COMPUTATIONAL OPTIMIZATION OF GAS COMPRESSOR STATIONS: MINLP MODELS VS. CONTINUOUS REFORMULATIONS

DANIEL ROSE[1], MARTIN SCHMIDT[2], MARC C. STEINBACH[1], BERNHARD M. WILLERT

Abstract. When considering cost-optimal operation of gas transport networks, compressor stations play the most important role. Proper modeling of these stations leads to complicated mixed-integer nonlinear and nonconvex optimization problems. In this article, we give an isothermal and stationary description of compressor stations, state MINLP and GDP models for operating a single station, and discuss several continuous reformulations of the problem. The applicability and relevance of different model formulations, especially of those without discrete variables, is demonstrated by a computational study on both academic examples and real-world stations.

1. Introduction

Natural gas is one of the most important energy sources. In 2013, it accounted for 25 % of the fossil energy used in Europe [18]. It is used in industrial processes, for heating, and, more recently, for natural gas vehicles. Especially in Germany, its low price leads to its role as a "bridging energy" during the transition to a future energy mix based primarily on regenerative energy. In Europe, natural gas is transported through pipeline networks with a total length of 100 000 km. Gas transport in pipeline networks is pressure-driven, i.e., the gas flows from higher to lower pressure. Thus, pipeline-based gas transport requires compressor stations. The power required to compress the gas is delivered by drives that use either electrical power or gas from the network itself. The energy consumption of compressors is responsible for a large fraction of the variable operating costs of a gas network.

In this article we focus on the stationary optimization of a single compressor station. More specifically, we consider fixed boundary conditions, i.e., fixed inflow and outflow pressures together with a fixed throughput, and ask the following questions:

- Can the station be operated in a way that satisfies the given boundary conditions? In other words: are those boundary conditions feasible?
- If the boundary conditions are feasible: What is a minimum cost operation that satisfies the boundary condition?

Transport of natural gas has been a rich source of mathematical optimization problems for roughly half a century. The first publications on optimization in gas networks address tree-like or gun-barrel topologies with dynamic programming as a solution technique [54, 55] (see [8] for a later survey). In the following, we give a brief overview on mathematical optimization in this field. More extensive reviews can be found in [25].

Date: February 26, 2015.

2010 *Mathematics Subject Classification.* 90-08,90C11,90C30,90C33,90C90.

Key words and phrases. Discrete-continuous nonlinear optimization, Gas networks, Gas compressor stations, Mixed-integer optimization, Continuous reformulations.

One main branch of research investigates the problems of *minimum cost operation* and *feasibility testing*. These problems require models of complete gas networks comprising various types of elements like pipes, compressors, (control) valves, etc. The combination of nonlinear gas physics with the switching of controllable network elements typically leads to mixed-integer nonlinear optimization models (MINLPs); see, e.g., [10, 12]. One standard approach for tackling these MINLPs is the application of (piecewise) linearizations of the nonlinearities in order to reduce the problems to mixed-integer *linear* models (MILPs) [21, 22, 31, 32, 33]. Other investigations focus on the nonlinear aspects under fixed discrete controls [16, 17, 43, 44, 46], or attempt to approximate discrete aspects by continuous reformulations [40, 41]. For more references and reviews in the areas of cost minimization and feasibility testing we refer to [25].

The second major research branch that we want to highlight considers selected types of network elements and studies them in more detail. For pipes, theoretical studies include the controllability and stabilization of the governing system of partial differential equations, the Euler equations (cf. [1, 2, 5]). The best-investigated problem for compressor stations is the one considered in this paper, which is of nonconvex MINLP type: minimum cost operation under given boundary conditions [7, 9, 24, 36, 56]. See also [28] for a simulation based model of compressor stations and [35] for a recent survey on modeling centrifugal gas compressors. As already mentioned, the compressor stations are the dominant variable cost factor in gas network operation.

In this article we study discrete-continuous models for the problem of minimum cost compressor station operation. We give an almost complete isothermal description of all relevant devices and their interplay. This description is comparable in accuracy with isothermal simulation models. However, we neglect some minor model aspects that would only complicate the presentation without influencing the solutions significantly. The specific structure of the resulting MINLP allows for a large variety of continuous reformulations that will be discussed in detail and applied to the problem under consideration. The approach of continuous reformulation is in line with recent publications that develop MPEC techniques for reformulating other discrete aspects of network optimization models [25, 37, 40, 41]. Thus, by combining the MPEC techniques from the cited publications with the reformulation schemes discussed in this paper, it is possible to formulate purely continuous NLP type models of problems of minimum cost operation or feasibility testing. The outcome of this achievement is twofold. First, it allows us to state highly detailed models of gas networks. This in particular is not viable with approaches based on linearizations since the resulting MILP models tend to be very hard for state-of-the-art MILP solvers. Second, it allows us to solve the resulting models with local NLP solvers, which are typically faster than global MI(N)LP solvers. Obviously, this comes at the price of finding only locally optimal solutions. For further applications of continuous reformulations of discrete-continuous optimization problems, especially in the field of process engineering, see [3, 26, 27, 45].

The paper is organized as follows. The problem of optimizing a gas compressor station under steady-state boundary conditions is presented in Sect. 2. Afterwards, Sect. 3 introduces a mixed-integer nonlinear formulation of the problem and briefly discusses an equivalent general disjunctive programming formulation. Then, in Sect. 4, the concept of pseudo NCP functions is introduced and a collection of continuous reformulation techniques is discussed that can be applied to the MINLP model of Sect. 3. These reformulations are applied to artificial and real-world instances of the problem in Sect. 5 and the solutions are compared. Finally, Sect. 6 gives some remarks on further work and open questions.

TABLE 1. Principal physical quantities and constants

Symbol	Explanation	Unit
p	Gas pressure	Pa
T	Gas temperature	K
ρ	Gas density	$\mathrm{kg\,m^{-3}}$
z	Compressibility factor	1
q	Mass flow rate	$\mathrm{kg\,s^{-1}}$
Q	Volumetric flow rate ($Q = q/\rho$)	$\mathrm{m^3\,s^{-1}}$
m	Average molar mass of gas mixture	$\mathrm{kg\,mol^{-1}}$
p_c	Pseudocritical pressure of gas mixture	Pa
T_c	Pseudocritical temperature of gas mixture	K
R	Universal gas constant	$\mathrm{J\,mol^{-1}\,K^{-1}}$
R_s	Specific gas constant ($R_\mathrm{s} = R/m$)	$\mathrm{J\,kg^{-1}\,K^{-1}}$

2. Problem Description

A compressor station hosts a fixed number of compressor machines called *units*. It can operate in finitely many discrete *states* that arise from the three *operation modes* (*closed*, *bypass mode*, and *active*) and a certain number of *configurations* in the active mode. Every configuration consists of a serial combination of parallel arrangements of compressor units, see Fig. 2. Every compressor unit has an associated *drive* that provides the power for compressing the gas. For ease of exposition we assume that every drive powers just a single compressor unit. We also neglect the frictional pressure loss caused by station piping, measurement devices, etc., which is usually modeled by fictitious elements called *resistors*.

In this section we present the required models of all types of compressor machines and drives and then describe their interplay. Full details can be found in [43] where models of all network elements have been developed. For later reference, the models are presented in constraint form, with constraint functions being denoted by c and super-indexed with an abbreviated name indicating the semantics of the constraint.

2.1. Physical Quantities.

The dynamics of natural gas is modeled in terms of the *mass flow* q, *pressure* p, *temperature* T, and *density* ρ, where p, T, ρ are coupled by an *equation of state*. In specific, we use the *thermodynamical standard equation*,

$$\rho(p, T, z) = \frac{p}{R_\mathrm{s} z T}.$$

Here the *compressibility factor* z models the deviation between real and ideal gas, for which we use an empirical formula of the American Gas Association (AGA),

$$z(p, T) = 1 + 0.257 p/p_\mathrm{c} - 0.533 \frac{p/p_\mathrm{c}}{T/T_\mathrm{c}}.$$

We will later formulate certain constraints using $\rho(p, T, z)$ and $z(p, T)$ with fixed T.

Finally we need the *volumetric flow* Q, given by the constraint

$$0 = c^{\text{flow-conv}}(q, \rho, Q) = q - \rho Q.$$

See Table 1 (or [43]) for explanations of the physical quantities and constants.

