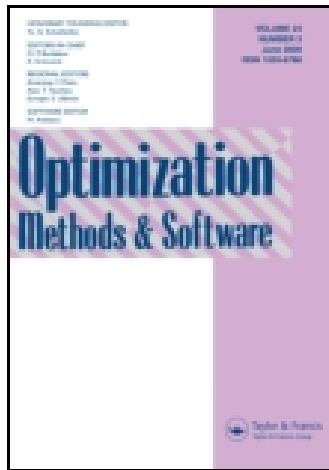


This article was downloaded by: [148.234.29.139]

On: 19 December 2014, At: 09:03

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



[Click for updates](#)

Optimization Methods and Software

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/goms20>

Validation of nominations in gas network optimization: models, methods, and solutions

Marc E. Pfetsch^a, Armin Fügenschuh^b, Björn Geißler^d, Nina Geißler^e, Ralf Gollmer^f, Benjamin Hiller^c, Jesco Humpola^c, Thorsten Koch^c, Thomas Lehmann^c, Alexander Martin^d, Antonio Morsi^d, Jessica Rövekamp^e, Lars Schewe^d, Martin Schmidt^g, Rüdiger Schultz^f, Robert Schwarz^c, Jonas Schweiger^c, Claudia Stangl^f, Marc C. Steinbach^g, Stefan Vigerske^c & Bernhard M. Willert^g

^a Department of Mathematics, Technische Universität Darmstadt, Darmstadt, Germany

^b Department of Mechanical Engineering, Helmut-Schmidt-Universität, Hamburg, Germany

^c Zuse Institute Berlin, Berlin, Germany

^d Department of Mathematics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany

^e Open Grid Europe GmbH, Essen, Germany

^f Department of Mathematics, Universität Duisburg-Essen, Duisburg, Germany

^g Institute of Applied Mathematics, Leibniz Universität Hannover, Hannover, Germany

Accepted author version posted online: 06 Feb 2014. Published online: 17 Mar 2014.

To cite this article: Marc E. Pfetsch, Armin Fügenschuh, Björn Geißler, Nina Geißler, Ralf Gollmer, Benjamin Hiller, Jesco Humpola, Thorsten Koch, Thomas Lehmann, Alexander Martin, Antonio Morsi, Jessica Rövekamp, Lars Schewe, Martin Schmidt, Rüdiger Schultz, Robert Schwarz, Jonas Schweiger, Claudia Stangl, Marc C. Steinbach, Stefan Vigerske & Bernhard M. Willert (2015) Validation of nominations in gas network optimization: models, methods, and solutions, Optimization Methods and Software, 30:1, 15-53, DOI: [10.1080/10556788.2014.888426](https://doi.org/10.1080/10556788.2014.888426)

To link to this article: <http://dx.doi.org/10.1080/10556788.2014.888426>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

Validation of nominations in gas network optimization: models, methods, and solutions

Marc E. Pfetsch^{a*}, Armin Fügenschuh^b, Björn Geißler^d, Nina Geißler^e, Ralf Gollmer^f, Benjamin Hiller^c, Jesco Humpola^c, Thorsten Koch^c, Thomas Lehmann^c, Alexander Martin^d, Antonio Morsi^d, Jessica Rövekamp^e, Lars Schewe^d, Martin Schmidt^g, Rüdiger Schultz^f, Robert Schwarz^e, Jonas Schweiger^c, Claudia Stangl^f, Marc C. Steinbach^g, Stefan Vigerske^c and Bernhard M. Willert^g

^aDepartment of Mathematics, Technische Universität Darmstadt, Darmstadt, Germany; ^bDepartment of Mechanical Engineering, Helmut-Schmidt-Universität, Hamburg, Germany; ^cZuse Institute Berlin, Berlin, Germany; ^dDepartment of Mathematics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany; ^eOpen Grid Europe GmbH, Essen, Germany; ^fDepartment of Mathematics, Universität Duisburg-Essen, Duisburg, Germany; ^gInstitute of Applied Mathematics, Leibniz Universität Hannover, Hannover, Germany

(Received 16 November 2012; accepted 9 January 2014)

In this article, we investigate methods to solve a fundamental task in gas transportation, namely the *validation of nomination problem*: given a gas transmission network consisting of passive pipelines and active, controllable elements and given an amount of gas at every entry and exit point of the network, find operational settings for all active elements such that there exists a network state meeting all physical, technical, and legal constraints. We describe a two-stage approach to solve the resulting complex and numerically difficult non-convex mixed-integer nonlinear feasibility problem. The first phase consists of four distinct algorithms applying mixed-integer linear, mixed-integer nonlinear, nonlinear, and methods for complementarity constraints to compute possible settings for the discrete decisions. The second phase employs a precise continuous nonlinear programming model of the gas network. Using this setup, we are able to compute high-quality solutions to real-world industrial instances that are significantly larger than networks that have appeared in the mathematical programming literature before.

Keywords: gas network optimization; nomination; mixed-integer nonlinear programming; gas transport optimization

1. Introduction

Natural gas is an energy resource of paramount importance for human society. In Europe it accounts for about 25% of the primary energy consumption and is distributed through a pipeline network with a total length of more than 100,000 km. The reliable and efficient operation of these pipeline networks is a permanent challenge asking for computer-based automated decision support.

The European legislative framework for gas transportation has undergone significant changes in the recent past. By European Union regulation [24], gas trading and transportation now have to

*Corresponding author. Email: pfetsch@opt.tu-darmstadt.de

be unbundled, i.e. performed by independent companies. Formerly, an integrated gas company, possibly owning the pipelines, storages, and consuming power stations, could conclude long-term supplier contracts and then operate its network in a dedicated point-to-point fashion. Today, gas trading takes place at virtual, liberalized, and non-discriminatory markets, with consumers, suppliers, storage, and transportation handled by different independent players. The network itself is owned and operated by so-called *gas transmission system operators* (TSO), who, on behalf of gas trading companies, have to ensure the delivery of the traded gas from suppliers to consumers.

The market participants can obtain rights (contracts) from a TSO to supply to or consume gas from the network at given entries or exits. The TSO decides on the total capacity of the rights sold for any particular entry or exit. Note that the rights for entries and exits are sold independently from each other.

For any group of entries and exits the owners of these rights can *nominate* any balanced (in fact, unbalanced nominations are allowed, but they have to be turned into balanced ones by using regulating energy, which incurs additional cost) amount of gas up to the contractual limit to be transported. When selling the rights to nominate at single points, the TSO already warranted that they are capable of fulfilling any such combined transportation request. The amount of nominated gas of all entry and exit points of the network constitutes a *nomination*, i.e. the total amount of gas that has to be transported at a given time. It further includes requirements on the gas composition and pressure bounds. For more details of the German legislation, see [11].

In the present article, we investigate a fundamental problem in gas transportation, namely the *validation of nomination problem* (NoVa): given a gas transmission network consisting of passive pipelines and active, controllable elements such as valves and compressors, and given a nomination (an amount of gas) at every entry and exit point, the task is to find an operational setting of the active elements of this network such that there exists a network state meeting all physical, technical, and legal constraints.

In practice, gas networks are operated/dispatched in a transient manner, i.e. over a continuous time horizon. A transient model of a gas network, thus, would require as input both initial states and future nomination profiles in continuous time. Since we consider mid- to long-term planning, both are unknown. Thus, in the present paper, we exclusively deal with stationary (steady-state) gas transportation.

Of course, a gas transportation operator must avoid granting rights of nominations that are legally correct but technically infeasible. This is closely related to NoVa. Since it is clearly desirable to grant as many nomination rights as possible, it is important to detect nominations that cannot be handled by the network.

Altogether, NoVa leads to a complex non-convex mixed-integer nonlinear feasibility problem, for which new solution approaches are proposed in the present paper.

The state of the art in practice is the verification of a given nomination by using simulation tools, i.e. an experienced planner tries to find suitable settings of the active elements by hand and uses a gas network simulator to test the outcome. With the models described in the present paper, we aim at automatizing this procedure. Furthermore, if the planner fails to find a suitable setting, it does not prove its non-existence. This is different for some of the models presented here, which can be solved to global optimality, thus proving infeasibility in case the solver is not able to find any feasible solution.

1.1 Outline of the paper

Since NoVa involves discrete decisions as well as nonlinearities, it is typically very hard to solve larger (real-world) instances. We propose a sequential approach, which is based on the

physical model presented in Section 2. In Section 3, we present four methods that aim at finding good discrete decisions, while approximating the nonlinearities. Then, we use these decisions (and corresponding network states) as input for a nonlinear optimization model that includes a detailed physical model (see Section 3.5). In this way, we can obtain high-quality solutions.

In Sections 4 and 5, we present an extensive computational study. In Section 4, we first evaluate each of the four approaches via computations on real-world instances and discuss their different features. It turns out that each approach has its strengths in different areas. In Section 5, we discuss the combination of these four approaches yielding a fairly successful solver for the NoVa problem. To the best of our knowledge, the successful solution of gas transportation problems of this size and complexity has never been reported in the literature, so far.

1.2 Related literature and our contributions

In this section, we will briefly highlight some of the related literature. For a recent survey on gas network optimization in general, we refer to [67].

Variants of NoVa, mostly for minimizing operating costs, have been studied by many researchers. Early approaches were often based on dynamic programming (DP). In [87], DP was applied to steady-state gas network optimization in cases where the underlying gas network consists of a single straight line; later, branched network structures were considered in [91]. A similar approach was described in [47], before Borráz-Sánchez and Ríos-Mercado [10] and Gilmour *et al.* [36] studied networks containing branches and loops of arbitrary size. A detailed overview on DP approaches to gas network optimization can be found in [16].

Since the NoVa problem contains nonlinearities, nonlinear programming (NLP) techniques were applied, too. In [43,46,82], subgradient methods were used to tackle the problem. Sequential linear programming techniques were used in [86]. Sequential quadratic programming techniques were applied in [23,30]. The problem was also studied with respect to primal-dual interior point methods [76]. In [77], locally linearized mixed-integer nonlinear programs (MINLPs) in combination with a receding horizon technique are used. Interval analysis techniques were used in [9] to minimize operation costs for instances without discrete decisions on the basis of the Belgian network. Moreover, mathematical program with equilibrium constraints (MPEC) techniques have been used by Baumrucker and Biegler [6].

Several papers deal with the extension or dimensioning of existing gas transportation networks using NLP/MINLP techniques. They implicitly handle NoVa problems via (simplified) gas transportation models, see, e.g., [2,3,39,85,90].

Mixed-integer linear programming (MILP) methods have already been successfully applied to steady-state optimization of gas networks in [55,57] and more recently also to the transient case [27,32,33,52,58]. A combination of both, MILP and NLP techniques, was suggested in [21].

Other techniques to tackle transient gas network optimization problems include simulated annealing [51,88], genetic algorithms [48], ant colony optimization [19], and hierarchical system theory [63]. The operating cost minimization problem can also be formulated as a non-cooperative game, where compressors and entries are the players, and communication is established through the network connectivity constraints. The solution is then given as a Nash-equilibrium found by an iterative algorithm, see [65,66].

The idea of variable elimination in models for flows adhering to Kirchhoff's first and second laws, which we elaborate in Section 3.3, can be traced back at least to [38] and was picked up repeatedly later on (see [53,68] and the monograph [62]). In our approach, this variable elimination is just a building block in a concerted feasibility testing in large meshed gas distribution networks with compressors, resistors, and control valves, i.e. elements whose control involves combinatorial decisions.

Unlike the existing literature, our approach focuses on discrete decisions and provides a higher level of (practical) detail. The MILP approach, which we present in Section 3.1, builds on our own developments and extends it toward solutions that can successfully be validated with a sophisticated gas network simulation. The spatial branching (SB) approach of Section 3.2 applies outer approximation techniques and adapts them to gas networks; the approach is based on a new approximation of compressors (compressor groups). The reduced NLP (REDNLP) approach of Section 3.3 provides a novel transshipment heuristic to find good discrete decisions. The MPEC approach applies a new two-stage solution technique. Finally, the combination of these four approaches with a validation by a detailed NLP provides to the best of our knowledge the best solution technique to date.

2. Detailed physical model

In this section, we describe a detailed physical and technical model of the gas transportation network and its components.

We model a gas network as a directed graph $G = (V, A)$ with nodes representing junctions of network elements. The arcs represent pipes, compressor groups, valves, control valves, and resistors, denoted by $A_{\text{pi}}, A_{\text{cg}}, A_{\text{va}}, A_{\text{cv}}, A_{\text{rs}}$, respectively. The set of arcs can be divided into active ($A_{\text{cg}}, A_{\text{va}}, A_{\text{cv}}$) and passive ($A_{\text{pi}}, A_{\text{rs}}$) ones. Active elements can be controlled directly and have several states of operation. This is not the case for passive arcs. We will treat each type in the presentation below.

For each node u , we introduce a gas pressure variable $p_u \geq 0$. Moreover, $q_a \in \mathbb{R}$ denotes the variable of the mass flow along arc a and describes the mass of gas passing through a given area per unit of time. We use the convention that $q_a > 0$ refers to gas flow in the direction of the arc and $q_a < 0$ indicates that the gas flows in the opposite direction. For each network element $a = (u, v)$, there is a relation between the mass flow q_a and the pressures p_u and p_v , depending on the type of a (see below for details). Due to technical limitations or contractual requirements, we may have pressure bounds $p_u \leq p_u \leq \bar{p}_u$ for each node u and mass flow bounds $\underline{q}_a \leq q_a \leq \bar{q}_a$ for each arc a ; note that \underline{q}_a and \bar{q}_a are allowed to be negative due to the direction of the flow.

We need the following two additional flow quantities: the volumetric flow $Q = q/\rho$ describes the volume of gas passing through a given area per time and depends on the gas density ρ . The normal volumetric flow $Q_0 = q/\rho_0$ is the volumetric flow under normal density ρ_0 , based on normal pressure $p_0 = 101,325$ Pa and normal temperature $T_0 = 273.15$ K.

Pressure, temperature, and density of the gas are coupled by so-called *equations of state* [50,59]. Several models exist—a common choice is the thermodynamical standard equation:

$$\rho = \frac{m}{R} \frac{p}{T z(p, T)}, \quad (1)$$

where R is the universal gas constant. The compressibility factor $z = z(p, T)$ is an approximation for the deviation of real gas from ideal gas. There exists no exact model, but several approximations. We use the formula of the American Gas Association (AGA), see, e.g., [49] in this paper:

$$z(p, T) = 1 + 0.257 \frac{p}{p_c} - 0.533 \frac{p/p_c}{T/T_c}. \quad (2)$$

See Papay [64] for another approximation.

At some nodes gas is supplied into the network, while it is discharged at other nodes. The mass flow $\mathbf{d} = (d_u)_{u \in V}$ arising from a nomination specifies the amount of gas entering ($d_u \geq 0$)

or leaving ($d_u \leq 0$) the network at each node u . At each junction, the gas flow is subject to the mass flow conservation condition

$$\sum_{a \in \delta^+(u)} q_a - \sum_{a \in \delta^-(u)} q_a = d_u \quad \forall u \in V, \quad (3)$$

where $\delta^+(u)$ and $\delta^-(u)$ are the arcs leaving and entering node u , respectively. In this paper, we assume a *homogeneous gas composition*, i.e. we approximate the molar mass m , pseudocritical pressure p_c , pseudocritical temperature T_c , normal density ρ_0 , and compressibility factor under normal conditions $z_0 = z(p_0, T_0)$ by constants. In this case, (3) can equivalently be expressed in Q_0 instead of q .

2.1 Modeling of pipes

Most network elements are pipes, and they are the only elements with a significant length. A pipe $a \in A_{pi}$ is a passive network element, i.e. the gas flow cannot be controlled directly. Instead, it results from the laws of physics, which are described by a system of partial differential equations: the continuity, momentum, and energy equation, see, e.g., [25,50].

As already mentioned, we deal with the stationary case in this paper, which allows to reduce these equations to a set of ordinary differential equations. As a consequence, the continuity equation reduces to the fact that the gas mass flow along a pipe is constant, which we already ensured by introducing a single mass flow variable for each pipe. Moreover, we assume a *constant gas temperature* T throughout. Hence, the energy equation can be neglected. This leaves us with the momentum equation:

$$\frac{\partial p_a}{\partial x} + \frac{q_a^2}{A_a^2} \frac{\partial 1}{\partial x \rho} + g \rho \frac{(h_v - h_u)}{L_a} + \lambda_a(q_a) \frac{|q_a| q_a}{2A_a^2 D_a \rho} = 0, \quad (4)$$

where x denotes the one-dimensional position along pipe a . The parameters L_a , D_a , and A_a specify the length, diameter, and cross-sectional area of a , respectively. Further, h_u and h_v are the normal heights of the nodes u and v , and g is the gravitational acceleration constant. Finally, $\lambda_a(q_a)$ is the friction factor that strongly depends on the vorticity of the flow. For *laminar* flow (i.e. small mass flow rates), we employ the model of Hagen–Poiseuille for the friction factor [26]:

$$\lambda_a^{\text{HP}}(q_a) = \frac{64}{\text{Re}(q_a)}. \quad (5)$$

With $\text{Re}(q_a)$ we denote the Reynolds number, which is a measure of the ratio of inertial forces to viscous forces and is also used to characterize the different flow regimes, i.e. laminar or turbulent flow. The Reynolds number $\text{Re}(q_a)$ is linear in $|q_a|$. For *turbulent* flow (i.e. large mass flow rates) no exact model is known. A highly accurate empirical model is the one of Prandtl–Colebrook [20,69,Chap. 9]:

$$\frac{1}{\sqrt{\lambda_a^{\text{PC}}(q_a)}} = -2 \log_{10} \left(\frac{2.51}{\text{Re}(q_a) \sqrt{\lambda_a^{\text{PC}}(q_a)}} + \frac{k_a}{3.71 D_a} \right), \quad (6)$$

where k_a denotes the (integral) roughness of the pipe.

Due to its complexity, this model cannot be solved directly for large-scale instances. Thus, some aspects of this detailed model are approximated in order to yield more accessible models, as used by the approaches presented in Section 3.

By assuming a constant compressibility factor, a constant mean temperature of gas, and by neglecting the ram pressure term $q_a^2/A_a^2 \partial/\partial x(1/\rho)$, it is possible to derive the following algebraic solution of the momentum equation (4) as an approximation of the behavior of each pipe $a = (u, v) \in A_{\text{pi}}$ [5,50]:

$$p_v^2 = \left(p_u^2 - \Lambda_a(q_a) |q_a| q_a \frac{e^{S_a} - 1}{S_a} \right) e^{-S_a}, \quad (7)$$

with

$$S_a = \frac{2g(h_v - h_u)m}{Rz_m T}, \quad (8)$$

$$\Lambda_a(q_a) = \left(\frac{4}{\pi} \right)^2 \frac{L_a R}{D_a^5 m} z_m T \lambda_a(q_a), \quad (9)$$

$$z_m = z(p_m, T), \quad (10)$$

$$p_m = \frac{2}{3} \left(p_u + p_v - \frac{p_u p_v}{p_u + p_v} \right). \quad (11)$$

The influence of the slope of a pipe is summarized in S_a according to (8).

2.2 Modeling of (control) valves

Valves are active elements used to control the gas flow. A closed valve prevents gas from flowing through this arc, decoupling the pressures at both sides of the valve. If the valve is open, however, no restriction on the flow through the valve applies. Since the physical length of a valve is negligible, we assume the pressures at both ends to be equal in this case. For a valve $a = (u, v) \in A_{\text{va}}$, we introduce a binary variable s_a to obtain

$$\begin{aligned} s_a = 0 &\implies q_a = 0, & p_u, p_v \text{ arbitrary,} \\ s_a = 1 &\implies q_a \text{ arbitrary,} & p_u = p_v. \end{aligned} \quad (12)$$

Control valves extend the model of valves by allowing to reduce the pressure within certain technical limits. They are usually found at the interconnection points of subnetworks with different pressure ranges. For a control valve $a = (u, v) \in A_{\text{cv}}$, we assume that the pressure reduction $\Delta_a = p_u - p_v \geq 0$ lies within $[\underline{\Delta}_a, \overline{\Delta}_a]$. The model of a valve is extended by an additional binary variable s_a^{act} , which determines whether the control valve is actively working or not. We obtain the following model:

$$s_a^{\text{act}} = 1 \implies 0 \leq \underline{\Delta}_a \leq \Delta_a \leq \overline{\Delta}_a, \quad q_a \geq 0, \quad s_a = 1. \quad (13)$$

This allows to model the following three control valve states: ‘closed’ ($s_a = 0, s_a^{\text{act}} = 0$), ‘active’ ($s_a = 1, s_a^{\text{act}} = 1$), and ‘bypass’ ($s_a = 1, s_a^{\text{act}} = 0$).

2.3 Modeling of compressor groups

While the gas pressure decreases along a pipe, groups of compressors allow to increase the pressure to facilitate long-distance gas transmission. Similar to control valves, a compressor group $a = (u, v) \in A_{\text{cg}}$ can take three states ‘closed’, ‘active’, and ‘bypass’. Thus, we add a binary variable s_a together with (12) and a binary variable s_a^{act} that controls whether a compressor group is active, i.e. compressing.

Each compressor group consists of at least one compressor, which increases the pressure of the gas, and additional technical devices, e.g. coolers. The set of *configurations* of a compressor group defines the valid combinations of the included compressors. For instance, if high throughput and moderate pressure increase is required, the compressor units may work in parallel, while in situations with moderate pressure increase some compressors may be deactivated. The configurations introduce an additional discrete aspect to the model of compressor groups. It is straightforward to model this by using further discrete variables.

The modeling of compressors is addressed, for instance, in [15,17,89]. The feasible range of the pressure increase of a single compressor depends on technical parameters and the volumetric flow. For a compressor, we call the set of all tuples (p_u, p_v, Q_a) , such that the compressor can increase the pressure from p_u to p_v for volumetric flow Q_a , its *operation range*. This is a nonlinearly bounded non-convex set, which is described using the so-called characteristic diagram of the compressor; see Figure 3 for an example. The four boundaries are usually approximated by quadratic polynomials that can be used as constraints in mathematical optimization models.

The energy needed to compress a unit of gas from p_u to p_v by a single compressor is described by the specific change in adiabatic enthalpy

$$H_{ad,a}(p_u, p_v) = z(p_u, T)T \frac{R}{m} \frac{\kappa}{\kappa - 1} \left(\left(\frac{p_v}{p_u} \right)^{(\kappa-1)/\kappa} - 1 \right), \quad (14)$$

with isentropic exponent κ , see, e.g., [18]. The power consumption of an active compressor is

$$P_a = \frac{H_{ad,a}}{\eta_{ad,a}} q_a, \quad (15)$$

where $\eta_{ad,a}$ denotes the adiabatic efficiency of a compressor.

2.4 Modeling of operation modes of subnetworks

Often, larger subnetworks of compressor groups, control valves, and valves are located at the intersection of several pipelines. The topology of such a subnetwork can be used to realize many operation modes by appropriately setting the active elements of this subnetwork. In these cases, usually very few of the possible switching combinations correspond to realistic operation modes. Thus, for such subnetworks our model includes additional linear constraints with discrete variables, enforcing so-called *Subnetwork Operation Modes (SOM)*, ensuring that only one of the permitted switching combinations is used: each SOM is encoded by a 0/1-vector $\omega \in \mathbb{R}^A$ that indicates the settings of the corresponding active elements. Let Ω be the set of all such vectors. We enforce $\sum_{\omega \in \Omega} s_\omega = 1$ to pick only one mode and add

$$\sum_{\omega \in \Omega} \omega_a s_\omega \geq s_a \quad \text{and} \quad \sum_{\omega \in \Omega} (1 - \omega_a) s_\omega \geq 1 - s_a \quad (16)$$

for all arcs a in the respective subnetworks to our models, which guarantee that the individual settings s_a have the right values, depending on the chosen mode.

2.5 Modeling of additional pressure losses

Besides the pressure loss resulting from friction of the flow through the pipes, there are some properties and components of the network inducing an additional pressure loss. Such a pressure loss is caused by, e.g. flow diversion and turbulences in shaped pieces, measurement devices,

curved parts within compressor groups, filter systems, reduced radii, and partially closed valves. Neither exact data nor exact models are available for most of these pressure loss effects. We represent them by inserting a virtual *resistor* element at affected locations of the network. Two different types of resistor models are used: either a constant pressure loss in flow direction

$$p_u - p_v = \Delta_a \operatorname{sgn}(q_a), \quad (17)$$

or a pressure loss according to an equation of Darcy–Weisbach type using the parameters drag factor ζ_a and a fictitious diameter D_a :

$$p_u - p_v = \frac{8\zeta_a}{\pi^2 D_a^4} \frac{|q_a| q_a}{\rho_u}. \quad (18)$$

At this point, we have proposed models for each network component, enabling us to formulate the nomination validation problem NoVa as a feasibility problem: given a gas network $G = (V, A)$ as defined above, can one realize a given nomination \mathbf{d} ?

3. Our solution approach

It turns out that even the simplified models as described in Section 2 cannot be solved for real-world instances with state-of-the-art black-box solvers, as we will demonstrate in Section 4.1. In this section, we therefore present an approach to solve NoVa that works in two steps: in the first step, we apply four approaches to handle the discrete decisions necessary to solve NoVa. These models use an approximation for each network component. The solutions provided by each of these approaches are validated in a second step to find accurate solutions for NoVa by fixing the discrete decisions and using the solutions as a starting point for solving a more accurate reference NLP model.

Of course, both the choice of a model and the choice of an approach imply certain tradeoffs regarding physical accuracy and computational tractability. Nevertheless, it is clear that a model that is sufficiently accurate for practical application will feature both discrete decisions and non-convex relationships between pressure and flow, i.e. one has to solve a MINLP problem.

3.1 The MILP approach

In this section, we develop a relaxation of the underlying nonlinear model. These relaxations are modeled in terms of mixed-integer linear constraints. The resulting mathematical programs can then be solved with the aid of a general purpose MILP solver. The core idea behind this approach is to first construct piecewise linear approximations of the nonlinearities together with the respective approximation errors. Then, we construct MILP relaxations via an extension of the so-called incremental model for piecewise linear functions. Thus, we exploit the strength of MILP solvers for NoVa via linearization of the nonlinearities.

We start with the model parts and formulas given in Section 2. Here, we have the basic variables \mathbf{p} , \mathbf{Q}_0 , \mathbf{P} and \mathbf{s} (recall that using \mathbf{Q}_0 is equivalent to using \mathbf{q} under our assumptions). Compared to constraints involving nonlinearities, combinatorial aspects arising from valves, control valves, compressors, or from SOM are fairly straightforward to integrate within an MILP model. In the following, we thus focus on the handling of the nonlinearities arising from pipes and compressors. Resistors are modeled with the same techniques.

To model a pipe $a = (u, v) \in A_{\text{pi}}$, we consider $\text{Re} \rightarrow \infty$ in (6) and obtain the constant friction factor

$$\lambda_a = \left(2 \log_{10} \left(\frac{D_a}{k_a} \right) + 1.138 \right)^{-2} \quad (19)$$

see, e.g., [40,56,60,61]. Moreover, instead of (10) and (11), we assume the following constant (mean) compressibility factor

$$z_m = z \left(\frac{\min \{ \underline{p}_u, \underline{p}_v \} + \max \{ \bar{p}_u, \bar{p}_v \}}{2}, T \right). \quad (20)$$

Under the assumption of a constant compressibility factor, Equation (7) is separable and can hence be expressed by the sum of three nonlinear univariate functions. The power consumption of a compressor (see (14) and (15)) is modeled by a recursive decomposition into three nonlinear univariate functions and one bivariate product. We remark here that from a theoretical point of view this decomposition is not necessary for our approach, but it is very valuable for the application to large-scale instances.

Our approach to integrate the nonlinearities in an MILP model is based on piecewise polyhedral relaxations of the nonlinearities, i.e. the non-convexities arising from nonlinear equality constraints in the MINLP model are transformed into non-convexities expressed by linear constraints and integrality conditions using a set of additional variables.

Since all basic variables (mainly \mathbf{p} and \mathbf{Q}_0) are bounded, we can assume that any occurring nonlinear function $f : D \rightarrow \mathbb{R}$ is defined on a compact set $D \subset \mathbb{R}^d$, typically a box. We triangulate D into simplices S_1, \dots, S_n and approximate f by a piecewise linear interpolation $g_i(x)$ for $x \in S_i$, g_i affine, $i = 1, \dots, n$, over the vertices of the simplices S_1, \dots, S_n . Additionally, we compute upper bounds ϵ_i , $i = 1, \dots, n$, for the approximation error on each simplex and replace each occurrence of f in the constraint set by its piecewise polyhedral outer approximation. The interpolations together with the bounds for the approximation errors are computed by an adaptive refinement algorithm, which makes use of convex underestimators in order to provide reliable error bounds. For a detailed description of the procedure, we refer to [31,33].

Finally, such a piecewise polyhedral outer approximation is modeled in terms of mixed-integer linear constraints by means of a modified version of the so-called incremental method of Markowitz and Manne [54]. The necessary modifications of the method have also been introduced in [31,33]. A crucial assumption for the application of the incremental approach is to order the simplices according to a Hamiltonian path in the dual graph of the triangulation. Simultaneously, an appropriate ordering of the vertices of each simplex must be constructed such that the first vertex of a simplex is equal to the last vertex of the predecessor simplex in the Hamiltonian path. While this is trivial in dimension 1, we refer to [31,33] for more details on how to obtain such an ordering in higher dimension.

The modified version of the incremental model is as follows:

$$x = \hat{x}_0^1 + \sum_{i=1}^n \sum_{j=1}^d (\hat{x}_j^i - \hat{x}_0^i) \delta_j^i, \quad (21)$$

$$y - e = \hat{y}_0^1 + \sum_{i=1}^n \sum_{j=1}^d (\hat{y}_j^i - \hat{y}_0^i) \delta_j^i, \quad (22)$$

$$-\epsilon_1 - \sum_{i=1}^{n-1} \omega_i (\epsilon_{i+1} - \epsilon_i) \leq e \leq \epsilon_1 + \sum_{i=1}^{n-1} \omega_i (\epsilon_{i+1} - \epsilon_i), \quad (23)$$

$$\sum_{j=1}^d \delta_j^i \leq 1, \quad 1 \leq i \leq n, \quad (24)$$

$$\delta_j^i \geq 0, \quad 1 \leq i \leq n, \quad 1 \leq j \leq d, \quad (25)$$

$$\sum_{j=1}^d \delta_j^{i+1} \leq \omega_i \leq \delta_d^i, \quad 1 \leq i \leq n-1, \quad (26)$$

$$\omega_i \in \{0, 1\}, \quad 1 \leq i \leq n-1. \quad (27)$$

Here, y denotes an approximation to $f(x)$ within the piecewise polyhedral outer approximation of the graph of f . The j th vertex of the i th simplex S_i of the triangulation is denoted by \hat{x}_j^i and $\hat{y}_j^i := f(\hat{x}_j^i)$. Constraints (23) ensure the error tolerance e over each simplex. The binary variables ω_i , $i = 1, \dots, n-1$, are used within Constraints (26) and (27) to express that if for the i th simplex some variable $\delta_j^i > 0$, $j \in \{1, \dots, d\}$, then $\delta_d^k = 1$ for all previous simplices $k = 1, \dots, i-1$ in the Hamiltonian path. The model (21)–(27) is applied to the formulas (7), (14), and (15)—with simplifications as outlined in the beginning of this section to obtain one-dimensional and two-dimensional approximations, respectively. As an example, a piecewise polyhedral relaxation of the function $f(Q_{0,a}) = \alpha_a |Q_{0,a}| Q_{0,a}$ with $D = [-4, 4]$, one of the univariate nonlinearities arising from the simplification of (7), is given in Figure 1. Here the simplices are the intervals specified by the breakpoints. The set of feasible solutions to (21)–(27) is depicted as the union of the parallelograms drawn with dotted lines in Figure 1.

Besides the function $f(Q_{0,a}) = \alpha_a |Q_{0,a}| Q_{0,a}$, the remaining univariate subexpressions arising from Equation (7) are p_u^2 and p_v^2 . We introduce additional variables for these squared pressures. In many cases, however, we can then remove the original pressure variables as they are not needed, e.g. in the trivial case when the only elements incident to a node are pipes. We decide a priori whether it is necessary to introduce an approximation of p_u^2 for a node u in an optimal way using an auxiliary integer linear program (ILP). The variables of the ILP express whether a pressure variable, a squared pressure variable, or both variables are needed. The constraints then enforce these requirements. For example, for both endpoints of a pipe, squared pressure variables are needed, and similarly pressure variables are required at both endpoints of a compressor group.

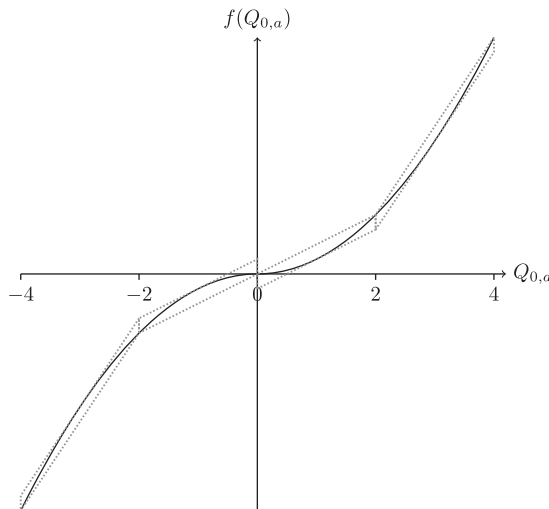


Figure 1. MILP relaxation of $f(Q_{0,a}) = \alpha_a |Q_{0,a}| Q_{0,a}$ with $\alpha_a = 0.25$ on $[-4, 4]$ with breakpoints $-4, -2, 0, 2, 4$.

For a MILP formulation of a valve, however, homogeneous variables at both endpoints are sufficient. The objective of the auxiliary ILP is then to minimize the sum of variables that express that both types are required and hence to minimize the number of nonlinearities for the coupling of pressure to squared pressure variables.

Characteristic diagrams of compressors are not modeled explicitly. We enforce any active compressor to operate at least within a convex relaxation of its operation range. This is achieved by dynamically cutting off solutions via the separation of tangential hyperplanes to this convex relaxation of the characteristic diagrams [45].

Moreover, in order to improve the chance that the corresponding MILP solution yields a feasible solution for more accurate NLP models (see Section 3.5), we compute a centroid point $(Q_{0,a}^*, \varepsilon_a^*)$ of the characteristic diagram, where ε_a^* is a ratio between inlet and outlet pressure. We then try to obtain a feasible solution by minimizing the maximum norm distance to the centroids of the characteristic diagrams of all active compressors. To this end, we introduce slack variables σ_a^+ , σ_a^- and add relations $Q_{0,a} - Q_{0,a}^* \leq M(1 - s_a) + \sigma_a^+$ and $Q_{0,a} - Q_{0,a}^* \geq M(1 - s_a) - \sigma_a^-$, where M is a suitably large constant; thus, these constraints are active iff the compressor is active, i.e. $s_a = 1$. Similar constraints are added for the ratios of the pressures. The objective is then to minimize the maximum of the σ 's. We remark that although operation ranges of compressors are typically not convex, the centroids of all the encountered characteristic diagrams are clearly feasible points. To choose among the different configurations of each compressor group, we incorporate special ordered set type-1 constraints on the respective binary decision variables.

3.2 The SB approach

The second approach solves an approximated version of NoVa with an outer approximation method. We use a tailored version of the branch-and-cut constraint integer programming framework SCIP [1,74,81], which implements a combination of convex (linear) outer approximation and SB to prove global optimality. This method is also used by state-of-the-art solvers for generic MINLP problems, see, e.g., [7,75,78–80].

Let us give a brief review about the concept of outer approximation and SB. Let $S \subseteq \mathbb{R}^n$ be the (non-convex) feasible set of the problem. A linear approximation of the feasible set is computed such that

$$S \subseteq \{x \mid Dx \leq d\}$$

for suitable $D \in \mathbb{Q}^{m \times n}$ and $d \in \mathbb{Q}^m$. This relaxation is used during the branch-and-bound algorithm and is successively refined by additional cutting planes. Branching on integer variables deals with the integrality requirements. When all integer variables take integral values, the relaxation is strengthened by recursive *spatial branching*, i.e. branching on continuous variables appearing in nonlinear terms. SB on variable i of the solution x^* to the linear relaxation refers to subdividing the previous linear relaxation into two parts

$$S \subseteq \{x \mid Dx \leq d, x_i \leq x_i^*\} \cup \{x \mid Dx \leq d, x_i \geq x_i^*\}.$$

For each part of the relaxation, a subproblem is created, and a tighter relaxation can be computed due to tighter variable bounds, see Figure 2 for an example. SB thus improves the convex relaxation, in particular, at places where the describing functions are non-convex. Branching is pursued until all integral variables take integral values and the convex relaxation is ‘close enough’ to the feasible region. This way, global bounds on the objective function can be computed and the problem can be solved to global optimality.

To apply this approach to NoVa, let us first consider the modeling of pipes. We use (7) and (8) formulated in norm volumetric flow Q_0 and apply the modifications (19) and (20). We introduce

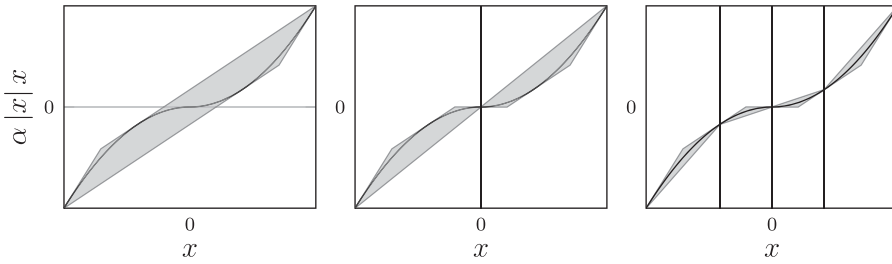


Figure 2. Successive improvement of the approximation of $x \mapsto \alpha |x|x$ by SB.

squared pressure variables $\pi_u := p_u^2$ for all nodes $u \in V$. Recall from Section 2 that all pressure values are non-negative (the pressure in the system is never smaller than the atmospheric pressure). Formulating (7) using the pressure square variables yields

$$\pi_u - \beta_a \pi_v = \alpha_a |Q_{0,a}| Q_{0,a}, \quad (28)$$

for suitable constants α_a and $\beta_a > 0$, which depend on the characteristics of the pipe, see (8), (19), and (20). To separate the nonlinear part, we introduce an additional variable y_a and obtain

$$\pi_u - \beta_a \pi_v = y_a \quad (29)$$

$$y_a = \alpha_a |Q_{0,a}| Q_{0,a}. \quad (30)$$

While (29) is linear, (30) is a nonlinear equation and thus a non-convex constraint. The nonlinear function is handled by linear outer approximation through a special SCIP plugin for this type of function performing separation of the convex hull and SB [28,81]. Figure 2 visualizes the treatment of the nonlinear function $x \mapsto \alpha |x|x$ by this approach.

Besides the nonlinear pressure loss equations for the pipelines, compressors are another source of nonlinear behavior. As mentioned in Section 2.3, the capability of a compressor is described by its characteristic diagram, which is represented by the set of feasible combinations of volumetric flow Q_a and adiabatic head $H_{ad,a}$ (Figure 3, left). These points are obtained by physical measurements on the compressor with different pressures and flow rates. However, these quantities cannot easily be computed in our model. But for a compressor $a = (u, v)$, they can approximately be mapped to the space $(p_u, p_v, Q_{0,a})$ of inlet and outlet pressure and normal flow through the compressor, respectively, via

$$\begin{pmatrix} p_u \\ p_v \\ Q_{0,a} \end{pmatrix} = p_u \begin{pmatrix} 1 \\ \left(\frac{H_{ad,a}}{c_2} + 1 \right)^{\kappa/(\kappa-1)} \\ \frac{Q_a}{c_1} \end{pmatrix}, \quad (31)$$

where c_1 and c_2 are constants (see (14) and (1), (2), (20)). Each point specifying the characteristic diagram can thus be translated into a ray of feasible combinations $\{(p_u, p_v, Q_{0,a}) \mid p_u \geq 0\}$ for the compressor. The convex hull of all these rays, which is a cone, approximates the operation range of the compressor. We further intersect the cone with

$$\{(p_u, p_v, Q_{0,a}) \mid \underline{p}_u \leq p_u \leq p_v \leq \bar{p}_v, \underline{Q}_{0,a} \leq Q_{0,a} \leq \bar{Q}_{0,a}\},$$

corresponding to the technical limitations of the compressor. This gives a bounded convex polyhedron. Figure 3 visualizes the different steps in this approximation procedure.

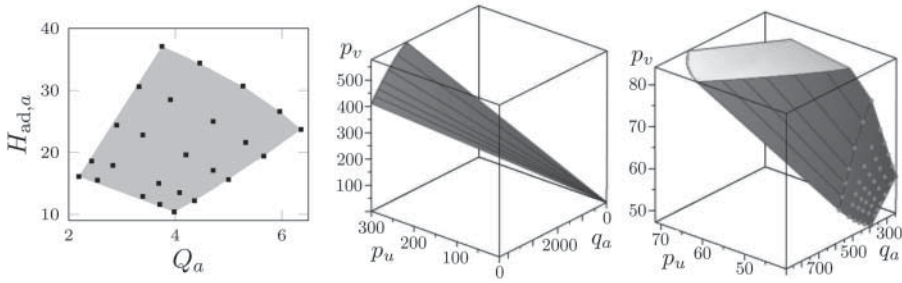


Figure 3. From the characteristic diagram of a compressor (left) over the (unbounded) cone (middle) to the operation range polytope (right).

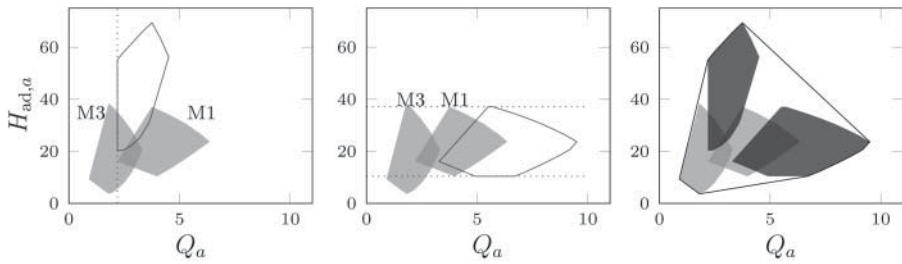


Figure 4. Resulting operation ranges for different combinations of two machines, M1 and M3: serial compression (first M3 then M1) (left); parallel compression (middle), and convex hull of all feasible points (right).

For compressor groups that consist of a single compressor machine, the approach outlined above already yields our approximation. The configurations of larger compressor groups are handled as follows. We do not model internal valves of a compressor group explicitly, but compute the operation ranges of the valid parallel and serial configurations (see Section 2.3). In Figure 4, we give an example for two machines. Then as a global approximation we use the convex hull of the union of all these resulting operation ranges, which can be thought of as a huge virtual compressor machine (see the right image in Figure 4). In our model, we embed its linear description as constraints. Technically, this leads to the computation of a projection of higher dimensional polytopes into the three-dimensional space, which is done by Fourier–Motzkin elimination, and numerically carried out by the Parma Polyhedra Library [4] during the setup phase of the model.

Note that the operation range polytope uses pressure variables, not squared pressure variables. Hence, explicit coupling constraints $\pi_u = p_u^2$ and $\pi_v = p_v^2$ are needed upstream and downstream of each compressor group. Algorithmically, we treat these constraints in the same way as (30).

Valves, control valves, and resistors are modeled as described in Section 2. SOMs are handled as sketched in Section 2.4. Moreover, our approach does not involve an objective function. We therefore stop the solution process as soon as a feasible solution is found and proceed with the detailed validation (see Section 3.5).

3.3 The REDNLP approach

The approach presented in this section relies on transforming nonlinearities into a more accessible form, reducing the problem dimension of the underlying NLP. This transformation approach is embedded into a heuristic procedure for finding promising switching decisions. The system of

(linear) flow conservation (3) and (nonlinear) pipe equation (7) is transformed into an equivalent nonlinear system, where most flow and pressure variables get eliminated, because they are explicit functions of a relatively small group of variables, consisting of one flow variable per fundamental network cycle and two pressure variables per active arc. Apart from the explicit formulae for flow and pressure variables, the transformed system contains implicit equations whose number equals the number of fundamental cycles of the network and whose unknowns are just the variables from the mentioned group. The approach aims at checking feasibility for a set of switching states of active elements, either predefined or resulting from a transshipment heuristic.

3.3.1 Transformation of nonlinearities

For a given setting of all the active elements (*switching state*) of the network, the directed graph $\bar{G} = (V, \bar{A}) \subset G$ models the relevant network, where \bar{A} denotes the set of all arcs not being in closed state.

Let \mathcal{A}^+ denote the node–arc incidence matrix and \mathcal{A} be the submatrix of full row rank arising from the deletion of one row, corresponding to a preselected, pipe-incident root node $\bar{u} \in V$ with pressure variable $p_{\bar{u}}$.

Let $\boldsymbol{\pi}$ and $|\mathcal{Q}_0| \mathcal{Q}_0$ denote the vectors with components $\pi_u, u \in V \setminus \bar{u}$, and $|\mathcal{Q}_{0,a}| \mathcal{Q}_{0,a}, a \in \bar{A}$. For $a = (u, v) \in A_{cg} \cup A_{cv}$, i.e. for compressor groups and control valves, π_u and π_v denote squared outgoing and ingoing pressures and $\Delta_a := \pi_v - \pi_u$. For $a \notin A_{cg} \cup A_{cv}$, we define $\Delta_a = 0$. The diagonal matrix $\alpha = \text{diag}(\Delta_a)$ has entries which either are the pipe specific values $\Lambda_a, a \in A_{pi}$, as defined in (9), or 0, otherwise. The vector of all ones is denoted by $\mathbb{1}$.

The equations for flow conservation at the nodes and pressure drop on the arcs can after some calculations be written as

$$\mathcal{A} \mathcal{Q}_0 = \mathbf{d}, \quad (32)$$

$$\boldsymbol{\Delta} - \mathcal{A}^T \boldsymbol{\pi} - \alpha |\mathcal{Q}_0| \mathcal{Q}_0 = -\pi_{\bar{u}} \mathcal{A}^T \mathbb{1}, \quad (33)$$

where (33) is built from the pressure loss equation (7) for pipes by neglecting geodetic heights and adding the vector $\boldsymbol{\Delta}$ for pressure changes in compressor groups or control valves. The compressibility factor z is approximated by (2). Splitting $\mathcal{A} = (\mathcal{A}_B, \mathcal{A}_N)$ into basis and non-basis parts according to a spanning tree, (32) and (33) are equivalent to

$$\mathcal{Q}_{0,B} = \mathcal{A}_B^{-1} \mathbf{d} - \mathcal{A}_B^{-1} \mathcal{A}_N \mathcal{Q}_{0,N}, \quad (34)$$

$$\boldsymbol{\pi} = \pi_{\bar{u}} \mathbb{1} - (\mathcal{A}_B^T)^{-1} (\alpha_B |\mathcal{Q}_{0,B}| \mathcal{Q}_{0,B} - \boldsymbol{\Delta}_B), \quad (35)$$

$$\alpha_N |\mathcal{Q}_{0,N}| \mathcal{Q}_{0,N} - \boldsymbol{\Delta}_N = \mathcal{A}_N^T (\mathcal{A}_B^T)^{-1} (\alpha_B |\mathcal{Q}_{0,B}| \mathcal{Q}_{0,B} - \boldsymbol{\Delta}_B). \quad (36)$$

Indeed, inserting (34) into (33) yields

$$\boldsymbol{\Delta}_B - \mathcal{A}_B^T \boldsymbol{\pi} - \alpha_B |\mathcal{Q}_{0,B}| \mathcal{Q}_{0,B} = -\pi_{\bar{u}} \mathcal{A}_B^T \mathbb{1}, \quad (37)$$

$$\boldsymbol{\Delta}_N - \mathcal{A}_N^T \boldsymbol{\pi} - \alpha_N |\mathcal{Q}_{0,N}| \mathcal{Q}_{0,N} = -\pi_{\bar{u}} \mathcal{A}_N^T \mathbb{1}. \quad (38)$$

Multiplying (37) with $-(\mathcal{A}_B^T)^{-1}$ implies (35), and inserting (35) into (38) yields (36).

Let us add some remarks on how these formulae relate to network issues:

- The arcs corresponding to the columns of \mathcal{A}_B form a spanning tree $T_B \in \bar{G}$.
- Each arc a belonging to a column of \mathcal{A}_N stands for a fundamental cycle with respect to T_B , i.e. the unique cycle in $T_B \cup a$.

- The columns of \mathcal{A}_B^{-1} mark the directed paths in the spanning tree from the root node to all other nodes, with entry +1 (−1) if the arc is directed in the same (opposite) way as the path from the root, and with 0 if the arc does not belong to the path.
- The columns of $\mathcal{A}_N^T(\mathcal{A}_B^T)^{-1}$ mark the tree arcs in fundamental cycles. The column corresponding to arc a is the difference of the columns of the paths from the root to the head of a and that to its tail.
- The representation of π in (35) states that the (squared) pressure at an arbitrary node $u \neq \bar{u}$ is determined by the pressure at the root minus the pressure drop along the unique path from \bar{u} to u and plus/minus the pressure changes by the active arcs along the path.
- The identity (36) states that the pressure change along the tree arcs of each fundamental cycle must equal the change along the arc that created the cycle.

Compared to (32) and (33), one ends up with a much smaller implicit part in the transformed system, namely (36). It has just as many equations as there are components of $\mathbf{Q}_{0,N}$ or as there are fundamental cycles in \bar{G} . For strongly meshed gas distribution networks, such as those met in many parts of Europe, this number still can be substantial. For weakly meshed gas transportation networks, however, values of 1 or 2 already become practically relevant. If the cycles of \bar{G} are arc disjoint, then (36) is separable with respect to the components of $\mathbf{Q}_{0,N}$.

Altogether, the feasibility system we use comprises (36), pressure bounds applied to the right-hand sides of (35), and the following specific conditions for resistors, control valves, and compressor groups.

Control valves allow for a pressure reduction, when traversed in the nominal direction. Compressor groups are modeled at an aggregated level, without resolution to individual machines. We add the following restrictions for $a = (u, v) \in A_{cg}$:

$$p_u \geq p_v, \quad (p_u - p_v) \cdot (-Q_{0,a}) \leq 0, \quad (39)$$

$$(p_u - p_v) \cdot (\underline{Q}_{0,a} - Q_{0,a}) \leq 0, \quad (p_u - p_v) \cdot (Q_{0,a} - \bar{Q}_{0,a}) \leq 0, \quad (40)$$

$$(p_u - p_v) \cdot \left(\underline{\varepsilon}_a - \frac{p_v}{p_u} \right) \leq 0, \quad (p_u - p_v) \cdot \left(\frac{p_v}{p_u} - \bar{\varepsilon}_a \right) \leq 0, \quad (41)$$

$$(p_u - p_v) \cdot (\underline{\Delta p}_a - p_v + p_u) \leq 0, \quad (p_u - p_v) \cdot (p_v + p_u - \bar{\Delta p}_a) \leq 0. \quad (42)$$

Inequalities (39) specify that compressor groups allow for a pressure increase in their active mode when traversed in the nominal direction. In order to approximate the characteristic diagram, lower and upper bounds ($\underline{Q}_{0,a}$ and $\bar{Q}_{0,a}$, respectively) are imposed on the flow for the active mode (see (40)). We impose lower and upper bounds ($\underline{\varepsilon}_a$ and $\bar{\varepsilon}_a$, respectively) on the pressure ratio (see (41)) and on the pressure increase ($\underline{\Delta p}_a$ and $\bar{\Delta p}_a$, see (42)). Resistors are modeled according to their type by the respective equation (17) or (18) in a non-smooth way due to the occurrence of the sign of the flow rate.

3.3.2 Search for promising switching decisions

To enable the approach described above, we have to fix switching states for control valves, valves, and compressor groups. We use two different techniques to fix these binary decisions. First we use a transshipment heuristic, and then we try some sets of given switching states for all active elements. These sets are constructed by using expert knowledge about the network and by collecting sets of switching states from the transshipment heuristic in cases where they led to a feasible solution for some nominations.

3.3.3 The transshipment heuristic

We construct a directed transshipment graph $\tilde{D} = (\tilde{V}, \tilde{A})$ with costs $\tilde{c} \in \mathbb{R}^{|\tilde{A}|}$ and flow requirements $\tilde{d} \in \mathbb{R}^{|\tilde{V}|}$. In the first step, copy the original graph to an auxiliary graph $\hat{D} = (\hat{V}, \hat{A})$. Start with $\tilde{V} = \tilde{A} = \emptyset$ and perform the following steps:

- For all entities being subject to SOM constraints, add one artificial node to \tilde{D} , representing the entity. Moreover, we add all its boundary nodes to \tilde{D} . The boundary nodes are assigned their original inflow or outflow requirements d_u . Assign the sum of the flow requirements of the entity's internal nodes to the new entity node. Add arcs to \tilde{A} between the entity node and its boundary nodes in both directions. All these arcs are assigned the same small artificial cost coefficient. Delete all internal nodes of the entity from \hat{V} and its arcs from \hat{A} .
- For all arcs a from A_{cg}, A_{va}, A_{cv} remaining in \hat{A} , add their endpoints with the original d_u to \tilde{V} . Depending on the signs of $\underline{Q}_{0,a}$ and $\bar{Q}_{0,a}$ add one or two arcs to \tilde{A} between their endpoints. Assign artificial cost coefficients—small ones for compressor groups and for control valves in forward direction and big ones for backward direction arcs (requiring bypass mode) and valves. Delete arc a from \hat{A} .
- For each connected component in \hat{D} , add one artificial node and assign the flow balance of the component's internal nodes to this component node. The boundary nodes of a component are boundary nodes of entities or end points of active arcs outside of entities, which were added to \tilde{V} in the previous steps. Add arcs in both directions to \tilde{A} , connecting the component node and the component's boundary nodes, assign the sum of the friction coefficients α_a within the component as the cost coefficient to these arcs.

The resulting graph \tilde{D} represents the flow situation and guides the usage of arcs in the transshipment by cost coefficients, such that flow through pressure increasing and pressure decoupling network elements is favored. The resulting transshipment model is

$$\begin{aligned}
 & \min \sum_{a \in \tilde{A}} \tilde{c}_a f_a \\
 & \sum_{a \in \delta^+(u)} f_a - \sum_{a \in \delta^-(u)} f_a = \tilde{d}_u \quad \forall u \in \tilde{V} \\
 & f_a \geq 0 \quad \forall a \in \tilde{A}. \quad (43)
 \end{aligned}$$

The moderate size of the transshipment problem allows for a rapid solution by any state-of-the-art linear programming solver. From an optimal solution, switching states for active arcs in the original network are derived. Namely, if no flow passes through an element outside an entity, we choose the off-state for it. Reflecting expert knowledge, for all the elements inside a specified entity, a suitable decision from the corresponding SOM is chosen by a set of rules based on the amount of flow and the flow direction through the entity.

3.4 The MPEC approach

As seen in Section 2, the problem of validation of nominations is a discrete-continuous nonlinear and non-smooth feasibility problem. The approach described here aims at finding feasible solutions by means of NLP techniques. Since these techniques require a continuous and sufficiently smooth model (C^2 in our case), we reformulate the discrete and non-smooth aspects in an appropriate way. For a more detailed explanation of the theory behind this approach, see [70,72]. For a recently published approach that also uses MPEC techniques in gas transport optimization (but for different aspects of the model), see [6].

3.4.1 Smoothing techniques

Many physical and technical aspects of NoVa are non-smooth, such as the pressure loss at pipes or resistors. Most of them can be handled by standard smoothing techniques for min, max, sgn, and absolute value functions (cf. [71]), but others require problem-tailored smoothing techniques. As an example, we present the smoothing of our non-smooth model of the pressure loss in pipes, which is based on the quadratic pressure loss model (7). For the compressibility factor we choose the AGA formula (2), see, e.g., [71]. Obviously, the term $|q_a| q_a$ in (7) has a second-order discontinuity at zero, while the composite friction model $\lambda^{\text{HPPC}}(q_a)$ defined by (5) and (6) has a jump discontinuity at the transition from laminar to turbulent flow (Figure 5, left). Both non-smooth aspects are handled by a global smooth approximation developed in [12,13,71]:

$$\phi(q_a) \approx \lambda^{\text{HPPC}}(q_a) |q_a| q_a. \quad (44)$$

Our choice of $\phi(q_a)$ is asymptotically correct for $|q_a| \rightarrow \infty$ (Figure 5, right).

3.4.2 Complementarity constraints for combinatorial aspects

Active network elements like compressor groups introduce discrete decisions into the validation problem. For a compressor group $a = (u, v) \in A_{\text{cg}}$, these discrete decisions result in states like *active* or *closed* that can be described in a simplified way by

$$\begin{aligned} a \text{ is active/in bypass} &\implies p_v - p_u - \Delta_a = 0, \quad \Delta_a \geq 0, \\ a \text{ is closed} &\implies q_a = 0, \end{aligned} \quad (45)$$

with Δ_a denoting the pressure increase. Standard mixed-integer approaches for modeling (45) use binary variables for the states (cf. Sections 3.1 and 3.2). Here we follow a different approach and replace (45) by the *complementarity constraint*

$$\tilde{p}_a q_a = 0, \quad \Delta_a \geq 0, \quad \tilde{p}_a := p_v - p_u - \Delta_a. \quad (46)$$

The model for the active state in (45) is not sufficiently detailed to be practically relevant. An active group can actually be operated in several configurations $k \in \mathcal{K}_a$ (see Section 2.3). Our heuristic approach attempts to determine the ‘most feasible’ configuration for a given flow–pressure situation (p_u, p_v, q_a) . To this end, we model the feasible sets \mathcal{F}_a^k for all configurations $k \in \mathcal{K}_a$ by a set of smooth nonlinear constraints (see [71] for details). We relax these constraints by applying a set of slack variables S_a^k yielding the relaxed feasible sets $\tilde{\mathcal{F}}_a^k$. Furthermore, we

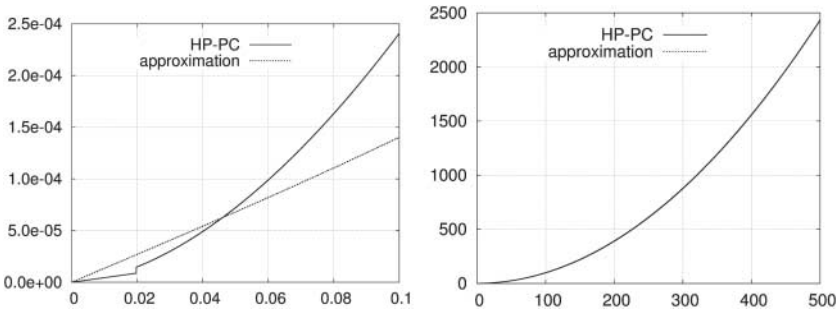


Figure 5. Discontinuous friction model λ^{HPPC} and its smooth approximation ϕ for small flow rates (left) and large flow rates (right).

make the fictitious assumption that the entire gas flow of the compressor group passes through every configuration, i.e. $q_a = q_a^k$ for all $k \in \mathcal{K}_a$. The pressure increase $\Delta_a = p_v - p_u$ is relaxed to a convex combination of pressure increases of the individual configurations, $\Delta_a^k = p_v^k - p_u^k$. In summary, the compressor group model (46) is extended by

$$\begin{aligned} \Delta_a &= \sum_{k \in \mathcal{K}_a} \sigma_a^k \Delta_a^k, & \sum_{k \in \mathcal{K}_a} \sigma_a^k &= 1, \\ \forall k \in \mathcal{K}_a : \sigma_a^k &\geq 0, & q_a &= q_a^k, & (p_u^k, p_v^k, q_a^k, S_a^k) &\in \tilde{\mathcal{F}}_a^k. \end{aligned} \quad (47)$$

To obtain the ‘most feasible’ configuration with the refined model (47), we minimize a suitable norm of the slack variables. Finally, we heuristically choose the active configuration in dependence of the values of the slack variables and the convex coefficients σ_a^k .

Using techniques similar to (46) for the remaining active elements, almost all discrete aspects can be represented by complementarity constraints. In combination with the smoothing techniques, the NoVa problem is thus transformed into an MPEC. We apply standard MPEC regularization schemes as in [29,41,73] to obtain a smooth and continuous NLP reformulation.

3.4.3 A two-stage solution approach

The NLP obtained from the MPEC approach combines highly nonlinear non-convex physical and technical phenomena with numerically problematic smoothing and penalization techniques. Numerical experiments show that solving all these aspects simultaneously on real-world networks is hardly possible with general purpose NLP codes like IPOPT [83] or SNOPT [35]. As a substantially more robust and reliable solution procedure, we propose a two-stage approach, where each stage solves an NLP with a different set of model aspects.

The first stage incorporates all previously described features except for the convexification of compressor groups (47). Thus, it decides the principal states (open or closed) of all active elements.

As mentioned above, various regularization schemes exist for the stage 1 MPEC. In our numerical experiments, the penalization scheme of Fukushima and Pang [29] proved to be the most appropriate choice for the specific class of problems. This scheme introduces an NLP sequence $\text{NLP}(\tau_k)$, $k = 1, 2, \dots$, with decreasing penalization parameter τ_k . While providing convergence theory, the approach suffers from significant practical drawbacks: the entire sequence needs to be solved to optimality, and the computation time is a multiple of a single instance. In our approach, τ is instead handled as an optimization variable that is driven to zero during the NLP solution process by additional constraints or penalization schemes. This *direct* approach is very stable for our practical purposes.

The solution of the first stage is analyzed and translated into discrete decisions. An ambiguous situation arises when one or more complementarity constraints are satisfied bi-actively, with both factors equal to zero. In this case, the most promising decision is chosen heuristically for stage 2, based on empirical experience. After the first stage, all open/closed states are decided and the overall flow situation in the network is determined; these will be fixed in the second stage.

In the second stage, the active configurations of the compressor groups are to be determined. The decisions of stage 1 are fixed, and the convexification model (47) is added. Moreover, we use the solution of stage 1 to initialize the NLP of stage 2.

Then, the solution of the second stage is analyzed to fix the remaining active configurations. If more than one configuration with vanishing slacks exist, the one with the largest convex coefficient is chosen. If no configuration with vanishing slack norm exists, the configuration with the smallest constraint violation is used.

3.5 Validation by NLP

Since the full NoVa problem becomes intractable if both a detailed physics model and discrete decisions are incorporated, each of the four solution approaches employs its specific approximations of certain physical and technical details. This raises the need for a posteriori feasibility checks with respect to some reference model that is trusted to provide a sufficiently accurate description of reality, such as the models used in commercial gas network simulation software.

We use a model that includes the pressure loss equation (7), with a global smooth approximation [13,71] replacing the piecewise friction model (5) and (6); compressor groups are modeled as accurately as possible, complete with drives, with operation ranges of individual units, and with arbitrary distributions of flow among parallel units [71]; only the fuel gas is neglected in the flow balances, and gas temperatures as well as gas quality parameters are considered constant in the entire NLP model; details can be found in [71]. Valves and control valves are modeled as in (12) and (13), and both resistor types of [71] arise.

The difficulty is that solution candidates with approximate physics from any approach will generically be infeasible in a strict sense; we can only expect approximate feasibility. Assuming that the discrete variables of a solution candidate are correct in the sense that suitable ‘small’ modifications of the continuous variables will yield exact feasibility, we proceed as follows to obtain a high-accuracy feasible solution: we fix all discrete decisions and all discrete states based on the given solution candidate. The discrete-continuous problem with a detailed physics model then reduces to a purely continuous feasibility problem consisting of linear and nonlinear equalities and inequalities with suitable smoothness properties (C^2 in our case). We introduce slack variables to relax all the nonlinear constraints. The minimization of some measures of the total constraints violation, specifically a weighted ℓ_1 -Norm of the slacks (with large weights on the slacks of compressor units), then yields a standard NLP. An initial solution estimate for this NLP is generated from the given solution candidate. If we are successful in computing a local NLP minimizer whose slack objective is zero or sufficiently small, we have a complete solution of the original problem, and we will regard the given candidate as a *valid* approximate solution. The final NLP solution can ultimately be verified with a suitable simulation tool.

Note that a different outcome of our NLP validation procedure does not provide any decisive information on the original problem. If a local minimizer with a nonzero slack objective is computed, we know that one or more constraints of the original problem are violated. In case of a ‘small’ objective, further (typically manual) checks may be carried out to decide whether the minimizer is practically acceptable or not. In case of a ‘large’ objective, we just know that the given solution candidate did not lead to a solution with zero slack. If the instance is feasible, one possibility is to improve the candidate by increasing the modeling accuracy in the first-stage approach.

4. Computational studies

To show the practical relevance of our approaches as a solver for the NoVa problem in gas networks, they are applied to two different types of nominations for country-size real-world gas networks arising at our project partner Open Grid Europe (OGE) (see Figure 6 for an illustration). The first set (*SN4*) contains 4227 automatically generated nominations based on contractual and statistical data by a sampling approach [44]. All these nominations are based on the same network with 592 nodes, 425 pipes, 35 valves, 23 control valves, and 6 compressor groups. The second group (*AB6*) consists of 44 hand-made worst-case nominations by OGE, including four definitely infeasible instances. The corresponding networks are variations of the above. They have about 660 nodes, 500 pipes and more than 30 valves, 25 control valves, and 7 compressor groups. In

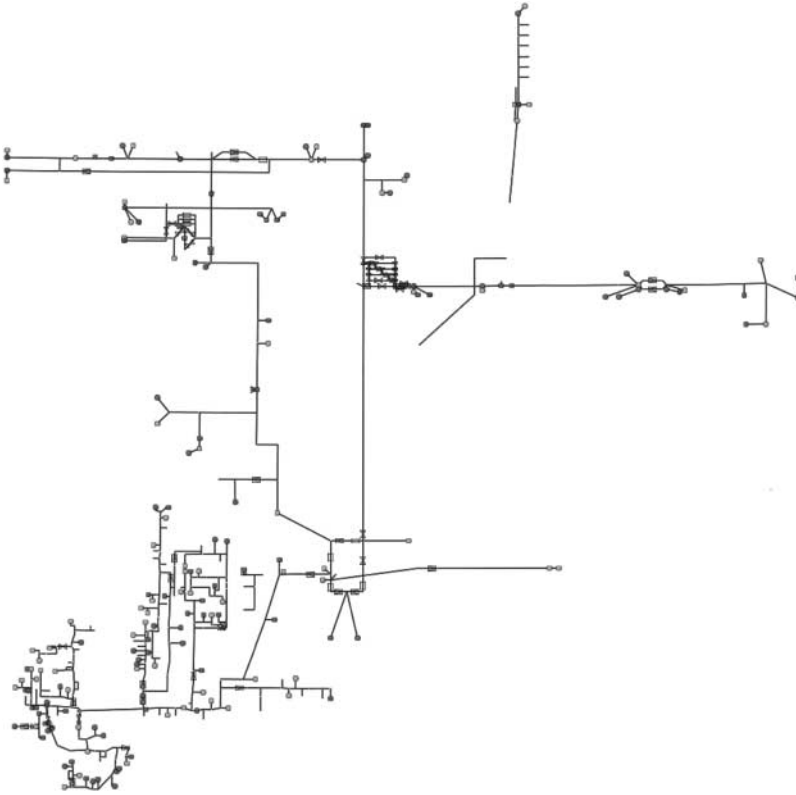


Figure 6. Illustration of one network used for the computations.

the appendix, we give detailed results for this second test set, including feasibility status. These nominations provide the mass flow demand $\mathbf{d} = (d_u)_{u \in V}$ for (3) and possibly pressure bounds at the entries and exits.

4.1 Solutions via black-box solvers

Since NoVa is a mixed-integer nonlinear problem, it is a fundamental question whether it is possible to solve our test instances with state-of-the-art MINLP solvers. To answer this question, we performed a computational study. The answer clearly depends on the mathematical model that is used to represent NoVa. We choose an MINLP model based on (7) with the simplifications (19), (20), (14), and (15). Moreover, we include the non-convex characteristic diagrams of compressors; the remaining model parts are very close to those of the MILP approach.

We are interested in how state-of-the-art solvers perform on such MINLP instances. If a local MINLP solver finds a feasible solution, NoVa is solved affirmatively. If a local solver is, however, not able to find a feasible solution, no conclusion for NoVa can be drawn. Thus, to prove infeasibility of a nomination—a crucial point for our application—the usage of a global solver is required. We apply state-of-the-art solvers of both classes: as global solvers for non-convex MINLPs, we select BARON [78–80] and SCIP [1,81]. These solvers mainly implement (linear) convexification techniques in combination with a spatial branch-and-bound technique.

As local MINLP solvers, we use BONMIN [8], ALPHA-ECP [84], and KNITRO [14]. All three solvers are exact for convex MINLPs, but can be used as heuristics in the non-convex

case. BONMIN implements different algorithms for convex MINLP, from which we choose two different variants, BB and Hyb, for our studies. Variant BB implements a nonlinear branch-and-bound search based on solving continuous nonlinear relaxations at the nodes. The hybrid algorithm Hyb is a combination of an outer approximation branch-and-cut algorithm and the BB algorithm. ALPHAECIP is developed for solving convex or pseudo-convex MINLPs by solving a sequence of MILPs, occasionally solving NLP subproblems, and generating cutting planes. From KNITRO we select the nonlinear branch-and-bound method.

All computations in this subsection were performed sequentially using a single thread on a machine with two six-core AMD Opteron CPUs with 2.6 GHz and 64 GB of RAM. We use GAMS 23.8.2 [34] to communicate over a common interface with all solvers. The respective solver versions included in this GAMS release are BARON 10.1, SCIP 2.1.1, BONMIN 1.5, ALPHAECIP 2.09.01, and KNITRO 8.0. All solvers are set to use their default parameters (if not stated otherwise above). Experiments with other parameters did not significantly improve the results. In addition, we use the nonlinear bound propagation preprocessing (except for SCIP), which is also used to tighten variable bounds for the MILP approach (see Section 4.3 for further details). For SCIP, this preprocessing dramatically worsens the results and sometimes even leads to false infeasibility detections. The other solvers did not behave in this way.

From the first test set (SN4), we pick a random subset of 50 instances to keep the computational effort within reasonable bounds. We set a time limit of four hours for each solver and instance, which is twice the time limit that is subsequently used for our specialized approaches. BARON solved 3 out of 50 instances by finding a feasible solution after 65, 112, and 162 minutes within the time limit. SCIP solved two instances in 3 and 10 minutes and ran into the time limit for all others. BONMIN, ALPHAECIP, and KNITRO are unable to find any feasible solution.

For the second test set (AB6), BARON solved 3 out of the 44 instances by finding a feasible solution after 84, 112, and 219 minutes. SCIP solved two instances in 1 and 63 minutes. Again all local solvers failed to find a feasible solution.

Interestingly, no global solver is able to prove infeasibility, although 12 out of the 50 instances in the first test set (SN4) are infeasible and at least four instances in the second test set (AB6).

We conclude from these computational experiments that large-scale instances of the NoVa problem cannot be solved with state-of-the-art black-box solvers. This motivates the approaches discussed in this paper.

4.2 Computational setup

In the following, we discuss the computational results of the four approaches presented in Section 3 and the corresponding validation step. The computations were performed on a Linux cluster. Each node has two Xeon 3.2 GHz quad core processors and 48 GB of RAM. We imposed a time limit of 2 h for the application of the approaches introduced in Section 3. We performed single-threaded computations, except for the MILP approach, which used eight threads (see Section 4.3 for a discussion of this fact). On each node, only one job is executed simultaneously. The run times for the four approaches to find good discrete decisions exclude the timings for NLP validation. All timings are wall clock times in seconds and include the time for reading the data and building the model, which for some instances consumes a major part of the running time.

Gurobi 5.0 [37] was used to solve the constructed problems by means of the MILP approach. The SB approach was implemented in a prerelease version of SCIP 3.0 [1,74,81]. The LP and NLP subproblems therein were solved using CPLEX 12.4 [42] and IPOPT 3.10 [83], respectively. IPOPT 3.10 was also used to solve the NLP problems in the REDNLP and MPEC approaches. Since the validation NLP can be tackled by several NLP solvers, we sequentially

tried the solvers IPOPT 3.10, CONOPT 3.15C, CONOPT 4.00 [22], and KNITRO 8.0.0 [14] until one of them converges to a feasible point. This last step could, of course, be parallelized. The NLPs of the MPEC and REDNLP approach as well as the NLP validation are solved using GAMS [34] version 23.8.2 as an interface.

Most of the results are illustrated by performance diagrams, where at each point on the x -axis (corresponding to some given measure, e.g. running time, slack value, etc.) we display on the y -axis the fraction of the total number of instances that were solved using at most the given measure on the x -axis. All times are reported in seconds.

4.3 Solutions of the MILP approach

In this section, we discuss numerical results of the MILP approach described in Section 3.1. All MILPs are solved with the branch-and-cut solver Gurobi with default parameter settings on eight threads, except that dual reductions and precrush are disabled, since we generate cutting planes on the fly.

Before the MILP model is constructed, a straightforward nonlinear bound propagation preprocessing is performed to improve unnecessarily large variable bounds. Since the sizes of the variable domains have a direct impact on the size of the linearization, this step is crucial. We refer to [31] for a detailed description of this propagation algorithm. The preprocessing never took more than 10s and is therefore negligible compared to the overall running time of this approach. Thus, we do not list presolving times explicitly. The piecewise linear relaxations are constructed to be within a deviation of at most 1.5 bar from the underlying nonlinear function. We remark that if this tolerance is relaxed, the resulting MILPs are solved faster, but the number of validated solutions decreases. Conversely, when the error tolerance is strengthened, the amount of non-zero slack validations declines, while the running time increases. The error tolerance value of 1.5 bar is an appropriate compromise based on our experience with different test sets.

An illustration of the results for the first test set (SN4), consisting of 4227 instances, is depicted in Figure 7. The variation of the sizes of the resulting MILP instances is small. On average the instances have about 18,302 constraints and 11,222 variables, about 3833 are binary variables.

For 3510 instances, a feasible solution to the MILP model is found, and 694 instances are proved to be infeasible during the time limit of 2 h. This leaves 23 undecided instances. Gurobi's running time is presented in Figure 7(a). The average running time is about 20 minutes, and the median is about 9 minutes. A subset of 3205 instances was solved to optimality, and 3444 instances are proved to be within 10% of the optimum. Of the 3510 feasible instances, the NLP

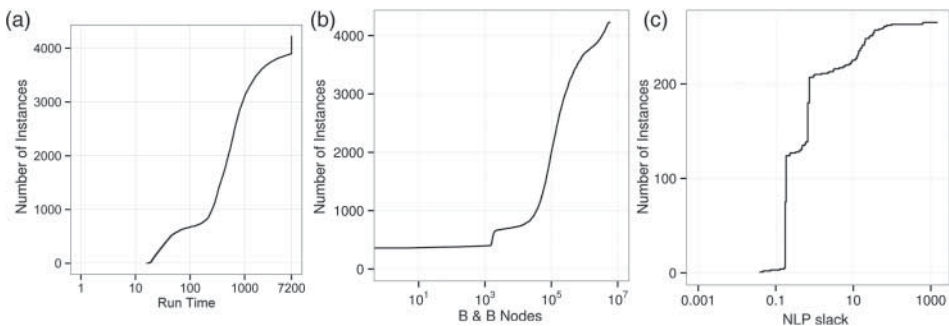


Figure 7. Numerical results on the first test set (SN4) for the MILP approach. (a) Running times (s), (b) number of branch-and-bound nodes, and (c) NLP slack sums for 265 MILP feasible instances with positive slack.

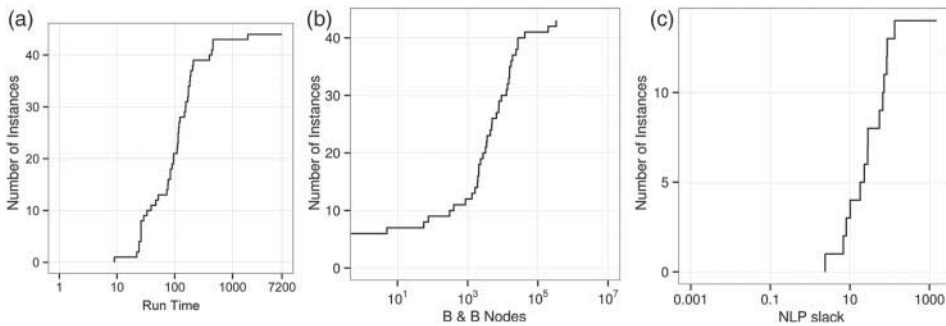


Figure 8. Numerical results on the second test set (AB6) for the MILP approach. (a) Running times (s), (b) number of branch-and-bound nodes, and (c) NLP slack sums for 14 MILP feasible instances with positive slack.

validation confirmed 3245 of them, i.e. resulted in a zero slack value. Figure 7(c) shows the distribution of the slack sum values for the NLP validation of the remaining 265 instances.

The average values are mainly influenced by the large time limit. Within a stricter time limit of 20 minutes, for example, we still find feasible solutions for 3285 instances. A similar behavior can be seen for the number of branch-and-bound nodes shown in Figure 7(b).

Almost all instances (389 out of 397) that were solved within less than 1000 branch-and-bound nodes (or within less than approximately 30 s) are infeasible instances. In general, only 40 of the 694 infeasible instances needed more than 10,000 nodes or more than approximately 1 minute. We conclude that the MILP model is suitable for detecting infeasibility, one of the main purposes why the model is constructed as a relaxation of the underlying nonlinear model.

We illustrate a typical solution process of the test set (SN4) on instance number 1872. The solution of this instance requires about 20 minutes. The first feasible solution is found after 214 s and 38851 branch-and-bound nodes, yielding an optimality gap of 98%. The dual bound is typically weak at the beginning, but this bound is significantly improved up to the point where the initial solution is found. Subsequently, the optimality gap is constantly reduced to 0.24% (by both improving the primal and dual bound) until the optimal solution is found after 615 s have passed and 104,874 nodes have been explored. The remaining search process, which is another 585 s and 379,938 nodes, is spent to prove optimality.

The average solution time can in principle be reduced, if the solution process is stopped as soon as the gap between the primal and dual bound is less than 1%; we have, however, not fully investigated this yet. We recall in this context that NoVa is a feasibility problem. The objective in the MILP approach has been added to overcome the gap between the underlying nonlinear model of the MILP approach and the physically more detailed nonlinear validation model. An optimality gap limit of 1% would also have a major impact on instances reaching the time limit. Most of those instances (188 out of 305) in fact have a similar behavior: a first feasible solution is found within the first 5 minutes and is improved to a gap of less than 1% within an overall running time of about 10 minutes. The remaining time until the time limit is then just used for trying to prove the last per cent of optimality.

As mentioned above, all results were obtained using eight threads, since this allowed to obtain the best performance using our computational resources. If we use only one thread with the same time limit, 3244 instances (instead of 3245) had a 0 slack value, 692 (instead of 694) were determined to be infeasible, and for 27 (instead of 23) we could not find an MILP solution. We conclude that using eight threads only slightly improves the results.

The results for the second test set (AB6) consisting of 44 expert nominations are shown in Figure 8. The MILP instances have on average about 23,824 constraints and 14,871 variables, 5484 of which are binary.

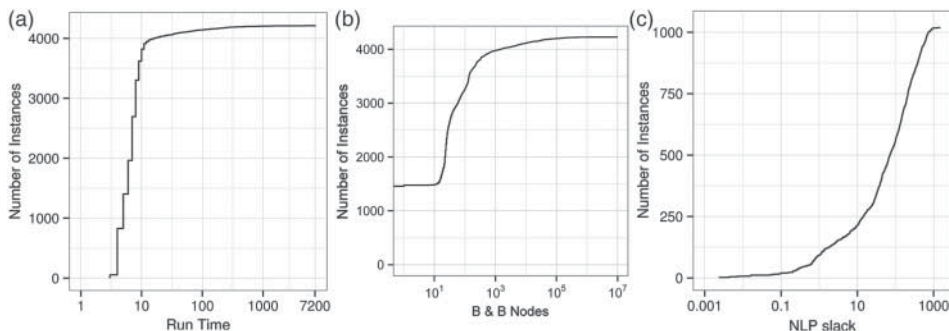


Figure 9. Numerical results on the first test set (SN4) for the SB approach. (a) Running times (s), (b) number of branch-and-bound nodes, and (c) NLP slack sums for 1036 SB feasible instances with positive slack.

For 39 instances, a feasible solution to the MILP model is found, and 5 instances are proved to be infeasible. The infeasibility of one instance is already detected by the nonlinear bound propagation preprocessing, while the remaining four infeasibilities are determined by Gurobi. Each instance is solved within a running time of no more than 30 minutes. The running times are depicted in Figure 8(a). The average running time is about 5 min, and the median is about 3 min. From the 39 feasible instances, 25 were confirmed with a zero slack by the NLP validation.

A more detailed view into the solution process of these instances shows that infeasibility is again rapidly detected (less than a few seconds). In contrast to the first test set, the dual bound is already improved in the root node or at least within the first 100 nodes of the search tree. A first primal solution is typically found faster, too—on average after approximately 100 s and exploring about 2000 nodes. Another observation is that the optimality proof in general does not require as much effort as in the first test set. Typically, just a few more nodes are explored.

An explanation for the remarkable difference between the solution process of instances of the test sets (SN4) and (AB6) might be their different origins. The nominations of (SN4) are based on ordinary, everyday gas delivery situations, which are typically realizable in many different ways and thus typically contain multiple feasible (nearly) symmetrical solutions. The nominations from (AB6), however, are based on expert knowledge to describe exceptional extreme situations, in which the number of admissible discrete decisions is much lower.

4.4 Solutions of the SB approach

In this section, numerical results for the SB approach (see Section 3.2) are presented. On the SN4 test set, 4213 instances are solved, leaving only 14 without solution or proof of infeasibility; 600 instances are proved to be infeasible within the model used by this approach. For 3613 instances, a feasible solution is found within the time limit. Figure 9 shows the performance profiles for the run time, branch-and-bound nodes, and NLP validation slack values of this solver on the SN4 test set. The profile shows that the approach solves 90% (3270) of the instances within less than 10 s. More than 97% of the instances are solved within 1 min.

Most solutions (2640 out of 3613) were found in the tree by the *subnlp* heuristic, which is included in SCIP by default. Whenever a solution is found that is integer feasible for the linear relaxation (either by solving the relaxation in a node or by applying a MILP heuristic), the *subnlp* heuristic applies a local solver (IPOPT) to the NLP that is obtained from the MINLP by fixing all integer variables. In our application, this proves to be extremely effective.

The picture looks similar for the 600 instances for which infeasibility could be proved. Here, in 484 of the instances SCIP presolving already detected the infeasibility. Overall the running time was less than 10 s for 92% of the 600 instances.

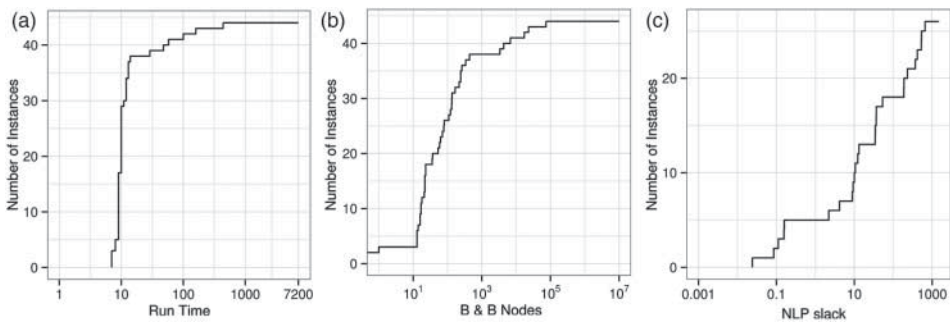


Figure 10. Numerical results on the second test set (AB6) for the SB approach. (a) Running times (s), (b) number of branch-and-bound nodes, and (c) NLP slack sums for 26 SB feasible instances with positive slack.

While the SB seems extremely fast on a large part of the test set, there are 14 instances for which neither a solution could be found nor infeasibility could be proved within the time limit of 2 h. Two main reasons can be identified. For two instances, integer feasibility of the relaxation is hard to reach. In one of these two instances, no integer feasible relaxation is found at all during the solution process. This instance can be proved to be infeasible with slightly different settings of the solver. The remaining unsolved instances spend more than half of the time trying to find a feasible solution in the subnlp heuristic. On these instances, substantial effort is made to strengthen the relaxation by SB; between 30% and 98% of the branchings are performed on continuous variables. However, with customized settings, all unsolved instances can be solved within 2 h resulting in 12 feasible and 2 infeasible instances.

On the vast majority of the instances, surprisingly small effort is made to strengthen the relaxation by SB. Only in 133 instances is SB applied at all. On the remaining test set, the relaxation is strengthened by cutting planes, but branching is not needed. Since we first branch on integer variables that have fractional values in the relaxation, this can be expected for the instances that are solved with only a few nodes. However, only 12 of the 100 instances that take most time to find a feasible solution apply SB. The solution of the most time consuming feasible instance (4649 seconds), for example, does not perform any SB. In this instance, 719,290 nodes are explored of which only three have an integer feasible solution of the linear relaxation. In the third integer feasible node, the subnlp heuristic finds a solution.

The SB approach was able to solve all instances from the AB6 test set. Four instances were recognized to be infeasible, while for the remaining 40 instances, feasible solutions could be computed. Performance profiles for this test set are depicted in Figure 10.

For two instances, infeasibility was proved in presolving. Proving infeasibility for the other two instances took 436 and 159 seconds, respectively. Interestingly, neither instance was solved using SB. All feasible solutions were found in less than 100 s, but only 65% in less than 10 s.

The analysis of the validation of the solutions produced by this approach is as follows. On the SN4 test set, 2577 out of 3613 (71%) solutions could be validated with slack zero, while for 1036 a positive slack remains in the NLP validation. On the AB6 test set, only 14 out of 40 solutions (35%) validate with slack zero. Since only one (almost randomly chosen) solution is passed as a candidate to the validation procedure, local improvement heuristics after a failed validation could help to increase the number of validated solutions.

4.5 Solutions of the RedNLP approach

For each nomination, the REDNLP approach first uses the binary decisions from the transshipment problem (43). If this is not successful, the algorithm is continued by testing 33 given

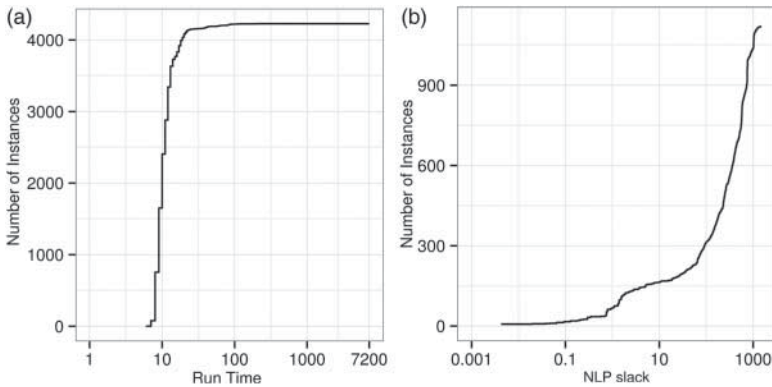


Figure 11. Numerical results on the first test set (SN4) for the REDNLP approach. (a) Running times (s) and (b) NLP slack sums for 1136 REDNLP feasible instances with positive slack.

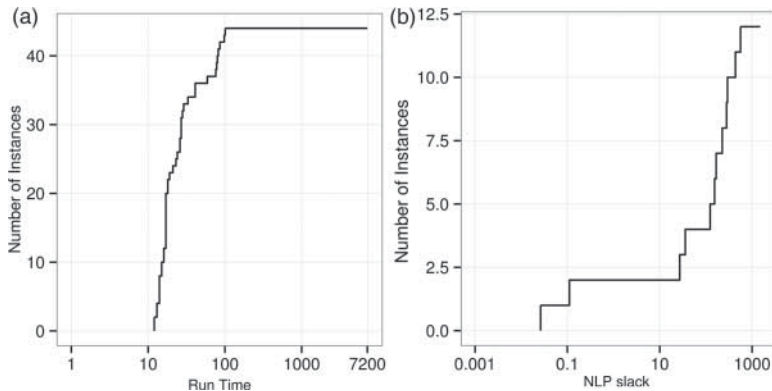


Figure 12. Numerical results on the second test set (AB6) for the REDNLP approach. (a) Running times (s) and (b) NLP slack sums for 12 REDNLP feasible instances with positive slack.

configurations, which successfully solved other nominations. The solution of the reduced NLP model is given as a starting point to the detailed validation NLP. The solution is called *confirmed* if a solution with zero slack could be found.

For the first test set (SN4) of 4227 nominations generated from statistical information, the reduced NLP model can solve 4194 nominations, 3731 of which were found using the switching decisions derived from the solution of the transshipment problem; 3058 of the solutions were confirmed by the detailed NLP slack model. The total computing times for the reduced NLP range from 6 to 208 s with an average of 11.78 s and a median of 10 s. Figure 11 shows the results for this test set.

Out of the 44 nominations in the second test set (AB6), a solution was found for 39 nominations, 27 of which were confirmed. Twenty-two of the solutions were found based on the transshipment solution. The computing times range from 12 to 101 s with an average of 31.64 s and a median of 18.5 s. Figure 12 presents the results.

The transshipment problem for this network has 198 variables and 108 equality constraints. The size of the reduced NLP depends on the chosen switching decisions. The number of variables varies between 1408 and 1441, the number of equality constraints between 1318 and 1346, and that of inequality constraints between 60 and 85.

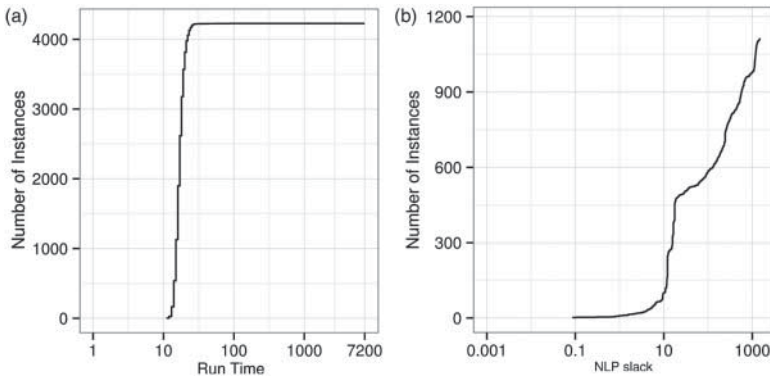


Figure 13. Numerical results on the second test set (SN4) for the MPEC approach. (a) Running times (s) and (b) NLP slack sums for 1178 MPEC feasible instances with positive slack.

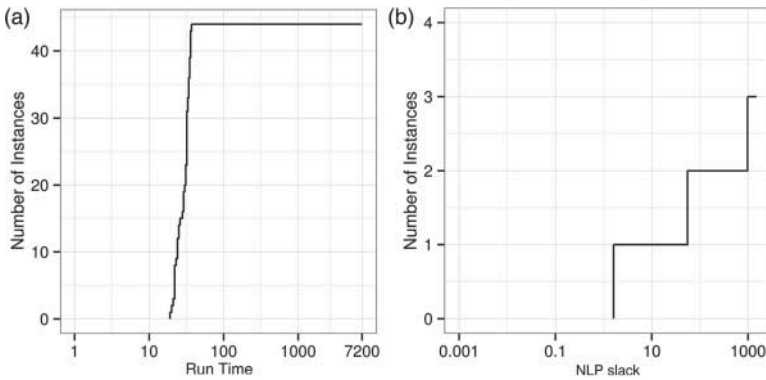


Figure 14. Numerical results on the second test set (AB6) for the MPEC approach. (a) Running times (s) and (b) NLP slack sums for four MPEC feasible instances with positive slack.

4.6 Solutions of the MPEC approach

The results of the MPEC approach are given in Figures 13 and 14. The 4227 instances of the first test set (SN4) require an average computing time of about 17 s, with a maximum of 89 s; 2927 of these instances are solved successfully. The final NLP stage requires about 7 s on average with a maximum of 42 s, and 1749 of the MPEC feasible instances are solved to optimality with vanishing slacks. All computing times include a preprocessing of variable bounds, which only takes a few seconds, (Section 4.3).

Constraints that couple discrete decisions of different active elements to select feasible SOMs (see Section 2.4) can currently not be handled within the MPEC model. These constraints do not fit the requirements necessary for the problem-tailored MPEC-based reformulation techniques that are used for the other discrete aspects of the model. Only 30 of the 2927 instances solved by the validation NLP satisfy these additional constraints. The success rate may be increased by postprocessing algorithms which revisit ambiguous decisions. These algorithms are subject of further research.

For the 44 nominations in the second test set (AB6), the MPEC model finds 10 MPEC feasible solutions, 6 of them with zero slack in the validation NLP. No SOM is fulfilled. The four infeasible instances are correctly identified as MPEC-infeasible, i.e. the MPEC did not converge to a feasible point. The computing time is 29 s on average, with median 31, and maximum 37 s.

Several typical reasons for unsuccessful runs can be observed. The penalization approach of stage 1 attempts to drive the violations of complementarity constraints to zero. If some violations remain positive in a local minimum, we have an infeasible solution (which may even involve technically impossible states such as ‘compressing’ control valves). In this case, stage 1 may be repeated with a different regularization scheme that treats complementarity as explicit constraints. Another difficulty arises from numerical inaccuracy: non-zero constraint values that are smaller than the thresholds of the decision heuristics of stage 1 may lead to wrong discrete decisions. For example, very small but non-zero flows into a subnetwork behind a valve may result in closing this valve. Finally, the simplified compressor group model may lead to an overly optimistic decision for a compressor group in stage 1. Especially when compressors need to be operated close to the boundary of their operation range, as in the worst-case nominations, this may produce first-stage solutions outside the domain of convergence of stage 2.

5. Comparison and combination of the approaches

Sections 4.3–4.6 present an individual analysis of the performance of our four approaches to obtain good discrete decisions and starting points for the NLP validation step. In this section, we compare the different results w.r.t. NLP validation and discuss their combination to yield a reliable solver for NoVa.

5.1 Comparison of the approaches and their results

As outlined in Section 3.5, we are validating the solutions of the different solvers using a detailed NLP model. Before passing the solution to the NLP, we check whether all discrete decisions are taken in accordance with the technical restrictions described by the SOM. The latter check is often the reason that a solution of the MPEC approach is rejected. (For the other approaches this test is satisfied by design—and only included to catch implementation errors.) For this reason, we will not compare the MPEC heuristic together with the other approaches in the following.

In line with the description of Section 3.5, we call solutions that yield an NLP slack below the precision of the validation NLP (which allows constraint violations of at most 10^{-5}) *valid* or *confirmed*. We repeat that this might exclude valid solutions, since the validation NLP might only find a local optimum.

On the test set SN4, the overall results of the MILP, SB, and REDNLP approaches are shown in Table 1 and Figure 15. Each solver can report at most one solution to be validated by the detailed NLP. The times now include the time needed for the validation NLP. All instances that are solved by the SB approach within 10 s are infeasible and therefore no validation with the NLP model is carried out. This explains the notable ‘bump’ in the graph of the SB approach in Figure 15. Moreover, when the MILP cannot prove optimality within the time limit, but has found a feasible solution, the best solution found is validated at the very end of the computation. This explains the ‘jump’ of the MILP success at the 7200 s line.

The computations on the test set AB6 are displayed in Table 2 and Figure 16.

When analyzing the results, we note that the SB and the REDNLP approach can both produce solutions for many instances in a rather short time. The MILP approach is by far slower, but has a significantly higher success rate. This is partly due to the parameter setting choice for the MILP approach to yield high accuracy.

The differences between the SB and the REDNLP approach become visible when comparing the numbers in Tables 1 and 2. The SB approach is able to detect infeasibility, whereas the REDNLP approach just reports that no solution has been found.

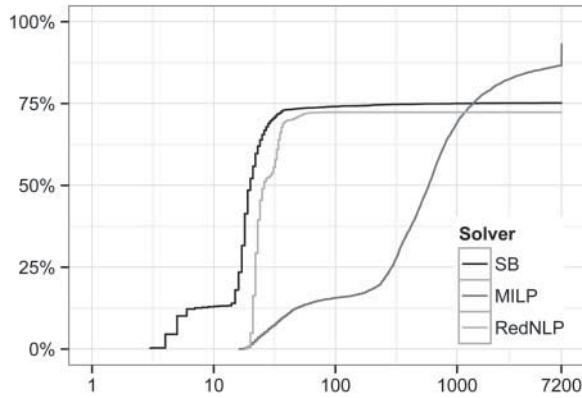


Figure 15. Profile of the run times (s) for test set SN4 including NLP validation; y-axis: percentage of confirmed or infeasible instances (total: 4227).

Table 1. Results of the solvers on test set SN4 (4227 total instances): the columns give the number of instances for which a solution with 0 slack could be found by NLP validation, the number of instances detected to be infeasible (REDNLP cannot detect infeasibility), the number of instances for which a solution with positive slack could be found, and the number of instances for which no solution could be found.

	Slack 0	Infeasible	Slack > 0	No solution
MILP	3245	694	265	23
SB	2577	600	1036	14
RedNLP	3058	0	1136	33

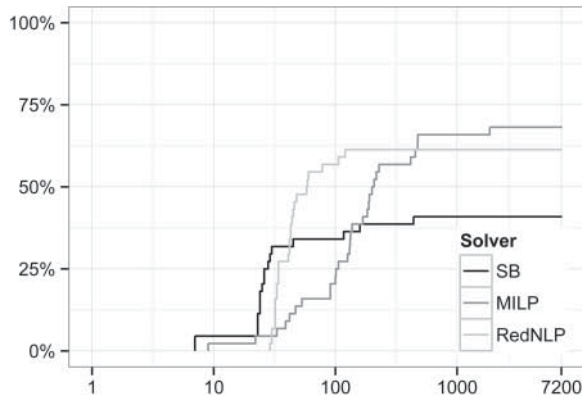


Figure 16. Profile of the run times (s) for test set AB6 including NLP validation; y-axis: percentage of confirmed or infeasible instances (total: 44).

Table 2. Results of the solvers on test set AB6 (44 instances); information as in Table 1.

	Slack 0	Infeasible	Slack > 0	No solution
MILP	25	5	14	0
SB	14	4	26	0
RedNLP	27	0	12	5

The differences between the MILP and SB approach with respect to their validation success might be explained as follows. One principal difference in the models is the compressor model. The MILP decides which configurations of the compressor groups are used in the model, while for the SB approach the capability of a compressor group is modeled by the convex hull over all configurations, and a feasible or a ‘least’ infeasible configuration is selected in a postprocessing step. This might include convexification errors and, thus, bad decisions in the surroundings of compressor groups. Apparently, especially on the AB6 instances, it is crucial to choose appropriate configurations for the compressor groups.

The results for the AB6 test set are similar to the ones for the SN4 test set, but the NLP validation results are worse for all approaches. One reason is that the ratio of feasible to infeasible instances is different in the two test sets. It seems to be easier for the SB and the MILP approach to detect infeasibilities than to find primal solutions for challenging instances. This also explains the comparatively better results of the REDNLP approach.

5.2 Combined solver

We have combined the MILP, SB, and REDNLP approaches presented in Section 3 to form a solver that can be used to reliably solve the NoVa problem for real-world gas network instances as described in the previous sections.

The analysis of the results is complicated by the fact that the models are not direct refinements of each other. All models differ at one or the other point from the model of the validation NLP. For instance, this holds for current implementation of the MILP model of Section 3.1: it could, in principle, be extended to form a proper relaxation of the MINLP corresponding to the validation NLP. This is, however, undesirable in practice: the non-differentiable functions like sgn and absolute values appearing in the nonlinear models usually have to be smoothed. An MILP model, on the other hand, can model these functions. The smoothing functions, however, may lead to problems in the MILP setting and inexactness. This means that different models are preferred for, say, the MILP model and the validation NLP.

Consequently, the approaches can contradict each other: one approach might find a feasible solution, while another might report the instance as infeasible. Our approach is, nevertheless, to run all solvers for all models in parallel. As above, each solver can report at most one solution to be validated by the detailed NLP. Recall once again that this approach might exclude valid solutions as we cannot be sure that the validation NLP is not stuck in a local optimum.

For the purpose of analysis, we call such a parallel run *successful*, if the solvers do not contradict each other and either at least one solver finds a solution with zero slack or at least one solver reports infeasibility. Otherwise such a run is *unsuccessful*. For the following computational experiments we used the same infrastructure as for the other tests. The parameters of the MILP model of Section 3.1 are calibrated to reliably yield feasible solutions at the cost of a long running time. We use the other solvers to generate feasible solutions quickly.

The results of the combined solver on the test set SN4 are given in Figure 17. Totally, 4157 (more than 98%) of the 4227 instances in the test set can be solved *successfully*, whereas we are *unsuccessful* on 70 of the instances. Here, 38 of the instances yield a contradictory result, and the remaining 32 failed to yield any definitive result. The times shown in Figure 17 are with respect to the first solver that reports the final result (now including the time needed to validate feasible solutions with the validation NLP).

The results of the combined approach on the test set AB6 are given in Figure 18 (using the same structure as Figure 17). Of the 44 instances in the test set, 38 (more than 86%) can be solved *successfully*, whereas we are *unsuccessful* on six of the instances. Here, none of the instances yield a contradictory result, and the remaining six fail to yield any definitive result.

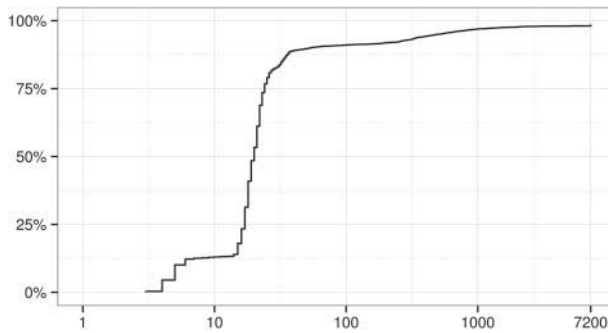


Figure 17. Profile of the run times (s) for the combined approach on test set SN4 including NLP validation; y-axis: percentage of confirmed or infeasible instances (total: 4227).

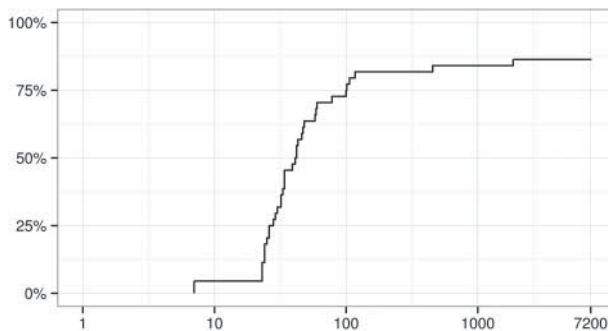


Figure 18. Profile of the run times (s) for the combined approach on test set AB6 including NLP validation; y-axis: percentage of confirmed or infeasible instances (total: 44).

6. Summary

Faced with a complex and numerically difficult mixed-integer non-convex nonlinear feasibility problem, we have shown how to utilize the full range of mathematical optimization technology (NLP, MILP, MINLP) in this context.

Our two-stage approach is finally able to successfully solve nearly 98% of the instances with high precision. The paper describes the two key reasons for this success:

- Splitting the problem into two phases: A first phase using approximate models for finding settings for the discrete decisions and then using the results of the first phase to compute highly precise solutions using an NLP.
- Combination of four quite distinct approaches to solve phase one: While all the approaches in their way are quite successful, none of them for itself is able to come near the combined performance.

Nevertheless, while the networks used in this paper are large in comparison to the state of art, they are small compared to the real European pipeline network. We are currently trying to further develop our approach to be able to solve networks which are at least five times bigger.

Another area that leaves room for improvement is the transition from the approximate models to the detailed NLP model. Due to the corresponding model differences, it is not entirely clear what the best objective for the approximate models of the first phase is. In principle, any feasible solution to the approximate models could be tried in the detailed model. This is one of the reasons why we included the MPEC approach. The model used here is the one which is most similar to the detailed NLP model. Consequently, those instances that can be solved typically exhibit no or

only very small slacks in the validation. The validation performance of the MPEC approach is currently not comparable to the other approaches, because of the missing subnetwork operation modes constraints. Thus, the MPEC approach is promising, if there is a way to address these issues.

Acknowledgements

We thank Klaus Spreckelsen from OGE GmbH for his support and Timo Berthold and Stefan Heinz from the MATH-EON B20 project for their support on adapting the SCIP solver. We thank the referees for their constructive comments that helped to improve the presentation of the paper. The coauthor Armin Fügenschuh acknowledges a Konrad-Zuse Fellowship.

Funding

This work has partially been supported by the Federal Ministry of Economics and Technology.

References

- [1] T. Achterberg, *SCIP: Solving constraint integer programs*, Math. Program. Comput. 1(1) (2009), pp. 1–41.
- [2] J. André, F. Bonnans, and L. Cornibert, *Optimization of capacity expansion planning for gas transportation networks*, Eur. J. Oper. Res. 197(3) (2009), pp. 1019–1027.
- [3] F. Babonneau, Y. Nesterov, and J.-P. Vial, *Design and operations of gas transmission networks*, Oper. Res. 60(1) (2012), pp. 34–47.
- [4] R. Bagnara, P. M. Hill, and E. Zaffanella, *The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems*, Sci. Comput. Program. 72(1–2) (2008), pp. 3–21.
- [5] P. Bales, *Hierarchische Modellierung der Eulerschen Flussgleichungen in der Gasdynamik*, Master’s thesis, Technische Universität Darmstadt, 2005.
- [6] B.T. Baumrucker and L.T. Biegler, *MPEC strategies for cost optimization of pipeline operations*, Comput. Chem. Eng. 34(6) (2010), pp. 900–913.
- [7] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optim. Methods Softw. 24(4–5) (2009), pp. 597–634.
- [8] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete Optim. 5(2) (2008), pp. 186–204.
- [9] J. Bonnans, G. Spiers, and J.-L. Vie, *Global optimization of pipe networks by the interval analysis approach: The Belgium network case*, Rapport de Recherche RR 7796, INRIA, 2011.
- [10] C. Borrás-Sánchez and R. Ríos-Mercado, *A non-sequential dynamic programming approach for natural gas network optimization*, WSEAS Trans. Syst. 3 (2004), pp. 1384–1389.
- [11] Bundesministerium der Justiz, *Verordnung über den Zugang zu Gasversorgungsnetzen (Gasnetzzugangsverordnung – GasNZV)*, July 2005.
- [12] J. Burgschweiger, B. Gnädig, and M.C. Steinbach, *Nonlinear programming techniques for operative planning in large drinking water networks*, Open Appl. Math. J. 3 (2009), pp. 14–28.
- [13] J. Burgschweiger, B. Gnädig, and M.C. Steinbach, *Optimization models for operative planning in drinking water networks*, Optim. Eng. 10(1) (2009), pp. 43–73.
- [14] R. Byrd, J. Nocedal, and R. Waltz, *Knitro: An integrated package for nonlinear optimization*, in *Large-Scale Nonlinear Optimization*, Volume 83 of Nonconvex Optimization and Its Applications, G. Pillo, M. Roma, and P. Pardalos, eds., Springer Berlin Heidelberg, 2006, pp. 35–59.
- [15] R. Carter, *Compressor station optimization: Computational accuracy and speed*, Tech. Rep. PSIG 9605, Pipeline Simulation Interest Group, 1996.
- [16] R.G. Carter, *Pipeline optimization: Dynamic programming after 30 years*, Proceedings of the 30th PSIG Annual Meeting, Denver, CO, 1998.
- [17] R. Carter, D. Schroeder, and T. Harbick, *Some causes and effects of discontinuities in modeling and optimizing gas transmission networks*, Tech. Rep. PSIG 9308, Pipeline Simulation Interest Group, 1993.
- [18] G. Cerbe, *Grundlagen der Gastechnik: Gasbeschaffung – Gasverteilung – Gasverwendung*, Hanser Verlag, Leipzig, 2008.
- [19] A. Chebouba, F. Yalaoui, A. Smati, L. Amodeo, K. Younsi, and A. Tairi, *Optimization of natural gas pipeline transportation using ant colony optimization*, Comput. Oper. Res. 36 (2009), pp. 1916–1923.
- [20] C.F. Colebrook, *Turbulent flow in pipes with particular reference to the transition region between smooth and rough pipe laws*, J. Inst. Civil Eng. 11 (1939), pp. 133–156.

- [21] P. Domschke, B. Geißler, O. Kolb, J. Lang, A. Martin, and A. Morsi, *Combination of nonlinear and linear optimization of transient gas networks*, INFORMS J. Comput. 23 (2011), pp. 605–617.
- [22] A.S. Drud, *CONOPT—a large-scale GRG code*, INFORMS J. Comput. 6(2) (1994), pp. 207–216.
- [23] K. Ehrhardt and M.C. Steinbach, *Nonlinear optimization in gas networks*, in *Modeling, Simulation and Optimization of Complex Processes*, H.G. Bock, E. Kostina, H.X. Phu, and R. Ranacher, eds., Springer Berlin Heidelberg, 2005, pp. 139–148.
- [24] European Parliament and Council, *Regulation (ec) no. 715/2009: Conditions for access to the natural gas transmission networks*, 13 July 2009; available at <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0036:0054:en:PDF>, Visited 10/2012.
- [25] M. Feistauer, *Mathematical Methods in Fluid Dynamics*, Volume 67 of Pitman Monographs and Surveys in Pure and Applied Mathematics Series, Longman Scientific & Technical, Harlow, 1993.
- [26] E.J. Finnemore and J.E. Franzini, *Fluid Mechanics with Engineering Applications*, 10th ed., McGraw-Hill, New York, 2002.
- [27] A. Fügenschuh, B. Geißler, A. Martin, and A. Morsi, *The transport PDE and mixed-integer linear programming*, in *Models and Algorithms for Optimization in Logistics*, C. Barnhart, U. Clausen, U. Lauther, and R.H. Möhring, eds., Number 09261 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009.
- [28] A. Fügenschuh, H. Homfeld, H. Schülldorf, and S. Vigerske, *Mixed-integer nonlinear problems in transportation applications*, in *Proceedings of the 2nd International Conference on Engineering Optimization (CD-ROM)*, H. Rodrigues, J. Herskovits, C. Mota Soares, J. Miranda Guedes, J. Folgado, A. Araújo, F. Moleiro, J.P. Kuzhichalil, J. Aguiar Madeira, and Z. Dimitrovová, eds., Instituto Superior Técnico, Lisboa, 2010.
- [29] M. Fukushima and J.-S. Pang, *Convergence of a smoothing continuation method for mathematical programming with complementarity constraints*, in *Ill-Posed Variational Problems and Regularization Technique, Lecture Notes in Economics and Mathematical Systems*, Vol. 477, M. Théra and R. Tichatschke, eds., Springer Berlin Heidelberg, 1999, pp. 99–110.
- [30] B. Furey, *A sequential quadratic programming-based algorithm for optimization of gas networks*, Automatica 29(6) (1993), pp. 1439–1450.
- [31] B. Geißler, *Towards globally optimal solutions for MINLPs by discretization techniques with applications in gas network optimization*, PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2011.
- [32] B. Geißler, O. Kolb, J. Lang, G. Leugering, A. Martin, and A. Morsi, *Mixed integer linear models for the optimization of dynamical transport networks*, Math. Methods Oper. Res. 73 (2011), pp. 339–362.
- [33] B. Geißler, A. Martin, A. Morsi, and L. Schewe, *Using piecewise linear functions for solving MINLPs*, in *Mixed Integer Nonlinear Programming*, Volume 154 of The IMA Volumes in Mathematics and Its Applications, J. Lee and S. Leyffer, eds., Springer, New York, 2012, pp. 287–314.
- [34] General Algebraic Modeling System (GAMS); software available at <http://www.gams.com/>.
- [35] P.E. Gill, W. Murray, and M.S. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM J. Optim. 12(4) (2002), pp. 979–1006.
- [36] B. Gilmour, C. Luongo, and D. Schroeder, *Optimization in natural gas transmission networks: A tool to improve operational efficiency*, Tech. Rep., Stoner Associates Inc., 1989.
- [37] Z. Gu, E. Rothberg, and R. Bixby, *Gurobi Optimizer Reference Manual, Version 5.0*, Gurobi Optimization Inc., Houston, TX, 2012.
- [38] Y. Hamam and A. Brameller, *Hybrid method for the solution of piping networks*, Proc. IEE 118(11) (1971), pp. 1607–1612.
- [39] C.T. Hansen, K. Madsen, and H.B. Nielsen, *Optimization of pipe networks*, Math. Program. 52(1–3) (1991), pp. 45–58.
- [40] P. Hofer, *Beurteilung von Fehlern in Rohrnetzrechnungen (error evaluation in calculation of pipelines)*, GWF Gas/Erdgas 11 (1973), pp. 113–119.
- [41] X.M. Hu and D. Ralph, *Convergence of a penalty method for mathematical programming with complementarity constraints*, J. Optim. Theory Appl. 123 (2004), pp. 365–390.
- [42] IBM Corporation, Armonk, USA, *User's Manual for CPLEX*, 12.4 ed., 2011.
- [43] Jerníček, *Steady-state optimization of gas transport*, Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modeling and Optimal Control of Large Scale Pipeline Networks, Prague, Czech Republic, 1993.
- [44] T. Koch, H. Leövey, R. Mirkov, W. Römisch, and I. Wegner-Specht, *Szenariogenerierung zur Modellierung der stochastischen Ausspeiselasten in einem Gastransportnetz*, VDI-Berichte: Optimierung in der Energiewirtschaft 2157 (2011), pp. 115–125.
- [45] T. Koch, D. Bargmann, M. Ebbers, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, U. Gotzes, C. Hayn, H. Heitsch, R. Henrion, B. Hiller, J. Humpola, I. Joormann, V. Köhl, T. Lehmann, R. Lenz, H. Leövey, A. Martin, R. Mirkov, A. Möller, A. Morsi, D. Oucherif, A. Pelzer, M.E. Pfetsch, L. Schewe, W. Römisch, J. Rövekamp, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, K. Spreckelsen, C. Stangl, M.C. Steinbach, A. Steinkamp, I. Wegner-Specht, and B.M. Willert, *From Simulation to Optimization: Evaluating Gas Network Capacities*, Book in preparation, 2012.
- [46] J. Králik, *Compressor Stations in SIMONE*, Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modeling and Optimal Control of Large Scale Pipeline Networks, Prague, Czech Republic, 1993.
- [47] H. Lall and P. Percell, *A dynamic programming based Gas Pipeline Optimizer*, in *Analysis and Optimization of Systems*, Volume 144 of Lecture Notes in Control and Information Sciences, A. Bensoussan and J. Lions, eds., Springer Berlin Heidelberg, 1990, pp. 123–132.

- [48] C. Li, W. Jia, Y. Yang, and X. Wu, *Adaptive genetic algorithm for steady-state operation optimization in natural gas networks*, J. Softw. 6 (2011), pp. 452–459.
- [49] LIWACOM Informations GmbH and SIMONE Research Group s.r.o, *Gleichungen und Methoden, Benutzerhandbuch*, 2004.
- [50] M.V. Lurie, *Modeling of Oil Product and Gas Pipeline Transportation*, Wiley-VCH, Weinheim, 2008.
- [51] D. Mahlke, A. Martin, and S. Moritz, *A simulated annealing algorithm for transient optimization in gas networks*, Math. Methods Oper. Res. 66(1) (2007), pp. 99–116.
- [52] D. Mahlke, A. Martin, and S. Moritz, *A mixed integer approach for time-dependent gas network optimization*, Optim. Methods Softw. 25(4) (2010), pp. 625–644.
- [53] J. Mallinson, A. Fincham, S. Bull, J. Rollet, and M. Wong, *Methods for optimizing gas transmission networks*, Ann. Oper. Res. 43 (1993), pp. 443–454.
- [54] H. Markowitz and A. Manne, *On the solution of discrete programming problems*, Econometrica 25 (1957), pp. 84–110.
- [55] A. Martin, M. Möller, and S. Moritz, *Mixed integer models for the stationary case of gas network optimization*, Math. Program. 105(2) (2006), pp. 563–582.
- [56] J. Mischner, *Notices about hydraulic calculations of gas pipelines*, GWF Gas/Erdgas 4 (2012), pp. 158–273.
- [57] M. Möller, *Mixed integer models for the optimisation of gas networks in the stationary case*, PhD thesis, Technische Universität Darmstadt, 2004.
- [58] S. Moritz, *A mixed integer approach for the transient case of gas network optimization*, PhD thesis, Technische Universität Darmstadt, 2007.
- [59] J. Murdock, *Fundamental Fluid Mechanics for the Practicing Engineer*, Mechanical Engineering, Marcel Dekker, New York, 1993.
- [60] J. Nikuradse, *Strömungsgesetze in rauhen Rohren*, Forschungsheft auf dem Gebiete des Ingenieurwesens, VDI-Verlag, Düsseldorf, 1933.
- [61] J. Nikuradse, *Laws of Flow in Rough Pipes*, Volume Technical Memorandum 1292, National Advisory Committee for Aeronautics, Washington, DC, 1950.
- [62] A.J. Osiadacz, *Simulation and Analysis of Gas Networks*, Spon, London, 1987.
- [63] A. Osiadacz and S. Swierczewski, *Optimal control of gas transportation systems*, Proceedings of IEEE Conference on Control Applications, Warsaw, Poland, Vol. 2, 1994, pp. 795–796.
- [64] I. Papay, OGIL Musz. Tud. Kozl., 1968.
- [65] T.A. Perdicoulis and L. Fletcher, *Decentralised dynamic optimisation of gas*, 31st Annual Meeting of Pipeline Simulation Interest Group, St. Louis, MI, USA, 1999.
- [66] N. Ramchandani, *Optimisation of gas networks using Nash equilibria derived from dynamic non-cooperative game theory*, PhD thesis, Stanford University, 1993.
- [67] R.Z. Ríos-Mercado and C. Borraz-Sánchez, *Optimization problems in natural gas transmission systems: A state-of-the-art survey*, Submitted, January 2012.
- [68] R.Z. Ríos-Mercado, S. Wu, L.R. Scott, and E.A. Boyd, *A reduction technique for natural gas transmission network optimization problems*, Ann. Oper. Res. 117(1) (2002), pp. 217–234.
- [69] J. Saleh, ed., *Fluid Flow Handbook*, McGraw-Hill Handbooks, McGraw-Hill, New York, 2002.
- [70] M. Schmidt, *A generic interior-point framework for nonsmooth and complementarity constrained nonlinear optimization*, PhD thesis, Gottfried Wilhelm Leibniz Universität Hannover, 2013.
- [71] M. Schmidt, M.C. Steinbach, and B.M. Willert, *High detail stationary optimization models for gas networks—Part 1: Model components*, IFAM Preprint 94, Institute of Applied Mathematics, Leibniz Universität, Hannover, 2012, to appear in Optim. Eng.
- [72] M. Schmidt, M. Steinbach, and B. Willert, *A primal heuristic for nonsmooth mixed integer nonlinear optimization*, in *Facets of Combinatorial Optimization*, M. Jünger and G. Reinelt, eds., Springer Verlag, Berlin, 2013, pp. 295–320.
- [73] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM J. Optim. 11(4) (2001), pp. 918–936.
- [74] SCIP: Solving constraint integer programs, 2012; software available at <http://scip.zib.de/>.
- [75] E. Smith and C. Pantelides, *A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs*, Comput. Chem. Eng. 23 (1999), pp. 457–478.
- [76] M.C. Steinbach, *On PDE solution in transient optimization of gas networks*, J. Comput. Appl. Math. 203(2) (2007), pp. 345–361.
- [77] O. Strusberg and S. Engell, *Optimal control of switched continuous systems using mixed-integer programming*, 15th IFAC World Congress of Automatic Control, Barcelona, Spain, 2002.
- [78] M. Tawarmalani and N. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Kluwer Academic Publishers, Boston, 2002.
- [79] M. Tawarmalani and N. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Math. Program. 99 (2004), pp. 563–591.
- [80] M. Tawarmalani and N. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Math. Program. 103 (2005), pp. 225–249.
- [81] S. Vigerske, *Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming*, PhD thesis, Humboldt-Universität zu Berlin, 2013.

- [82] Z. Vostrý, *Transient optimization of gas transport and distribution*, Proceedings of the 2nd International Workshop SIMONE on Innovative Approaches to Modeling and Optimal Control of Large Scale Pipeline Networks, Prague, Czech Republic, 1993.
- [83] A. Wächter and L.T. Biegler, *On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming*, Math. Program. 106(1) (2006), pp. 25–57.
- [84] T. Westerlund and R. Pörn, *Solving pseudo-convex mixed integer optimization problems by cutting plane techniques*, Optim. Eng. 3 (2002), pp. 253–280.
- [85] D. de Wolf and Y. Smeers, *Optimal dimensioning of pipe networks with application to gas transmission networks*, Oper. Res. 44(4) (1996), pp. 596–608.
- [86] D. De Wolf and Y. Smeers, *The gas transmission problem solved by an extension of the simplex algorithm*, Manag. Sci. 46(11) (2000), pp. 1454–1465.
- [87] P.J. Wong and R.E. Larson, *Optimization of natural-gas pipeline systems via dynamic programming*, IEEE Trans. Automat. Control 15(5) (1968), pp. 475–481.
- [88] S. Wright, M. Somani, and C. Ditzel, *Compressor station optimization*, Tech. Rep., Pipeline Simulation Interest Group, Denver, CO, 1998.
- [89] S. Wu, R.Z. Ríos-Mercado, E.A. Boyd, and L.R. Scott, *A reduction technique for natural gas transmission network optimization problems*, Math. Comput. Model. 31(2) (2000), pp. 197–220.
- [90] J. Zhang and D. Zhu, *A bilevel programming method for pipe network optimization*, SIAM J. Optim. 6(3) (1996), pp. 838–857.
- [91] H. Zimmer, *Calculating optimum pipeline operations*, Tech. Rep. SAND2009-5066C, El Paso Natural Gas Company, 1975.

Appendix

In this section, we collect detailed results on the 44 instances in the AB6 test set for the MILP, SB, REDNLP, and MPEC approaches described in Section 3, see Tables A1, A2, A3, and A4, respectively. The tables contain data that are specific to the respective approach. In all tables, ‘Name’ refers to the name of the instance, and ‘Time’ shows the time (in seconds) needed for the specific approach, excluding the time for NLP validation. For each approach, ‘Status’ refers to the status of the NLP validation: it displays the value of the NLP validation slack, if the approach produced a solution. If the approach proved infeasibility or found no solution, it displays ‘inf’ or ‘nosol’, respectively. The last column (‘Validation time’) displays the time need for NLP validation (in seconds).

Additionally, in Table A1 for the MILP approach, ‘Nodes’ refers to the number of branch-and-bound nodes, ‘Variables’ the total number of variables, ‘Binary’ the number of binary variables, and ‘Constraints’ the number of constraints. Table A2 for the SB approach additionally gives the number of branching on integer variables in ‘Variable Branchings’ and in column ‘Spatial Branchings’ the number of SBs. Table A3 for the REDNLP displays in the column labeled ‘Transshipment’ whether the binary decisions found by the transshipment heuristic lead to a feasible solution. The column ‘Tested Configs’ gives the number of tested preselected configurations. Table A4 for the MPEC approach additionally shows the status after the stage X in ‘Status X’ and the running time of IPOPT in seconds in ‘Time X’.

Table A1. Detailed results of the MILP approach on test set AB6.

Name	Time	Status	Nodes	Variables	Binary	Constraints	Validation time
01-1011	76	0.00	56	14,873	5482	23,835	9
02-1011	210	0.00	4835	15,187	5638	24,308	9
04-1011	26	84.01	304	14,851	5478	23,781	8
03-1011	52	6.80	1904	14,805	5455	23,712	8
05-1011	123	132.55	4970	14,501	5296	23,277	8
06-1011	96	8.05	3447	15,436	5768	24,676	8
07-1011	119	0.00	2331	14,843	5463	23,778	10
08-1011	437	0.00	42,335	14,577	5342	23,390	10
09-1011	73	0.00	2077	14,719	5412	23,583	9
09-1011-inf	9	inf	–	–	–	–	–
10-1011	90	0.00	1324	14,659	5377	23,520	9
01-1112	110	0.00	7674	14,865	5477	23,820	8
02-1112	189	0.00	15,626	15,179	5634	24,296	9
04-1112	29	86.45	866	14,871	5488	23,811	8
03-1112	26	71.77	76	14,851	5478	23,781	8
05-1112	95	55.03	3204	14,551	5321	23,352	9
06-1112	185	22.88	26,767	15,432	5766	24,670	7
06-1112-inf	39	inf	0	15,432	5766	24,670	–
07-1112	147	10.13	14,837	14,835	5459	23,766	8
08-1112	1838	0.00	333,372	14,571	5339	23,381	10
09-1112	84	0.00	4411	14,865	5485	23,802	10
10-1112	168	0.00	17,231	14,629	5362	23,475	8
01-1213	115	0.00	2735	14,903	5496	23,877	9
02-1213	182	0.00	23,920	15,175	5632	24,290	10
04-1213	26	66.75	402	14,871	5488	23,811	8
04-1213-inf	22	inf	0	14,871	5488	23,811	–
04-1213-inf-m	47	inf	5	14,908	5505	23,887	–
03-1213	24	0.00	0	14,801	5453	23,706	10
05-1213	33	inf	0	14,575	5333	23,388	–
06-1213	155	27.86	18,846	15,334	5717	24,523	8
07-1213	115	0.00	6580	14,829	5456	23,757	9
08-1213	459	0.00	26,949	14,605	5356	23,432	10
09-1213	398	0.00	196,466	14,863	5484	23,799	9
10-1213	173	0.00	13,594	14,617	5356	23,457	9
01-1314	204	0.00	1943	14,867	5478	23,823	9
02-1314	153	0.00	7797	15,171	5630	24,284	9
04-1314	24	2.38	0	14,873	5489	23,814	6
03-1314	26	0.00	0	14,801	5453	23,706	8
05-1314	77	18.08	1618	14,549	5320	23,349	8
06-1314	112	28.27	15,513	15,378	5739	24,589	7
07-1314	117	0.00	3584	14,829	5456	23,757	8
08-1314	458	0.00	9135	14,603	5355	23,429	10
09-1314	84	0.00	12,742	14,863	5484	23,799	9
10-1314	172	0.00	2067	14,615	5355	23,454	10

Table A2. Detailed results of the SB approach on test set AB6.

Name	Time	Status	Variable branchings	SBs	Nodes	Validation time
01-1011	10	0.16	0	0	0	10
02-1011	13	34.54	96	1	136	9
04-1011	10	233.00	35	7	53	8
03-1011	10	12.26	38	0	63	9
05-1011	13	656.57	101	0	167	9
06-1011	13	53.45	251	0	439	8
07-1011	12	0.00	92	58	263	10
08-1011	10	526.43	24	0	22	8
09-1011	10	0.09	44	4	59	9
09-1011-inf	7	inf	3	0	1	-
10-1011	9	0.00	17	0	17	10
01-1112	29	0.00	1728	0	3366	10
02-1112	9	0.00	30	0	36	10
04-1112	9	370.67	14	0	16	8
03-1112	9	190.47	49	0	80	9
05-1112	100	0.00	4449	4181	17,194	10
06-1112	10	0.16	122	0	219	8
06-1112-inf	438	inf	37,548	0	74,648	-
07-1112	14	0.00	202	0	341	9
08-1112	12	187.67	161	0	243	10
09-1112	8	10.04	22	0	22	8
10-1112	8	0.00	22	1	21	9
01-1213	10	34.58	15	0	17	10
02-1213	11	0.00	63	0	109	10
04-1213	9	2.21	46	0	71	9
04-1213-inf	7	inf	0	0	0	-
04-1213-inf-m	159	inf	11,653	0	23,042	-
03-1213	10	8.89	75	0	133	12
05-1213	48	419.13	2255	0	4325	9
06-1213	9	4.16	25	0	23	9
07-1213	9	36.65	18	0	18	14
08-1213	9	0.00	11	0	13	9
09-1213	7	13.17	14	0	14	8
10-1213	9	0.00	22	0	22	10
01-1314	9	36.74	12	0	13	9
02-1314	10	0.00	70	0	123	9
04-1314	9	9.32	49	0	78	8
03-1314	10	10.54	15	0	16	14
05-1314	58	533.70	3464	0	6790	9
06-1314	12	0.11	142	1	242	9
07-1314	12	0.00	87	0	134	10
08-1314	9	0.00	13	0	13	10
09-1314	10	0.02	24	9	37	9
10-1314	10	0.00	25	0	23	10

Table A3. Detailed results of the REDNLP approach on test set AB6.

Name	Time	Status	Transshipment	Tested configs	Validation time
01-1011	17	0.00	Yes	0	9
02-1011	15	0.00	Yes	0	9
04-1011	27	0.00	No	4	7
03-1011	24	0.00	No	4	8
05-1011	29	166.59	No	2	9
06-1011	76	nosol	–	33	–
07-1011	17	0.00	Yes	0	8
08-1011	12	436.10	Yes	0	44
09-1011	59	0.00	No	26	9
09-1011-inf	98	nosol	–	33	–
10-1011	17	0.00	Yes	0	9
01-1112	16	0.00	Yes	0	8
02-1112	14	0.00	Yes	0	8
04-1112	41	0.00	No	18	9
03-1112	27	0.00	No	4	9
05-1112	21	27.07	Yes	0	7
06-1112	18	153.81	No	1	8
06-1112-inf	78	nosol	–	33	–
07-1112	17	0.00	Yes	0	9
08-1112	13	560.38	Yes	0	7
09-1112	101	0.00	No	26	9
10-1112	16	0.00	Yes	0	8
01-1213	17	0.00	Yes	0	10
02-1213	14	0.00	Yes	0	8
04-1213	41	0.00	No	18	9
04-1213-inf	80	nosol	–	33	–
04-1213-inf-m	82	nosol	–	33	–
03-1213	26	0.00	No	4	9
05-1213	23	35.78	Yes	0	8
06-1213	18	0.11	No	1	8
07-1213	19	0.00	Yes	0	9
08-1213	13	279.50	Yes	0	8
09-1213	86	0.00	No	26	9
10-1213	14	0.00	Yes	0	9
01-1314	17	0.00	Yes	0	8
02-1314	15	0.00	Yes	0	9
04-1314	33	0.00	No	4	8
03-1314	26	0.00	No	4	9
05-1314	27	225.91	No	2	8
06-1314	17	123.70	No	1	8
07-1314	17	0.00	Yes	0	9
08-1314	12	292.22	Yes	0	9
09-1314	28	0.03	No	4	8
10-1314	14	0.00	Yes	0	8

Table A4. Detailed results of the MPEC approach on test set AB6 (instances with violated SOMs in *italics*).

Name	Time	Status	Status 1	Time 1	Status 2	Time 2	Validation Time
01-1011	33	nosol	optimal	3	infeasible	2	–
02-1011	35	<i>0.00</i>	optimal	4	optimal	1	9
04-1011	36	nosol	optimal	4	infeasible	2	–
03-1011	29	nosol	optimal	2	infeasible	2	–
05-1011	36	nosol	optimal	5	infeasible	1	–
06-1011	37	nosol	optimal	4	infeasible	7	–
07-1011	35	<i>55.28</i>	optimal	3	optimal	2	11
08-1011	35	nosol	optimal	5	infeasible	1	–
09-1011	28	nosol	optimal	2	optimal	2	–
09-1011-inf	21	nosol	infeasible	3	–	–	–
10-1011	36	nosol	optimal	4	infeasible	1	–
01-1112	32	nosol	optimal	3	optimal	1	–
02-1112	34	nosol	optimal	4	infeasible	3	–
04-1112	29	nosol	optimal	3	optimal	1	–
03-1112	32	nosol	optimal	3	optimal	1	–
05-1112	31	nosol	optimal	2	infeasible	1	–
06-1112	36	nosol	optimal	5	infeasible	3	–
06-1112-inf	32	nosol	optimal	5	infeasible	1	–
07-1112	33	nosol	optimal	2	optimal	2	–
08-1112	31	nosol	optimal	4	infeasible	1	–
09-1112	32	nosol	optimal	4	infeasible	2	–
10-1112	34	<i>1.62</i>	optimal	5	optimal	1	8
01-1213	32	nosol	optimal	1	infeasible	2	–
02-1213	31	<i>0.00</i>	optimal	3	optimal	1	10
04-1213	32	nosol	optimal	3	infeasible	2	–
04-1213-inf	29	nosol	optimal	2	infeasible	2	–
04-1213-inf-m	30	nosol	optimal	2	infeasible	1	–
03-1213	32	nosol	optimal	4	optimal	1	–
05-1213	34	<i>978.74</i>	optimal	4	optimal	2	9
06-1213	32	nosol	optimal	3	infeasible	2	–
07-1213	24	<i>0.00</i>	optimal	4	optimal	1	10
08-1213	22	<i>1618.63</i>	optimal	3	optimal	1	9
09-1213	20	nosol	optimal	4	optimal	0	–
10-1213	22	nosol	optimal	2	infeasible	1	–
01-1314	22	<i>0.00</i>	optimal	2	optimal	1	8
02-1314	22	nosol	optimal	2	infeasible	1	–
04-1314	19	nosol	optimal	3	optimal	1	–
03-1314	22	nosol	optimal	4	optimal	1	–
05-1314	24	nosol	optimal	3	infeasible	2	–
06-1314	26	nosol	optimal	7	infeasible	2	–
07-1314	24	<i>0.00</i>	optimal	4	optimal	1	9
08-1314	25	nosol	optimal	5	infeasible	2	–
09-1314	23	nosol	optimal	6	infeasible	1	–
10-1314	25	<i>0.00</i>	optimal	5	optimal	1	8