{b}}$	bar
$\sigma^{d+}_{v,t} \in \mathbb{R}_{\geq 0}$	Positive flow slack for boundary node $v \in \mathcal{V}^{\text{b}}$	kg/s
$\sigma^{d-}_{v,t} \in \mathbb{R}_{\geq 0}$	Negative flow slack for boundary node $v \in \mathcal{V}^{\text{b}}$	kg/s
$\delta^{\text{om}}_{t} \in \{0, 1\}$	Operation mode change	1
$\delta^{\text{rg}}_{a,t} \in \{0, 1\}$	Mode change for regulator $a \in \mathcal{A}^{\text{rg}}$	1
$\delta^{\text{us}}_{u,t} \in \{0, 1\}$	Start of compressor unit $u \in \mathcal{U}_a$ for $a \in \mathcal{A}^{\text{cs}}$	1
$\delta^{\text{rg-pl}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of incoming pressure of active $a \in \mathcal{A}^{\text{rg}}$	bar
$\delta^{\text{rg-pr}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of outgoing pressure of active $a \in \mathcal{A}^{\text{rg}}$	bar
$\delta^{\text{rg-q}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of flow over of active $a \in \mathcal{A}^{\text{rg}}$	kg/s
$\delta^{\text{cs-pl}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of incoming pressure of active $a \in \mathcal{A}^{\text{cs}}$	bar
$\delta^{\text{cs-pr}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of outgoing pressure of active $a \in \mathcal{A}^{\text{cs}}$	bar
$\delta^{\text{cs-q}}_{a,t} \in \mathbb{R}_{\geq 0}$	Change of flow over of active $a \in \mathcal{A}^{\text{cs}}$	kg/s

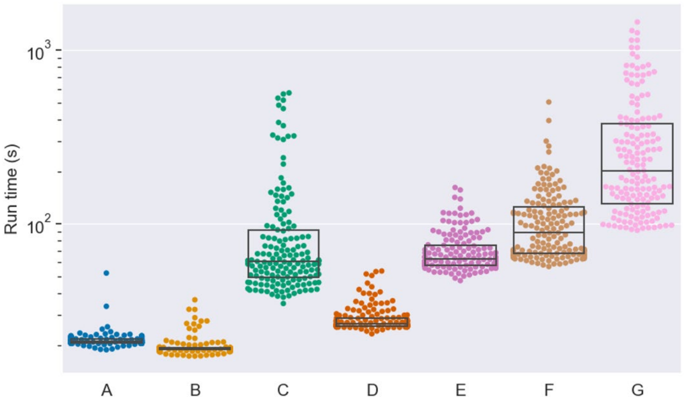


Fig. 9 Run times of instances with 24 time steps, sorted by station. One dot represents one instance, however, dots may overlap when too many have similar run time. In addition, we draw a typical whiskerless box plot. Therefore the lines represent the 25th percentile, the median, and the 75th percentile

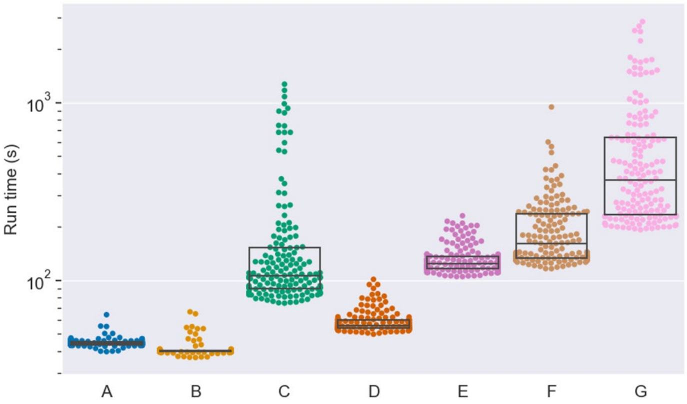


Fig. 10 Run times of instances with 48 time steps, sorted by station. One dot represents one instance, however, dots may overlap when too many have similar run time. In addition, we draw a typical whiskerless box plot. Therefore the lines represent the 25th percentile, the median, and the 75th percentile

