A Consensus-Based Alternating Direction Method for Mixed-Integer and PDE-Constrained Gas Transport Problems

Richard Krug[1], Günter Leugering[1], Alexander Martin[1],
Martin Schmidt[2], Dieter Weninger[1]

Abstract. We consider dynamic gas transport optimization problems, which lead to large-scale and nonconvex mixed-integer nonlinear optimization problems (MINLPs) on graphs. Usually, the resulting instances are too challenging to be solved by state-of-the-art MINLP solvers. In this paper, we use graph decompositions to obtain multiple optimization problems on smaller blocks, which can be solved in parallel and which may result in simpler classes of optimization problems since not every block necessarily contains mixed-integer or nonlinear aspects. For achieving feasibility at the interfaces of the several blocks, we employ a tailored consensus-based penalty alternating direction method. Our numerical results show that such decomposition techniques can outperform the baseline approach of just solving the overall MINLP from scratch. However, a complete answer to the question of how to decompose MINLPs on graphs in dependence of the given model is still an open topic for future research.

1. Introduction

Many countries around the world strive towards a carbon-free energy supply but this energy turnaround is not yet accomplished. Hence, modern societies and economies still need to use energy carriers with a significant carbon footprint such as coal or natural gas, where the latter is often characterized as some kind of a bridging technology. To achieve their energy and climate goals, these societies and economies need to use resources like natural gas as efficiently as possible—an aspect that got even more important due to Russia's 2022 attack on Ukraine and the subsequent energy crisis in Europe. Obviously, not only the energy-saving usage of natural gas is important but also a highly efficient operation of the transport infrastructure is necessary. In this paper, we consider the latter aspect and derive novel algorithms to dynamically control gas transport networks.

From a mathematical point of view, this leads to very challenging optimization problems due to at least three reasons. First, appropriate models of gas physics are highly nonlinear. Second, the control of certain elements in gas transport networks such as valves or compressor stations introduce mixed-integer aspects. Third, dynamic effects lead to large-scale problems after the discretization of the considered time horizon. This all results in large-scale and nonconvex mixed-integer nonlinear models (MINLPs) that need to be solved on large transport networks if real-world problems are considered. Usually, the resulting instances are beyond the scope of what can be tackled even with today's most sophisticated MINLP solvers.

Our contribution is to exploit the special structure of the studied MINLPs that is induced by the transport networks, which are modeled by graphs. We decompose these graphs so that multiple smaller optimization problems can be solved on the resulting blocks. In addition

Date: August 14, 2023.

2020 *Mathematics Subject Classification.* 90Cxx, 90C11, 90C35, 90C90.

Key words and phrases. Gas transport networks, Mixed-integer nonlinear optimization, Alternating direction methods, Graph decomposition, Penalty methods.

to the reduced size of the graphs, and depending on the specific graph decomposition, the resulting optimization problems on the smaller blocks may also be in a simpler class of optimization problems because they, e.g., only contain mixed-integer linear, or nonlinear but purely continuous models. Of course, the graph decompositions lead to couplings between the different blocks that need to be satisfied to obtain a feasible solution on the overall transport network. We ensure this by a problem-tailored and consensus-based penalty alternating direction method (PADM) that allows for solving the problems on the separate components in parallel in every iteration of the PADM.

The main idea just sketched is motivated by related works on the infinite-dimensional level, where non-overlapping domain decompositions have been studied a lot in the last years [13, 14, 16]. In this contribution, however, we focus on finite-dimensional MINLPs obtained by a full discretization as they are also considered in, e.g., [2, 10, 26]. In contrast to such non-overlapping decomposition methods, overlapping methods have also been applied with great success in the recent years; see [24] for an application to power networks, [25] for decentralized schemes for general graph-structured problems, [17] for applications in nonlinear optimal control, [23] for some theoretical foundations for these overlapping methods, and [12] for a recent Julia implementation of the general ideas. At the core of many of these contributions are the key ideas of alternating directions methods of multipliers (ADMMs); see [1] for an overview. In the latter paper, the authors discuss so-called sharing and consensus methods, which we also exploit in our contribution. However, due to the mixed-integer nature of the problems we want to solve, ADMMs are not directly applicable and we thus resort to alternating direction methods (ADMs) that do not rely on duality reasoning, which is not available in our setting. ADMs have been already used to solve gas transport problems; see, e.g., [8, 9]. However, the decomposition of the model used in [9] is not graph- but physics-based and in [8], very specific graph-decompositions are used that we generalize in the present paper. Moreover, both papers tackle the stationary case whereas we now consider a time-dependent model based on the partial differential equations (PDEs) given by the Euler equations.

The remainder of this paper is structured as follows. In Section 2, we present the PDE-constrained optimization model that we consider and also discuss the used discretization of the Euler equations to obtain a finite-dimensional MINLP. Afterward, we review the basics of penalty alternating direction methods in Section 3 and then show in Section 4 how we can reformulate our original model so that it fits into the format to which the PADM can be applied. The numerical results are presented and discussed in Section 5 and the paper closes with a brief discussion of open research questions for future work in Section 6.

2. Gas Network Modeling

We consider a fixed and finite time horizon $[0, T]$ in which we control a gas network. The network itself is given as a directed and weakly connected graph (V, A). The nodes V consist of entries V_+, exits V_-, and inner nodes V_0, i.e.,

$$V = V_+ \cup V_- \cup V_0,$$

while the arcs are made up of pipes, short pipes, valves, compressor stations, and control valves:

$$A = A_{\mathrm{p}} \cup A_{\mathrm{sp}} \cup A_{\mathrm{v}} \cup A_{\mathrm{cs}} \cup A_{\mathrm{cv}}.$$

In what follows, we introduce models for the separate gas network components and state possible objective functions. To this end, we require some main physical quantities that will appear in all of the components' models: gas pressure $p = p(x, t)$ and gas mass flow

$q = q(x,t)$, which both depend on space x (in pipes) and time t. Often, we omit these dependencies in the following for the ease of better reading.

2.1. **Pipes.** The one-dimensional semi-linear Euler equations that we use to model the gas flow in a pipe $a \in A_{\mathrm{p}}$ are given by

$$\partial_t p_a + \frac{c^2}{A_a} \partial_x q_a = 0, \qquad (t,x) \in (0,T) \times (0,L), \tag{1a}$$

$$\partial_t q_a + A_a \partial_x p_a = -\frac{\lambda_a c^2}{2 D_a A_a} \frac{q_a |q_a|}{p_a} - \frac{g A_a h'_a}{c^2} p_a, \qquad (t,x) \in (0,T) \times (0,L); \tag{1b}$$

see, e.g., [11]. They consist of the continuity equation (1a) and the momentum equation (1b). Here and in what follows, A, D, L, and h' describe the cross-sectional area, the diameter, the length, and the slope of the pipe. Furthermore, c and g are the speed of sound in natural gas and the gravitational acceleration. In what follows, the notation $p_{a,u}(t)$ stands for the pressure in the pipe a at time t directly at the interface to node u.

There is a multitude of ways to model the friction coefficient λ_a; see, e.g., [6, 22]. For the modeling in this paper, we use the flow-independent law of Nikuradse, i.e.,

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

where the parameter k_a describes the roughness of the pipe's inner wall.

Finally, we assume that the mass flow inside of every pipe and every other element of the network is uniformly (over time) bounded from below and above, i.e.,

$$q_a(t,x) \in \left[q_a^-, q_a^+ \right] \tag{2}$$

holds on $(0,T) \times (0,L)$. The same holds true for pressures, i.e.,

$$p_a(t,x) \in \left[p_a^-, p_a^+ \right] \tag{3}$$

holds on $(0,T) \times (0,L)$.

2.2. **Short Pipes.** A short pipe $a = (u,v) \in A_{\mathrm{sp}}$ is an artificial modeling tool to directly connect two nodes so that there is no pressure loss, i.e.,

$$p_{a,u}(t) = p_{a,v}(t), \quad t \in [0,T], \tag{4}$$

and there is only one mass flow variable $q_a(t) = q_{a,u}(t) = q_{a,v}(t)$ for every point in time $t \in T$.

2.3. **Valves.** At each point in time $t \in [0,T]$, a valve $a = (u,v) \in A_{\mathrm{v}}$ is either open or closed. If it is open, it is modeled as a short pipe. If the valve is closed, gas flow is blocked and the pressures at both incident nodes are decoupled. Thus we have $p_{a,u}(t) = p_{a,v}(t)$ if the valve a is open and $q_a(t) = 0$, if the valve a is closed. This can be modeled by using a binary variable $o_a(t) \in \{0,1\}$ that indicates if the valve is open ($o_a(t) = 1$) or closed ($o_a(t) = 0$). The mixed-integer model for valves then reads

$$p_{a,u}(t) \geq p_{a,v}(t) - (p_a^+ - p_a^-)(1 - o_a(t)), \tag{5a}$$

$$p_{a,u}(t) \leq p_{a,v}(t) + (p_a^+ - p_a^-)(1 - o_a(t)), \tag{5b}$$

$$q_a(t) \geq q_a^- o_a(t), \tag{5c}$$

$$q_a(t) \leq q_a^+ o_a(t) \tag{5d}$$

for all $a \in A_{\mathrm{v}}$ and all $t \in [0,T]$.

2.4. **Compressor Stations.** Similar to valves, compressor stations $a \in A_{\text{cs}}$ can be open or closed. If they are open, they can be either active or in bypass mode. Active compressor stations can increase the pressure by a controlled amount $\Delta p_a \in [0, \Delta p_a^+]$, while compressor stations in bypass mode behave like short pipes. The flow direction of a compressor station $a \in A_{\text{cs}}$ is fixed, i.e., $q_a^- = 0$. It holds

$$\begin{aligned} p_{a,u}(t) + \Delta p_a(t) &= p_{a,v}(t), && \text{if the compressor station } a \text{ is active,}\\ p_{a,u}(t) &= p_{a,v}(t), && \text{if the compressor station } a \text{ is in bypass mode,}\\ q_a(t) &= 0, && \text{if the compressor station } a \text{ is closed.} \end{aligned}$$

To simplify the modeling, we assume that compressor stations are always open, which allows us to use the linear compressor station model

$$p_{a,u}(t) + \Delta p_a(t) = p_{a,v}(t), \tag{6}$$

where $\Delta p_a(t) = 0$ corresponds to the bypass mode.