2.2. Boundary Conditions.

We are interested in feasibility testing and in computing cost optimal controls of a compressor station for given boundary values. In accordance with our general network models [43], a compressor station is seen as an

TABLE 2. Compressor quantities

Symbol	Explanation	Unit
H_{ad}	Specific change in adiabatic enthalpy	$\mathrm{kg\,J^{-1}}$
η_{ad}	Adiabatic efficiency	1
P	Compressor input power	W
q^{fc}	Fuel consumption	$\mathrm{kg\,s^{-1}}$
b	Specific energy consumption	W
H_{u}	Lower calorific value	$\mathrm{J\,mol^{-1}}$
n	Compressor speed	$\mathrm{s^{-1}}$
κ	Isentropic exponent	1
M	Shaft torque	$\mathrm{N\,m}$
V_{op}	Operating volume	$\mathrm{m^3}$

arc $a = (u, v)$ of a directed network graph $G = (\mathbb{V}, \mathbb{A})$. The *boundary conditions* at which the compressor station a has to operate consist of a mass flow range,

$$q_a \in [q_a^-, q_a^+],$$

and of inlet and outlet pressure ranges,

$$p_u \in [p_u^-, p_u^+], \quad p_v \in [p_v^-, p_v^+].$$

Each of the values q_a, p_u, p_v can be fixed by setting its bounds to identical values.

2.3. **Compressor Machines.** We distinguish *turbo compressors* and *piston compressors* and start with the common parts of their models. For an overview of relevant compressor quantities see Table 2.

The compression of a gas stream from p_{in} to p_{out} results in a *specific change in adiabatic enthalpy*, H_{ad}:

$$0 = c^{\text{ad-head}}(H_{\mathrm{ad}}, p_{\mathrm{in}}, p_{\mathrm{out}}, z_{\mathrm{in}}) = H_{\mathrm{ad}} - z_{\mathrm{in}} T_{\mathrm{in}} R_{\mathrm{s}} \frac{\kappa}{\kappa - 1} \left(\left(\frac{p_{\mathrm{out}}}{p_{\mathrm{in}}} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right).$$

Here $z_{\mathrm{in}} = z(p_{\mathrm{in}}, T_{\mathrm{in}})$, and we assume that the *isentropic exponent* κ is constant. The *power* P required for compressing the mass flow q is given by

$$0 = c^{\text{power}}(P, q, H_{\mathrm{ad}}, \eta_{\mathrm{ad}}) = P - \frac{q H_{\mathrm{ad}}}{\eta_{\mathrm{ad}}}.$$

Its upper limit is the *maximal power* P^+ that the associated drive can deliver:

$$0 \le c^{\text{power-limit}}(P, P^+) = P^+ - P. \tag{1}$$

2.4. **Turbo Compressors.** Turbo compressors are modeled by *characteristic diagrams* in (Q, H_{ad})-space, see Fig. 1 (left). The curves are defined using biquadratic and quadratic polynomials $F_2(x, y; A)$ and $F_1(z; b)$ whose coefficients A and b are obtained from least-squares fits for given measurements of the compressor:

$$F_2(x, y; A) = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}^T \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{12} & a_{22} \end{bmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \end{pmatrix}$$

and

$$F_1(z; b) = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}^T \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix}.$$

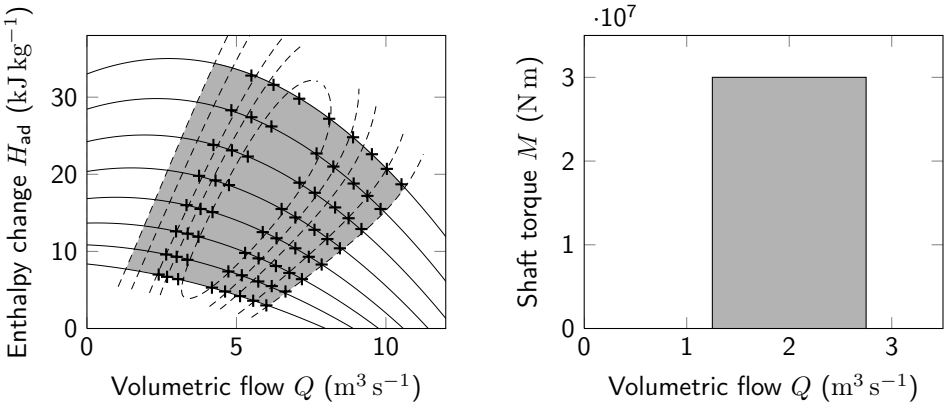

FIGURE 1. Characteristic diagrams of a turbo compressor (left) and a piston compressor (right)

The isolines of speed n (solid) and adiabatic efficiency η_{ad} (dashed) are given by:

$$0 = c^{\text{iso-speed}}(H_{\text{ad}}, Q, n) = H_{\text{ad}} - F_2(Q, n; A_{H_{\text{ad}}}),$$
$$0 = c^{\text{iso-eff}}(\eta_{\text{ad}}, Q, n) = \eta_{\text{ad}} - F_2(Q, n; A_{\eta_{\text{ad}}}).$$

The compressor's *operating range* (grey area) is bounded by the isolines of minimal and maximal speeds $n^{\pm}$, by the *surgeline* (left), and by the *chokeline* (right),

$$0 \le c^{\text{surge}}(Q, H_{\text{ad}}) = F_1(Q; b_{\text{surge}}) - H_{\text{ad}},$$
$$0 \le c^{\text{choke}}(Q, H_{\text{ad}}) = H_{\text{ad}} - F_1(Q; b_{\text{choke}}).$$

2.5. **Piston Compressors.** Piston compressors are modeled by characteristic diagrams in (Q, M)-space, see Fig. 1 (right). The volumetric flow Q satisfies

$$0 = c^{\text{op-vol}}(Q, n) = Q - V_{\text{op}} n,$$

and the *shaft torque* M (with constant efficiency η_{ad}) is given by

$$0 = c^{\text{torque}}(M, H_{\text{ad}}, \rho_{\text{in}}) = M - \frac{V_{\text{op}} H_{\text{ad}}}{2\pi\eta_{\text{ad}}} \rho_{\text{in}}.$$

The speed bounds $n \in [n^-, n^+]$ induce left and right bounds of the operating range, $Q \in [V_{\text{op}} n^-, V_{\text{op}} n^+]$. The upper limit can be given in one of three forms depending on the specific compressor: either by a maximum compression ratio ε^+,

$$0 \le c^{\text{limit}}(p_{\text{in}}, p_{\text{out}}) = \varepsilon^+ - \frac{p_{\text{out}}}{p_{\text{in}}},$$

by a maximum pressure increase Δp^+,

$$0 \le c^{\text{limit}}(p_{\text{in}}, p_{\text{out}}) = p_{\text{in}} + \Delta p^+ - p_{\text{out}},$$

or by a maximum shaft torque M^+,

$$0 \le c^{\text{limit}}(M) = M^+ - M.$$

2.6. **Drives.** The four most frequently used drive types are *gas turbines*, *gas driven motors*, *electric motors*, and *steam turbines*. Here we focus on the common model aspects and consider a *generic* "catchall" drive model that incorporates the two main aspects: the maximal power P^+ that the drive can deliver (cf. (1)),

$$0 = c^{\text{max-power}}(P^+, n) = P^+ - F_2(n, T_{\text{amb}}; A_{P^+}), \tag{2}$$

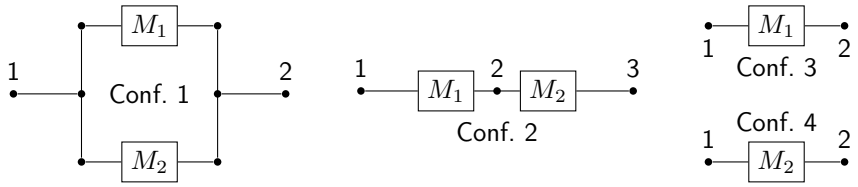

FIGURE 2. Complete set of possible configurations of a compressor station with two machines (a real station may offer only a subset)

and its specific energy consumption b,

$$0 = c^{\text{spec-ener-cons}}(b, P) = b - F_1(P; a_b). \tag{3}$$

For specific drive types, (2) or (3) may simplify or vanish completely. The fuel consumption of a gas-powered drive is the mass flow q^{fc},

$$0 = c^{\text{fuel-cons}}(q^{\text{fc}}, b) = q^{\text{fc}} - \frac{bm}{H_{\text{u}}}.$$

The fuel consumption of an electric drive is zero, $0 = c^{\text{fuel-cons}}(q^{\text{fc}}, b) = q^{\text{fc}}$.