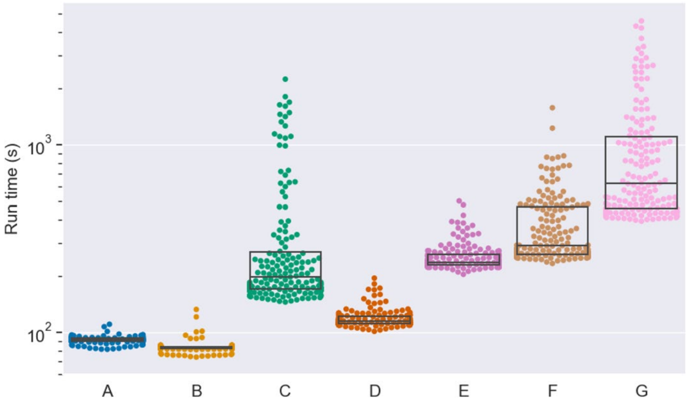
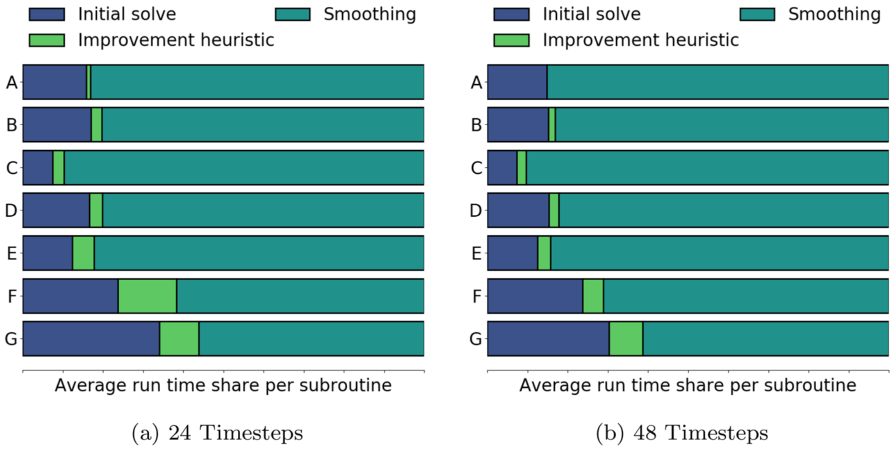


Fig. 11 Run times of instances with 96 time steps, sorted by station. One dot represents one instance, however, dots may overlap when too many have similar run time. In addition, we draw a typical whiskerless box plot. Therefore the lines represent the 25th percentile, the median, and the 75th percentile



(a) 24 Timesteps

(b) 48 Timesteps

Fig. 12 Portion of run time spent in the single subroutines of Algorithm 1 displayed for each of the single stations. Figure 12a represents the results for 24 time steps, Fig. 12b the results for 48 time steps

References

Balas E (1985) Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. SIAM J Algeb Discrete Methods 6(3):466–486. https://doi.org/10.1137/0606047

Balas E (2018) The convex hull of a disjunctive set. Disjunctive programming. Springer, Cham, pp 17–39

Bonami P, Lodi A, Tramontani A, Wiese S (2015) On mathematical programming with indicator constraints. Math Prog 151(1):191–223. https://doi.org/10.1007/s10107-015-0891-4

Brouwer J, Gasser I, Herty M (2011) Gas pipeline models revisited: model hierarchies, nonisothermal models, and simulations of networks. Multisc Model Simul 9(2):601–623. https://doi.org/10.1137/100813580

Burlacu R, Egger H, Groß M, Martin A, Pfetsch M, Schewe L, Sirvent M, Skutella M (2019) Maximizing the storage capacity of gas networks: a global MINLP approach. Optim Eng 20(2):543–573. https://doi.org/10.1007/s11081-018-9414-5

Domschke P, Geißler B, Kolb O, Lang J, Martin A, Morsi A (2011) Combination of nonlinear and linear optimization of transient gas networks. INFORMS J Comput 23(4):605–617

Domschke P, Hiller B, Lang J, Tischendorf C (2017) Modellierung von Gasnetzwerken: Eine Übersicht

Ehrhardt K, Steinbach MC (2003) Nonlinear optimization in gas networks. Technical Report 03-46, ZIB, Takustr. 7, 14195 Berlin

Federal Ministry for Economic Affairs and Energy (2019) Still indispensable for a reliable energy supply. https://www.bmwi.de/Redaktion/EN/Dossier/conventional-energy-sources.html. Accessed 31 Oct 2019

Fügenschuh A, Geißler B, Gollmer R, Morsi A, Pfetsch ME, Rövekamp J, Schmidt M, Spreckelsen K, Steinbach MC (2015) Physical and technical fundamentals of gas networks. In: Koch et al. (2015)