If we want the pressure increase to be bounded via $0 < \Delta p_a^- \leq \Delta p_a(t) \leq \Delta p_a^+$ if the compressor station is active, we have to introduce a binary variable $b_a(t) \in \{0, 1\}$. This variable indicates if the compressor station is active ($b_a(t) = 1$) or in bypass mode ($b_a(t) = 0$). The mixed-integer compressor station model then reads

$$\Delta p_a(t) \geq \Delta p_a^- b_a(t), \tag{7a}$$

$$\Delta p_a(t) \leq \Delta p_a^+ b_a(t), \tag{7b}$$

$$p_{a,u}(t) + \Delta p_a(t) = p_{a,v}(t). \tag{7c}$$

We will analyze the computational impact of the different presented compressor models later in our numerical study. Note, however, that there are much more realistic nonlinear compressor station models in the literature; see, e.g., [18, 19]. These highly accurate models are, however, out of the scope of this paper.

2.5. **Control Valves.** We model control valves $a = (u, v) \in A_{\text{cv}}$ similar to compressor stations, but instead of increasing the pressure they can decrease it. This can be modeled by using the compressor station model (6) or (7) with $\Delta p \in [\Delta p^-, 0]$.

2.6. **Nodes.** At nodes $u \in V$, we have to model pressure continuity, which is given by

$$p_u(t) = p_{a,u}(t) \quad \text{for all } a \in \delta(u) \tag{8}$$

for all $t \in [0, T]$. It states that the node pressure $p_u(t)$ is equal to the adjacent arc pressures $p_{a,u}(t)$ at the interface to the respective node u. Here, we use $\delta(u) = \delta^{\text{in}}(u) \cup \delta^{\text{out}}(u)$, where $\delta^{\text{in}}(u)$ and $\delta^{\text{out}}(u)$ are the sets of in- and outgoing arcs of node u. In addition, the pressure at all nodes are bounded from below and above via

$$p_u(t) \in \left[p_u^-, p_u^+\right] \tag{9}$$

for all $t \in [0, T]$.

Furthermore, at nodes, we have to model mass balance via

$$q_u(t) + \sum_{a \in \delta^{\text{in}}(u)} q_{a,u}(t) - \sum_{a \in \delta^{\text{out}}(u)} q_{a,u}(t) = 0 \tag{10}$$

for $u \in V$ and $t \in [0, T]$, where $q_u(t)$ is the mass flow that is supplied or withdrawn from the network at node u. It holds

$$q_u(t) \begin{cases} \geq 0 & \text{for entries } u \in V_+, \\ \leq 0 & \text{for exits } u \in V_-, \\ = 0 & \text{for inner nodes } u \in V_0 \end{cases}$$

and the mass flow at nodes is also bounded from below and above, i.e.,

$$q_u(t) \in \left[q_u^-, q_u^+\right] \tag{11}$$

for all $u \in V$ and $t \in [0, T]$.

2.7. **Objective Function.** A classic objective function in gas transport optimization is to minimize the operational costs of the compressor stations, i.e., one wants to minimize

$$f_{\mathrm{c}} = \sum_{a \in A_{\mathrm{cs}}} \int_0^T \Delta p_a(t) \,\mathrm{d}t. \tag{12}$$

Another possibility is the tracking-type objective function on entries V_+ and exits V_-,

$$f_{\mathrm{t}} = \sum_{u \in V_+ \cup V_-} \eta \left(p_u(T) - \hat{p}_u\right)^2 + \theta \left(q_u(T) - \hat{q}_u\right)^2, \tag{13}$$

for given factors η, θ and target pressures $\hat{p}$ and mass flows $\hat{q}$.

2.8. **Problem Statement.** Combining all the parts presented before, we arrive at the optimization problem

$$\begin{aligned} \min \quad & (12) \text{ or } (13) \\ \text{s.t.} \quad & \text{Euler equations: } (1), \\ & \text{short pipe model: } (4), \\ & \text{valve model: } (5), \\ & \text{compressor station and control valve model: } (6) \text{ or } (7), \\ & \text{pressure continuity equations: } (8), \\ & \text{mass balance equation: } (10), \\ & \text{variable bounds: } (2), (3), (9), (11). \end{aligned} \tag{14}$$

The variables of this problem are the following. We have pressure $p_a(\cdot, \cdot)$ and mass flow variables $q_a(\cdot, \cdot)$ for all pipes $a \in A_{\mathrm{p}}$. Moreover, the model includes variables $p_{a,u}$, $p_{a,v}$, and q_a for short pipes, valves, compressor stations, and control valves, i.e., for $a = (u, v) \in A_{\mathrm{sp}} \cup A_{\mathrm{v}} \cup A_{\mathrm{cs}} \cup A_{\mathrm{cv}}$. Finally, we have variables o_a for valves $a \in A_{\mathrm{v}}$, Δp_a for compressor stations $a \in A_{\mathrm{cs}}$ and control valves $a \in A_{\mathrm{cv}}$ as well as pressure variables p_u for all nodes $u \in V$. Depending on the modeling choice, we may also include the variables b_a for compressor stations $a \in A_{\mathrm{cs}}$.

2.9. **Discretization.** Model (14) is a time-dependent mixed-integer optimization problem that is further constrained by partial differential equations. In this paper, we follow the first-discretize-then-optimize approach. To this end, we have to discretize the Euler equations (1) in space and time. For both dimensions, we apply the implicit Euler method. For time, we use the grid given by the points t_κ, $\kappa = 0, \ldots, N_t$, with $t_0 = 0$, $t_{N_t} = T$, and $t_\kappa < t_{\kappa+1}$. We assume that all variables at time point $t_0 = 0$ are fixed as they constitute the initial state of the system. For the spatial discretization, we use the grid given by the points $x_{a,\nu}$, $\nu = 0, \ldots, N_x^a$, with $x_{a,0} = 0$, $x_{a,N_x^a} = L_a$, and $x_{a,\nu} < x_{a,\nu+1}$. Note that the spatial grid depends on the specific pipe $a \in A_{\mathrm{p}}$. For the ease of presentation, we consider an equidistant discretization in time and space with a fixed time step

$$\Delta t = T/N_t = t_{\kappa+1} - t_\kappa \quad \text{for all } \kappa = 0, \ldots, N_t - 1$$

and spatial step

$$\Delta x_a = L_a/N_x^a = x_{a,\nu+1} - x_{a,\nu} \quad \text{for all } \nu = 0, \ldots, N_x^a - 1,\ a \in A_{\mathrm{p}}.$$

Using this notation, we get the fully discretized set of Euler equations

$$\frac{p_a(t_{\kappa+1}, x_{a,\nu+1}) - p_a(t_\kappa, x_{a,\nu+1})}{\Delta t} + \frac{c^2}{A_a} \frac{q_a(t_{\kappa+1}, x_{a,\nu+1}) - q_a(t_{\kappa+1}, x_{a,\nu})}{\Delta x_a} = 0, \tag{15a}$$

$$\begin{aligned} &\frac{q_a(t_{\kappa+1}, x_{a,\nu+1}) - q_a(t_\kappa, x_{a,\nu+1})}{\Delta t} + A_a \frac{p_a(t_{\kappa+1}, x_{a,\nu+1}) - p_a(t_{\kappa+1}, x_{a,\nu})}{\Delta x_a} \\ &\qquad + \frac{\lambda_a c^2}{2 D_a A_a} \frac{q_a |q_a|}{p_a}(t_{\kappa+1}, x_{a,\nu+1}) + \frac{g A_a h'_a}{c^2} p_a(t_{\kappa+1}, x_{a,\nu+1}) = 0 \end{aligned} \tag{15b}$$

for all $a \in A_{\mathrm{p}}$, $\kappa = 0, \dots, N_t$, and $\nu = 0, \dots, N_x^a$. We now replace (1) in Model (14) with (15) and further discretize the remaining algebraic constraints by simply writing them for all time points t_κ with $\kappa = 1, \dots, N_t$. Moreover, we discretize the objective function (12) via the right Riemannian sum

$$f_{\mathrm{c}} = \sum_{a \in A_{\mathrm{cs}}} \sum_{\kappa=1}^{N_t} \Delta t \Delta p_a(t_\kappa).$$

Note that $\Delta p_a(t_0)$ is given and thus not part of the sum in the objective function. The tracking-type objective function (13) stays unchanged because it only depends on the time point $t_{N_t} = T$. In this way, we obtain a large-scale but finite-dimensional and structured mixed-integer nonlinear optimization problem. Here, the nonlinearities are all contained in the model of the pipes, namely in the discretized momentum equation (15b).

3. Penalty Alternating Direction Method

In this section, we briefly review penalty alternating direction methods (PADM). We first sketch the original version of the method as introduced in [7] and then present a modified variant for so-called quasi-separable problems, which we will then apply to the discretized gas transport problem in the next section.

3.1. Classic Penalty Alternating Direction Methods.

For the moment, let us consider the optimization problem

$$\min_{x,y} \quad f(x, y) \tag{16a}$$

$$\text{s.t.} \quad g(x, y) = 0, \tag{16b}$$

$$x \in X, \ y \in Y \tag{16c}$$

with $X \subseteq \mathbb{R}^{n_x}$ as well as $Y \subseteq \mathbb{R}^{n_y}$ being non-empty and compact sets and $f : \mathbb{R}^{n_x + n_y} \to \mathbb{R}$ as well as $g : \mathbb{R}^{n_x + n_y} \to \mathbb{R}^m$ are continuous functions. Classic PADMs try to compute so-called partial minima of this problem, which are points (x^*, y^*) that satisfy

$$\begin{aligned} f(x^*, y^*) &\leq f(x, y^*) \quad \text{for all } x \in X \text{ with } g(x, y^*) = 0, \\ f(x^*, y^*) &\leq f(x^*, y) \quad \text{for all } y \in Y \text{ with } g(x^*, y) = 0. \end{aligned}$$

For what follows, we introduce the penalized objective function

$$\phi(x, y; \mu) = f(x, y) + \sum_{j=1}^m \mu_j g_j(x, y)^2 \tag{17}$$

with penalty parameters $\mu \in \mathbb{R}^m_{>0}$ and the penalty problem

$$\min_{x,y} \quad \phi(x, y; \mu) \quad \text{s.t.} \quad x \in X, \ y \in Y. \tag{18}$$

Note that we have a separate penalty parameter for every constraint. With this, we can state the PADM, which is formally given in Algorithm 1.

Algorithm 1 Penalty Alternating Direction Method

```
Require: $x^{0,0} \in X$, $y^{0,0} \in Y$, and penalty parameters $\mu^0 \in \mathbb{R}^m_{>0}$.
for $k = 0, 1, 2, \dots$ do
    Set $l = 0$.
    while $(x^{k,l}, y^{k,l})$ is not a partial minimum of (18) do
        Compute $x^{k,l+1} \in \arg\min_x \{\phi(x, y^{k,l}; \mu^k) \colon x \in X\}$.
        Compute $y^{k,l+1} \in \arg\min_y \{\phi(x^{k,l+1}, y; \mu^k) \colon y \in Y\}$.
        Set $l \leftarrow l + 1$.
    end while
    Choose new penalty parameters $\mu^{k+1} \geq \mu^k$.
    Set $(x^{k+1,0}, y^{k+1,0}) \leftarrow (x^{k,l}, y^{k,l})$.
end for
```

Let us also briefly sketch the convergence properties of Algorithm 1. To this end, we first make the following assumption that has already been mentioned above.

Assumption 1. *The objective function $f : \mathbb{R}^{n_x+n_y} \to \mathbb{R}$ and the constraint function $g : \mathbb{R}^{n_x+n_y} \to \mathbb{R}^m$ are continuous and the sets X and Y are non-empty and compact.*

With this assumption, we can state the main convergence theorem for Algorithm 1, which is taken from [7].

Theorem 1. *Suppose that Assumption 1 holds and that $\mu_j^k \nearrow \infty$ for all $j = 1, \dots, m$. Moreover, let (x^k, y^k) be a sequence of partial minima of (18) (for $\mu = \mu^k$) generated by Algorithm 1 with $(x^k, y^k) \to (x^*, y^*)$. Then, there exist $\bar{\mu} \geq 0$ such that (x^*, y^*) is a partial minimizer of the weighted ℓ_1 feasibility measure*

$$\chi_{\bar{\mu}}(x, y) := \sum_{j=1}^{m} \bar{\mu}_j |g_j(x, y)|.$$

If, in addition, (x^*, y^*) is feasible for the original problem (16), the following holds:

(a) *If f is continuous, then (x^*, y^*) is a partial minimum of (16).*
(b) *If f is continuously differentiable, then (x^*, y^*) is a stationary point of (16).*
(c) *If f is continuously differentiable as well as convex and if the feasible set of Problem (16) is convex, then (x^*, y^*) is a global optimum of (16).*

3.2. A Penalty Alternating Direction Method for Quasi-Separable Problems.

We now present a reformulation for problems of the form

$$\min_{x^1, \dots, x^\ell, z} \quad \sum_{i=1}^{\ell} f_i(x^i) \tag{19a}$$

$$\text{s.t.} \quad (x^i, z) \in X_i \quad \text{for } i = 1, \dots, \ell, \tag{19b}$$

$$z \in Z \tag{19c}$$

with $X_i \subseteq \mathbb{R}^{n_i} \times \mathbb{R}^{n_z}$, $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ for $i = 1, \dots, \ell$, and a convex set $Z \subseteq \mathbb{R}^{n_z}$. The problem consists of ℓ constraint blocks X_i that are connected via the so-called linking variables $z \in Z$. In the next section, we will see that we can put the fully discretized gas transport problem (14) in this format in order to then apply a tailored PADM to solve it. Informally speaking, we call such problems "quasi-separable" if the dimension of the coupling variables in (19c) is not too large.

We reformulate problem (19) by introducing a copy z^i of z for each block, which yields the equivalent problem

$$\begin{aligned}
\min_{x^1,\dots,x^\ell,z^1,\dots,z^\ell,z} \quad & \sum_{i=1}^{\ell} f_i(x^i) && \text{(20a)}\\
\text{s.t.} \quad & (x^i, z^i) \in X_i \quad \text{for } i = 1,\dots,\ell, && \text{(20b)}\\
& z \in Z, && \text{(20c)}\\
& z^i = z \quad \text{for } i = 1,\dots,\ell, && \text{(20d)}\\
& z^i \in Z \quad \text{for } i = 1,\dots,\ell. && \text{(20e)}
\end{aligned}$$

Next, we apply Algorithm 1 to Problem (20) with $x = (x^1, z^1, \dots, x^\ell, z^\ell)$, $y = z$, $X = (X_1 \cap (\mathbb{R}^{n_1} \times Z)) \times \cdots \times (X_\ell \cap (\mathbb{R}^{n_\ell} \times Z))$, and $Y = Z$. Moreover, the constraints g are identified with the equality constraints (20d). The optimization problem in Step 4 of Algorithm 1 is then given by

$$\begin{aligned}
\min_{x^1,\dots,x^\ell,z^1,\dots,z^\ell} \quad & \sum_{i=1}^{\ell} \left(f_i(x^i) + \sum_{j=1}^{n_z} \mu_j^i \left(z_j^i - (z_j)^{k,l}\right)^2 \right)\\
\text{s.t.} \quad & (x^i, z^i) \in X_i \quad \text{for } i = 1,\dots,\ell,\\
& z^i \in Z \quad \text{for } i = 1,\dots,\ell.
\end{aligned}$$

This problem is completely separable. Therefore, we can instead solve

$$\begin{aligned}
\min_{x^i,z^i} \quad & f_i(x^i) + \sum_{j=1}^{n_z} \mu_j^i \left(z_j^i - (z_j)^{k,l}\right)^2 && \text{(21a)}\\
\text{s.t.} \quad & (x^i, z^i) \in X_i, && \text{(21b)}\\
& z^i \in Z && \text{(21c)}
\end{aligned}$$

for all blocks $i = 1,\dots,\ell$, which can be done in parallel. The optimization problem in Step 5 of Algorithm 1 reads

$$\min_{z\in Z} \sum_{i=1}^{\ell} \sum_{j=1}^{n_z} \mu_j^i \left(\left(z_j^i\right)_k^{l+1} - z_j \right)^2 .$$

The solution for this so-called consensus problem is the weighted arithmetic mean

$$z_j = \frac{1}{\sum_{i=1}^{\ell} \mu_j^i} \sum_{i=1}^{\ell} \mu_j^i \left(z_j^i\right)_k^{l+1}$$

for all $j = 1,\dots,n_z$, since Z is convex and $\left(z^i\right)_k^{l+1} \in Z$ for all $i = 1,\dots,\ell$ because of (20e); see, e.g., [1].

4. PADM Tailored Reformulation of the Gas Network Problem

In this section, we describe how we can state the gas transport problem introduced in Section 2 in the format derived in the last section to then apply the PADM.

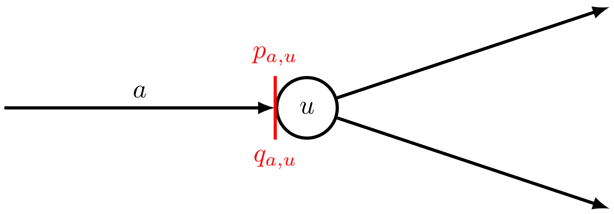

FIGURE 1. The vertical red line represents a possible decomposition at node u that minimizes the number of linking variables.

4.1. **Decomposition of Gas Networks.** To efficiently apply the PADM to the discretized version of Problem (14), we need to identify a quasi-separable structure as in (19). To this end, we can utilize the fact that we model the problem on a graph. As each variable can be associated with a certain location in this graph it follows that decomposing the graph yields corresponding blocks (indexed with i) of variables with respective constraint sets X_i. If the network is decomposed in a way that the resulting sub-networks are only coupled in a sparse way, then the number of linking variables z stays small and we have the desired quasi-separable structure.

The question is still open on how to decompose the graph. One might assume that it is a good idea to decompose a network at selected nodes of the network. This, however, significantly increases the number of linking variables if the degree of the respective nodes is large since the pressures and mass flows of all adjacent arcs are connected across the node by the pressure continuity equation (8) and the mass balance equation (10). This undesired aspect can be avoided by splitting the network at points that lie on arcs instead of nodes. To still assign each arc to exactly one sub-network, we decide to only split an arc at the interface to its start or end node. Figure 1 depicts a possible decomposition at the interface to a node of degree three. The only linking variables in this situation are $p_{a,u}$ and $q_{a,u}$ for all time points t_κ.

More precisely, we decompose the graph (V, A) into blocks (V_i, A_i) for $i = 1, \dots, \ell$ with

$$V = \bigcup_{i=1}^{\ell} V_i, \quad A = \bigcup_{i=1}^{\ell} A_i$$

and

$$\begin{aligned} V_i \cap V_j &= \emptyset \quad \text{for } i, j = 1, \dots, \ell,\ i \neq j, \\ A_i \cap A_j &= \emptyset \quad \text{for } i, j = 1, \dots, \ell,\ i \neq j. \end{aligned}$$

Thus, each node $u \in V$ and each arc $a \in A$ belongs to exactly one block (V_i, A_i). Note that a block (V_i, A_i) itself is not a graph, as $a = (u, v) \in A_i$ does not imply $u \in V_i$ and $v \in V_i$. Vice versa, it can happen that $a \notin A_i$ for some arc $a \in \delta^{\text{in}}(u) \cup \delta^{\text{out}}(u)$ with $u \in V_i$. These cases appear exactly at the interfaces of the blocks. We define the interfaces of a single block $i \in \{1, \dots, \ell\}$ as

$$I_i := \{(u, a) \in V_i \times A \colon a \in \delta(u) \text{ and } a \notin A_i\} \cup \{(u, a) \in V \times A_i \colon a \in \delta(u) \text{ and } u \notin V_i\}.$$

The set of all interfaces is then given by

$$I := \bigcup_{i=1}^{\ell} I_i.$$

We now use this graph decomposition for gas networks. The constraint system of Problem (14) can then be written as

Euler equations: (1) for $A_{\mathrm{p}} \cap A_i$,
short pipe model: (4) for $A_{\mathrm{sp}} \cap A_i$,
valve model: (5) for $A_{\mathrm{v}} \cap A_i$,
compressor station and control valve model: (6) or (7) for $(A_{\mathrm{cs}} \cup A_{\mathrm{cv}}) \cap A_i$,
pressure continuity equations: (8) for V_i,
mass balance equation: (10) for V_i,
variable bounds: (2), (3), (9), (11) for (V_i, A_i).

to fit to the constraints (20b) and (20e) for all $i = 1, \dots, \ell$. The objectives (12) and (13) can be rewritten as

$$f_{\mathrm{c}} = \sum_{i=1}^{\ell} \sum_{a \in A_{\mathrm{cs}} \cap A_i} \int_0^T \Delta p_a(t) \,\mathrm{d}t$$

and

$$f_{\mathrm{t}} = \sum_{i=1}^{\ell} \sum_{u \in (V_+ \cup V_-) \cap V_i} \eta \left(p_u(T) - \hat{p}_u\right)^2 + \theta \left(q_u(T) - \hat{q}_u\right)^2$$

to fit into the form of (20a). The linking constraints (20d) read

$$\begin{aligned} p^i_{a,u} &= p_{a,u} \quad \text{for } (u,a) \in I_i, \; i = 1, \dots, \ell, \\ q^i_{a,u} &= q_{a,u} \quad \text{for } (u,a) \in I_i, \; i = 1, \dots, \ell. \end{aligned}$$

Finally, the constraints (20c) of the consensus problem are given by

$$\begin{aligned} p_{a,u} &\in \left[p_a^-, p_a^+\right] \quad \text{for } (u,a) \in I, \\ q_{a,u} &\in \left[q_a^-, q_a^+\right] \quad \text{for } (u,a) \in I. \end{aligned}$$

It is easy to see that the resulting optimization problem satisfies Assumption 1 if the original gas network problem (14) is feasible. The objective function and the constraint functions are all continuous. The blocks X and Y are non-empty because a feasible point of (14) can directly be transformed into a feasible point of the reformulated problem. Compactness follows from the variable bounds as well as the constraints in (14). Therefore, the convergence result of Theorem 1 can still be applied. Note, however, that the optimization problems in Step 4 and Step 5 of Algorithm 1 need to be solved to global optimality. For general nonconvex NLPs or MINLPs, this can be a challenging task in practice. Nevertheless, even if we do not solve all sub-problems to global optimality, we can use Algorithm 1 to compute feasible solutions of good quality if the algorithm converges. We will see in Section 5 that this is almost always the case for the instances that we test in our numerical experiments.

Remark 1. (1) *Note that there are multiple strategies to decompose a gas network. If more sub-networks are created, the number of linking variables increases, which potentially makes it harder to find a point that is feasible at all the sub-networks' interfaces. This can lead to a higher number of iterations required by Algorithm 1 and, therefore, longer computation times. However, multiple smaller problems may be easier to solve than a few larger ones. This trade-off needs to be considered when choosing the number of sub-networks.*

(2) *Another relevant question is which properties the sub-networks should have for Algorithm 1 to converge quickly. One of the approaches that we later test is to separate the active elements of the network, i.e., valves, compressor stations, and control valves, from the passive pipe network. The problems on the resulting sub-networks then contain either discrete variables and no nonlinearities, or no discrete variables but nonlinear constraints. Therefore, these problems are MIPs or NLPs but not MINLPs anymore, which are usually much harder to solve. One drawback of this approach is that the sub-networks containing active elements are rather small compared to larger passive sub-networks, which may lead to many iterations that are required to obtain points that are feasible at the blocks' interfaces. A different approach is to create problems on sub-networks that are roughly equally hard to solve. Then, the fact that the sub-problems (21) can be solved in parallel can be utilized most effectively. We will explore these approaches further in multiple case studies in Section 5.*

4.2. **Penalty Update Strategies.** Theorem 1 requires the penalty parameters μ to go to infinity for Algorithm 1 to converge. This is ensured by updating μ in Step 8 of the algorithm. However, it is not obvious what a good update strategy for these parameters is. One the one hand, large μ can lead to numerical instability and worse objective values as the penalty term will dominate the original objective function in (17). On the other hand, slowly growing μ may lead to slow convergence rates because the feasibility violations at the interfaces of the blocks are not penalized strong enough. Therefore, it is reasonable to increase those μ_j^i stronger for which $((z_j^i)_k^{l+1} - z_j)^2$ is large while increasing those μ_j^i with small $((z_j^i)_k^{l+1} - z_j)^2$ only a little.

While it is theoretically possible to use a separate penalty parameter μ_j^i for each linking variable z_j^i of block i, it might be a good idea to use a combined parameter μ for multiple variables z_j^i of the same block that represent similar quantities. In the case of gas networks, the linking variables consist of pressure values $p_{a,u}^i$ and mass flow values $q_{a,u}^i$ at different time points and different locations in the network. Our preliminary numerical tests showed that having different penalty parameters for different pressure variables (or flow variables, respectively) can lead to strange and unintuitive controls in a block. To avoid this, it therefore makes sense to introduce a single penalty parameter μ_p^i for the pressures and a single penalty parameter μ_q^i for the mass flows of each block i. Note that for different blocks we still use different penalty parameters as these blocks should be clearly separated from each other while only exchanging information through the linking variables at their interfaces.

The combination of these ideas leads to the following update formulas for the penalty parameters in Step 8 of Algorithm 1:

$$\left(\mu_p^i\right)_{k+1} = \left(1 + \omega_\mu \frac{m_{p,i}}{\max_{j=1}^{\ell} m_{p,j}}\right) \left(\mu_p^i\right)_k, \quad i = 1, \dots, \ell,$$

$$\left(\mu_q^i\right)_{k+1} = \left(1 + \omega_\mu \frac{m_{q,i}}{\max_{j=1}^{\ell} m_{q,j}}\right) \left(\mu_q^i\right)_k, \quad i = 1, \dots, \ell,$$

with

$$m_{p,i} := \max_{(u,a)\in I_i, \kappa=0,\dots,N_t} \left(p_{a,u}^i(t_\kappa) - p_{a,u}(t_\kappa)\right)^2, \quad i = 1, \dots, \ell,$$

$$m_{q,i} := \max_{(u,a)\in I_i, \kappa=0,\dots,N_t} \left(q_{a,u}^i(t_\kappa) - q_{a,u}(t_\kappa)\right)^2, \quad i = 1, \dots, \ell,$$

where we omitted the iteration indices k and l on the pressure and mass flow variables for the ease of presentation. Finally, $\omega_\mu > 0$ serves as a scaling factor.

5. Numerical Results

In this section, we apply the PADM for quasi-separable problems to transient gas network optimization. To this end, we implemented the method in Python 3.8. We use GAMS 36.2.0 [3] to model the optimization problems, which we then solve using the solver CPLEX 20.1.0.1 [5] for LPs and MIPs and the solver KNITRO 12.4.0 [4] for NLPs and MINLPs.

Let us briefly comment on the choice of KNITRO as the (MI)NLP solver. We also tested other MINLP solvers but KNITRO delivered the best results. However, KNITRO is a not a global solver but only solves nonconvex (MI)NLPs to local optimality. Since the different decompositions tested in the following lead to different models, the used solver might end up in different local minimizers, which is important for understanding the reported objective function values. In our opinion, the choice of KNITRO, even if it is a local solver, is legitimate since we are mainly interested in running times and since the overall objective values of KNITRO are rather good.

All computations were done on nodes of the HPC cluster Woody at FAU Erlangen-Nürnberg with Xeon E3-1240 v6 CPUs on 4 cores, 3.7 GHz, and 32 GB RAM. For the discretization, we choose a time step size of $\Delta t = 3600\,\mathrm{s} = 1\,\mathrm{h}$ and a spatial step size of $\Delta x = 5000\,\mathrm{m}$. For each problem, we set a time limit of 1000 s.

To speed up the convergence of Algorithm 1, we limited the number of ADM loops (Steps 3–7) to 5. Additionally, we stop the ADM loop if the progress of two consecutive iterations is smaller than a given threshold $\varepsilon_1 \geq 0$, i.e., if

$$\left\| z^{i,k,l+1} - z^{i,k,l} \right\|_\infty \leq \varepsilon_1 \quad \text{for } i = 1, \dots, \ell.$$

In our case, we choose $\varepsilon_1 := 0.01$ (which is either bar or $\mathrm{kg\,s^{-1}}$). We consider a solution to be feasible if all copies z^i of the linking variables z are close enough to z, i.e., if

$$\|z^i - z\|_\infty \leq \varepsilon_2 \quad \text{for } i = 1, \dots, \ell \tag{22}$$

holds with $\varepsilon_2 := 0.1$ (again in bar or $\mathrm{kg\,s^{-1}}$). For each considered gas network we will compare multiple decompositions that split the network at different places and into differently many blocks. It follows that the relaxation (22) is not applied in the same way for each decomposition, which could lead to an unfair advantage to some of them. To counteract this, we introduce additional relaxations as follows: To each non-linking variable x of a given decomposition that is a linking variable in another decomposition we add a slack variable $s \in [-\varepsilon_2, \varepsilon_2]$. This gives x the same tolerance as the corresponding linking variable in (22), which makes the results comparable again.

We initialize all penalty parameters μ with $\Delta t / T$. In this way, a different time discretization does not change the overall weight of the penalty term in the objective function. For a similar reason, we also divide the cost objective (12) by T. For the tracking-type objective (13), we use the factors $\eta = 1\,\mathrm{bar}^{-2}$ and $\theta = 1\,\mathrm{s^2\,kg^{-2}}$. The penalty scaling factor is set to $\omega_\mu = 2$. If a penalty parameter μ reaches a value of 10^9, all penalty parameters get re-scaled by a factor of 10^{-6}; see, e.g., [20], where a similar re-scaling was done with the help of sigmoidal functions. The problem data, Python code, and result data used in this section can be found in [15].

5.1. Transient Gas Network Data.

The gas network instances for our numerical experiments are taken from the GasLib [21]. It contains network data as well as stationary scenarios for these networks. We use this information to create transient scenarios on a time horizon of $T = 24\,\mathrm{h}$. For each network, we fix the mass flow q_u of each exit node $u \in V_-$ and the

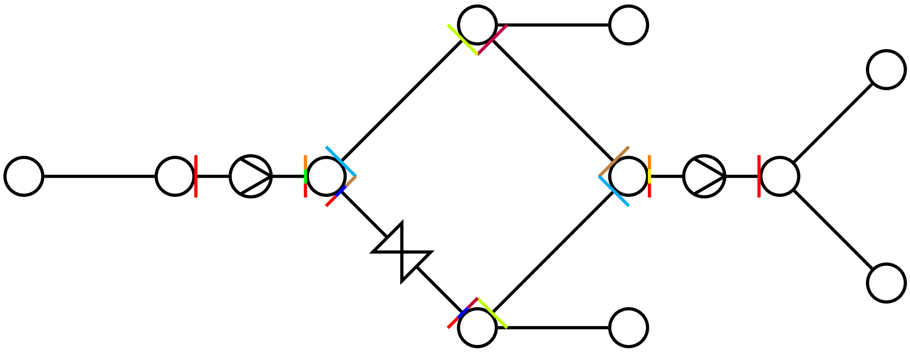

FIGURE 2. Decompositions of GasLib-11: Active (red), Valve (blue), Left Arm (green), Right Arm (yellow), Both Arms (orange), 1 o'clock (purple), 2 o'clock (brown), 4 o'clock (cyan), 5 o'clock (lime)

pressure p_u for each entry node $u \in V_+$. For the mass flow at an exit node at $t = 0$ we use the value given in the stationary scenario. Then, we apply a vertically shifted sine curve with period T and an amplitude of 0.1 to this initial value to receive the mass-flow curve for the complete time horizon:

$$q_u(t) = \left(1 + 0.1 \sin\left(t\frac{2\pi}{T}\right)\right) q_u(0), \quad u \in V_-, \, t \in [0, T].$$

The pressure at the entry nodes stays constant over the time horizon. For the first three test networks we consider, namely GasLib-11, GasLib-24, and GasLib-40, which come with only one stationary scenario, we choose these values manually. For the last instance, GasLib-134, which represents the gas network of Greece, however, there are 1234 different stationary scenarios given in the GasLib. Here, we first solve all stationary scenarios by using the discretized version of Problem (14) with only one time point $t = 0 = T$. To this end, we use the stationary Euler equations

$$\begin{aligned} \partial_x q_a &= 0, \\ \partial_x p_a &= -\frac{\lambda_a c^2}{2D_a A_a^2} \frac{q_a |q_a|}{p_a} - \frac{g h_a'}{c^2} p_a, \end{aligned}$$

where all time derivatives vanished, the linear compressor station model (6), and the objective function (12), which now reads

$$f_{\mathrm{c}} = \sum_{a \in A_{\mathrm{cs}}} \Delta p_a.$$

Then, we fix the entry pressures to the corresponding values of the obtained stationary solution. Since for two of the stationary problems, the solver reported infeasibility, this results in 1232 transient scenarios for the GasLib-134 network.