2.7. **Configurations.** Recall that a compressor station can be closed, in bypass mode, or active. We denote the set of configurations of an active compressor station by $\mathcal{C} = \{1, \dots, n_{\mathcal{C}}\}$, the set of serial stages of configuration $i \in \mathcal{C}$ by $\mathcal{S}_i = \{1, \dots, n_{\mathcal{S}_i}\}$, and the set of parallel units of stage $j \in \mathcal{S}_i$ by $\mathcal{P}_{ij} = \{1, \dots, n_{\mathcal{P}_{ij}}\}$. For instance, in Fig. 2 we have $\mathcal{C} = \{1, \dots, 4\}$ and

$$n_{\mathcal{S}_1} = n_{\mathcal{S}_3} = n_{\mathcal{S}_4} = 1, \quad n_{\mathcal{S}_2} = 2,$$
$$n_{\mathcal{P}_{11}} = 2, \quad n_{\mathcal{P}_{21}} = n_{\mathcal{P}_{22}} = n_{\mathcal{P}_{31}} = n_{\mathcal{P}_{41}} = 1.$$

In what follows, we refer to individual compressor units by the index triple ijk.

Given a triple (q_a, p_u, p_v) of boundary values, the selected configuration i must satisfy the following: the flow q_j of every serial stage $j \in \mathcal{S}_i$ equals q_a and is distributed over the $n_{\mathcal{P}_{ij}}$ parallel units:

$$q_a = q_j = \sum_{k \in \mathcal{P}_{ij}} q_{ijk} \quad \text{for all} \quad j \in \mathcal{S}_i.$$

Moreover, all parallel units in stage $j \in \mathcal{S}_i$ have to increase the gas pressure to a common outflow value $p_{i,j+1}$, which becomes the inflow pressure of stage $j+1$. The first and last stages must satisfy $p_{i1} = p_u$ and $p_{i,n_{\mathcal{S}_i}+1} = p_v$, respectively.

2.8. **Objective Function.** There are various reasonable objective functions in our context, which can mainly be categorized as *feasibility* or *optimization goals*. If one merely wishes to know whether given boundary conditions are feasible, it suffices to use a zero objective, $f \equiv 0$. If the boundary conditions are infeasible, it may be useful to add slack variables to a certain set of constraints and minimize the total infeasibility, measured by a suitable norm of the vector of slack variables. The reader interested in problem-specific slack variable formulations for gas network planning is referred to [44].

If one is convinced of having feasible boundary conditions, it is straightforward to minimize operating costs, power, or fuel consumption. Specific objective functions will be formulated after stating the optimization models in the following section.

3. Mixed-Integer and General Disjunctive Programming Models

The most natural way to model the problem described in the previous section is presented in Sect. 3.1: a mixed-integer nonlinear program (MINLP) that incorporates binary variables for the discrete states of the compressor station. An equivalent general disjunctive programming (GDP) formulation of the problem is given in Sect. 3.2. Continuous reformulations of the MINLP and GDP models are discussed later in Sect. 4.

One further possibility of tackling the problem is to enumerate all states of the compressor station and solve all the resulting (continuous) problems by a local or global method. However, this is only suitable when single compressor stations are considered, whereas continuous reformulations can also be used for models of the complete transport network. Another way is to apply suitable heuristics, see, e.g., [42].

We denote individual continuous variables by x and discrete ones by s. Variable vectors are written with bold letters, like $\mathbf{x}$ or $\mathbf{s}$. Sub-indices refer to corresponding elements of the compressor station or to sets of elements. In the constraint notation of Sect. 2, we now also use sub-indices with the same meaning as for variables. Additional sub-indices $\mathcal{E}$ or $\mathcal{I}$ distinguish equality and inequality constraints.

3.1. MINLP Formulation.

With $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s} \in \{0,1\}^m$, the optimization problem under consideration takes the general MINLP form

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

The boundary conditions correspond to the variable sub-vector $\mathbf{x}_{\mathcal{B}} = (q_a, p_u, p_v)$.

For a suitable formulation of the discrete decisions and their implications on other parts of the model, we review the concept of *indicator constraints*.

3.1.1. *Indicator Constraints.*

Indicator constraints use a binary *indicator variable* to control whether a certain constraint is enabled or disabled. If $c(\mathbf{x})$ is any equality or inequality constraint with indicator variable s, we denote the associated indicator constraints generically by

$$c^{\text{ind}}_{\mathcal{E}}(c(\mathbf{x}), s) = 0, \quad c^{\text{ind}}_{\mathcal{I}}(c(\mathbf{x}), s) \geq 0. \tag{4}$$

The formulation (4) is chosen in such a way that the original constraint $c(\mathbf{x}) = 0$ or $c(\mathbf{x}) \geq 0$ must hold if $s = 1$ whereas it becomes irrelevant if $s = 0$. In this article we use two specific types: *big-M* and *bilinear* indicator constraints. The big-M indicator constraints of the equality constraint $c(\mathbf{x}) = 0$ take the form

$$-M_c^-(1-s) \leq c(\mathbf{x}) \leq M_c^+(1-s). \tag{5}$$

For the inequality constraint $c(\mathbf{x}) \geq 0$ we obtain the single inequality

$$-M_c^-(1-s) \leq c(\mathbf{x}). \tag{6}$$

Bilinear indicator constraints for equality and inequality constraints, respectively, are given by

$$0 = sc(\mathbf{x}) \quad \text{and} \quad 0 \leq sc(\mathbf{x}). \tag{7}$$

The big-M formulation offers the advantage that convex constraints yield convex indicator constraints. However, the big-M constants must be chosen sufficiently large to ensure that the constraint is properly enabled or disabled: $M_c^- \geq |\min_{\mathbf{x}} c(\mathbf{x})|$ and $M_c^+ \geq |\max_{\mathbf{x}} c(\mathbf{x})|$ in (5), and $M_c^- \geq |\min_{\mathbf{x}} c(\mathbf{x})|$ in (6). Numerical difficulties are to be expected if $M_c^{\pm}$ are very large. The bilinear formulation avoids this but has the disadvantage that even convex constraints yield nonconvex indicator constraints. We later review these issues when discussing the numerical results of different problem formulations in Sect. 5.

Finally we need two more notions: an *indicator expression* y is a term whose value is $y \geq 1$ if the associated state is enabled and zero otherwise, like s in (7). Conversely, a *negation expression* is a term whose value is $y \geq 1$ if the associated state is disabled and zero otherwise, like $1 - s$ in (5), (6).

3.1.2. *Discrete States.* To represent the set of states of a compressor station, we introduce the binary variable vector

$$\mathbf{s} = ((s_i)_{i\in\mathcal{C}}, s_{\mathrm{bp}}, s_{\mathrm{cl}}) \in \{0,1\}^m, \quad m = |\mathcal{C}| + 2,$$

with the following interpretation:

- the station is active in configuration $i \in \mathcal{C}$ if and only if $s_i = 1$;
- the station is in bypass mode if and only if $s_{\mathrm{bp}} = 1$;
- the station is closed if and only if $s_{\mathrm{cl}} = 1$.

Since precisely one state must be selected, we add the special-ordered-set-1 (SOS-1) constraint

$$0 = c^{\text{op-mode}}(\mathbf{s}) = 1 - \left(s_{\mathrm{bp}} + s_{\mathrm{cl}} + \sum_{i\in\mathcal{C}} s_i \right).$$

In bypass mode, the inflow and outflow pressures have to be identical,

$$0 = c^{\mathrm{bypass}}(p_u, p_v) = p_u - p_v,$$

yielding indicator constraints

$$0 = c_{\mathcal{E}}^{\text{bypass-ind}}(c^{\mathrm{bypass}}(p_u, p_v), s_{\mathrm{bp}}), \quad 0 \leq c_{\mathcal{I}}^{\text{bypass-ind}}(c^{\mathrm{bypass}}(p_u, p_v), s_{\mathrm{bp}}).$$

Likewise, if the compressor station is closed, we have

$$0 = c^{\mathrm{closed}}(q_a) = q_a,$$

yielding

$$0 = c_{\mathcal{E}}^{\text{closed-ind}}(c^{\mathrm{closed}}(q_a), s_{\mathrm{cl}}), \quad 0 \leq c_{\mathcal{I}}^{\text{closed-ind}}(c^{\mathrm{closed}}(q_a), s_{\mathrm{cl}}).$$

3.1.3. *Configurations.* Now we turn to the details of individual configurations. The continuous variable vector of configuration $i \in \mathcal{C}$ reads

$$\mathbf{x}_i = \begin{pmatrix} q_i^{\mathrm{fc}} \\ (p_{ij})_{j=1,\dots,n_{\mathcal{S}_i}+1} \\ (z_{ij})_{j\in\mathcal{S}_i} \\ (\rho_{ij})_{j\in\mathcal{S}_i} \\ (\mathbf{x}_{ijk})_{j\in\mathcal{S}_i, k\in\mathcal{P}_{ij}} \end{pmatrix}.$$

Here q_i^{fc} is the total fuel consumption, p_{ij} and $p_{i,n_{\mathcal{S}_i}+1}$ denote the inflow pressure of stage j and the outflow pressure of the last stage, respectively, and z_{ij}, ρ_{ij} represent the inflow compressibility factor and inflow gas density of stage j. Finally, $\mathbf{x}_{ijk}$ is the variable vector of compressor unit ijk and its associated drive, see Sect. 3.1.4.

The inflow quantities z_{ij} and ρ_{ij} are coupled to the physical inflow conditions by the constraints

$$0 = c_{ij}^{\mathrm{compr}}(z_{ij}, p_{ij}) = z_{ij} - z(p_{ij}, T),$$
$$0 = c_{ij}^{\mathrm{dens}}(\rho_{ij}, p_{ij}, z_{ij}) = \rho_{ij} - \rho(p_{ij}, T, z_{ij}).$$

The total fuel consumption q_i^{fc} is the sum over all compressor units:

$$0 = c_i^{\text{fuel-1}}(q_i^{\mathrm{fc}}, (q_{ijk}^{\mathrm{fc}})_{j\in\mathcal{S}_i, k\in\mathcal{P}_{ij}}) = q_i^{\mathrm{fc}} - \sum_{j\in\mathcal{S}_i} \sum_{k\in\mathcal{P}_{ij}} q_{ijk}^{\mathrm{fc}}.$$

This constraint must be enabled if and only if configuration i is active:

$$0 = c^{\text{ind-fuel-1}}_{\mathcal{E},i}(c^{\text{fuel-1}}_i(q^{\text{fc}}_i, (q^{\text{fc}}_{ijk})_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}}), s_i),$$
$$0 \le c^{\text{ind-fuel-1}}_{\mathcal{I},i}(c^{\text{fuel-1}}_i(q^{\text{fc}}_i, (q^{\text{fc}}_{ijk})_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}}), s_i).$$

Otherwise configuration i is inactive,

$$0 \le c^{\text{fuel-0}}_i(q^{\text{fc}}_i, s_i) = s_i(q^{\text{fc}}_i)^+ - q^{\text{fc}}_i,$$

where $(q^{\text{fc}}_i)^+$ is a suitable upper bound of q^{fc}_i. Next, the inflow and outflow pressures of the active configuration must equal the inflow and outflow pressures of the station:

$$0 = c^{\text{ind-p-in}}_{\mathcal{E},i}(c^{\text{p-in}}_i(p_{i1}, p_u), s_i),$$
$$0 \le c^{\text{ind-p-in}}_{\mathcal{I},i}(c^{\text{p-in}}_i(p_{i1}, p_u), s_i),$$
$$0 = c^{\text{ind-p-out}}_{\mathcal{E},i}(c^{\text{p-out}}_i(p_{i,n_{\mathcal{S}_i}+1}, p_v), s_i),$$
$$0 \le c^{\text{ind-p-out}}_{\mathcal{I},i}(c^{\text{p-out}}_i(p_{i,n_{\mathcal{S}_i}+1}, p_v), s_i),$$

where

$$0 = c^{\text{p-in}}_i(p_{i1}, p_u) = p_{i1} - p_u,$$
$$0 = c^{\text{p-out}}_i(p_{i,n_{\mathcal{S}_i}+1}, p_v) = p_{i,n_{\mathcal{S}_i}+1} - p_v.$$

The flow distribution over the parallel units of every stage j finally completes the configuration model:

$$0 = c^{\text{flow}}_{ij}(q_a, (q_{ijk})_{k\in\mathcal{P}_{ij}}) = q_a - \sum_{k\in\mathcal{P}_{ij}} q_{ijk}, \quad j \in \mathcal{S}_i.$$

In summary, the equality constraints of configuration i read

$$0 = c_{\mathcal{E},i}(\mathbf{x}_{\mathcal{B}}, \mathbf{x}_i, s_i) = \begin{pmatrix} (c^{\text{compr}}_{ij}(z_{ij}, p_{ij}))_{j\in\mathcal{S}_i} \\ (c^{\text{dens}}_{ij}(\rho_{ij}, p_{ij}, z_{ij}))_{j\in\mathcal{S}_i} \\ c^{\text{ind-fuel-1}}_{\mathcal{E},i}(q^{\text{fc}}_i, (q^{\text{fc}}_{ijk})_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}}, s_i) \\ c^{\text{ind-p-in}}_{\mathcal{E},i}(p_{i1}, p_u, s_i) \\ c^{\text{ind-p-out}}_{\mathcal{E},i}(p_{i,n_{\mathcal{S}_i}+1}, p_v, s_i) \\ (c^{\text{flow}}_{ij}(q_a, (q_{ijk})_{k\in\mathcal{P}_{ij}}))_{j\in\mathcal{S}_i} \end{pmatrix}$$

and the inequality constraints are given by

$$0 \le c_{\mathcal{I},i}(\mathbf{x}_{\mathcal{B}}, \mathbf{x}_i, s_i) = \begin{pmatrix} c^{\text{ind-fuel-1}}_{\mathcal{I},i}(q^{\text{fc}}_i, (q^{\text{fc}}_{ijk})_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}}, s_i) \\ c^{\text{ind-fuel-0}}_{\mathcal{I},i}(q^{\text{fc}}_i, s_i) \\ c^{\text{ind-p-in}}_{\mathcal{I},i}(p_{i1}, p_u, s_i) \\ c^{\text{ind-p-out}}_{\mathcal{I},i}(p_{i,n_{\mathcal{S}_i}+1}, p_v, s_i) \end{pmatrix}.$$

3.1.4. *Compressor Units and Drives.* Lastly, we formulate MINLP models of turbo and piston compressors within a configuration. The continuous variable vector of a turbo compressor $\nu = ijk$ with its associated drive reads

$$\mathbf{x}_\nu = (H_{\text{ad},\nu}, q_\nu, Q_\nu, n_\nu, P_\nu, q^{\text{fc}}_\nu, b_\nu, P^+_\nu, \eta_{\text{ad},\nu}),$$

and for a piston compressor ν with associated drive it reads

$$\mathbf{x}_\nu = (H_{\text{ad},\nu}, q_\nu, Q_\nu, n_\nu, P_\nu, q^{\text{fc}}_\nu, b_\nu, P^+_\nu, M_\nu).$$

The equality constraints of a turbo compressor are then given by

$$0 = c_{\mathcal{E},\nu}(\mathbf{x}_i, s_i) = \begin{pmatrix} c_\nu^{\text{flow-conv}}(q_\nu, \rho_{ij}, Q_\nu) \\ c_\nu^{\text{ad-head}}(H_{\text{ad},\nu}, p_{ij}, p_{i,j+1}, z_{ij}) \\ c_\nu^{\text{fuel-cons}}(q_\nu^{\text{fc}}, b_\nu) \\ c_{\mathcal{E},\nu}^{\text{ind-power}}(P_\nu, q_\nu, H_{\text{ad},\nu}, \eta_{\text{ad},\nu}, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-iso-speed}}(H_{\text{ad},\nu}, Q_\nu, n_\nu, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-iso-eff}}(\eta_{\text{ad},\nu}, Q_\nu, n_\nu, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-max-power}}(P_\nu^+, n_\nu, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-spec-ener-cons}}(b_\nu, P_\nu, s_i) \end{pmatrix},$$

and its inequality constraints read

$$0 \le c_{\mathcal{I},\nu}(\mathbf{x}_i, s_i) = \begin{pmatrix} c_{\mathcal{I},\nu}^{\text{ind-surge}}(Q_\nu, H_{\text{ad},\nu}, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-choke}}(Q_\nu, H_{\text{ad},\nu}, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-power}}(P_\nu, q_\nu, H_{\text{ad},\nu}, \eta_{\text{ad},\nu}, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-iso-speed}}(H_{\text{ad},\nu}, Q_\nu, n_\nu, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-iso-eff}}(\eta_{\text{ad},\nu}, Q_\nu, n_\nu, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-power-limit}}(P_\nu, P_\nu^+, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-max-power}}(P_\nu^+, n_\nu, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-spec-ener-cons}}(b_\nu, P_\nu, s_i) \end{pmatrix}.$$

Here we introduce indicator constraints to disable the associated limits of physical and technical quantities of inactive configurations: otherwise *irrelevant* infeasible values of those inactive configurations would render the entire compressor model infeasible.

The equality constraints of a piston compressor are given by

$$0 = c_{\mathcal{E},\nu}(\mathbf{x}_i, s_i) = \begin{pmatrix} c_\nu^{\text{flow-conv}}(q_a, \rho_{ij}, Q_\nu) \\ c_\nu^{\text{ad-head}}(H_{\text{ad},\nu}, p_{ij}, p_{i,j+1}, z_{ij}) \\ c_\nu^{\text{fuel-cons}}(q_\nu^{\text{fc}}, b_\nu) \\ c_\nu^{\text{torque}}(M_\nu, H_{\text{ad},\nu}, \rho_{ij}) \\ c_{\mathcal{E},\nu}^{\text{ind-power}}(P_\nu, q_\nu, H_{\text{ad},\nu}, \eta_{\text{ad},\nu}, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-op-vol}}(Q_\nu, n_\nu, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-max-power}}(P_\nu^+, n_\nu, s_i) \\ c_{\mathcal{E},\nu}^{\text{ind-spec-ener-cons}}(b_\nu, P_\nu, s_i) \end{pmatrix},$$

and the inequality constraints read

$$0 = c_{\mathcal{I},\nu}(\mathbf{x}_i, s_i) = \begin{pmatrix} c_\nu^{\text{limit}}(p_{ij}, p_{i,j+1}) \text{ or } c_\nu^{\text{limit}}(M_\nu) \\ c_{\mathcal{I},\nu}^{\text{ind-power}}(P_\nu, q_\nu, H_{\text{ad},\nu}, \eta_{\text{ad},\nu}, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-op-vol}}(Q_\nu, n_\nu, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-power-limit}}(P_\nu, P_\nu^+, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-max-power}}(P_\nu^+, n_\nu, s_i) \\ c_{\mathcal{I},\nu}^{\text{ind-spec-ener-cons}}(b_\nu, P_\nu, s_i) \end{pmatrix}.$$

Here the indicator constraints are introduced for the same reason as for turbo compressors.

3.1.5. *Model Summary.* As already mentioned, we may choose the objective $f \equiv 0$ if we are only interested in feasibility testing. If we aim at minimum cost operation, a suitable objective is

$$f(\mathbf{x}) = \sum_{i \in \mathcal{C}} s_i \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{P}_{ij} \cap \mathcal{U}^{\text{fuel}}} \omega^{\text{fuel}} q_{ijk}^{\text{fc}} + \sum_{i \in \mathcal{C}} s_i \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{P}_{ij} \cap \mathcal{U}^{\text{el}}} \omega^{\text{el}} P_{ijk}, \tag{8}$$

where $\mathcal{U}^{\text{fuel}}$ and $\mathcal{U}^{\text{el}}$ are the sets of fuel gas and electricity driven compressor units, respectively, and ω^{fuel} and ω^{el} represent fuel gas and electricity costs. The complete MINLP model is then obtained with the variable vector

$$\mathbf{x} = (\mathbf{x}_{\mathcal{B}}, (\mathbf{x}_i)_{i\in\mathcal{C}})$$

and the constraints

$$c_{\mathcal{E}}(\mathbf{x},\mathbf{s}) = \begin{pmatrix} c^{\text{op-mode}}(\mathbf{s}) \\ (c_{\mathcal{E},i}(\mathbf{x}_{\mathcal{B}},\mathbf{x}_i,s_i))_{i\in\mathcal{C}} \\ (c_{\mathcal{E},ijk}(\mathbf{x}_i,s_i))_{i\in\mathcal{C},j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}} \\ c_{\mathcal{E}}^{\text{bypass-ind}}(p_u,p_v,s_{\text{bp}}) \\ c_{\mathcal{E}}^{\text{closed-ind}}(q_a,s_{\text{cl}}) \end{pmatrix},$$

$$c_{\mathcal{I}}(\mathbf{x},\mathbf{s}) = \begin{pmatrix} (c_{\mathcal{I},i}(\mathbf{x}_{\mathcal{B}},\mathbf{x}_i,s_i))_{i\in\mathcal{C}} \\ (c_{\mathcal{I},ijk}(\mathbf{x}_i,s_i))_{i\in\mathcal{C},j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}} \\ c_{\mathcal{I}}^{\text{bypass-ind}}(p_u,p_v,s_{\text{bp}}) \\ c_{\mathcal{I}}^{\text{closed-ind}}(q_a,s_{\text{cl}}) \end{pmatrix}.$$

3.2. **General Disjunctive Programming Formulation.** Here we present a general disjunctive programming (GDP) formulation [23, 39] of the cost minimization problem:

$$\min \quad \gamma_{\text{bp}} + \gamma_{\text{cl}} + \sum_{i\in\mathcal{C}} \gamma_i \tag{9a}$$

$$\text{s.t.} \quad \bigvee_{i\in\mathcal{C}} \begin{pmatrix} s_i \\ \begin{pmatrix} c_{\mathcal{E},i}(\mathbf{x}_{\mathcal{B}},\mathbf{x}_i) \\ (c_{\mathcal{E},ijk}(\mathbf{x}_i))_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}} \end{pmatrix} = 0 \\ \begin{pmatrix} c_{\mathcal{I},i}(\mathbf{x}_{\mathcal{B}},\mathbf{x}_i) \\ (c_{\mathcal{I},ijk}(\mathbf{x}_i))_{j\in\mathcal{S}_i,k\in\mathcal{P}_{ij}} \end{pmatrix} \geq 0 \\ \gamma_i = \omega q_i^{\text{fc}} \end{pmatrix} \tag{9b}$$

$$\vee \begin{pmatrix} s_{\text{bp}} \\ c^{\text{bypass}}(p_u,p_v) = 0 \\ \gamma_{\text{bp}} = 0 \end{pmatrix} \vee \begin{pmatrix} s_{\text{cl}} \\ c^{\text{closed}}(q_a) = 0 \\ \gamma_{\text{cl}} = 0 \end{pmatrix}, \tag{9c}$$

$$s_{\text{bp}} + s_{\text{cl}} + \sum_{i\in\mathcal{C}} s_i = 1. \tag{9d}$$

GDP models like this are generally built from separate sets of "local" constraints and objective terms that are combined in a logical disjunction. Every part of the disjunction is controlled by a decision variable (here s_{bp}, s_{cl}, or s_i, $i \in \mathcal{C}$): the local constraints are enabled if and only if the associated decision variable is true ($s_i = 1$), and the local objective terms γ_i are set to zero otherwise. The SOS-1 constraint (9d) ensures that precisely one decision variable is true. In a general GDP, any set of feasible combinations of decision variables could be defined by a suitable logical expression. Moreover, "global" constraints and objective terms might be added.

4. Continuous Reformulations

As discussed in Sect. 1, it is reasonable to study continuous reformulations of the MINLP and GDP models of Sect. 3.1 and 3.2 in order to tackle optimization problems for compressor stations or entire gas networks with continuous (local) optimization methods, which tend to be faster than global methods for mixed-integer nonlinear and nonconvex problems.

This section discusses five model reformulation schemes from the literature that can be applied to any model with binary variables that exhibit the structure of a logical disjunction, such as the problem under consideration.

First we introduce the concepts of NCP and pseudo NCP functions in Sect. 4.1. Then we present all reformulation schemes and discuss their feasible sets and regularity properties with a view towards solving them by local algorithms.

4.1. **Pseudo NCP Functions.** NCP functions are bivariate functions $\varphi\colon \mathbb{R}^2 \to \mathbb{R}$ with the property that

$$\varphi(a,b) = 0 \iff a,b \geq 0, \quad ab = 0. \tag{10}$$

Classical NCP functions include, e.g., the minimum function $\varphi_{\min}(a,b) = \min(a,b)$ and the Fischer–Burmeister function [19],

$$\varphi_{\mathrm{FB}}(a,b) = \sqrt{a^2+b^2} - (a+b).$$

See [47] and the references therein for an overview of NCP functions. NCP functions can be used to replace binary variables with continuous variables. However, the right-hand side of (10) introduces some difficulties if it is considered from an NLP perspective. The constraints $a,b \geq 0$, $ab = 0$ lead to so-called Mathematical Programs with Equilibrium Constraints (MPECs). The problem is that standard NLP regularity concepts like the linear independence constraint qualification (LICQ) or the Mangasarian–Fromovitz constraint qualification (MFCQ) are violated at every feasible point of an MPEC [57] (see [30] for an overview of the theory and applications of MPECs). However, the nonnegativity constraints in (10) are not needed for the reformulations discussed below. This leads us to extend the class of NCP functions: We call a function $\phi\colon \mathbb{R}^2 \to \mathbb{R}$ a *pseudo NCP function* if

$$\phi(a,b) = 0 \implies a = 0 \quad \text{or} \quad b = 0.$$

In the remainder of this article we use the following two pseudo NCP functions:

$$\phi_{\mathrm{prod}}(a,b) := ab, \quad \phi_{\mathrm{FB}}(a,b) := \varphi_{\mathrm{FB}}(a,b).$$

Note that the Fischer–Burmeister ϕ_{FB} function has a lack of regularity, whereas the product formulation ϕ_{prod} is favorable since we do not have to impose nonnegativity constraints here as in the classical NCP setting.

4.2. **Reformulation Schemes.** In this section we discuss five schemes that allow to reformulate the MINLP and GDP models of Sect. 3.1 and 3.2 with continuous variables and additional smooth constraints. We discuss the geometry of the feasible regions of the resulting continuous models and their regularity properties.

To be applicable in the context of the model of Sect. 3.1, the reformulations have to possess the following properties:

- Every feasible solution of the reformulation has to be uniquely translatable into a feasible solution of the original MINLP (or GDP). This means that there exists a mapping (a left-total right-unique relation) from the feasible set of the reformulation to the feasible set of the original model.
- For every binary variable $s \in \{0,1\}$, the reformulation has variables from which *indicator* and *negation expressions* can be constructed (cf. Sect. 3.1.1).
- The reformulation has variables from which the SOS-1 constraint can be constructed.

In what follows, we only present the representations of a set of binary variables $s_1,\dots,s_m \in \{0,1\}$ together with the SOS-1 constraint $\sum_{i=1}^m s_i = 1$, rather than stating complete reformulated models. The latter can easily be reproduced from Sect. 3.1. We frequently use the sets $\mathcal{M} := \{1,\dots,m\}$ and $\mathcal{M}_i := \mathcal{M} \setminus \{i\}$.

As already stated in Sect. 1, all reformulation schemes below can be found in the existing literature (in particular, see [3, 27, 45]), except that we use pseudo NCP functions rather than NCP functions.

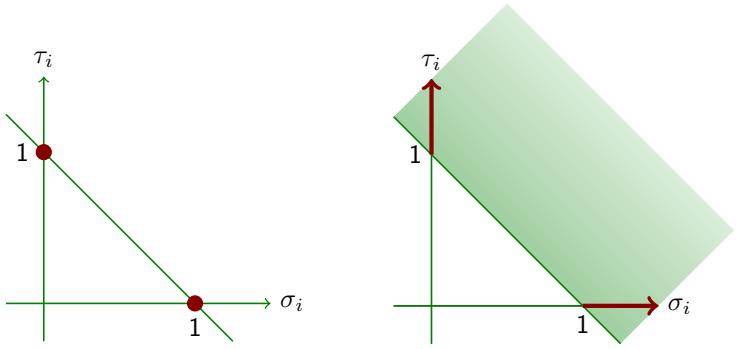

FIGURE 3. Left: Feasible set of the exact bivariate reformulation. The axes are the feasible sets of (11a). Together with (11b) the feasible set $\{(1,0),(0,1)\}$ (bullets) remains. Right: Feasible set of the approximate bivariate reformulation. The axes are the feasible sets of (12a) and the shaded area is the feasible region of (12b). The intersection consists of the two disjoint bold lines.

4.2.1. *Exact Bivariate Reformulation.* The first reformulation requires two continuous variables $(\sigma_i, \tau_i) \in \mathbb{R}^2$ per binary variable s_i together with the constraints

$$\phi(\sigma_i, \tau_i) = 0 \quad \text{for all} \quad i \in \mathcal{M}, \tag{11a}$$
$$\sigma_i + \tau_i = 1 \quad \text{for all} \quad i \in \mathcal{M}, \tag{11b}$$
$$\sum_{i \in \mathcal{M}} \sigma_i = 1, \tag{11c}$$

where ϕ is an arbitrary pseudo NCP function. Clearly, the constraints (11) imply $(\sigma_i, \tau_i) \in \{(1,0),(0,1)\}$, giving the required mapping:

$$(\sigma_i, \tau_i) = (1,0) \mapsto s_i = 1, \quad (\sigma_i, \tau_i) = (0,1) \mapsto s_i = 0.$$

An indicator expression is σ_i and valid negation expressions are τ_i or $(1 - \sigma_i)$. The feasible region of (11) is illustrated in Fig. 3 (left).

Since this scheme adds $2m$ continuous variables and $2m + 1$ equality constraints, it introduces a lack of regularity (in the LICQ and MFCQ sense) in the (σ_i, τ_i)-space. However, (11) can be equivalently reformulated by replacing (11a) with

$$\phi(\sigma_i, \tau_i) = 0 \quad \text{for all} \quad i \in \mathcal{M}_1, \quad \sigma_1 \geq 0.$$

For $\phi \in \{\phi_{\text{prod}}, \phi_{\text{FB}}\}$, it can be shown that the latter formulation satisfies the LICQ if and only if $\sigma_1 \neq 0$. Of course, $\mathcal{M}_1$ can be replaced with every other $\mathcal{M}_j$.

4.2.2. *Approximate Bivariate Reformulation.* The second reformulation works like the first one but relaxes (11b):

$$\phi(\sigma_i, \tau_i) = 0 \quad \text{for all} \quad i \in \mathcal{M}, \tag{12a}$$
$$\sigma_i + \tau_i \geq 1 \quad \text{for all} \quad i \in \mathcal{M}, \tag{12b}$$
$$\sum_{i \in \mathcal{M}} \sigma_i = 1. \tag{12c}$$

The feasible set is illustrated in Fig. 3 (right): it is readily seen that $(\sigma_i, \tau_i) \in (\{0\} \times [1, \infty)) \cup ([1, \infty) \times \{0\})$, and that exactly one $i \in \mathcal{M}$ exists with $\sigma_i = 1$ and $\sigma_j = 0$ for all $j \neq i$. Here we have added $2m$ variables with $m + 1$ equality and m inequality constraints. Depending on the activity status of inequalities, the LICQ might be satisfied or again be violated. An indicator expression is σ_i and a valid

FIGURE 4. Left: Feasible set $\{0,1\}$ of the exact univariate reformulation (13) and the alternative exact univariate reformulation (14). Right: Feasible set $\{0\} \cup [1,\infty)$ of the alternative univariate reformulation (15).

negation expression is given by τ_i. This scheme cannot be reformulated again like the exact bivariate reformulation since one would lose the property of being able to formulate indicator and negation expressions.

4.2.3. *Exact Univariate Reformulation.* This scheme requires only one continuous variable σ_i per binary variable s_i and adds the following constraints:

$$\phi(\sigma_i, 1-\sigma_i) = 0 \quad \text{for all} \quad i \in \mathcal{M}, \tag{13a}$$

$$\sum_{i\in\mathcal{M}} \sigma_i = 1. \tag{13b}$$

The feasible set is shown in Fig. 4 (left). Clearly we have $\sigma_i \in \{0,1\}$ for all $i \in \mathcal{M}$. An indicator expression is σ_i and a negation expression is $(1-\sigma_i)$. The number of equality constraints exceeds the number of variables again by one ($m+1$ vs. m), but (13) can be equivalently replaced with

$$\phi(\sigma_i, 1-\sigma_i) = 0 \quad \text{for all} \quad i \in \mathcal{M}_1, \quad \sigma_1 \geq 0, \quad \sum_{i\in\mathcal{M}} \sigma_i = 1,$$

satisfying the LICQ for $\phi \in \{\phi_{\text{prod}}, \phi_{\text{FB}}\}$ if and only if $\sigma_1 \neq 0$.

4.2.4. *Alternative Exact Univariate Reformulation.* Here, the set of constraints reads

$$\phi(\sigma_i, \textstyle\sum_{j\in\mathcal{M}_i} \sigma_j) = 0 \quad \text{for all} \quad i \in \mathcal{M}, \tag{14a}$$

$$\sum_{i\in\mathcal{M}} \sigma_i = 1. \tag{14b}$$

These constraints imply $\phi(\sigma_i, 1-\sigma_i) = 0$, yielding $\sigma_i \in \{0,1\}$ for all i. As before, there is one more equality constraint than variables ($m+1$ vs. m), and we have the equivalent reformulation

$$\phi(\sigma_i, \textstyle\sum_{j\in\mathcal{M}_i} \sigma_j) = 0 \quad \text{for all} \quad i \in \mathcal{M}_1, \quad \sigma_1 \geq 0, \quad \sum_{i\in\mathcal{M}} \sigma_i = 1.$$

In both cases, an indicator expression is σ_i and a negation expression is $(1-\sigma_i)$ or, equivalently, $\sum_{j\in\mathcal{M}_i} \sigma_j$. The feasible set is the same as before, see Fig. 4 (left). Again, the LICQ is satisfied for $\phi \in \{\phi_{\text{prod}}, \phi_{\text{FB}}\}$ if and only if $\sigma_1 \neq 0$.

4.2.5. *Approximate Univariate Reformulation.* The set of constraints for the final scheme is given by

$$\phi(\sigma_i, \textstyle\sum_{j\in\mathcal{M}_i} \sigma_j) = 0 \quad \text{for all} \quad i \in \mathcal{M}, \tag{15a}$$

$$\sum_{i\in\mathcal{M}} \sigma_i \geq 1. \tag{15b}$$

The feasible set is illustrated in Fig. 4 (right). Clearly we have $\sigma_i \in \{0\} \cup [1,\infty)$ for all i. An indicator expression is σ_i, and a negation expression is $\sum_{j\in\mathcal{M}_i} \sigma_j$. This reformulation has the disadvantage that it always violates the LICQ for both ϕ_{prod} and ϕ_{FB}, even if the inequality constraint (15b) is not active. Again, (15) can be equivalently reformulated by replacing (15a) with

$$\phi(\sigma_i, \textstyle\sum_{j\in\mathcal{M}_i} \sigma_j) = 0 \quad \text{for all} \quad i \in \mathcal{M}_1, \quad \sigma_1 \geq 0.$$

TABLE 3. Summary of all reformulation schemes. Index 1 denotes the first version and index 2 denotes the second version of the reformulation.

Section	Variables	$\lvert\mathcal{E}_1\rvert$	$\lvert\mathcal{E}_2\rvert$	$\lvert\mathcal{I}_1\rvert$	$\lvert\mathcal{I}_2\rvert$	exact/approx.
4.2.1	$2m$	$2m+1$	$2m$	0	1	exact
4.2.2	$2m$	$m+1$	—	m	—	approx.
4.2.3	m	$m+1$	m	0	1	exact
4.2.4	m	$m+1$	m	0	1	exact
4.2.5	m	m	$m-1$	1	2	approx.

This formulation satisfies the LICQ for $\phi \in \{\phi_{\text{prod}}, \phi_{\text{FB}}\}$ if and only if $\sigma_1 \neq 0$.

Table 3 lists the main properties of the five reformulation schemes.

5. Computational Study

In the preceding sections we have discussed the problem of compressor station optimization and we have presented several model formulations. With these formulations at hand, the question arises whether all models are comparably well suited for numerical computations, or whether there are benefits or disadvantages for any of the models. In this section we present an extensive computational study and compare the results of local and global solvers applied to different model formulations. We will see that the continuous reformulations work quite well when compared to MINLP approaches.

Section 5.1 introduces two compressor stations with boundary conditions as test instances and describes the hardware and software used in the study. Section 5.2 then discusses performance profiles for measuring performance and robustness of different model formulations and solvers. Subsequently, Sects. 5.3 and 5.4 present the numerical results, and Sect. 5.5 gives a summary of the computational study.

5.1. **Test Instances and Computational Setup.** We consider minimum cost problems using the objective (8) with cost coefficients $\omega^{\text{fuel}} = 0.024\,€/(\text{kg/s})$ and $\omega^{\text{el}} = 0.14\,€/\text{kW}$. This objective is combined with all presented models for two different compressor stations. The first station, called GasLib-582 station in the following, is `compressorStation_5` from the network GasLib-582 [20]. It contains one turbo compressor and one piston compressor and can be operated in three configurations. This station is comparatively small and serves as a proof of concept for the applicability of the continuous reformulations. Moreover, the data of this test set are publicly available, so that other researchers can compare their models or algorithms on the same data. The second station, called HG station in the following, is a real-world compressor station of our former industry partner Open Grid Europe[1] (OGE). It is one of OGE's largest compressor stations, containing five turbo compressors that can be operated in 14 configurations in our model. The results on this station illustrate the applicability of the presented models on real-world data.

The boundary values for our test instances are constructed as follows. We always prescribe the in- and outflow pressures as well as the flow through the station. For both stations, the set of inflow pressures (in bar) is $\mathcal{P}_{\text{in}} = \{20, 50, 80\}$. For every inflow pressure $p_{\text{in}} \in \mathcal{P}_{\text{in}}$, we then construct a set of outflow pressures $\mathcal{P}_{\text{out},i}(p_{\text{in}}) = \{p_{\text{in}} + k\Delta p_i \colon k = 0, \dots, 3\}$, $i = 1, 2$, where $\Delta p_1 = 13.3\,\text{bar}$ (GasLib-582 station) and $\Delta p_2 = 20\,\text{bar}$ (HG station). The sets of flows (in $10^3\,\text{Nm}^3\,\text{h}^{-1}$) are $\mathcal{Q}_1 = \{0, 375, 750, \dots, 2250\}$ (GasLib-582 station) and $\mathcal{Q}_2 = \{0, 800, 1600, \dots, 4800\}$

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

(HG station). Thus, the complete set of boundary conditions for the GasLib-582 station is

$$\mathcal{T}_1 = \{(p_{\text{in}}, p_{\text{out}}, Q_0) \colon p_{\text{in}} \in \mathcal{P}_{\text{in}}, p_{\text{out}} \in \mathcal{P}_{\text{out},1}(p_{\text{in}}), Q_0 \in \mathcal{Q}_1\},$$

and the corresponding set for the HG station is

$$\mathcal{T}_2 = \{(p_{\text{in}}, p_{\text{out}}, Q_0) \colon p_{\text{in}} \in \mathcal{P}_{\text{in}}, p_{\text{out}} \in \mathcal{P}_{\text{out},2}(p_{\text{in}}), Q_0 \in \mathcal{Q}_2\}.$$

All values are chosen based on our experience with the technical capabilities of the stations and with typical values in gas networks. Note that the flow values for the test sets are given in terms of volumetric flow under normal conditions, Q_0, measured in 1000 normal cubic meters per hour, as this is the standard technical unit in gas transportation. It can easily be converted to mass flow via $q = cQ_0\rho_0$, where $c = 1000/3600$, and ρ_0 is the gas density under normal conditions. The sizes of the test sets are $|\mathcal{T}_1| = |\mathcal{T}_2| = 84$.