Gugat M, Leugering G, Martin A, Schmidt M, Sirvent M, Wintergerst D (2018) MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems. Comput Optim Appl 70(1):267–294. https://doi.org/10.1007/s10589-017-9970-1

Gurobi Optimization L (2018) Gurobi optimizer reference manual, version 8.1.0. http://www.gurobi.com

Hahn M, Leyffer S, Zavala VM (2017) Mixed-integer PDE-constrained optimal control of gas networks. Preprint

Hart WE, Watson JP, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in Python. Math Prog Comput 3(3):219–260

Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) Pyomo-optimization modeling in python, vol 67, 2nd edn. Springer, Berlin

Hennings F (2018) Benefits and limitations of simplified transient gas flow formulations. In: Operations research proceedings 2017. Springer, pp 231–237

Hoppmann K, Hennings F, Lenz R, Koch T (2019) Optimal operation of transient gas transport networks. Technical Report 19-23, ZIB, Takustr. 7, 14195 Berlin

Humpola J, Fügenschuh A, Hiller B, Koch T, Lehmann T, Lenz R, Schwarz R, Schweiger J (2015) The specialized MINLP approach. In: Koch et al. (2015)

Koch T, Hiller B, Pfetsch ME, Schewe L (eds) (2015) Evaluating gas network capacities, MOS-SIAM series on optimization, vol 21. SIAM

Kolb O, Lang J, Bales P (2010) An implicit box scheme for subsonic compressible flow with dissipative source term. Numeri Algorith 53(2–3):293–307

Mahlke D, Martin A, Moritz S (2007) A simulated annealing algorithm for transient optimization in gas networks. Math Methods Oper Res 66(1):99–115. https://doi.org/10.1007/s00186-006-0142-9

Mak TW, Van Hentenryck P, Zlotnik A, Hijazi H, Bent R (2016) Efficient dynamic compressor optimization in natural gas transmission systems. In: American control conference (ACC), 2016, IEEE, pp 7484–7491. https://doi.org/10.1109/ACC.2016.7526855

Moritz S (2007) A mixed integer approach for the transient case of gas network optimization. PhD thesis, Technische Universität Darmstadt, Darmstadt

Nikuradse J (1950) Laws of flow in rough pipes. National Advisory Committee for Aeronautics Washington

Odom FM, Muster GL (2009) Tutorial on modeling of gas turbine driven centrifugal compressors. In: PSIG-09A4, Pipeline Simulation Interest Group

Osiadacz AJ (1996) Different transient flow models-limitations, advantages, and disadvantages. In: PSIG-9606, Pipeline Simulation Interest Group

Papay (1968) A termeléstechnológiai paraméterek változása a gáztelepek muvelése során. OGIL Musz Tud Kozl

Pfetsch ME, Fügenschuh A, Geißler B, Geißler N, Gollmer R, Hiller B, Humpola J, Koch T, Lehmann T, Martin A, Morsi A, Rövekamp J, Schewe L, Schmidt M, Schultz R, Schwarz R, Schweiger J, Stangl C, Steinbach MC, Vigerske S, Willert BM (2014) Validation of nominations in gas network optimization: models, methods, and solutions. Optim Methods Softw. https://doi.org/10.1080/10556788.2014.888426

Ríos-Mercado RZ, Borraz-Sánchez C (2015) Optimization problems in natural gas transportation systems: a state-of-the-art review. Appl Energy 147:536–555

Saleh JM (2002) Fluid flow handbook. McGraw-Hill, London

Walther T, Hiller B (2017) Modelling compressor stations in gas networks. Technical Report 17-67, ZIB, Takustr. 7, 14195 Berlin

Zlotnik A, Chertkov M, Backhaus S (2015) Optimal control of transient flow in natural gas networks. In: 54th IEEE conference on decision and control (CDC), IEEE, pp 4563–4570. https://doi.org/10.1109/CDC.2015.7402932

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Felix Hennings[1,2] · **Lovis Anderson**[1] · **Kai Hoppmann-Baum**[1,2] · **Mark Turner**[1,2] · **Thorsten Koch**[2,1]

Lovis Anderson
anderson@zib.de

Kai Hoppmann-Baum
hoppmann-baum@zib.de

Mark Turner
turner@zib.de

Thorsten Koch
koch@zib.de

[1] Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany

[2] Chair of Software and Algorithms for Discrete Optimization, Institute of Mathematics, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany