Optim Eng (2016) 17:437–472
DOI 10.1007/s11081-015-9300-3

High detail stationary optimization models for gas networks: validation and results

Martin Schmidt[1,2] · Marc C. Steinbach[3] · Bernhard M. Willert[4]

Received: 9 December 2014 / Revised: 29 July 2015 / Accepted: 25 November 2015 /
Published online: 29 December 2015
© Springer Science+Business Media New York 2015

Abstract Due to strict regulatory rules in combination with complex nonlinear physics, major gas network operators in Germany and Europe face hard planning problems that call for optimization. In part 1 of this paper we have developed a suitable model hierarchy for that purpose. Here we consider the more practical aspects of modeling. We validate individual model components against a trusted simulation tool, give a structural overview of the model hierarchy, and use its large variety of approximations to devise robust and efficient solution techniques. An extensive computational study demonstrates the suitability of our models and techniques for previously unsolvable problems in gas network planning.

Keywords Gas networks · Stationary operation · High-detail modeling · Model validation · Sequential NLP solving

Mathematics Subject Classification 90B10 · 90C06 · 90C30 · 90C90

✉ Martin Schmidt
mar.schmidt@fau.de

Marc C. Steinbach
mcs@ifam.uni-hannover.de

1 Department Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

2 Energie Campus Nürnberg, Fürther Str. 250, 90429 Nuremberg, Germany

3 Institute for Applied Mathematics, Leibniz Universiät Hannover, Welfengarten 1, 30167 Hannover, Germany

4 Ronnenberg, Germany

1 Introduction

In part one of this paper (Schmidt et al. 2014) we have presented detailed models for stationary optimization in gas transport networks, consisting of physical components that are mostly associated with network wide gas dynamics, and of technical components that are usually associated with specific types of network elements. We have also developed smoothing techniques for nonsmooth components in order to apply optimization algorithms that use first and second order derivatives, and we have presented simplifications and approximations of highly complex components to reduce the computational effort when a lower accuracy is acceptable. This leads to a complicated variety of possible combinations of the component models. In this paper we will first provide a structural overview of the possible combinations in Sect. 2. The components will then be checked for correctness and accuracy in Sect. 3 by comparing computational results with the results of a commercial gas network simulation software. In Sect. 4 we consider general optimization techniques for computing practically useful and reliable results efficiently. This includes penalty relaxations to obtain additional information when no feasible solutions can be found, and using the variety of simplified and approximated models to construct sequences of warm-started NLPs for tackling highly detailed and nonsmooth problems that cannot be solved without these techniques. Results will be presented in Sect. 5 for a set of hard problem instances on a real-world network, and for a large set of publicly available instances on an artificial but realistic network. In combination, the model validation and computational results demonstrate that our NLP models are indeed applicable and useful in practice. An extensive literature review has already been given in part one (cf. Schmidt et al. 2014); a broader review can be found in Koch et al. (2015).

In the following subsection we introduce a fundamental application problem that will be referred to throughout the paper.

1.1 Validation of nominations (NoVa)

The *validation of nominations* is one of the key problems in the day-to-day work of gas network planners. In a nutshell, the NoVa problem is this:

> Given a transport network and a nomination of all entry and exit load flows, determine whether there is a technically and physically feasible operation of the network that satisfies this nomination.

The details are as follows. Consider a set of booking contracts with entry and exit customers. The booked capacities define maximal load flows that the customers may *nominate* for actual transport. Now, a *nomination* is defined as a complete set of balanced entry and exit load flows together with specific restrictions on gas pressures and with prescribed values for all quality parameters of the supplied gas, like calorific value, etc. The problem of validating such a nomination is to decide whether it is technically and physically feasible, i.e., whether all load flows can be realized. Throughout the paper we only consider the continuous feasibility problem

that is obtained after fixing all discrete decisions. For more information on the NoVa problem and on our overall solution approach see, e.g., Koch et al. (2015), Pfetsch et al. (2015).

2 NLP model variants

In part one of this paper (cf. Schmidt et al. 2014), we have presented component models for the required element types of gas networks and for basic physical phenomena that are relevant to the entire network. The component models often consist of several sub-models describing different physical phenomena or technical processes that we refer to as model *aspects*, and many of these aspects admit a choice among several modeling variants that we refer to as model *concretizations*. This leads to a large variety of possible NLP models, each of which is determined by choosing a complete set of concretizations for all model aspects.

In this section we provide a structural description of the sets of component models, aspects and concretizations, to obtain an overview and better understanding of the complete set of NLP models and their interrelations. This is complicated by the fact that the selection of concretizations for certain *global* aspects determines the sets of choices of other aspects, whereas the remaining selections are mostly independent of each other, even for different network elements of the same type. To capture these interdependencies formally, we will arrange aspects and concretizations in a directed (meta-)graph so that every possible NLP model will ultimately correspond to a forest in the meta-graph: a set of trees that satisfy certain properties. These properties will now be determined.

To this end, let us first give an overview of the component models and some illustrative examples of model aspects and concretizations. The basic physical phenomena include: gas compressibility, the equation of state for real gas, the heat capacity of gas, the interdependence of changes in gas pressure and temperature (Joule–Thomson effect), and mixing of different gas compositions. The required element types include: nodes (entries, exits, junctions), passive arcs (pipes, resistors, short cuts), and active arcs (compressor groups, control valve stations, valves). Typical model aspects are the pressure drop along a pipe, the fuel consumption of a compressor, or the equation of state. For instance, the aspect "equation of state" has two concretizations, "thermodynamical standard equation" and "Redlich–Kwong". The former depends on another model aspect, "compressibility", which in turn has two concretizations, "AGA" and "Papay"; see Fig. 1. Here we consider only the concretizations that we use in our NLP models, with details given in part one; further variants can be added as needed.

There are two *global* aspects whose concretizations determine sets of choices of other aspects: gas temperature and gas composition, the latter being represented by seven *gas quality parameters*. In a coarse model we may fix the values of gas temperature and gas quality parameters globally, but if we decide to take into account changes at *some* network elements, then we must compute changes throughout the network. This is the case since both temperature and composition are globally coupled: fixing certain values at some network elements and computing changes at other

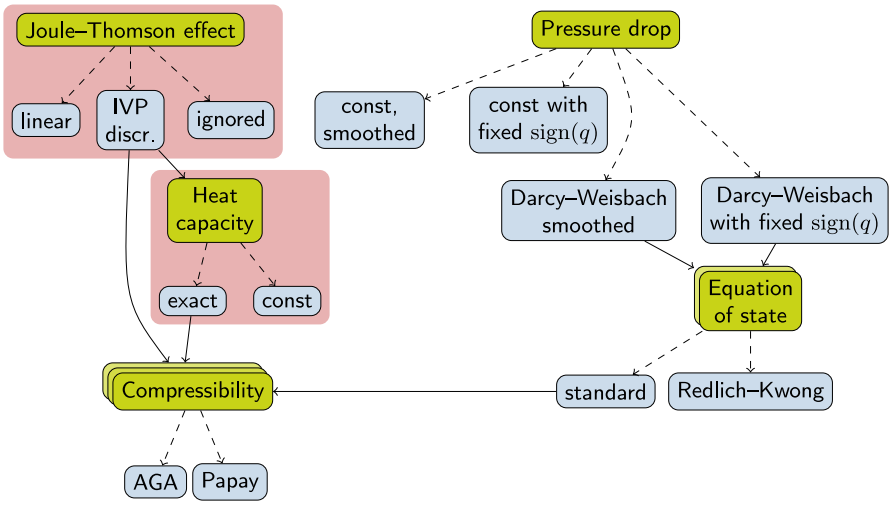

Fig. 1 Model aspect graph of resistors. The nodes in shaded blocks exist only in non-isothermal models

elements would inevitably lead to model inconsistencies. Taking the changes into account is known as *temperature tracking* and *gas quality tracking*, respectively. Practitioners tend to use it only when necessary as it complicates the models and computations considerably. Note finally that the seven gas quality parameters (molar mass m, calorific value H_c, pseudocritical pressure and temperature p_c, T_c, and the coefficients of isobaric molar heat capacity A, B, C) consist of four independent subsets that can be selected for tracking individually, namely m, H_c, (p_c, T_c), and (A, B, C). Tracking has to be enabled if at least one subset is selected, but only the selected subsets appear in the required mixing equations. The two global aspects thus have 2^5 concretizations arising as combinations of "fixed" or "tracking" for the temperature and the four parameter groups.

Next, we start describing the above-mentioned meta-graph to formalize relations between model aspects and concretizations. As before, let the directed graph $G = (\mathbb{V}, \mathbb{A})$ model the gas network with node set $\mathbb{V}$ and arc set $\mathbb{A}$. We denote model aspect nodes of the meta-graph by $\alpha \in \mathcal{A}$ and concretization nodes by $\gamma \in \mathcal{C}$; their union $\mathcal{A} \cup \mathcal{C}$ defines the node set of the meta-graph. An arc $a = \alpha\gamma$ from α to γ exists iff γ is a concretization of the model aspect α. These arcs in the set $\mathcal{A} \times \mathcal{C}$ are called "can-be-modeled-by"-arcs. For instance, there is an arc from the model aspect node $\alpha =$ "equation of state" to the concretization node $\gamma =$ "Redlich–Kwong". In addition, the meta-graph also contains arcs from the set $\mathcal{C} \times \mathcal{A}$. These occur for concretizations that depend on other model aspects. For instance, the concretization "thermodynamical standard equation" depends on the sub-aspect "compressibility" (with concretizations "AGA" and "Papay"). We refer to these arcs in the set $\mathcal{C} \times \mathcal{A}$ as "contains-aspect"-arcs. In contrast, there are no arcs in $\mathcal{C} \times \mathcal{C}$ or in $\mathcal{A} \times \mathcal{A}$ since concretizations do not need to be concretized further and there is no need to consider aspects containing sub-aspects. (For instance, considering the specific change in adiabatic enthalpy H_{ad} as an aspect with sub-aspects "compressibility" and "isentropic

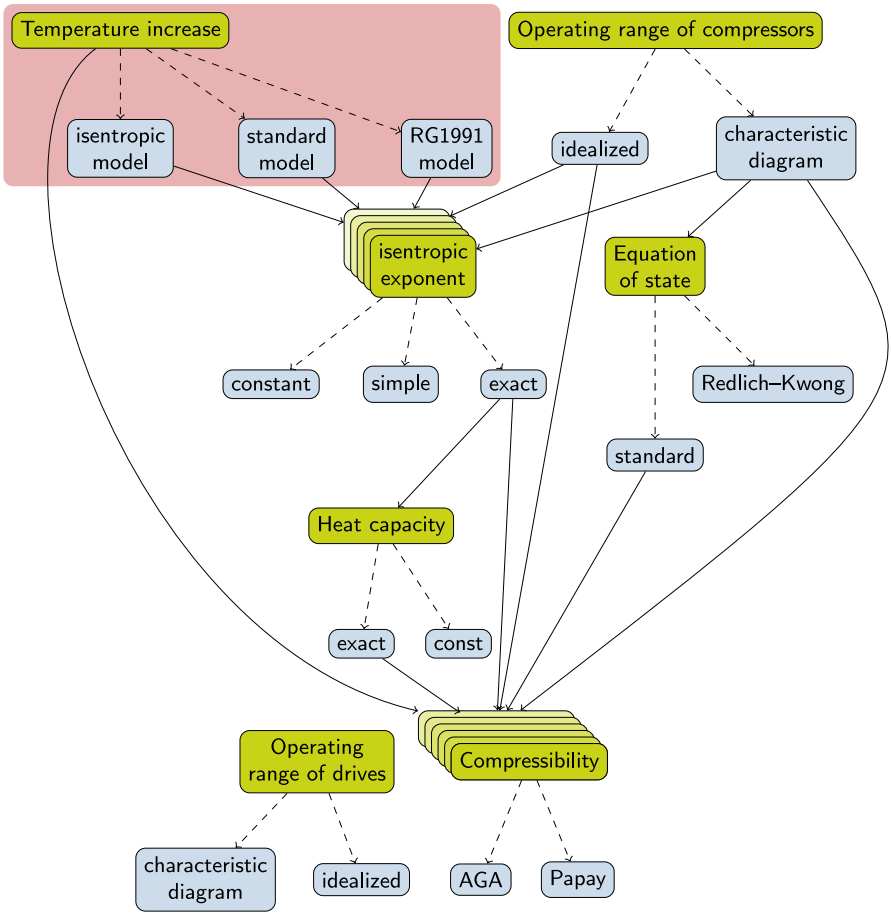

Fig. 2 Model aspect graph of compressors. The nodes in the shaded block exist only in non-isothermal models

exponent" would be physically reasonable but would unnecessarily complicate the meta-graph, cf. Fig. 2). Thus, every aspect has at least one concretization and every concretization belongs to some aspect, implying that aspect nodes always have nonzero out-degree and concretization nodes always have nonzero in-degree. We define *fundamental* model aspects as aspects with in-degree zero, and *terminal* concretizations as concretizations with out-degree zero. Those form the respective sets $\bar{\mathcal{A}} := \{\alpha \in \mathcal{A} : |\delta_\alpha^-| = 0\}$ and $\bar{\mathcal{C}} := \{\gamma \in \mathcal{C} : |\delta_\gamma^+| = 0\}$.

In contrast to the two global model aspects, gas temperature and gas composition, we call all other model aspects *local*. The selection of concretizations of local aspects does not influence the sets of choices of other local aspects (except for sub-aspects of the concretizations, of course), and the selections are independent for all individual network elements. For instance, we may choose to concretize the compressibility factor model with the AGA formula at pipes in regional (low-pressure) sub-networks but choose to model the compressibility factor with the

Papay formula on large (high-pressure) transport pipelines. Similar considerations may apply to the choice between "approximation" or "discretization" of the governing ODEs for pipes. We thus speak of *global* and *local* selections of concretizations.

To select a specific NLP model variant, we first have to fix the global selections and subsequently the local selections for all nodes and arcs of the network graph. Let us denote the global selections by $\sigma_g \in \Sigma_g$ where Σ_g is the (finite) set of global choices. Then, for every network element $\ell \in \mathbb{V} \cup \mathbb{A}$, the global selections σ_g determine a *local model aspect graph* $\mathcal{M}_\ell(\sigma_g)$ describing all possible aspects, concretizations, and sub-aspects of ℓ; see, e.g., Figs. 1 and 2 for the local model aspect graphs of resistors and compressors. Fixing the local selections for ℓ means that, for every fundamental aspect node $\bar{\alpha}$ in $\mathcal{M}_\ell(\sigma_g)$, we select a tree $T_{\ell,\bar{\alpha}}$ rooted in $\bar{\alpha}$ such that every aspect node α in $T_{\ell,\bar{\alpha}}$ has out-degree one (select one concretization of α) whereas every concretization node γ in $T_{\ell,\bar{\alpha}}$ has the same out-degree as in $\mathcal{M}_\ell(\sigma_g)$ (cover every sub-aspect of γ). Obtaining trees here requires that identical aspects reached from several concretizations are separate nodes in $\mathcal{M}_\ell(\sigma_g)$, such as "compressibility" in Figs. 1 and 2. Thus $\mathcal{M}_\ell(\sigma_g)$ is actually a forest, and every tree $T_{\ell,\bar{\alpha}}$ is a subtree of the tree rooted in $\bar{\alpha}$. On the other hand, we always select identical concretizations for every instance of a "multiple" aspect, and we do not see any good reason to do otherwise. (Therefore we depict every such aspect as a single node with a visual indication of its multiplicity.) Now, the union $\bigcup_{\bar{\alpha}} T_{\ell,\bar{\alpha}}$ is a forest in $\mathcal{M}_\ell(\sigma_g)$ representing a specific local model selection. We denote those local selections by $\sigma_\ell \in \Sigma_\ell(\sigma_g)$ where $\Sigma_\ell(\sigma_g)$ is the set of choices (local forests) at element ℓ determined by σ_g. Given network elements ℓ_1, ℓ_2 of the same type, the local model aspect graphs $\mathcal{M}_{\ell_1}(\sigma_g)$, $\mathcal{M}_{\ell_2}(\sigma_g)$ are clearly identical whereas the local selections $\sigma_{\ell_1}, \sigma_{\ell_2}$ may differ. A formal definition of the complete set of possible selections (or NLP model variants) thus reads

$$\Sigma := \bigcup_{\sigma_g \in \Sigma_g} \left[\{\sigma_g\} \times \prod_{\ell \in \mathbb{V} \cup \mathbb{A}} \Sigma_\ell(\sigma_g) \right].$$

The elements of Σ will be denoted by $\sigma = (\sigma_g, \sigma_{\mathbb{V}\cup\mathbb{A}})$ with $\sigma_{\mathbb{V}\cup\mathbb{A}} = (\sigma_\ell)_{\ell \in \mathbb{V}\cup\mathbb{A}}$; each corresponds to a forest in the meta-graph constructed as a union of local forests. The respective subsets of global concretizations and of local concretizations at element $\ell \in \mathbb{V} \cup \mathbb{A}$ selected by σ_g and σ_ℓ will be denoted by $\mathcal{C}_g, \mathcal{C}_\ell$.

With a selection $\sigma \in \Sigma$ at hand, we can now formulate the corresponding NLP model variant of the complete network. To this end, let $\bar{\mathcal{E}}_g, \bar{\mathcal{I}}_g$ and $\bar{\mathcal{E}}_\ell, \bar{\mathcal{I}}_\ell$ denote complete index sets of equality and inequality constraints that may appear in concretizations of the global aspects and of the aspects of element $\ell \in \mathbb{V} \cup \mathbb{A}$, respectively. Likewise, let $\bar{\mathcal{V}}_g, \bar{\mathcal{V}}_\ell$ denote the respective index sets of variables that appear in these concretizations. Here we require that corresponding constraint index sets of different elements $\ell_1, \ell_2 \in \mathbb{V} \cup \mathbb{A}$ are mutually disjoint, even if ℓ_1, ℓ_2 are elements of the same type,

$$\bar{\mathcal{E}}_{\ell_1} \cap \bar{\mathcal{E}}_{\ell_2} = \emptyset, \quad \bar{\mathcal{I}}_{\ell_1} \cap \bar{\mathcal{I}}_{\ell_2} = \emptyset.$$

In addition, we require that every local variable index is associated with a unique network element ℓ. This is possible since variables of arc $\ell \in \mathbb{A}$ can only appear in the constraints of arc ℓ and of its head and tail nodes whereas variables of node $\ell \in \mathbb{V}$ can only appear in the constraints of node ℓ and of its incident arcs. Thus, we obtain the following complete sets of constraints and variables that may appear in the NLP model:

$$\bar{\mathcal{E}} = \bar{\mathcal{E}}_g \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \bar{\mathcal{E}}_\ell, \quad \bar{\mathcal{I}} = \bar{\mathcal{I}}_g \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \bar{\mathcal{I}}_\ell, \quad \bar{\mathcal{V}} = \bar{\mathcal{V}}_g \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \bar{\mathcal{V}}_\ell.$$

Now, every model selection σ determines certain global and local index subsets corresponding to the constraints and variables of the concretizations $\mathcal{C}_g, \mathcal{C}_\ell$,

$$\mathcal{E}_{\sigma_g} \subseteq \bar{\mathcal{E}}_g, \quad \mathcal{E}_{\sigma_\ell} \subseteq \bar{\mathcal{E}}_\ell, \qquad \mathcal{I}_{\sigma_g} \subseteq \bar{\mathcal{I}}_g, \quad \mathcal{I}_{\sigma_\ell} \subseteq \bar{\mathcal{I}}_\ell, \quad \mathcal{V}_{\sigma_g} \subseteq \bar{\mathcal{V}}_g, \quad \mathcal{V}_{\sigma_\ell} \subseteq \bar{\mathcal{V}}_\ell.$$

The unions of these sets yield the complete index subsets associated with $\sigma \in \Sigma$,

$$\mathcal{E}_\sigma = \mathcal{E}_{\sigma_g} \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \mathcal{E}_{\sigma_\ell}, \quad \mathcal{I}_\sigma = \mathcal{I}_{\sigma_g} \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \mathcal{I}_{\sigma_\ell}, \quad \mathcal{V}_\sigma = \mathcal{V}_{\sigma_g} \cup \bigcup_{\ell \in \mathbb{V} \cup \mathbb{A}} \mathcal{V}_{\sigma_\ell}.$$

The selected NLP variant finally reads

$$\min_{x_\sigma} \quad f(x_\sigma) \quad \text{s.t.} \quad c_{\mathcal{E}_\sigma}(x_\sigma) = 0, \quad c_{\mathcal{I}_\sigma}(x_\sigma) \geq 0, \tag{1}$$

where

$$x_\sigma := (x_i)_{i \in \mathcal{V}_\sigma}, \quad c_{\mathcal{E}_\sigma} := (c_i)_{i \in \mathcal{E}_\sigma}, \quad c_{\mathcal{I}_\sigma} := (c_i)_{i \in \mathcal{I}_\sigma}.$$

The objective function f, although not part of the model variant, is restricted by its set of variables: it can be any smooth function that depends only on x_σ.

We remark that some of the NLP model variants contain additional continuous choices that we will not formalize here. Those appear in concretizations that involve smoothing techniques or discretizations of differential equations, with smoothing parameters or grid parameters that vary in continuous sets. Examples of such concretizations include "Darcy–Weisbach smoothed" for resistors (see Fig. 1), or the concretization "discretization" of the model aspect "Momentum equation" for pipes (see Fig. 21 in Appendix 2).

Figures 1 and 2 and 17, 18, 19, 20, and 21 in Appendix 2 show the local model aspect graphs for resistors, compressors, nodes, control valves, and pipes. As already mentioned, the given sets of concretizations correspond to the model variants developed in part one of this paper (cf. Schmidt et al. 2014); further variants are conceivable and may be added as needed.

Let us finally remark that the occurrence of model aspects with several concretizations is not restricted to the case of gas network optimization. Our experience shows that other problems from engineering and physics share similar properties. The framework formalized in this section can thus be used in other fields of applied optimization as well.

3 Model validation

For a proper validation of the model components developed in the first part of this paper (cf. Schmidt et al. 2014), one would ideally set up inverse problems to perform parameter estimations based on measurements from a real gas network. Although possible in principle, this is prohibitively expensive in practice. Instead, we validate our model components against a commercial simulation software by a systematic comparison. This is a natural approach since our industrial partners rely on the same simulation software to check the results of the optimization methods.

For a stationary simulation, one generally has to prescribe a suitable set of quantities so that the resulting system of nonlinear differential and algebraic equations has a unique solution. In our case this includes discrete and continuous control settings of active network elements, flows at all but one entry and exit nodes in every connected component of the network, and in addition the pressure at one node in every hydraulically connected area. The simulation problem is typically solved by a Newton framework using numerical integrators. In practical planning applications, the solution is then manually checked against inequality constraints like pressure limits at nodes or the velocity limit on pipes.

Modern simulation methods can compute solutions with high accuracy. For increased computational speed, some simulation tools offer model simplifications, such as replacing heat dynamics by a constant temperature (see Záworka 1993) or deactivating the tracking of gas quality parameters like molar mass or calorific value.

To analyze the physical and technical accuracy of our NLP model components, we now carry out a consistency test against the commercial gas network simulation software SIMONE v5.73 (2004).[1] The relevant components are pipes, resistors, control valves, and compressors: entries, exits, junctions, valves, and short cuts have trivial models and are therefore excluded. For the comparison, each of the four element types is tested with several technical and physical settings, and possibly with several model variants. The variant offering the highest common level of detail between the NLP model and SIMONE is always included in the test set. In specific, we use a non-isothermal model of gas physics with gas mixing at nodes, discretized ODEs for the gas flow in pipes, and maximally detailed models of compressors and drives; see Sect. 3 in Schmidt et al. (2014) for the details of the NLP model.

When generating the test sets, we aim at choosing realistic parameters of network elements. Ideally, we would like to use the elements of a real-world network for the comparison, but due to limitations of the SIMONE API (2009), some parameters can only be entered manually via the graphical user interface, which is impractical for our large number of tests. The test set of every network element type is therefore designed as follows. For every technical parameter (except pipe inclination, which cannot be set with the SIMONE API) we take four quantiles from the distribution of values of a real-world gas network: the 10, 35, 65, and 90 % quantiles. The gas network considered is the northern high-calorific network of our industry partner Open Grid Europe (OGE)[2] (see Sect. 5 for more details of this network). All other

[1] SIMONE software. http://www.liwacom.de.

[2] https://www.open-grid-europe.com.

parameters of our test sets like pressure or flow values are suitably chosen from typical ranges based on our experience. The Cartesian product of the resulting sets of parameters then defines the test set. Excluding the lower and upper 10 % of parameter values helps to avoid unrealistic parameter combinations that would otherwise distort the analysis. We also exclude those cases from the analysis where SIMONE or the NLP solver do not converge or where SIMONE yields a solution that violates the velocity limit on a pipe. The remaining successful cases will be referred to as *valid* tests.

As measures of deviation for a physical or technical quantity x (such as pressure or temperature) we will consider the absolute and relative deviations of x itself,

$$d_{\text{abs}}(x) = |x_{\text{NLP}} - x_{\text{Sim}}|, \qquad d_{\text{rel}}(x) = \frac{|x_{\text{NLP}} - x_{\text{Sim}}|}{|x_{\text{Sim}}|},$$

and, if applicable, the relative deviation of the change of x along a network arc,

$$d_{\text{rel}}(\Delta x), \quad \Delta x = x_{\text{in}} - x_{\text{out}}.$$

Here, x_{Sim} and x_{NLP} denote the respective solution values of SIMONE and the NLP model. To obtain a meaningful comparison, the component parameters are always identical in the NLP and in SIMONE, similarly the transport situations (typically defined by Q_0, p_{in}, T_{in}), and the model variants are selected to agree as well as possible. To avoid that exceptional cases dominate the comparison, we will discuss arithmetic mean deviations rather than maximum deviations.

3.1 Pipes

For the validation of pipes we select four model variants: the combinations of two pressure loss models (quadratic approximation and ODE discretization) with two models of the compressibility factor (AGA formula and Papay's formula). For each of these four variants, 12,288 test cases are obtained as combinations of the parameters given in Table 1. The geodesic height of the inflow node is fixed at 0 m, so that 8192 physically possible test cases remain (with $|h_{\text{out}}| < L$). The valid tests are used to calculate average deviations of $x \in \{p_{\text{out}}, T_{\text{out}}\}$. Here $d_{\text{rel}}(\Delta x)$ is only computed if the change Δx is at least 1% of the inflow value, to avoid numerical noise dominating in cases where the change is insignificant and hence the denominator Δx becomes very small, which could bias the comparison substantially. Table 2 shows the resulting deviations. As expected, the deviations between the discretized ODE model and SIMONE are smaller than in case of the quadratic approximation. Note that SIMONE always applies an implicit integration over time for the Euler equations of gas; dynamics; cf. Králik et al. (1988) and Záworka (1993). In the stationary case, the PDE solution approaches the solution of the spatial ODE asymptotically, and SIMONE simply uses a large time interval to obtain a highly accurate solution. Nevertheless, the absolute deviation of pressure with the quadratic approximation is only about 0.1 bar, which appears to be sufficiently small for practical purposes in mid- to long-term planning tasks. Figure 3 shows logarithmically scaled histograms of the absolute deviations of outflow pressure and temperature for the model using the discretized

Table 1 Parameters for test cases of pipes

Quantity	Symbol	Tested values	Unit
Length	L	0.01, 0.1, 0.9, 46.0	km
Diameter	D	150, 310, 405, 1185	mm
Roughness	k	0.006, 0.02, 0.1, 0.5	mm
Geodesic height of head	h_{out}	−500, 0, 500	m
Inflow pressure	p_{in}	3.9, 15.3, 53.8, 74.4	bar
Inflow temperature	T_{in}	288.15, 298.15, 308.15, 318.15	K
Normal volumetric flow	Q_0	50, 250, 500, 750	1000 Nm^3/h

Table 2 Deviations of outflow pressure (p_{out}, bar) and outflow temperature (T_{out}, K) of pipes

Model variant	Valid tests	Symbol	$d_{\text{abs}}(x)$	$d_{\text{rel}}(x)$	$d_{\text{rel}}(\Delta x)$
Approximation, AGA	2419	p_{out}	0.1090	0.0086	0.0581
Approximation, Papay	2651	p_{out}	0.1010	0.0094	0.0591
ODE, AGA	2658	p_{out}	0.0462	0.0031	0.0323
ODE, Papay	2659	p_{out}	0.0504	0.0033	0.0325
Approximation, AGA	2419	T_{out}	1.1000	0.0038	0.1610
Approximation, Papay	2651	T_{out}	1.0000	0.0035	0.1500
ODE, AGA	2658	T_{out}	0.2160	0.0007	0.0457
ODE, Papay	2659	T_{out}	0.1480	0.0005	0.0287

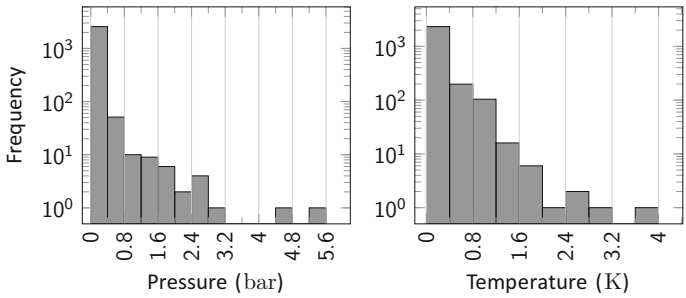

Fig. 3 Logarithmic histograms of outflow pressure and temperature for the model variants "ODE discretization" and "Papay's formula"

ODE and Papay's formula. For the outflow pressure the leftmost bin contains 2574 of 2659 samples (97 %) and for outflow temperature the leftmost bin contains 2330 of 2659 samples (88 %).

Let us now discuss a specific test instance in more detail. Figure 4 shows some exemplary pressure profiles (outflow pressure vs. flow) of SIMONE and two NLP model variants, each combined with Papay's formula. As one would expect, the outflow pressure deviations between the models grow with increasing pressure loss. In fact, almost arbitrarily large deviations can be generated if the pipe parameters

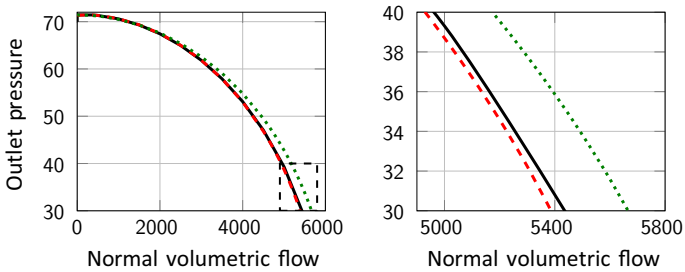

Fig. 4 Outflow pressure profile of an exemplary pipe (L = 46 km, D = 1185 mm, k = 0.006 mm, s = 0.01, p_{in} = 74.44 bar, T_{in} = 318.15 K and T_{soil} = 284.15 K) computed using Papay's formula in SIMONE v5.73 (——), NLP with ODE discretization (- - -), and NLP with a quadratic approximation (···). Figure on the *right* is a zoom of the dashed frame in the *left* figure

Table 3 Extremal pipe example

Parameters
Length L = 46 km
Diameter D = 850 mm
Roughness k = 1 mm
Slope s = 0.01
Normal volumetric flow $Q_0 = 1944 \times 10^3$ Nm^3/h
Inflow pressure p_{in} = 199.1 bar
Results (outflow pressure p_{out})
p_{out}(SIMONE) = 4 bar
p_{out}(NLP, ODE discr.) = 30.6 bar
p_{out}(NLP, quadr. approx.) = 42.5 bar

and boundary values are chosen such that huge pressure losses result: for instance, a gap of about 38.5 bar between SIMONE and the quadratic approximation is obtained for the example in Table 3. In this (totally unrealistic) case, the pipe is rather long and its inner wall is extremely rough.

Turning to the outflow temperatures in Fig. 5, we observe that all three profiles are roughly comparable but visibly different. An interesting aspect is the difference between the NLP models based on approximation and ODE discretization. As can be seen in Fig. 5, the approximation model does not catch the convex curvature at small flows. Nevertheless, the average absolute deviation of 1 K (cf. Table 2) lies within the range of data accuracy: environmental forecasts, soil temperatures, and heat transfer coefficients of pipe walls are inherently inaccurate, and data imprecision is likely to introduce larger errors than 1 K.

Figure 5 also shows a strange behavior of SIMONE for flow values close to zero: there is a sharp peak in the outflow temperature profile that cannot actually represent physical behavior. Therefore we tested the example of Fig. 5 again with the more recent software release SIMONE v5.83. This version appears to fix the problem at

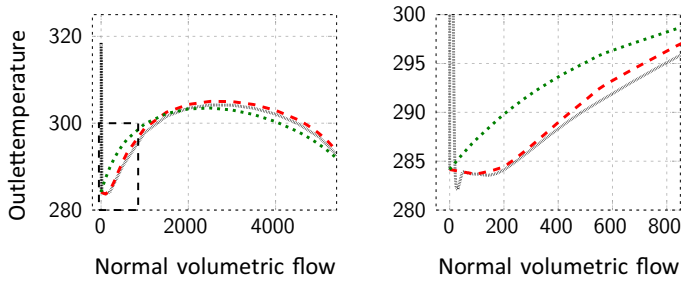

Fig. 5 Outflow temperature profile for the pipe of Fig. 4

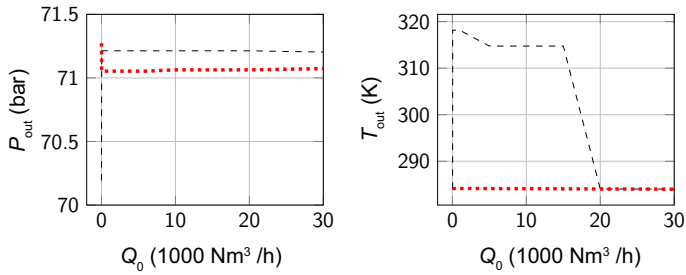

Fig. 6 Differences between SIMONE v5.73 (– – –) and SIMONE v5.83 (····) versions for small flows

small flows, see Fig. 6 (right). However, both versions exhibit a discontinuity of the outflow pressure at zero flow, see Fig. 6 (left).

Moreover, comparing both versions at large flows in the example of Fig. 6 yields deviations up to 1.5 bar for outflow pressure and up to 0.93 K for outflow temperature. These deviations are larger than most of the deviations between the NLP model and SIMONE v5.73. Thus, regarding temperature, the NLP model appears to match SIMONE as well as the two versions of SIMONE match each other.

Finally, although the quadratic pressure loss approximation is a substantial simplification of the PDE model solved by SIMONE, its results appear to be sufficiently accurate for practical use in mid- to long-term planning tasks: the deviations discussed above are mostly smaller than data inaccuracies caused by uncertainty of measurements and unknown network data. In any case, if higher physical accuracy is required, the approximation model can easily be replaced by the ODE model.

3.2 Control valves and resistors

In order to obtain unique solutions for control valves and resistors, the values of inflow pressure, inflow temperature, and flow are fixed, and in case of control valves also the pressure reduction. For the compressibility factor we consider again both the AGA formula and Papay's formula. The temperature change in control valves and resistors is calculated by the same model, cf. Eq. (100) of Schmidt et al. (2014), hence similar behavior is to be expected in the comparisons. The definition of the test set is given in Tables 4 and 5 and the results can be seen in Tables 6 and 7.

Table 4 Parameters for test cases of control valves

Quantity	Symbol	Tested values	Unit
Pressure decrease	Δp	0, 5, 10	bar
Inflow pressure	p_{in}	20, 40, 60	bar
Inflow temperature	T_{in}	283.15, 300.15, 318.15	K
Normal volumetric flow	Q_0	500, 1000	1000 Nm^3/h

Table 5 Parameters for test cases of resistors

Quantity	Symbol	Tested values	Unit
Inflow pressure	p_{in}	20, 40, 60	bar
Inflow temperature	T_{in}	283.15, 300.15, 318.15	K
Normal volumetric flow	Q_0	500, 1000	1000 Nm^3/h
Pressure drop	ξ	2, 4, 6, 8, 10	bar
Drag factor	ζ	5, 20, 50, 70, 90	1
Diameter	D	300, 525, 775, 1000	mm

Resistors with a linear pressure loss model are described by the pressure drop ξ and those with a nonlinear (Darcy–Weisbach type) pressure loss model are described by the drag factor ζ and the diameter D

Table 6 Outflow temperature (T_{out}) deviations of control valves

Model variant	Valid tests	$d_{abs}(x)$	$d_{rel}(x)$	$d_{rel}(\Delta x)$
AGA	54	0.7390	0.0025	0.3380
Papay	54	0.0938	0.0003	0.0409

Table 7 Outflow pressure (p_{out}) and temperature (T_{out}) deviations of resistors: linear pressure loss model (above) and Darcy–Weisbach type model (below)

Model variant	Valid tests	Quantity	$d_{abs}(x)$	$d_{rel}(x)$	$d_{rel}(\Delta x)$
AGA	90	T_{out}	0.8850	0.0030	0.3370
Papay	90	T_{out}	0.0983	0.0003	0.0357
AGA	317	p_{out}	0.0060	0.0006	0.0013
Papay	318	p_{out}	0.0083	0.0008	0.0018
AGA	317	T_{out}	0.6830	0.0024	0.3410
Papay	318	T_{out}	0.2280	0.0008	0.0740

In one respect they are similar to the case of pipes: Papay's formula yields very small deviations for the outflow temperature whereas the AGA formula yields deviations up to 8 times as large. However, the absolute deviations are always reasonably small at less than 1 K.

For control valves and for resistors with a linear pressure loss model, the outflow pressure depends linearly on the preset values by Eq. (99) of Schmidt et al. (2014) and Eq. (12) of Schmidt et al. (2014). Thus the NLP model and SIMONE should

Table 8 Parameters for test cases of piston and turbo compressors

Quantity	Symbol	Tested values	Unit
Pressure increase	Δp	10, 15, 20	bar
Inflow pressure	p_{in}	45, 50, 55	bar
Inflow temperature	T_{in}	283.15, 300.15, 318.15	K
Normal volumetric flow	Q_0 (piston compr.)	300, 400, 500	1000 Nm^3/h
Normal volumetric flow	Q_0 (turbo compr.)	400, 500, 600	1000 Nm^3/h

agree almost within machine precision, but actually the outflow pressures show an average absolute deviation of about 5×10^{-5} bar. Since the NLP results can be shown to be correct, the deviation possibly indicates an output precision of the SIMONE API of about 1×10^{-4}. The precision of internal computations is not known for SIMONE.

For resistors with a Darcy–Weisbach type pressure loss model, the outflow pressure is determined by the nonlinear eq. (13) of Schmidt et al. (2014). Although all values in this equation are given (except for the outflow pressure), we observe again small but measurable deviations between the NLP and SIMONE. The mean absolute deviation is 0.006 bar with the AGA formula and 0.008 bar with Papay's formula.

3.3 Compressors

For the examination of turbo and piston compressors, we fix the values of inflow pressure, inflow temperature, outflow pressure, and flow (see Table 8 for the specific values). The test set is constructed from 18 combinations of physical models of the compressibility factor, the isentropic exponent, and the temperature rise equations, yielding 1458 test cases both for turbo and piston compressors. We compare computed values of the specific change in adiabatic enthalpy H_{ad} (kJ kg^{-1}), required compressor power P (MW), and compressor outflow temperature T_{out} (K). Table 9 shows the results for turbo and piston compressors. Since H_{ad} and P do not have

Table 9 Results of the comparison of turbo compressors (above) and piston compressors (below) concerning specific change in adiabatic enthalpy (H_{ad}), power (P), and outflow temperature (T_{out})

Compr. factor	Valid tests	Quantity	$d_{abs}(x)$	$d_{rel}(x)$	$d_{rel}(\Delta x)$
AGA	444	H_{ad}	0.0729	0.0019	–
Papay	435	H_{ad}	0.0388	0.0010	–
AGA	444	P	0.0058	0.0009	–
Papay	435	P	0.0105	0.0016	–
AGA	444	T_{out}	0.6810	0.0021	0.0199
Papay	435	T_{out}	0.6200	0.0019	0.0188
AGA	513	H_{ad}	0.0721	0.0021	–
Papay	513	H_{ad}	0.0437	0.0012	–
AGA	513	P	0.0044	0.0014	–
Papay	513	P	0.0053	0.0017	–
AGA	513	T_{out}	0.6600	0.0021	0.0263
Papay	513	T_{out}	0.6060	0.0019	0.0243

inflow and outflow values, $d_{\mathrm{rel}}(\Delta x)$ does not exist for these quantities. The results of the NLP model and SIMONE agree very well for the compressors: the relative deviations $d_{\mathrm{rel}}(x)$ are no larger than 0.2 %, and even the relative deviations $d_{\mathrm{rel}}(\Delta x)$ with respect to the temperature change are just about 2 %. A possible reason for the remaining small temperature difference may lie in the model of the isentropic exponent or in differences of the adiabatic efficiency. These two quantities cannot be accessed with the SIMONE API and are therefore not compared.

3.4 Conclusion

Of course, one cannot expect that computational results of any simulation software and of the NLP model agree with high precision. Differences in numerical algorithms, discretizations, and implementations make deviations almost inevitable. In addition, it is likely that there are differences in data handling and in the precision of data input and output. Obviously, these aspects may add to the deviations caused by the numerical schemes.

As we have seen, it is difficult to validate our highly accurate NLP model against a closed-source simulation package like SIMONE because the code of the latter cannot be inspected. This is the reason why in certain cases we can only observe but not analyze differences in the computational results. For instance, we have no clue why the outflow temperatures of several network element types show unexpectedly large deviations if we use the AGA formula for the compressibility factor, whereas deviations are very small with Papay's formula. A possible reason might be that the correction term for the heat capacity (see Eq. (38) of Schmidt et al. 2014) is handled in different ways. As stated in part 1 of this paper, that term evaluates to zero with the AGA formula, so the value is *exact* in our NLP model. However, it is possible that SIMONE evaluates the integral numerically rather than simply dropping it. Differences like those may cause many of the unexplained deviations, but we have no way to tell.

In summary, the observed differences between the NLP model and SIMONE are surprisingly small. Thus, we view the comparison presented above as a successful validation of our model components for the intended use in optimization-based mid-term to long-term planning in gas networks.

4 Optimization techniques

This section addresses two types of techniques that are important when solving the problem of validation of nominations in practice: penalty-based relaxations in several standard and problem-specific variants and sequences of simplified NLPs that are used to obtain a good initial estimate for the NLP of interest.

As it is the case for the framework of model aspect graphs in Sect. 2, the techniques presented in this section are not restricted to the field of gas network optimization but can also be applied to other problems from engineering or physics.

4.1 Penalty formulations

Penalty formulations provide an essential tool for the modeler as well as for the practitioner who uses optimization software to solve real-world problems. For the modeler it is often difficult to find model errors or inconsistencies, particularly if he or she is not also a practitioner or an expert in the area of application. More importantly, the modeler (or practitioner) is often troubled with erroneous, incomplete, or inconsistent data. These data problems may lead to infeasible instantiations of the optimization model and are typically hard to detect. Sophisticated penalty formulations relax certain constraints of the optimization model, thus giving a chance to detect the area of the transport network in which a model or data problem may be located. For the practitioner, penalty formulations are also very useful in situations where he or she is not only interested in *optimal* solutions but also in other *feasible* solutions.

Furthermore, penalty formulations are often instrumental in speeding up the solution process by finding an (almost) feasible solution first. It is usually easier and faster to solve the original problem with such a near-feasible solution as initial estimate than solving it from scratch (cf. Sect. 4.2).

In what follows, we present some penalty-based relaxation schemes that are used in gas transport applications. The presented relaxation schemes will be used in Sect. 4.2, and numerical results are finally given in Sect. 5.

4.1.1 Relaxation schemes for gas network optimization models

Suppose that a specific instantiation of interest of the model hierarchy presented in Sect. 3 of Schmidt et al. (2014) is given in standard NLP form,

$$\min_x f(x) \quad \text{s.t.} \quad c_{\mathcal{E}}(x) = 0, \quad c_{\mathcal{I}}(x) \geq 0. \tag{2}$$

Here and in what follows, $\mathcal{E}$ and $\mathcal{I}$ are the respective index sets of equality and inequality constraints. With slack variables $s = (s_{\mathcal{E}}^+, s_{\mathcal{E}}^-, s_{\mathcal{I}}^+)$, the associated standard ℓ_1 penalty model is then defined as

$$\min_{x,s} ||s||_1 \quad \text{s.t.} \quad c_{\mathcal{E}}(x) + s_{\mathcal{E}}^+ - s_{\mathcal{E}}^- = 0, \quad c_{\mathcal{I}}(x) + s_{\mathcal{I}}^+ \geq 0, \quad s \geq 0. \tag{3}$$

Likewise, the standard ℓ_∞ penalty model reads

$$\min_{x,s} ||s||_\infty \quad \text{s.t.} \quad c_{\mathcal{E}}(x) + s_{\mathcal{E}}^+ - s_{\mathcal{E}}^- = 0, \quad c_{\mathcal{I}}(x) + s_{\mathcal{I}}^+ \geq 0, \quad s \geq 0. \tag{4}$$

Both models are full relaxations of (2) and hence feasible by construction. The ℓ_1 penalty model (3) is actually smooth since $||s||_1$ is just the sum of the components of s, while the ℓ_∞ penalty model is easily converted to an equivalent smooth formulation by adding an extra variable z for $||s||_\infty$ and inequality constraints $z - s_i \geq 0$ for all components of s.

Minimizing the ℓ_1 norm has the well-known property that it tends to generate *sparse* solutions, i.e., solutions with only a few nonzero entries in s (see Elad (2010),

Wotao and Zhang (2008), and the references therein). This property is very useful in detecting and analyzing potential reasons of infeasibility.

Minimizing the ℓ_∞ norm is useful if one is interested in determining the smallest possible upper bound on the constraint violations. In contrast to (3), solutions of (4) tend to have the property that many components of s do not vanish but often have significantly smaller values than the nonzero components of solutions to the corresponding ℓ_1 penalty model.

These two basic relaxation types are now used to develop problem-specific relaxation schemes. In practice, one often does not wish to relax all constraints but only certain classes. For instance, such a class may contain all pressure loss modeling equations on pipes or all constraints modeling the borders of characteristic diagrams of compressors. For handling selected classes of constraints, we introduce the index set $\mathcal{R} \subseteq \mathcal{E} \cup \mathcal{I}$ indicating which constraints are to be relaxed. The required slack variables are $s = (s^+_{\mathcal{E}\cap\mathcal{R}}, s^-_{\mathcal{E}\cap\mathcal{R}}, s^+_{\mathcal{I}\cap\mathcal{R}})$. With this notation, the partly relaxed ℓ_1 penalty model is given by

$$\begin{aligned} \min_{x,s} \ ||s||_1 \quad \text{s.t.} \quad & c_{\mathcal{E}\setminus\mathcal{R}}(x) = 0, \quad c_{\mathcal{E}\cap\mathcal{R}}(x) + s^+_{\mathcal{E}\cap\mathcal{R}} - s^-_{\mathcal{E}\cap\mathcal{R}} = 0, \\ & c_{\mathcal{I}\setminus\mathcal{R}}(x) \geq 0, \quad c_{\mathcal{I}\cap\mathcal{R}}(x) + s^+_{\mathcal{I}\cap\mathcal{R}} \geq 0, \quad s \geq 0. \end{aligned} \tag{5}$$

The corresponding partly relaxed ℓ_∞ norm penalty model reads

$$\begin{aligned} \min_{x,s} \ ||s||_\infty \quad \text{s.t.} \quad & c_{\mathcal{E}\setminus\mathcal{R}}(x) = 0, \quad c_{\mathcal{E}\cap\mathcal{R}}(x) + s^+_{\mathcal{E}\cap\mathcal{R}} - s^-_{\mathcal{E}\cap\mathcal{R}} = 0, \\ & c_{\mathcal{I}\setminus\mathcal{R}}(x) \geq 0, \quad c_{\mathcal{I}\cap\mathcal{R}}(x) + s^+_{\mathcal{I}\cap\mathcal{R}} \geq 0, \quad s \geq 0. \end{aligned} \tag{6}$$

Note that these two relaxations are not necessarily feasible, except of course with the choice $\mathcal{R} = \mathcal{E} \cup \mathcal{I}$ where they become identical to the standard penalty models.

The partly relaxed penalty models (5) and (6) are of special interest in practice, because a suitable choice of the index set $\mathcal{R}$ may allow to detect reasons of infeasibility. Of course, from the mathematical viewpoint, there is no such thing as a "reason" of infeasibility: infeasibility is simply caused by a set of incompatible constraints, and in general there may be a variety of possibilities to achieve feasibility by relaxing different subsets of those constraints. In practice, however, different constraints often have different relevance (depending on the specific situation). Then, if the violation of a lower-priority constraint makes the problem otherwise feasible, that lower-priority constraint may be interpreted as "the reason" of infeasibility, and a sufficiently small violation will be tolerated.

Suppose that a practitioner is confronted with an infeasible instance. If, say, a relaxation of compressor group constraints leads to feasibility, the reason might be that a lower pressure bound in a downstream part of the network is too tight. Another reason might be that the operating ranges of certain active compressor units are not sufficiently large for generating an outflow pressure that satisfies the pressure bounds in the downstream network. The opposite situation (too tight upper pressure bounds) may be detected by relaxing control valve constraints that limit the pressure decrease from below. Here, the "reason" depends on the interpretation of the situation by the practitioner, who may decide that the pressure bounds or the

borders of the characteristic diagrams of the compressors are more important, respectively. Another relaxation that is often used in practice is to relax the supplied and discharged amounts of flow at entry and exit nodes. Given a solution to this relaxed model with non-vanishing slack variables, the practitioner obtains the information of how much the contracts with one or more entry or exit customers have to be modified in order to achieve a feasible flow situation.

Going further, it may be useful for practitioners to partition the index set $\mathcal{R}$ into several mutually disjoint constraint classes $\mathcal{R}_k$, $k \in \mathcal{K}$,

$$\mathcal{R} = \bigcup_{k \in \mathcal{K}} \mathcal{R}_k \subseteq \mathcal{E} \cup \mathcal{I},$$

and allow a specific maximum constraint violation $\bar{s}_k$ for each constraint class under consideration. Letting $s_k = (s^+_{\mathcal{E}\cap\mathcal{R}_k}, s^-_{\mathcal{E}\cap\mathcal{R}_k}, s^+_{\mathcal{I}\cap\mathcal{R}_k})$ and $s = (s_k)_{k\in\mathcal{K}}$, this leads to the modified ℓ_∞ feasibility problem

$$\begin{aligned} \text{Find} \quad (x,s) \quad \text{s.t.} \quad & c_{\mathcal{E}\setminus\mathcal{R}}(x) = 0, \quad c_{\mathcal{E}\cap\mathcal{R}}(x) + s^+_{\mathcal{E}\cap\mathcal{R}} - s^-_{\mathcal{E}\cap\mathcal{R}} = 0, \\ & c_{\mathcal{I}\setminus\mathcal{R}}(x) \geq 0, \quad c_{\mathcal{I}\cap\mathcal{R}}(x) + s^+_{\mathcal{I}\cap\mathcal{R}} \geq 0, \\ & s \geq 0, \quad \bar{s}_k - ||s_k||_\infty \geq 0 \quad \text{for } k \in \mathcal{K}. \end{aligned} \tag{7}$$

This refined relaxation plays an important role in the presence of *soft constraints* (see Joormann et al. 2015 for a detailed discussion). For instance, the gas flow can excite vibrations of the pipes, which in turn may generate undesired noise in populated areas and, in serious cases, may even destroy the pipes. As there is no suitable quantitative model for these phenomena, a simple speed limit for the gas flow often serves as a crude practical measure to prevent vibrations. In this situation, the network operator may formulate the goal that the speed bound should be satisfied ($v \leq v^+$), although he will accept small violations if necessary. So he can define the gas velocity constraints to be one of the sets $\mathcal{R}_k$ and can specify an additional quantity $\bar{s}_k$ that softens the former bound to $v \leq v^+ + \bar{s}_k$. Soft constraints like these often play an important role in real-world problems and can be covered by penalty reformulations like (7).

4.2 Sequential NLP solving

As one might imagine, industrial users would always like to use the most accurate available model. Unfortunately but naturally, the most detailed and accurate model variants presented in Sect. 3 of Schmidt et al. (2014) are the most nonlinear and nonconvex ones. In addition, some of them are nonsmooth, i.e., the standard $\mathcal{C}^2$ assumption is violated unless one incorporates numerically challenging smoothing techniques. All these facts illustrate that it is an ambitious task to solve these models from scratch. Since it is impractical for industrial users of optimization methods to adapt the model or to tune the parameters of the solver for individual problem instances, there is a general need for numerically robust solution techniques.

The model hierarchy presented in Sect. 3 of Schmidt et al. (2014) suggests a natural way to achieve this goal. The large variety of NLP model variants allows us

to set up *sequences* of NLPs that can be solved successively. The key aspect is to arrange each sequence in a way such that successive NLPs differ only slightly in terms of model size and in increase of nonlinearity and non-convexity, thus allowing for warm starts.

4.2.1 NLP sequences

We consider finite sequences (NLP_k), $k \in \mathcal{N}$, that are chosen from the family of NLPs given in (1). Recall that $\mathcal{A}$ is the set of all model aspects where aspect $\alpha \in \mathcal{A}$ has concretizations $\gamma_\alpha \in \mathcal{C}_\alpha$. With this notation, we require that a sequence of NLPs has increasing *accuracy order*, defined by

$$\mathsf{NLP}_k \leq \mathsf{NLP}_{k+1} :\Longleftrightarrow \gamma_\alpha^k \leq \gamma_\alpha^{k+1} \quad \text{for all} \quad \alpha \in \mathcal{A}, \quad k, k+1 \in \mathcal{N}.$$

Here, the notation "$\gamma_\alpha^k \leq \gamma_\alpha^{k+1}$" means that the concretization γ_α^k of model aspect α used in NLP_k is no more accurate than the concretization γ_α^{k+1} used in NLP_{k+1}. The notion of "more accurate" is not always defined in a strict mathematical sense (the Papay formula is considered more accurate than the AGA formula, for instance), but in any case we require that all variables of a model variant are also present in a "more accurate" variant. Thus, the numbers of NLP variables in a sequence satisfy $n_k \leq n_{k+1}$.

4.2.2 Increasing robustness and convergence acceleration

When solving an NLP sequence we make use of three key ideas that improve the solution process by making it more robust and by accelerating convergence of the individual NLPs:

1. In most of the sequences used in practice, at least the first NLP is a penalty formulation of the *target NLP* that we actually want to solve; thus solving the first NLP is analogous to phase 1 of the simplex method. The benefit of this approach is twofold. First, solving a penalty model is much easier than solving the *original* model in many cases, according to our experience. Second, if a given instance is infeasible, the penalty formulation allows for a better analysis of possible reasons (cf. Sect. 4.1). In view of this, similar techniques have also been used in optimization for water networks; cf. Burgschweiger and Steinbach (2009).
2. Nonsmooth model aspects appear primarily in the model of gas parameter mixing, because flow directions in the network are initially unknown (cf. Eq. (9) of Schmidt et al. 2014). When solving an NLP of the sequence that incorporates mixing of gas parameters, we fix the flow directions on all network arcs based on the solution of the previous NLP, as follows. Let $q_a^k \in \left[q_a^-, q_a^+\right]$ denote the bounded flow variable of arc a in NLP_k and let $(q_a^k)^*$ denote its optimal value in NLP_k. The flow direction for NLP_{k+1} is then fixed by setting properly restricted bounds,

$$q_a^{k+1} \in \begin{cases} \mathbb{R}_{\geq 0} \cap \left[q_a^-, q_a^+\right], & (q_a^k)^* > 0, \\ \mathbb{R}_{\leq 0} \cap \left[q_a^-, q_a^+\right], & (q_a^k)^* < 0, \\ \{0\}, & (q_a^k)^* = 0. \end{cases} \tag{8}$$

Similar ideas can be used for other nonsmooth aspects, like (de-)activating gas coolers (see Eq. (68) of Schmidt et al. 2014) and preheaters (see Eq. (103) of Schmidt et al. 2014). It would be reasonable to consider the optimal values of the dual variables of NLP_k in order to relax the third case in (8), yielding restricted bounds $\mathbb{R}_{\geq 0} \cap \left[q_a^-, q_a^+\right]$ or $\mathbb{R}_{\leq 0} \cap \left[q_a^-, q_a^+\right]$ instead of fixing the flow to zero. We have not done this in the current study because of the technical effort for accessing the dual variables from many NLP codes with different interfaces that we used, and because some solvers do not even provide dual variables.

3. We initialize the variables of NLP_{k+1}, $x^{k+1} \in \mathbb{R}^{n_{k+1}}$, based on the optimal solution of NLP_k, $(x^k)^* \in \mathbb{R}^{n_k}$. This is done as follows. After some re-ordering we can write $x^{k+1} = (x^k, \tilde{x}^{k+1})$ with $\tilde{x}^{k+1} \in \mathbb{R}^{\tilde{n}}$, $\tilde{n} = n_{k+1} - n_k$. The key idea is to fix the x^k part at $(x^k)^*$ and to determine $\tilde{x}^{k+1}$ by simple techniques so that $((x^k)^*, \tilde{x}^{k+1})$ satisfies the constraints of NLP_{k+1} as well as possible. For instance, consider a constraint $c(x^k, \tilde{x}_i^{k+1}) = 0$ that can be solved explicitly for the "new" variable $\tilde{x}_i^{k+1}$ in terms of the "old" variables,

$$0 = c(x^k, \tilde{x}_i^{k+1}) \Longleftrightarrow \tilde{x}_i^{k+1} = \tilde{c}(x^k) =: \tilde{x}_i^*.$$

In this case we initialize $\tilde{x}_i^{k+1}$ with $\tilde{x}_i^*$. This technique is successively applied to new variables that depend on x^k and already initialized components $\tilde{x}_i^{k+1}$. In other situations, we simply initialize the new variables with suitable constants.

4.2.3 Stopping criterion

In the finite sequences considered thus far, $\mathsf{NLP}_{|\mathcal{N}|}$ is the most accurate model that we attempt to solve. It is also possible to extend the presented technique to infinite sequences of NLPs. For instance, one may wish to increase the physical accuracy of the model of gas dynamics in pipes by discretizing the ODEs on successively finer grids, which calls for a *stopping criterion*. Various criteria are possible, and a natural requirement is that they should be based on changes in variables that exist in every member NLP_k of the sequence. Meaningful quantities of this type are pressures at nodes and mass flows on arcs of the network. With the vectors

$$p_{\mathbb{V}}^k := (p_i^k)_{i \in \mathbb{V}}, \quad q_{\mathbb{A}}^k := (q_a^k)_{a \in \mathbb{A}},$$

and tolerances $\varepsilon_p, \varepsilon_q > 0$, a suitable stopping criterion is given by

$$||p_{\mathbb{V}}^k - p_{\mathbb{V}}^{k-1}||_\infty < \varepsilon_p \quad \text{and} \quad ||q_{\mathbb{A}}^k - q_{\mathbb{A}}^{k-1}||_\infty < \varepsilon_q. \tag{9}$$

4.2.4 Specific NLP sequences

In this section we give three examples of sequences that are useful in gas network optimization. The first sequence mainly addresses the nonsmoothness of the gas quality parameter mixing model given in Sect. 3.3 of Schmidt et al. (2014). The second one deals with highly detailed isothermal modeling of gas dynamics in pipes and the last one can be used in order to incorporate heat dynamics in the problem. Computational results of the sequential NLP approach are presented in Sect. 5. For all sequences, we assume that all discrete decisions are given.

An NLP sequence for gas parameter tracking. The finite sequence for gas quality parameter tracking is informally defined as follows:

- NLP_1 Simple mass conservation model at nodes without gas parameter mixing (see Eq. (7) of Schmidt et al. 2014), isothermal quadratic approximation of gas dynamics in pipes (see Eq. (55) of Schmidt et al. 2014), full isothermal compressor group model with machine models as described in Sects. 3.4.10 and 3.4.15 of Schmidt et al. (2014), isothermal control valve model (see Eq. (99) of Schmidt et al. 2014), standard resistor, short cut, and valve models, arbitrary penalty formulation (see Sect. 4.1). Globally constant gas density under normal conditions.
- NLP_2 Like NLP_1 but with an extended node model incorporating mixing of molar masses (see Eq. (9) of Schmidt et al. 2014).
- NLP_3 Like NLP_2 but with an extended node model incorporating mixing of calorific values.
- NLP_4 Like NLP_3 but with an extended node model incorporating mixing of pseudocritical pressures and temperatures.
- NLP_5 Like NLP_4 but with gas density under normal conditions depending on the quality parameters.

To obtain smooth formulations of the mixing models it is necessary to fix the flow directions on all network arcs, as already discussed in the previous section.

An NLP sequence for highly detailed isothermal gas dynamics modeling. In this sequence the main focus is on highly detailed modeling of isothermal gas dynamics in pipes. The sequence is given as follows:

- NLP_1 Identical to NLP_1 in the sequence above.
- NLP_2 Like NLP_1 but with the momentum equation discretized on initial grids Δ_a^0 for all pipes $a \in \mathbb{A}_{\mathrm{pi}}$; cf. Sect. 3.4.7 of Schmidt et al. (2014).
- $\mathsf{NLP}_k, k > 2$ Like NLP_{k-1} but with the momentum equation discretized on refined grids $\Delta_a^{k-2} \supset \Delta_a^{k-3}$ for all pipes $a \in \mathbb{A}_{\mathrm{pi}}$.

The canonical stopping criterion for this infinite sequence is (9). To reduce the numerical effort during iterative grid refinement ($k > 2$), one might also employ an adaptive discretization scheme. Here *adaptive* means that the discretization on pipe $a = ij \in \mathbb{A}_{\mathrm{pi}}$ is only refined in NLP_k if

$$\left\| \begin{pmatrix} p_i \\ p_j \end{pmatrix}^k - \begin{pmatrix} p_i \\ p_j \end{pmatrix}^{k-1} \right\|_\infty > \varepsilon_p \quad \text{or} \quad |q_a^k - q_a^{k-1}| > \varepsilon_q. \tag{10}$$

Of course, it is also possible to use pipe-specific tolerances, i.e., to replace ε_p and ε_q in (10) by $\varepsilon_{p,a}$ and $\varepsilon_{q,a}$.

An NLP sequence for temperature dynamics modeling. Here we incorporate all temperature dynamics constraints in the model. The sequence is defined as follows:

- NLP_1 Identical to NLP_1 in the sequences above.
- NLP_2 Like NLP_1 but with an extended node model incorporating the mixing of the coefficients A, B, C of the molar heat capacity of ideal gas.
- NLP_3 Like NLP_2 but with non-isothermal models of all elements of the network (including heat capacity models).

Notice that none of the three sequences above has a cost minimization objective: all of them solve ℓ_1 penalty-based relaxation models in which we relax the flow balance constraint of all entry and exit nodes. However, all sequences can readily be extended or modified to incorporate a cost minimization objective.

5 Computational results

In order to show the practical relevance of our NLP models and solution techniques on large-scale real-world networks, we now present an extensive computational study for the problem of validation of nominations (NoVa). A complete NoVa model contains discrete aspects of controllable elements as well as detailed models of gas dynamics and technical devices of the network. This combination leads to hard mixed-integer nonlinear optimization (or feasibility) models that are intractable for state-of-the-art general-purpose MINLP solvers on real-world network sizes. In the research project ForNe we have developed custom solution approaches for NoVa. The main idea is to split up the solution process into two stages. The first stage solves a model containing all discrete aspects and simplified variants of the underlying gas physics and technical network elements. Afterwards, the resulting discrete controls are fixed throughout the network in order to obtain a purely continuous and highly detailed NLP model that "validates" the fixed decisions. In this stage, penalty formulations as described in Sect. 4.1 are used to prove (ε-)feasibility of the underlying MINLP, or to obtain information on possible reasons of infeasibility otherwise. Here we use the sequential NLP (SNLP) approach for solving NLP models with a high level of physical and technical accuracy.

This section deals with that second stage of the NoVa solution approach and presents detailed numerical results for it. We will indeed see that the NoVa problem is solvable even with high detail and accuracy, and that the SNLP approach is a key tool for increasing robustness. Additional information concerning the outcomes of the ForNe project and extensive computational studies of the overall approach are given in Fügenschuh et al. (2014), Martin et al. (2011), Pfetsch et al. (2015) and in the book Koch et al. (2015).

The remainder of this section is organized as follows. In Sect. 5.1 we present the computational setup including the description of used modeling languages, solvers, and other software and hardware issues. Sect. 5.2 then describes both the real-world network and the publicly available academic network that we used in our computational experiments. The following Sects. 5.3 and 5.4 present and discuss the respective numerical results on the real-world and the academic test set. Finally, Sect. 5.5 concludes with a discussion of the results.

For brevity, we only present the numerical results for the NLP sequences "parameter tracking" and "temperature dynamics" here. The SNLP approach has also been run successfully on the "ODE Discretization" sequence for highly detailed isothermal gas dynamics modeling (Sect. 4.2.4), with results that are qualitatively comparable to the results of both other sequences.

5.1 Computational setup

All models and algorithms are implemented using the framework LaMaTTO++ (2015) for modeling and solving mixed-integer nonlinear optimization problems on networks. The computational study of the NLP validation stage presented in this paper is run on a desktop PC with a six-core AMD Opteron Processor 2435 with 2600 MHz and 64 GB RAM. The operating system is Debian 7.5 and the C++ code LaMaTTO++ is compiled using GCC 4.7.2. All models are implemented in GAMS 24.1.3[3], cf. Rosenthal (2008).

The computational results presented below are obtained using the NLP solver CONOPT 3.15L, because it performs best on the given problem class and on the networks considered in the ForNe project. The solvers we have tested include Ipopt by Wächter and Biegler (2006), CONOPT and CONOPT4 (1996), KNITRO (2006), SNOPT (2002), and MINOS (1993). Note that the comparison of these solvers is not a topic of this paper.

5.2 Test instances

We evaluate and test our models on two different test sets based on country-sized transport networks. The first test set, called HN, contains 30 difficult expert instances that arise at OGE. The corresponding network is the northern high-calorific gas network of OGE. The second test set, called GasLib-582, is publicly available (cf. Humpola et al. 2015) and is approximately of the same size as the HN network. These two test sets appear under the same names in other publications as well. Table 10 gives the numbers of elements of the two networks. In order to limit the computational effort to a reasonable amount, we randomly chose 500 out of the 4227 GasLib-582 nominations that are freely available at http://gaslib.zib.de. The first stage solvers of our NoVa approach obtained feasible solutions for 394 of the 500 random nominations. We perform the NLP validation on these 394 instances; see Table 13 in Appendix 3 for the complete list.

[3] General Algebraic Modeling System (GAMS). http://www.gams.com/.

Table 10 Elements of the HN network and the GasLib-582 network

Element type	HN Number	GasLib-582 Number
Vertex	747	582
Entry	30	31
Exit	139	129
Junction	578	422
Arc	786	609
Pipe	505	278
Compressor group	8	5
Control valve	27	23
Valve	50	26
Resistor	37	8
Short cut	159	269

The results on the HN test set will demonstrate the practicability of our NLP models on hard instances that arise in the day-to-day work at German gas network planning departments whereas the GasLib-582 test set provides publicly available instances on which other researchers can test and compare their algorithms.

5.3 The HN test set

Here we present the computational results for the SNLP sequences "parameter tracking" and "temperature dynamics" (see Sect. 4.2) applied to the expert instances on the HN test set, and compare them with the *direct approach*, i.e., the attempt to solve the final NLP of each sequence (the *target NLP*) from scratch. SNLP results are determined by the target NLP (exit state, accuracy) or by the entire sequence (iteration count, computing time).

5.3.1 Parameter tracking problem

Recall that we expect that solving a sequence of NLPs while successively adding new gas parameters to the model leads to better convergence than the direct approach. This is indeed the case, as can be seen in Table 11: the exit states show that the direct approach solves only one third of all instances to local optimality whereas the SNLP approach solves every instance.

Since the NoVa problem is a feasibility problem, our objective is to minimize the constraint violation (using the relaxed NLP in Sect. 4.1). Figure 7 presents empirical distribution functions of the respective objective values for the SNLP sequence and the direct approach, giving for every constraint violation the percentage of instances not exceeding that value. The plot clearly shows that the direct approach is much less successful: it yields substantially larger constraint violations and in fact many "false infeasibles". Moreover, only for 2 out of the 30 instances, it computes a zero constraint violation where the SNLP approach fails to do so.

Table 11 Exit states of direct approach and of SNLP sequence (HN test set, both sequence types)

Exit state	"param. tracking"		"temp. dynamics"	
	Direct	SNLP	Direct	SNLP
Locally optimal	10	30	8	30
Locally infeasible	19	0	6	0
Intermediate nonoptimal	1	0	1	0
Intermediate infeasible	–	–	15	0

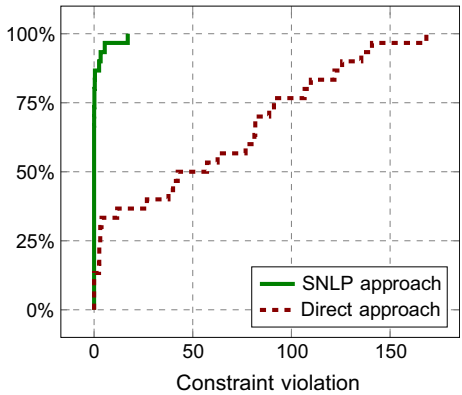

Fig. 7 Constraint violations of direct approach and of SNLP sequence (HN test set, "parameter tracking")

Figure 8 shows the empirical distribution functions for total iteration counts and computing times (i.e., accumulated over all NLPs in case of the SNLP sequence). Despite the fact that the SNLP approach solves five NLPs rather than just one, both approaches require roughly the same effort. While the SNLP approach needs a slightly larger number of iterations than the direct approach, it does not possess as extreme outliers as the latter. In summary, it can be clearly stated that the SNLP approach is much more effective than the direct approach in producing feasible solutions while the computational cost is comparable.

The above observations are confirmed by Fig. 9, which displays the distributions of iteration counts and computing times of individual NLPs in the sequence over the HN test set. Except for a few outliers, all iteration counts are below 100, which is approximately 1/5 of the average iteration count of the direct approach. Outliers above 100 are only present in the first, third, and fifth NLP of the sequence. Possible explanations are as follows. The first model does not contain any mixing aspects, and the outliers thus may arise from "correcting" the continuous part of the solution given by the first stage of the NoVa solution approach. The third NLP adds the calorific value as a new gas quality parameter to the model. Since the HN test set includes heat power bounds at the entries and exits (in addition to the mass flow bounds), activating the tracking of the calorific value plays a crucial role for the feasibility of the problem. Finally, the fifth and last NLP adds the gas density under normal conditions as a variable to the model. Since this quantity occurs in many

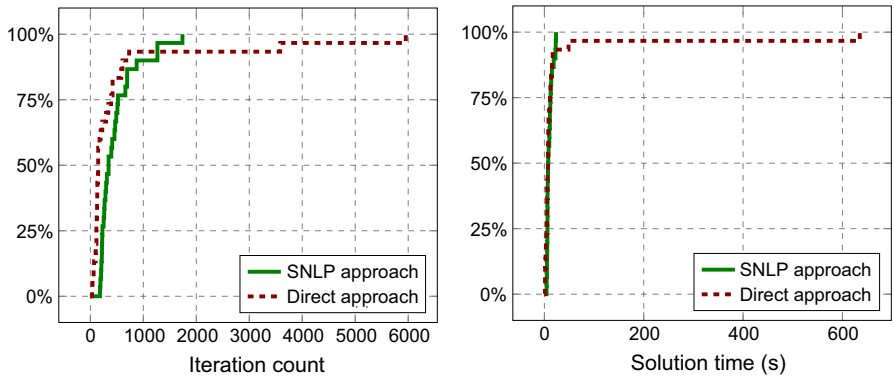

Fig. 8 Iteration counts and computing times (s) of direct approach and of SNLP sequence (HN test set, "parameter tracking")

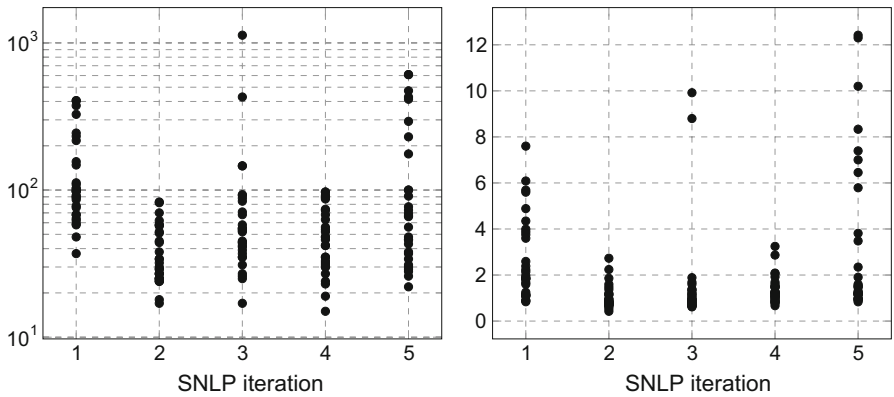

Fig. 9 Iteration counts (*left*) and computing times (s, *right*) of NLPs of the SNLP sequence (HN test set, "parameter tracking")

constraints—especially in all flow conversion constraints—the worst-case difficulty of the model increases, which might be the reason for outliers in SNLP iteration 5. Turning to the computing times, we see that the majority are below 4 s, and we observe a similar qualitative behavior as for the iteration counts.

Let us now consider some model statistics of the NLPs of the SNLP sequence. In all NLPs the number of variables is only slightly larger than the number of constraints: after fixing the discrete decisions only a few degrees of freedom remain, resulting from continuous control variables of active network elements. Both the numbers of variables and constraints strictly increase within the SNLP sequence, lying between 5000 and 15,000. The number of non-constant entries and the total number of entries in the constraints Jacobian also strictly increase, lying between 10,000 and 20,000 and between 20,000 and 40,000, respectively. The fraction of non-zero entries in the Jacobian is approximately 0.025 %.

5.3.2 *Temperature dynamics problem*

The results for "temperature dynamics" are similar to the results for "parameter tracking". Again, the direct approach solves only an unsatisfactory share of instances to local optimality (26.67 %) whereas the SNLP approach solves all instances (see Table 11). Moreover, the SNLP approach reduces the constraint violation much stronger than the direct approach, as Fig. 10 clearly shows. Only two instances are solved with a smaller constraint violation by the direct approach than by the SNLP approach. Furthermore, the SNLP approach is substantially more efficient. Figure 11 compares iteration counts and computing times using empirical distribution functions. On average, the SNLP approach is faster and does not possess as strong outliers as the direct approach does.

Iteration counts of the individual NLPs of the SNLP sequence are displayed in Fig. 12. In particular, the first NLP (isothermal model) appears to be the problem with the hardest instances of the sequence, as in "parameter tracking": the number of outliers of NLP 1 clearly exceeds the numbers of outliers of NLPs 2 and 3.

5.4 The GasLib-582 test set: parameter tracking and temperature dynamics

Next we describe the results for the SNLP sequences "parameter tracking" and "temperature dynamics" on the random subset of all GasLib-582 instances given in Table 13 of Appendix 3.

The characteristics of the computational results for "parameter tracking" and "temperature dynamics" differ significantly from the corresponding results on the HN test set. While it is very hard to compute feasible solutions with the direct approach on the latter (cf. Sect. 5.3), this is not the case for the GasLib-582 test set. Table 12 shows the associated exit states for all sequences. The percentage of instances solved to local optimality by the direct approach is 96.4 % for "parameter tracking" and 95.7 % for "temperature dynamics". By way of comparison, the SNLP approach solves 98.2 % and 98.5 % to local optimality, respectively. This shows that the SNLP approach is again more robust, but this time only by a small

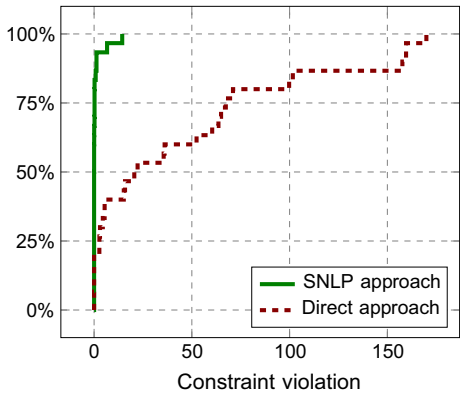

Fig. 10 Constraint violations of direct approach and of SNLP sequence (HN test set, "temperature dynamics")

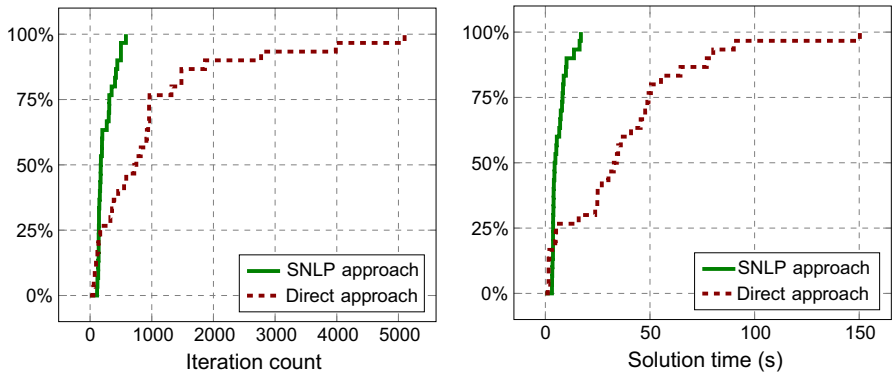

Fig. 11 Iteration counts and computing times (s) of direct approach and of SNLP sequence (HN test set, "temperature dynamics")

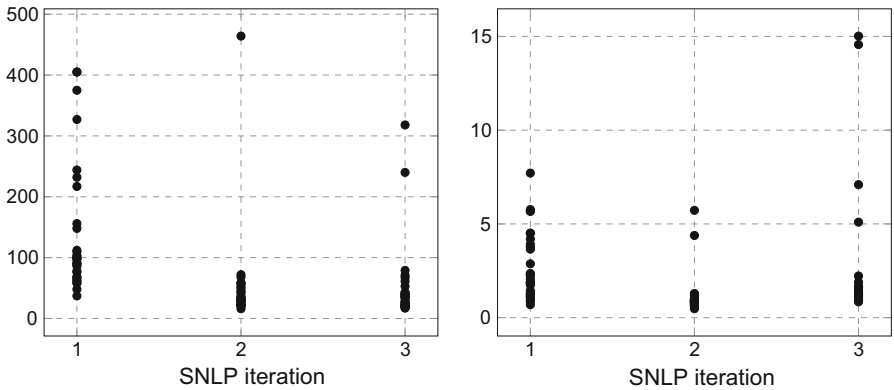

Fig. 12 Iteration counts (*left*) and computing times (s, *right*) of NLPs of the SNLP sequence (HN test set, "temperature dynamics")

Table 12 Exit states of direct approach and of SNLP sequence (GasLib-582 test set, both sequence types)

Exit state	"param. tracking"		"temp. dynamics"	
	Direct	SNLP	Direct	SNLP
Locally optimal	380	387	377	388
Broken sequence	–	7	–	6
Locally infeasible	14	0	15	0
Intermediate nonopt.	0	0	2	0

margin. In contrast to the HN test set there are now also a few instances (7 for "parameter tracking" and 6 for "temperature dynamics") on which the SNLP approach fails completely because one NLP of the corresponding sequence cannot

be solved at all (see the row "broken sequence" in Table 12). All other results of these two problems look very similar to the corresponding results on the HN test set. We only discuss the results for "parameter tracking"—all figures for "temperature dynamics" are given in Appendix 1. As Figs. 13 and 14 show, both approaches behave quite similar in terms of constraint violations, iteration counts, and computing times. The advantage of the SNLP approach is that it does not possess extreme outliers. However, these outliers of the direct approach are not as extreme as for the HN test set. The qualitative behavior of individual iteration counts and numbers of variables, constraints, etc., of the SNLP approach is similar to the behavior on the HN test set and therefore not further discussed here.

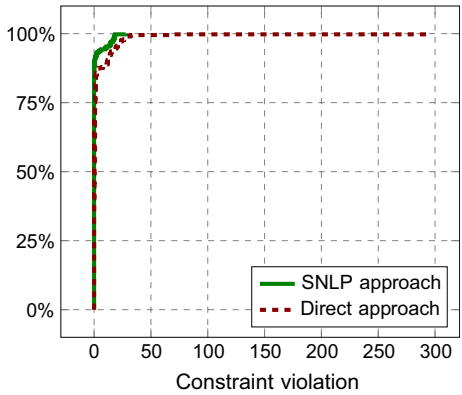

Fig. 13 Constraint violations of direct approach and of SNLP sequence (GasLib-582 test set, "param. tracking")

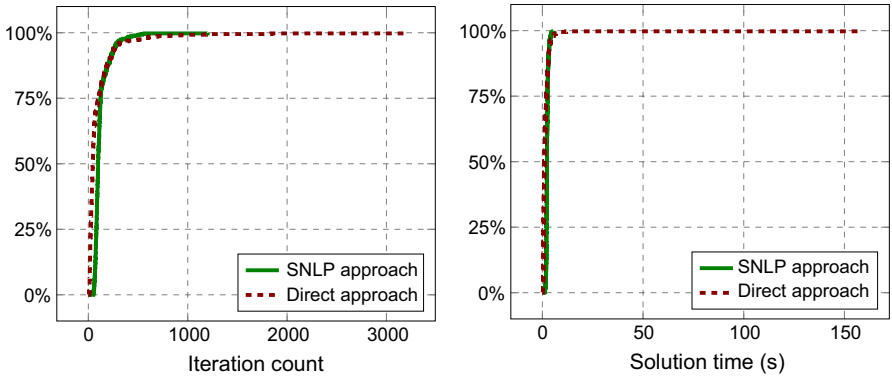

Fig. 14 Iteration counts and computing times of direct approach and SNLP sequence (GasLib-582 test set, "parameter tracking")

5.5 Discussion

In this section we have presented an extensive computational study of our NLP models on two different test sets involving the solution of about 4500 NLPs. The NLP models under consideration are hard to solve due to their high degree of nonlinearity and non-convexity and because they possess only a small number of free variables.

It turns out that the expert instances of the HN test set are much harder than the instances of the GasLib-582 test set. The problems of both tested sequences ("parameter tracking" and "temperature dynamics") are hardly solvable from scratch, whereas the SNLP approach proves robust and entirely practical: it solves all instances of the HN test set successfully, while iteration counts and solution times are comparable to the direct approach when the latter is successful.

6 Summary

In part 1 of this paper we describe detailed models for stationary optimization in gas transport networks. This paper provides an analysis of their structural interplay and possible combinations of these models and shows that their level of detail and accuracy is comparable with today's commercial gas network simulation software. The former aspect is realized by the analysis of so-called model aspect graphs whereas the latter aspect is illustrated by a comparison with a state-of-the-art simulation software on single network elements. Finally, we present tailored optimization techniques and show that—using these techniques—it is possible to solve the resulting NLPs on real-world networks. In particular, by using NLP sequences we are able to include model aspects that are usually omitted or substantially simplified to obtain tractable problems or to reduce the solution effort: the mixing of different gas species and the heat dynamics. This demonstrates that optimization models and solvers can provide a valuable tool for gas network planners that adds new capabilities to existing simulation tools.

Acknowledgments This work has been supported by the German Federal Ministry of Economics and Technology owing to a decision of the German Bundestag. The responsibility for the content of this publication lies with the authors. This research has been performed as part of the Energie Campus Nürnberg and supported by funding through the "Aufbruch Bayern (Bavaria on the move)" initiative of the state of Bavaria. We would also like to thank our industry partner Open Grid Europe GmbH and the project partners in the ForNe consortium. Finally, we thank Björn Geißler and Antonio Morsi for many suggestions for improvement.

Appendix 1: Computational results for temperature dynamics on the GasLib-582 test set

See Figs. 15 and 16.

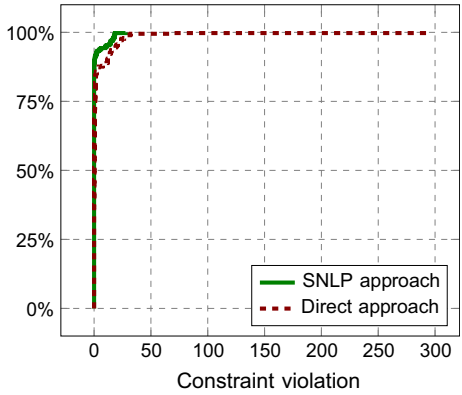

Fig. 15 Constraint violations of direct approach and of SNLP sequence

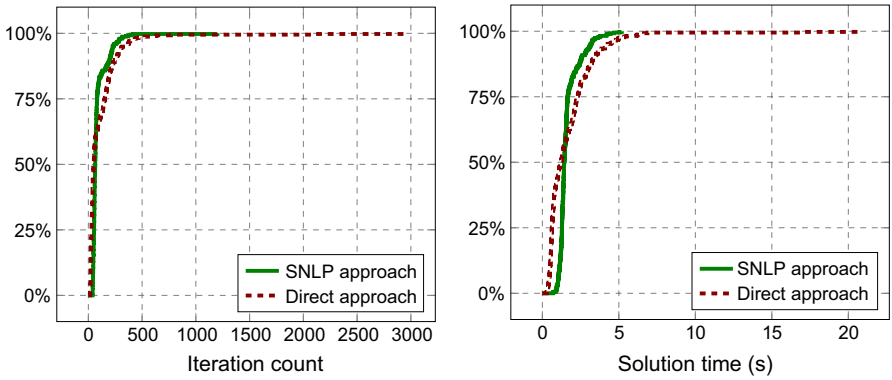

Fig. 16 Iteration counts and computing times (s) of direct approach and of SNLP sequence

Appendix 2: Model aspect graphs

See Figs. 17, 18, 19, 20, and 21.

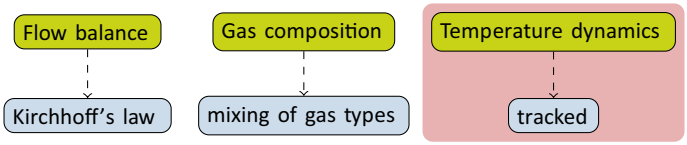

Fig. 17 Model aspect graph of nodes (global aspect: mixing of gas types). The nodes in the shaded block exist only in non-isothermal models

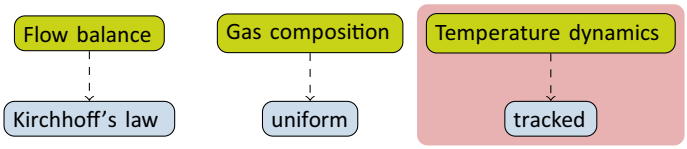

Fig. 18 Model aspect graph of nodes (global aspect: uniform gas composition). The nodes in the shaded block exist only in non-isothermal models

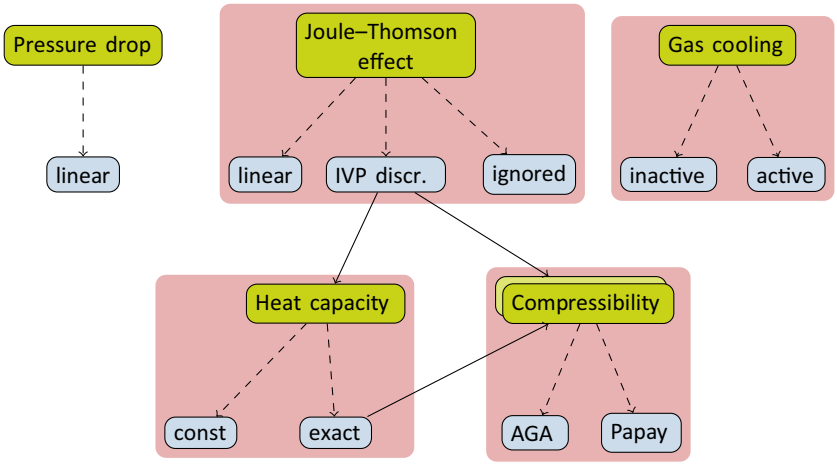

Fig. 19 Model aspect graph of control valves. The nodes in shaded blocks exist only in non-isothermal models

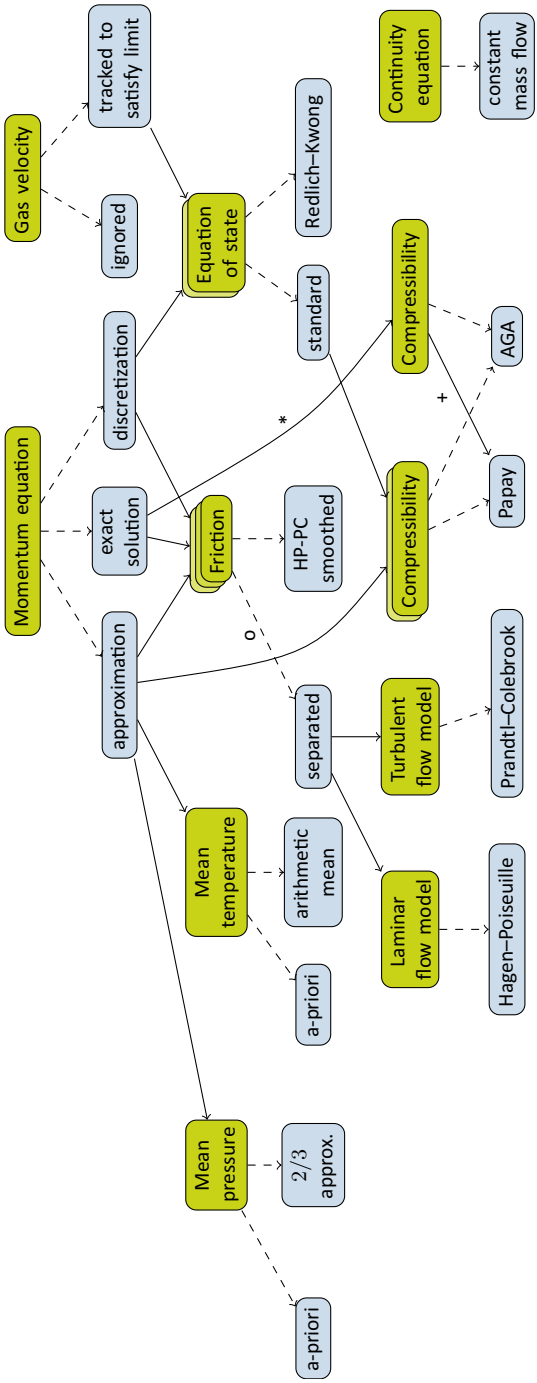

Fig. 20 Model aspect graph of pipes (isothermal). (*) If the exact solutions of the momentum equation are chosen as a concretization, not every choice of the compressibility factor is possible, cf. Schmidt et al. (2014). In addition, not every combination of pipe slope and compressibility factor is possible. (o) To achieve a smooth NLP model, flow bound strengthening has to be applied. (+) Only possible for horizontal pipes

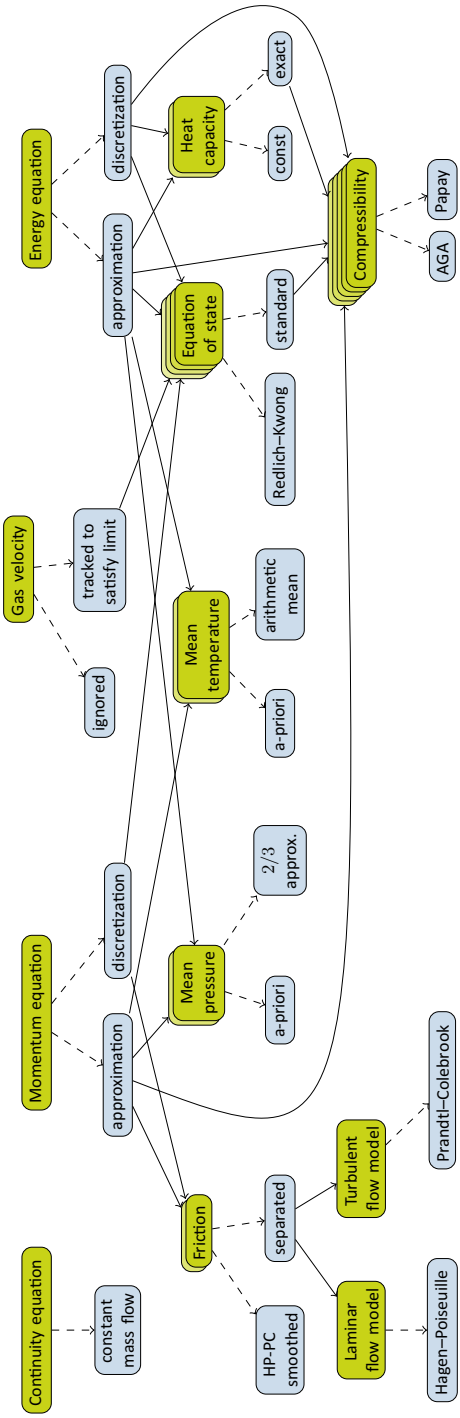

Fig. 21 Model aspect graph of pipes (non-isothermal)

Appendix 3: GasLib-582 instances

See Table 13.

Table 13 ID's of GasLib-582 instances used in the computational study of Sect. 5.4. The set of instances can also be downloaded at www.ifam.uni-hannover.de/mcs

Nomination_freezing_<ID>													
35	51	119	121	164	238	362	402	692	714	728	792	831	1023
1093	1106	1135	1191	1233	1263	1416	1505	1568	1588	1599	1775	1889	2005
2055	2110	2145	2206	2236	2293	2351	2378	2466	2491	2567	2629	2724	2969
3044	3078	3126	3225	3383	3465	3496	3500	3592	3673	3853	3910	4028	4182
Nomination_cold_<ID>													
20	72	276	298	435	486	527	586	595	611	641	681	712	716
796	823	890	893	1033	1056	1132	1156	1309	1311	1327	1363	1380	1432
1439	1461	1466	1468	1543	1689	1703	1748	1765	1792	1828	1851	1910	1951
1971	2038	2067	2077	2124	2147	2182	2189	2192	2195	2311	2316	2406	2425
2438	2478	2486	2510	2516	2551	2556	2627	2686	2763	2791	2799	2857	2891
2898	3064	3086	3136	3155	3202	3387	3395	3397	3407	3528	3539	3543	3566
3620	3624	3707	3769	3800	3824	3878	3987	4038	4105	4129	4146	4176	4200
Nomination_cool_<ID>													
3	23	89	97	122	141	148	161	201	222	235	277	412	416
456	526	543	565	578	585	628	662	699	724	790	815	821	845
875	879	958	995	1030	1234	1294	1324	1373	1391	1393	1397	1398	1452
1453	1500	1501	1586	1607	1628	1666	1676	1680	1733	1766	1770	1772	1854
1895	1896	1929	1954	2028	2045	2059	2083	2109	2171	2208	2238	2254	2270
2275	2335	2386	2453	2529	2535	2544	2592	2608	2654	2671	2740	2764	2828
2859	2895	2901	2966	3035	3040	3100	3103	3113	3172	3209	3254	3321	3354
3374	3400	3409	3420	3421	3458	3522	3595	3656	3681	3770	3771	3779	3791
3830	3885	3908	3909	3929	3947	3986	4031	4043	4067	4071	4084	4115	4120
4173	4187	4192											
Nomination_mild_<ID>													
2	18	39	47	92	173	241	246	305	521	630	686	711	749
928	952	1011	1016	1187	1203	1281	1307	1344	1386	1422	1459	1472	1481
1559	1646	1743	1840	1853	1961	1964	2044	2157	2495	2531	2539	2636	2897
2950	3124	3151	3182	3197	3230	3419	3480	3524	3552	3661	3701	3734	3766
3848	3890	4170											
Nomination_warm_<ID>													
120	136	244	306	441	540	684	786	844	880	916	1062	1080	1215
1521	1523	1556	1656	1747	1803	1812	1975	2060	2140	2221	2259	2276	2314
2356	2399	2584	2646	2648	2689	2718	2824	2988	3048	3050	3164	3235	3331
3341	3501	3718	3742	3749	3784	3956	3962	4160	4168				

References

Burgschweiger J, Gnädig B, Steinbach MC (2009) Nonlinear programming techniques for operative planning in large drinking water networks. Open Appl Math J 3:14–28

Byrd RH, Nocedal J, Waltz RA (2006) KNITRO: an integrated package for nonlinear optimization. In: Di Pillo G, Roma M (eds) Large scale nonlinear optimization. Springer, New York, pp 35–59

Drud AS (1996) CONOPT: a system for large scale nonlinear optimization, reference manual for CONOPT subroutine library. Technical Report, ARKI Consulting and Development A/S, Bagsvaerd

Elad M (2010) Sparse and redundant representations: from theory to applications in signal and image processing. Springer, New York

Fügenschuh A, Geißler B, Gollmer R, Hayn C, Henrion R, Hiller B, Humpola J, Koch T, Lehmann T, Martin A, Mirkov R, Morsi A, Rövekamp J, Schewe L, Schmidt M, Schultz R, Schwarz R, Schweiger J, Stangl C, Steinbach MC, Willert BM (2014) Mathematical optimization for challenging network planning problems in unbundled liberalized gas markets. Energy Syst 5:449–473

Gill PE, Murray W, Saunders MA (2002) SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM J Optim 12:979–1006

Humpola J, Joormann I, Oucherif D, Pfetsch ME, Schewe L, Schmidt M, Schwarz R (2015) GasLib—a library of gas network instances. http://www.optimization-online.org/DB_HTML/2015/11/5216.html

Joormann I, Schmidt M, Steinbach MC, Willert BM (2015) What does "feasible" mean? In: Koch T, Hiller B, Pfetsch ME, Schewe L (eds) Evaluating gas network capacities, chapter 11., SIAM-MOS series on optimization, SIAM, Philadelphia, pp 211–232

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

Králik J, Stiegler P, Vostrý Z, Záworka J (1988) Dynamic modeling of large-scale networks with application to gas distribution, vol 6., studies in automation and control. Elsevier Science Publishers, New York

LaMaTTO++: a framework for modeling and solving mixed-integer nonlinear programming problems on networks (2015) http://www.mso.math.fau.de/edom/projects/lamatto.html

LIWACOM Informations GmbH and SIMONE Research Group s.r.o. (2004) Gleichungen und Methoden, Benutzerhandbuch

LIWACOM Informations GmbH and SIMONE Research Group s.r.o. (2009) SIMONE API Interface documentation

Martin A, Geißler B, Hayn C, Hiller B, Humpola J, Koch T, Lehmann T, Morsi A, Pfetsch M, Schewe L, Schmidt M, Schultz R, Schwarz R, Schweiger J, Steinbach MC, Willert BM (2011) Optimierung Technischer Kapazitäten in Gasnetzen. In: Optimierung in der Energiewirtschaft, vol. 2157 of VDI-Berichte, pp 105–114

Murtagh BA, Saunders MA (1993) Minos 5.4 user's guide. Technical Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford

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 (2015) Validation of nominations in gas network optimization: models, methods, and solutions. Optim Methods Softw 30:15–53

Rosenthal RE (2008) GAMS—a user's guide. GAMS Development Corporation

Schmidt M, Steinbach MC, Willert BM (2014) High detail stationary optimization models for gas networks. Optim Eng 16:131–164

Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106:25–57

Yin W, Zhang Y (2008) Extracting salient features from less data via ℓ_1-minimization. SIAG/OPT Views-and-News 19:11–19

Záworka J (1993) Project SIMONE—achievements and running development. In: Proceedings of 2nd international workshop SIMONE on innovative approaches to modeling and optimal control of large scale pipeline networks. Prague, pp 1–24