For the initial conditions of each scenario we used the solution of the corresponding stationary model. The same holds for the terminal condition that are needed for the tracking-type objective function (13). This stationary solution is also used to initialize all variables including the linking variables.

5.2. **GasLib-11.** The first test network we consider is the GasLib-11. It contains 11 nodes including 3 entries and 3 exits, 8 pipes, 2 compressor stations, and 1 valve. The total pipe length is 440 km. We fixed the entry pressures to 53 bar, 51 bar, and 52 bar, respectively.

GasLib-11 has a cycle containing the valve in the middle and one arm containing a compressor station on the left and right side, respectively. We apply several decompositions to this network, which we depict through colored cuts in the network graph in Figure 2. If there are multiple cuts at the same location, the corresponding line has multiple colors. Each decomposition gets its own name to be easily identifiable.

The first decomposition, which we apply to every network, is called "Active", which creates a block for each active element (compressor stations, valves, and control valves) and each connected sub-network of the other elements (pipes, short pipes, and nodes). The resulting blocks of active elements are all LPs (or QPs) and MIPs (or MIQPs), depending on the chosen objective function, and the remaining pipe networks are NLPs, which, in the case of GasLib-11, do not contain cycles. The next decomposition approach, called "Valve", separates the valve from the rest of the network. The two resulting blocks are an MIP and an NLP or MINLP, depending on the compressor station model and the chosen objective function. Both blocks do not contain a cycle. Then, there are the decompositions "Left Arm", "Right Arm", and "Both Arms" which separate either one or two arms containing a compressor station from the cycle in the middle of the network. The block containing the cycle is always an MINLP and the separated arm-blocks are NLPs or MINLPs, depending on the compressor station model. Next, we introduce four decompositions that create cycle-free blocks by cutting the cycle at two opposite locations. The position of the first cut in the cycle inspire the names "1 o'clock", "2 o'clock", "4 o'clock", and "5 o'clock". The block containing the valve is again an MINLP and the other block is again an NLP or MINLP, depending on the compressor station model. Finally, we tried applying no decomposition ("No decomp.") and, thus, solving the MINLP as a whole to have a baseline comparison for our method applied to the different graph decompositions. Here, the same relaxations as for the decompositions are used to ensure that the results are comparable w.r.t. computation time and objective value; see also the discussion of this aspect at the beginning of this section.

Now, we apply these decompositions to all combinations of compressor station (CS) models and objective functions. The numerical results are given in Table 1. It contains the computation time (Time (s)), the objective value without the penalty term (Objective), the largest final constraint violation due to the linking variables in the pressure $\max_i |p^i_{a,u} - p_{a,u}|$ (called Violation p) and the mass flow $\max_i |q^i_{a,u} - q_{a,u}|$ (called Violation q), the number of penalty iterations (#outer), and the total number of ADM-loops (#inner). The fastest computation time and the best objective value for each part of the table are printed in bold font. If for a decomposition no feasible solution was found within the time limit of 1000 s, the corresponding row does not appear in the table.

To better understand the given objective values we recall their meaning. The "CS cost" objective value is the mean value over time of the pressure increases Δp in all compressor stations in bar. For the tracking-type objective we use the squared distance of the pressure and mass flow at entry and exit nodes to given target values. Here, we use the factors $\eta = 1\,\mathrm{bar}^{-2}$ and $\theta = 1\,\mathrm{s}^2\,\mathrm{kg}^{-2}$, respectively, for scaling the different terms.

For the LP compressor station model, a solution was found for every decomposition. For the compressor station cost objective, the "Valve" decomposition was by far the fastest with 8.42 s, followed by the "Active" decomposition with 45.78 s, and "No decomp." with 54.93 s. It even has a better objective value compared to "No decomp."; see again the discussion of KNITRO being a local MINLP solver at the beginning of this section. The best objective value is achieved by the "2 o'clock" decomposition, though at the price of a much longer computation time. For the tracking-type objective, most computations terminate in less than 5 s while "1 o'clock" is the fastest with only 1.47 s. The best objective value belongs to

Table 1. Numerical results for GasLib-11

Decomposition	Time (s)	Objective	Violation p	Violation q	#outer	#inner
LP CS model; CS cost objective						
No decomp.	54.93	0.48312	–	–	–	–
Active	45.78	1.01738	0.09695	0.08587	6	28
Valve	**8.42**	0.47833	0.07905	0.05382	1	4
Left Arm	571.24	0.59130	0.09372	0.06290	3	14
Right Arm	944.77	0.62175	0.08152	0.06134	4	16
Both Arms	464.27	0.74183	0.09675	0.05359	4	17
1 o'clock	64.98	0.49287	0.07973	0.04720	3	11
2 o'clock	127.68	**0.45954**	0.08531	0.08029	3	13
4 o'clock	343.93	0.63797	0.09706	0.07133	4	17
5 o'clock	333.57	0.56293	0.06911	0.06172	4	17
LP CS model; tracking objective						
No decomp.	4.15	0.00000	–	–	–	–
Active	32.92	0.10707	0.03590	0.08650	5	22
Valve	2.23	0.00000	0.00012	0.00004	1	1
Left Arm	3.97	**0.00000**	0.00137	0.00137	1	1
Right Arm	3.65	0.00002	0.00035	0.00110	1	1
Both Arms	118.26	0.00109	0.07458	0.08709	1	3
1 o'clock	**1.47**	0.00000	0.01910	0.00113	1	1
2 o'clock	20.35	0.00007	0.06902	0.02955	1	3
4 o'clock	3.28	0.00000	0.06841	0.03208	1	2
5 o'clock	3.02	0.00000	0.05693	0.01174	1	2
MIP CS model; CS cost objective						
No decomp.	**289.26**	0.80878	–	–	–	–
Active	301.96	4.42851	0.09718	0.09317	40	195
Valve	timeout	**0.79566**	0.05970	0.07864	1	5
1 o'clock	326.31	0.82131	0.08855	0.09632	7	34
MIP CS model; tracking objective						
No decomp.	5.37	0.00000	–	–	–	–
Active	45.24	0.24734	0.09959	0.05930	6	28
Valve	4.38	**0.00000**	0.00016	0.00003	1	1
Left Arm	5.03	0.00000	0.00006	0.00019	1	1
Right Arm	292.04	0.00076	0.07938	0.06592	1	1
Both Arms	182.14	0.00111	0.08576	0.08568	1	4
1 o'clock	49.28	0.00095	0.04168	0.04569	2	6
2 o'clock	87.79	0.00364	0.01343	0.08440	2	6
4 o'clock	5.60	0.00000	0.06059	0.02642	1	2
5 o'clock	**3.27**	0.00000	0.07170	0.01790	1	1

the “Left Arm” decomposition but most values are very close to 0 so that the differences are less than the displayed accuracy of 10^{-5}.

Now, we take a closer look at the MIP compressor station model. Here, the combination with the compressor station cost objective seems to be the hardest problem as only four decompositions find a feasible point. The fastest solution comes from “No decomp.” with 289.26 s but “Active” with 301.96 s and “1 o’clock” with 326.31 s are at least comparable. The “Valve” decomposition does not terminate within the time limit of 1000 s and is stopped after 5 ADM-loops.

For the tracking-type objective, all decompositions terminate within the time limit. Some of them terminate in a few seconds while “5 o’clock” is the fastest with 3.27 s. Again, most objective values are very close to 0.

For all cases we can see that the “Active” decomposition finds a solution with a comparably bad objective value. This could be due to the fact that the active elements that are used to control the network now only have a very small scope, i.e., their respective blocks are very small. The decomposition “Both Arms” is never competitive due to its slow solution times and bad objective values.

Next, we take a closer look at the development of linking variables over the iterations of Algorithm 1. To this end, we plot all copies and the consensus of the mass flow variables of the right compressor station in the beginning of for four different penalty iterations in Figure 3. Here, we used the “Active” decomposition with the LP compressor station model and the compressor station cost objective, which took 6 penalty iterations and overall 28 ADM-loop iterations to produce a feasible point. In the figure, one can clearly see the consensus as the weighted arithmetic mean of the three variables and how these variables change until they converge to the same curve.

5.3. **GasLib-24.** Next, we consider the GasLib-24 network. It contains 24 nodes including 3 entries and 5 exits, 19 pipes, 1 short pipe, 3 compressor stations, 1 control valve, and 1 resistor, which we replace with a short pipe. There are two cycles in the network. The total pipe length is 820.01 km. For this network, we fixed the pressure of the second entry to 49 bar.

We introduce the decompositions as depicted in Figure 4. We, again, use “No decomp.” as well as the decomposition “Active”. Besides this, the decomposition “Halves” splits the network in two parts of roughly the same size. In the same manner, the decomposition “Thirds” splits the network in three parts of roughly the same size. Finally, the decomposition “Tree” cuts both cycles in a way that the resulting three blocks are trees.

Table 2 contains the numerical results for the aforementioned decompositions. For the LP compressor station model, using “No decomp.” produces solutions very fast. In the case of the compressor station cost objective, the second-fastest option is “Tree” with 71.70 s. For the tracking-type objective, however, the decomposition “Halves” is able to beat the “No decomp.” time of 3.98 s with 2.82 s and basically the same objective value. It seems that the initial values of the variables at the interface were already so close to the optimal solution that it took only one iteration.

For the MIP compressor station model with compressor station cost objective, all decompositions find a global optimal solution with an objective of 0. “No decomp.” takes the shortest amount of time with 88.26 s followed by “Active” with 128.24 s. In the case of the MIP compressor station model combined with the tracking-type objective, the “Tree” decomposition scores the fastest time (21.69 s). However, the much slower “Halves” approach (653.80 s) produces a slightly better objective value.

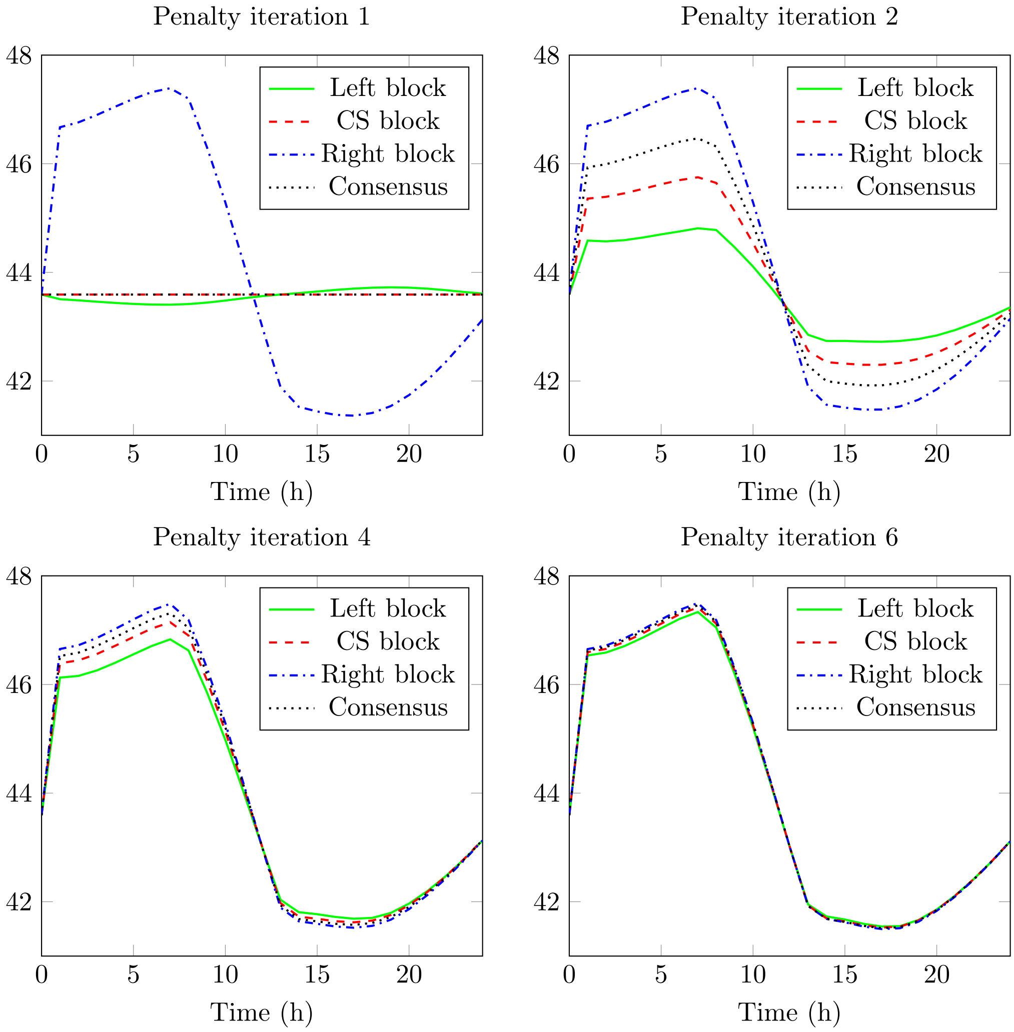

FIGURE 3. The linking variables representing the mass flow q_a (in $\mathrm{kg\,s^{-1}}$) in the left compressor station of GasLib-11 for the decomposition "Active" at the beginning of different penalty iterations

5.4. **GasLib-40.** The last of the three test networks of the GasLib library is GasLib-40. It contains 40 nodes including 3 entries and 29 exits, 39 pipes, and 6 compressor stations. Its total pipe length is 1112.47 km. The network contains multiple cycles. One key point is a node of degree 5, which connects the passive lower part of the network with the three arms that contain the entries and all compressor stations. While the left and upper arm have one compressor station each, the right arm has four of them. All three entry pressures are fixed to 67 bar.

Figure 5 shows our decompositions of GasLib-40. Next to "No decomp." and "Active", we introduce four additional decompositions. The first one, "All Arms", separates the three arms containing compressor stations from the passive lower part of the network. The second decomposition, "Lower", does something similar but cuts towards the passive lower part

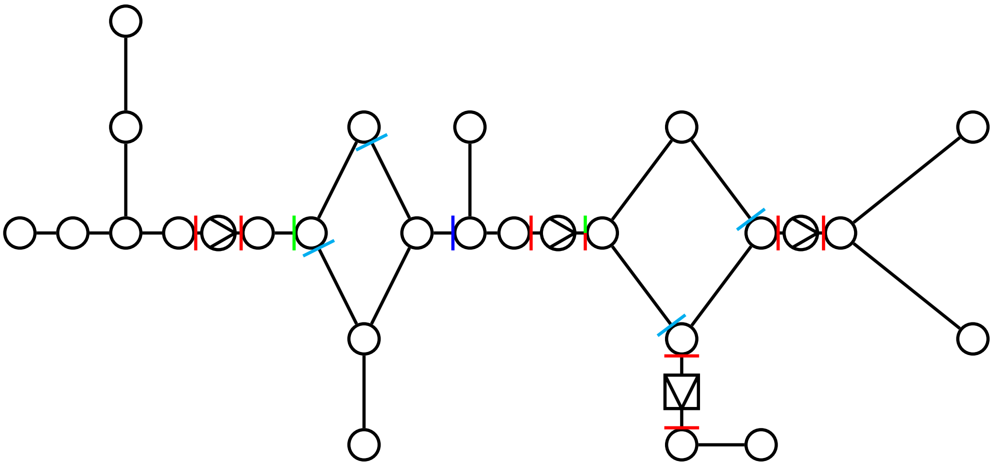

FIGURE 4. Decompositions of GasLib-24: Active (red), Halves (blue), Thirds (green), Tree (cyan)

TABLE 2. Numerical results for GasLib-24

Decomposition	Time (s)	Objective	Violation p	Violation q	#outer	#inner
LP CS model; CS cost objective						
No decomp.	**4.48**	0.00000	–	–	–	–
Active	127.36	0.13531	0.09777	0.08709	10	47
Halves	112.23	0.00000	0.09792	0.08936	6	28
Thirds	112.97	**0.00000**	0.09882	0.08582	7	34
Tree	71.70	0.07795	0.09862	0.08832	6	26
LP CS model; tracking objective						
No decomp.	3.98	**0.00000**	–	–	–	–
Active	39.06	0.00006	0.08032	0.07988	3	13
Halves	**2.82**	0.00000	0.02660	0.00041	1	1
Thirds	11.43	0.00006	0.06521	0.01035	1	4
Tree	7.61	0.00010	0.08357	0.03889	1	3
MIP CS model; CS cost objective						
No decomp.	**88.26**	**0.0**	–	–	–	–
Active	128.24	**0.0**	0.09955	0.06807	10	48
Halves	173.47	**0.0**	0.09550	0.08749	6	26
Thirds	295.76	**0.0**	0.09694	0.08652	7	32
Tree	259.25	**0.0**	0.09951	0.07350	5	25
MIP CS model; tracking objective						
No decomp.	284.93	0.00001	–	–	–	–
Active	66.12	0.01578	0.09921	0.06549	5	22
Halves	653.80	**0.00000**	0.01120	0.00006	1	1
Thirds	317.83	0.00516	0.09836	0.03645	3	14
Tree	**21.69**	0.00006	0.08277	0.03525	1	3

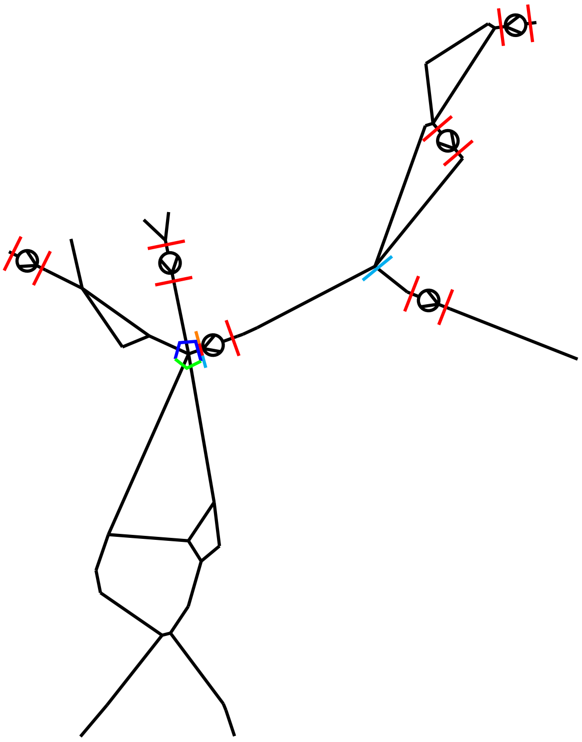

FIGURE 5. Decompositions of GasLib-40: Active (red), All Arms (blue), Lower (green), Right Arm (orange), Right Arm 2 (cyan)

instead of the three arms. The third one, "Right Arm", only separates the right arm from the rest of the network because it is the most complicated one containing the most compressor stations. The last decomposition, "Right Arm 2", follows the same idea but also adds an additional cut in the right arm to further simplify the resulting blocks.

Since GasLib-40 is already rather large, not all details can be displayed in Figure 5. Hence, we also show the block information of the different decompositions in Table 3. Here, the numbers of nodes, entries, exits, pipes, and compressor stations are given together with the model type of each block, which might still depend on the compressor station model in use.

We show the numerical results for GasLib-40 in Table 4. For the LP compressor station model together with the compressor station cost function, a feasible solution was found for only three settings. The fastest time of 126.81 s is achieved by applying "Right Arm 2". The three objective values are all really close but the best one belongs to "No decomp.". For the tracking-type objective, the clear winner w.r.t. computation time and objective value is "No decomp.".

In the case of the MIP compressor station model, only "No decomp." produces solutions. However, the time limit of 1000 s was reached in both cases and only a feasible point with no optimality guarantee is returned. For all four combinations for GasLib-40, the decomposition "Active" never resulted in a feasible solution within the time limit. This could be due to the small scope of the active elements and the large size of the remaining passive network. The decompositions "All Arms" and "Lower" also did not perform very well.

Table 3. Information on the decompositions of GasLib-40

Block Nr.	#nodes	#entries	#exits	#pipes	#cs	Type
			No decomp.			
1	40	3	29	39	6	MINLP
			Active			
1	3	1	2	2	0	NLP
2–3	1	1	0	0	0	LP
4	22	0	20	25	0	NLP
5	2	0	1	1	0	NLP
6	11	0	6	11	0	NLP
7–12	0	0	0	0	1	LP/MIP
			All Arms			
1	4	1	2	3	1	NLP/MINLP
2	6	1	4	6	1	NLP/MINLP
3	14	1	7	12	4	NLP/MINLP
4	16	0	16	18	0	NLP
			Lower			
1	15	0	15	18	0	NLP
2	25	3	14	21	6	NLP/MINLP
			Right Arm			
1	14	1	7	12	4	NLP/MINLP
2	26	2	22	27	2	NLP/MINLP
			Right Arm 2			
1	3	0	2	2	1	NLP/MINLP
2	26	2	22	27	2	NLP/MINLP
3	11	1	5	10	3	NLP/MINLP

5.5. **GasLib-134.** In our case studies for the test networks GasLib-11, GasLib-24, and GasLib-40, we have seen that applying Algorithm 1 to transient gas network problems can yield a significant benefit w.r.t. the solution times as well as the objective value. However, the results can vary strongly and are dependent on a number of factors including the compressor station model, the objective function, the network's structure, and the chosen decomposition. Therefore, the generalizability of these results is limited, especially since there was only a single transient scenario for each of the networks. Thus, we now apply our decomposition strategies to the real-world gas network of Greece GasLib-134. For this network, we have constructed 1232 different transient scenarios based on the stationary scenarios given in the GasLib. Each of these scenarios represents a day between the 1st of November 2011 and the 17th of February 2016.

GasLib-134 consists of 134 nodes including 3 entries and 45 exits, 86 pipes, 45 short pipes, 1 compressor station, and 1 control valve. The total pipe length is 1447.02 km. The network is a tree with the compressor station in the north and the control valve in the south east.

TABLE 4. Numerical results for GasLib-40

Decomposition	Time (s)	Objective	Violation p	Violation q	#outer	#inner
LP CS model; CS cost objective						
No decomp.	174.45	**5.43501**	–	–	–	–
Right Arm	**126.81**	5.69835	0.08847	0.04462	4	17
Right Arm 2	146.31	5.81410	0.09957	0.06121	4	20
LP CS model; tracking objective						
No decomp.	**184.93**	**8.44500**	–	–	–	–
All Arms	793.95	69.33519	0.07043	0.09671	24	119
Lower	501.64	13.53845	0.09875	0.08132	8	39
Right Arm	310.75	12.49074	0.08609	0.09528	7	33
Right Arm 2	239.61	15.59417	0.08633	0.06746	8	36
MIP CS model; CS cost objective						
No decomp.	**timeout**	**11.15985**	–	–	–	–
MIP CS model; tracking objective						
No decomp.	**timeout**	**3874.66251**	–	–	–	–

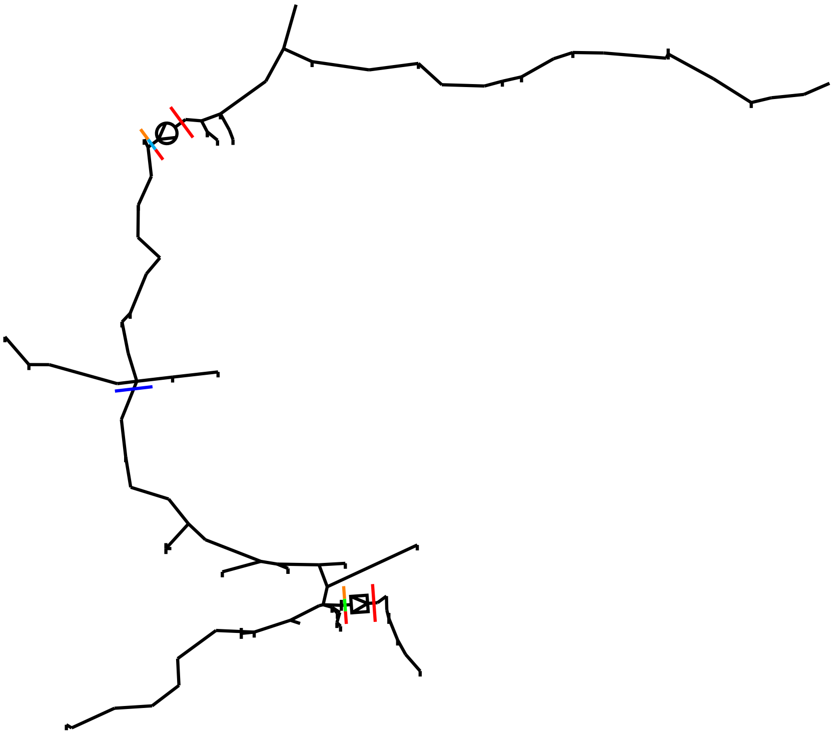

FIGURE 6. Decompositions of GasLib-134: Active (red), Halves (blue), CS Arm (cyan), CV Arm (green), Both Arms (orange)

Table 5. Decompositions of GasLib-134

Block Nr.	#nodes	#entries	#exits	#pipes	#sp	#cs	#cv	Type
				No decomp.				
1	134	3	45	86	45	1	1	NLP/MINLP
				Active				
1	42	2	13	28	13	0	0	NLP
2	81	1	28	52	28	0	0	NLP
3	11	0	4	6	4	0	0	NLP
4	0	0	0	0	0	1	0	LP/MIP
5	0	0	0	0	0	0	1	LP/MIP
				Halves				
1	65	1	23	41	23	0	1	NLP/MINLP
2	69	2	22	45	22	1	0	NLP/MINLP
				CV Arm				
1	11	0	4	6	4	0	1	NLP/MINLP
2	123	3	41	80	41	1	0	NLP/MINLP
				CS Arm				
1	42	2	13	28	13	1	0	NLP/MINLP
2	92	1	32	58	32	0	1	NLP/MINLP
				Both Arms				
1	42	2	13	28	13	1	0	NLP/MINLP
2	11	0	4	6	4	0	1	NLP/MINLP
3	81	1	28	52	28	0	0	NLP

Figure 6 shows GasLib-134 and the decompositions that we use. We again use the already known “No decomp.” and the “Active” decomposition. Additionally, we introduce the “Halves” decomposition which splits the network in two blocks of roughly the same size. Then, there are “CS Arm” and “CV Arm”, which split the network directly after the compressor station in the north or directly before the control valve in the south. The decomposition “Both Arms” combines the two aforementioned ideas. Since GasLib-134 is a large network, all block details of the proposed decompositions are shown in Table 5. This includes the number of nodes, entries, exits, pipes, short pipes, compressor stations, and control valves as well as the model type of each block.

Figure 7 shows the percentage of instances that can be solved in which amount of time for each decomposition, compressor station model, and objective. For the cases with the LP compressor station model, applying no decomposition is dominating most of the time. Consequently, there is not much gain by using a decomposition. However, the picture is completely different for the MIP compressor station model. Here, the decompositions “Active”, “Halves”, “CV Arm”, and “Both Arms” outperform “No decomp.” for most of the scenarios. For both objective models, “CV Arm” and “Both Arms” perform especially well while the overall worst results are produced by “CS Arm” and “No decomp.”. For the tracking-type objective, also “Halves” can lead to some benefit for some scenarios‘.

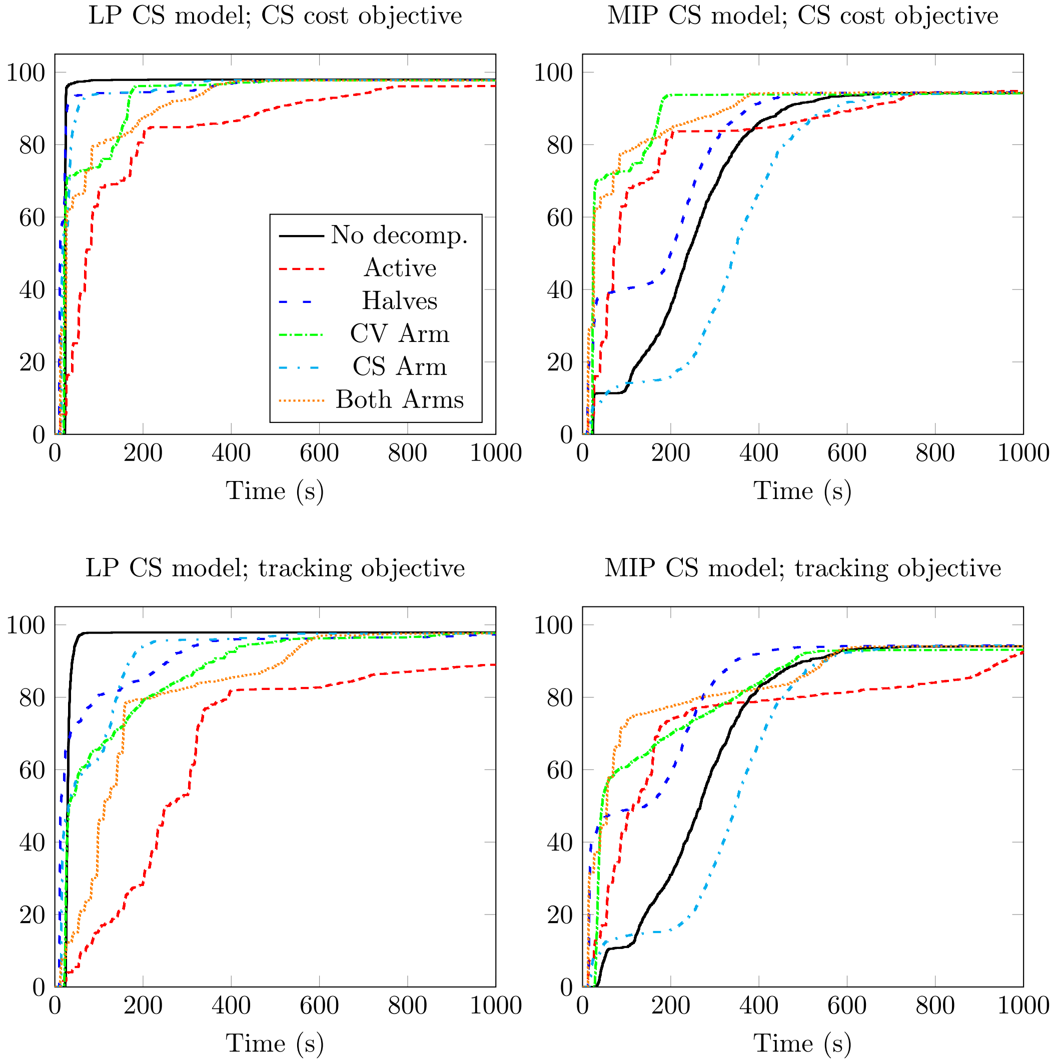

FIGURE 7. Percentage of solved instances of GasLib-134 in the time limit of 1000 s for different decompositions, the LP (6) and MIP (7) compressor station (CS) models, and compressor station cost (12) and tracking-type (13) objective functions.

For most of the scenarios, “No decomp.”, “Halves”, and “CS Arm” only produce mediocre results even though they were the best performers in the LP compressor station model cases.

Noteworthy is also that for the CS cost objective, the “Active” decomposition solves the least amount of scenarios within the time limit for the LP compressor station model but the most scenarios for the MIP compressor station model. The corresponding curves in Figure 7 stay almost identical when the compressor station model is changed. For the tracking objective, the computation times even decrease when the MIP compressor station model is used. While the MIP model has a negative impact on the other decompositions, this does not seem to be the case for "Active".

Overall, the two decompositions "CV Arm" and "Both Arms" are the clear winners for the MIP compressor station model. Separating the mixed-integer parts of the control valve from the large middle part of GasLib-134 seems to generate blocks that are not too hard to solve while simultaneously do not require too much time for getting feasible couplings at the blocks' interfaces.

6. Conclusion

We considered the dynamic optimization of gas transport networks. After a full discretization in space and time, the resulting model is a nonconvex MINLP that needs to be solved on large-scale transport networks. This puts a challenging burden even on today's most sophisticated MINLP solvers. Hence, we proposed a graph decomposition and a tailored consensus-based PADM to compute feasible controls of good quality as fast as possible.

The interpretation of the numerical results is somewhat tricky. On the one hand, we clearly show that using graph decompositions together with a tailored solution method can outperform the baseline method, which just solves the large MINLP from scratch using a state-of-the-art MINLP solver. However, whether this is the case seems to depend on very many aspects such as the chosen objective function, the chosen decomposition, or the chosen model for compressor stations (continuous vs. mixed-integer). This directly leads to the main open research question (for which we think that answering it is out of the scope of this paper): Given a large-scale MINLP defined on a graph, what is the best way to decompose the network so that a tailored decomposition method outperforms the baseline method? Although this paper provides a first step to understanding the relevant effects, we leave a complete answer to this question for future work. An interesting further question in this context is to analyze if multiple decompositions can be started in parallel so that simply the fastest method can be used—either as a standalone approach or for warm-starting the overall MINLP.

Acknowledgments

The authors thank the Deutsche Forschungsgemeinschaft for their support within projects A05 and B08 in the Sonderforschungsbereich/Transregio 154 "Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks".

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122. DOI: 10.1561/2200000016.

[2] R. Burlacu, H. Egger, M. Groß, A. Martin, M. E. Pfetsch, L. Schewe, M. Sirvent, and M. Skutella. "Maximizing the storage capacity of gas networks: a global MINLP approach." In: *Optimization and Engineering* 20.2 (2019), pp. 543–573. DOI: 10.1007/s11081-018-9414-5.

[3] M. R. Bussieck and A. Meeraus. "General algebraic modeling system (GAMS)." In: *Modeling languages in mathematical optimization*. Springer, 2004, pp. 137–157. DOI: 10.1007/978-1-4613-0215-5_8.

[4] R. H. Byrd, J. Nocedal, and R. A. Waltz. "KNITRO: An integrated package for nonlinear optimization." In: *Large Scale Nonlinear Optimization, 35–59, 2006*. Springer Verlag, 2006, pp. 35–59. DOI: 10.1007/0-387-30065-1_4.

[5] CPLEX, IBM ILOG. *V12. 1: User's Manual for CPLEX*. 53. 2009, p. 157.

[6] A. Fügenschuh, B. Geißler, R. Gollmer, A. Morsi, M. E. Pfetsch, J. Rövekamp, M. Schmidt, K. Spreckelsen, and M. C. Steinbach. "Physical and technical fundamentals of gas networks." In: *Evaluating Gas Network Capacities*. Ed. by T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. SIAM-MOS series on Optimization. SIAM, 2015. Chap. 2, pp. 17–43. DOI: 10.1137/1.9781611973693.ch2.

[7] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Penalty Alternating Direction Methods for Mixed-Integer Optimization: A New View on Feasibility Pumps." In: *SIAM Journal on Optimization* 27.3 (2017), pp. 1611–1636. DOI: 10.1137/16M1069687.

[8] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Solving Highly Detailed Gas Transport MINLPs: Block Separability and Penalty Alternating Direction Methods." In: *INFORMS Journal on Computing* 30.2 (2018), pp. 309–323. DOI: 10.1287/ijoc.2017.0780.

[9] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Solving power-constrained gas transportation problems using an MIP-based alternating direction method." In: *Computers & Chemical Engineering* 82 (2015), pp. 303–317. DOI: 10.1016/j.compchemeng.2015.07.005.

[10] M. Hahn, S. Leyffer, and V. M. Zavala. *Mixed-integer PDE-constrained optimal control of gas networks*. Tech. rep. 2017. URL: https://www.mcs.anl.gov/papers/P7095-0817.pdf.

[11] F. M. Hante, G. Leugering, A. Martin, L. Schewe, and M. Schmidt. "Challenges in Optimal Control Problems for Gas and Fluid Flow in Networks of Pipes and Canals: From Modeling to Industrial Applications." In: *Industrial Mathematics and Complex Systems: Emerging Mathematical Models, Methods and Algorithms*. Ed. by P. Manchanda, R. Lozi, and A. H. Siddiqi. Industrial and Applied Mathematics. Singapore: Springer Singapore, 2017, pp. 77–122. DOI: 10.1007/978-981-10-3758-0_5.

[12] J. Jalving, S. Shin, and V. M. Zavala. "A graph-based modeling abstraction for optimization: concepts and implementation in Plasmo.jl." In: *Mathematical Programming Computation* (2022). DOI: 10.1007/s12532-022-00223-3.

[13] R. Krug, G. Leugering, A. Martin, M. Schmidt, and D. Weninger. "Time-Domain Decomposition for Optimal Control Problems Governed by Semilinear Hyperbolic Systems." In: *SIAM Journal on Control and Optimization* 59.6 (2021), pp. 4339–4372. DOI: 10.1137/20M138329X.

[14] R. Krug, G. Leugering, A. Martin, M. Schmidt, and D. Weninger. "Time-Domain Decomposition for Optimal Control Problems Governed by Semilinear Hyperbolic Systems with Mixed Two-Point Boundary Conditions." In: *Control and Cybernetics* 50.4 (2021), pp. 427–455. DOI: 10.2478/candc-2021-0026.

[15] R. Krug and D. Weninger. *gas-network-opti-padm*. URL: https://github.com/RichardKrug/gas-network-opti-padm.

[16] G. Leugering, A. Martin, M. Schmidt, and M. Sirvent. "Nonoverlapping Domain Decomposition for Optimal Control Problems governed by Semilinear Models for Gas Flow in Networks." In: *Control and Cybernetics* 46.3 (2017), pp. 191–225.

[17] S. Na, S. Shin, M. Anitescu, and V. M. Zavala. "On the Convergence of Overlapping Schwarz Decomposition for Nonlinear Optimal Control." In: *IEEE Transactions on Automatic Control* (2022), pp. 1–16. DOI: 10.1109/TAC.2022.3194087.

[18] M. E. Pfetsch, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, A. Morsi, J. Rövekamp, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. C. Steinbach, S. Vigerske, and B. M. Willert. "Validation of nominations in gas network optimization: models, methods, and solutions." In: *Optimization Methods and Software* 30.1 (2015), pp. 15–53. DOI: 10.1080/10556788.2014.888426.

[19] D. Rose, M. Schmidt, M. C. Steinbach, and B. M. Willert. "Computational optimization of gas compressor stations: MINLP models versus continuous reformulations." In: *Mathematical Methods of Operations Research* 83.3 (2016), pp. 409–444. DOI: 10.1007/s00186-016-0533-5.

[20] L. Schewe, M. Schmidt, and D. Weninger. "A decomposition heuristic for mixed-integer supply chain problems." In: *Operations Research Letters* 48.3 (2020), pp. 225–232. DOI: 10.1016/j.orl.2020.02.006.

[21] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent. "GasLib—A Library of Gas Network Instances." In: *Data* 2.4 (2017). DOI: 10.3390/data2040040.

[22] M. Schmidt, M. C. Steinbach, and B. M. Willert. "High detail stationary optimization models for gas networks." In: *Optimization and Engineering* 16.1 (2015), pp. 131–164. DOI: 10.1007/s11081-014-9246-x.

[23] S. Shin, M. Anitescu, and V. M. Zavala. "Exponential Decay of Sensitivity in Graph-Structured Nonlinear Programs." In: *SIAM Journal on Optimization* 32.2 (2022), pp. 1156–1183. DOI: 10.1137/21M1391079.

[24] S. Shin, P. Hart, T. Jahns, and V. M. Zavala. "A Hierarchical Optimization Architecture for Large-Scale Power Networks." In: *IEEE Transactions on Control of Network Systems* 6.3 (2019), pp. 1004–1014. DOI: 10.1109/TCNS.2019.2906917.

[25] S. Shin, V. M. Zavala, and M. Anitescu. "Decentralized Schemes With Overlap for Solving Graph-Structured Optimization Problems." In: *IEEE Transactions on Control of Network Systems* 7.3 (2020), pp. 1225–1236. DOI: 10.1109/TCNS.2020.2967805.

[26] V. M. Zavala. "Stochastic optimal control model for natural gas networks." In: *Computers & Chemical Engineering* 64 (2014), pp. 103–113. DOI: 10.1016/j.compchemeng.2014.02.002.

(R. Krug, G. Leugering, A. Martin, D. Weninger) [1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Department of Data Science, Cauerstr. 11, 91058 Erlangen, Germany
Email address: richard.krug@fau.de
Email address: guenter.leugering@fau.de
Email address: alexander.martin@fau.de
Email address: dieter.weninger@fau.de

(M. Schmidt) [2]Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany
Email address: martin.schmidt@uni-trier.de