All models are implemented using the modeling language GAMS [34] and the C++ software framework LaMaTTO++ [29]. As global solvers for the MINLP model and its continuous (NLP type) reformulations we use BARON 12.3.3 [48, 49, 50] and SCIP 3.0 [51, 52]. Additionally, we use the convex MINLP solver KNITRO 8.1.1 [6] as a heuristic for the nonconvex MINLPs and as NLP solver for the continuous reformulations. As local solvers for the continuous reformulations we use the interior-point code Ipopt 3.11 [53] and the reduced-gradient code CONOPT4 [13, 14, 15] as well as the three MINLP solvers. The solvers are run with default settings throughout, even for solution tolerances, as it is virtually impossible to find settings that make the results comparable in a strict mathematical sense.

All computations are executed on a six-core AMD Opteron Processor 2435 with 2600 MHz and 64 GB RAM. The operating system is Debian 7.5.

5.2. **Measuring Performance and Robustness.** To compare the computing times of different combinations of model formulations and solvers, we use standard performance profiles [11]. To this end, let us define the performance measure

$$t_{p,s} := \text{computing time required to solve problem } p \in \mathcal{T}_i \text{ by } s \in \mathcal{S},$$

where the set $\mathcal{S}$ contains all 106 combinations of model formulations and solvers: the MINLP model in big-M and bilinear product formulation combined with BARON, SCIP, and KNITRO ($1 \times 2 \times 3 = 6$), and each of the five continuous reformulations in four variants (big-M and bilinear product with both pseudo NCP functions each) combined with all five solvers ($5 \times 4 \times 5 = 100$). We consider only the subsets of feasible boundary values, $\mathcal{F}_i \subset \mathcal{T}_i$, defined to contain those instances for which at least one combination $s \in \mathcal{S}$ produces a feasible solution. Now the performance ratio $r_{p,s}$ associated with $t_{p,s}$ is

$$r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s'} \colon s' \in \mathcal{S}\}} \in [1, \infty).$$

Moreover, we set $r_{p,s} = r_M := \max\{r_{p',s'} \colon p' \in \mathcal{F}_i,\ s' \in \mathcal{S}_i\}$ for those instances p that cannot be solved by s. The performance profile is finally given by

$$\rho_s(\tau) := \frac{1}{|\mathcal{F}_i|}|\{p \in \mathcal{F}_i \colon r_{p,s} \le \tau\}| \in [0, 1].$$

5.3. **The GasLib-582 Test Set.** The set $\mathcal{F}_1$ contains 48 instances of $\mathcal{T}_1$, hence 36 of the 84 instances are infeasible. The computing times are up to 62 s in cases where solutions are found and up to 59 s in cases where a solver detects infeasibility. Some of the instances are solved in fractions of a second. This can happen, for instance, in the preprocessing of BARON due to bound strengthening. As one would expect, the largest differences of computing times are observed for the global solvers.

TABLE 4. GasLib-582 test set: numbers of false infeasibility reports and solver failures (top: solver does not find a feasible solution with any model, bottom: no solver finds a feasible solution with any variant of given model)

BARON	SCIP	KNITRO	Ipopt	CONOPT4	
0	7	7	7	8	
MINLP	EBR	ABR	EUR	AEUR	AUR
7	1	0	0	2	5

First we discuss the objective values that are obtained by the different formulations and solvers. In theory, the global solvers BARON and SCIP should produce identical objective values whereas the local optima found by the local solvers can be larger by arbitrary amounts. However, our results show that the values of BARON and SCIP differ in some cases and that all model formulations and solvers produce *almost* identical values on the current test set.

For every problem $p \in \mathcal{F}_1$, we compute the minimum and maximum objective values $f^*_{p,\min}$ and $f^*_{p,\max}$ as well as the maximal absolute gap $g^*_p = f^*_{p,\max} - f^*_{p,\min}$. The minimal values $f^*_{p,\min}$ range from 0 to 0.008 735 (operating cost in €/s), with maximal gaps ranging from 0 to 0.008 765. The average maximal gap over $p \in \mathcal{F}_1$ is approximately 0.0012, but most of the individual gaps are actually zero. We assume that differing objective values are mainly caused by different numerical properties of the solvers. We also compared the discrete states of the compressor stations in the optimal solutions for every instance. Different active states are found for 9 of the 48 feasible instances, 3 of which have boundary values of the form $Q_0 = 0$ and $p_u = p_v$, where the bypass mode and the closed mode are both feasible and globally optimal with zero cost. The remaining 6 instances have different active configurations. Thus, we have the surprising observation that on the current test set the local solvers always yield optimal values close to the global minima.

Next, we turn to the issue of infeasibility detection. A substantial fraction of the boundary values of our test set are infeasible: 36 out of 84. In theory, if a global solver detects infeasibility of an instance, this is considered as an infeasibility proof. However, due to numerical inaccuracy, this is not always true in practice. To give a more detailed overview, we also list the numbers of feasible instances that are not solved (i.e., infeasibility is reported or the solver simply fails) in Table 4. Among the solvers, BARON clearly shows the most reliable results: for every feasible instance there is at least one model formulation for which BARON produces a solution. All other solvers report false infeasible results or fail in 7 or 8 cases. Surprisingly, this is also true for the global solver SCIP. Regarding the different continuous reformulations, the approximate bivariate and exact univariate reformulations are solved for every feasible instance (at least by one solver). The worst result is obtained for the MINLP model (7 failures). However, this result has to be carefully interpreted because the MINLP is handled by only 3 solvers whereas all 5 solvers can handle the continuous reformulations. Within the set of reformulations, the approximate univariate reformulation has by far the largest number of false infeasibility reports and solver failures.

Let us now investigate the performance (measured in computing time) of all combinations $s \in \mathcal{S}$. We define the fastest combination to be the one that has the largest number of instances that it solves at least as fast as all other combinations, i.e., the combination s with the largest value $\rho_s(0)$ in the corresponding logarithmically scaled performance profile. We use this value because it is a generally accepted quality measure. Of course, one might also be interested in other measures such as the

TABLE 5. GasLib-582 test set: fastest solvers for every combination of MINLP model or continuous reformulation with indicator constraint type and pseudo NCP function (BP: bilinear product)

Model	big-M/ϕ_{FB}	big-M/ϕ_{prod}	BP/ϕ_{FB}	BP/ϕ_{prod}
MINLP	BARON		**BARON**	
Exact bivar. reform.	BARON	**BARON**	BARON	BARON
Approx. bivar. reform.	KNITRO	**BARON**	BARON	BARON
Exact univar. reform.	BARON	**BARON**	BARON	BARON
Alt. exact univar. reform.	Ipopt	BARON	BARON	**CONOPT4**
Approx. univar. reform.	BARON	BARON	BARON	**CONOPT4**

total runtime of a model/solver combination s on all instances. The solution quality (objective value) does not play a role here. To give a visual illustration, we need some aggregation and compare the 106 combinations in two stages: in the first stage we determine the best combination of a solver with indicator constraint type and pseudo NCP function for each of the 6 basic models: MINLP and 5 reformulations. In the second stage, only the 6 best combinations of the first stage are compared. Table 5 lists the fastest solvers for the MINLP model and the 5 reformulations. The solver printed in bold is the fastest solver of the entire row, i.e., over all combinations of indicator constraint types and pseudo NCP functions. Note that the MINLP model does not involve any pseudo NCP functions, hence the first row has only two entries (big-M and BP).

First, it can be seen that the global solver BARON is the fastest solver for 18 of the 22 model formulations. The local solvers Ipopt or CONOPT4 are faster in only three cases. CONOPT4 is the overall fastest solver for the alternative exact and approximate univariate reformulations. Second, it is apparent that the bilinear product used as pseudo NCP function yields clearly faster runs than the Fischer–Burmeister function: all the bold model/solver combinations use the bilinear product. Third, a best choice of the type of indicator constraints (big-M vs. bilinear product) is not apparent by this criterion. The full data set actually shows that the choice does not have a significant impact here.

Figure 5 shows the 6 performance profiles of the fastest model/solver combinations (bold) for each row of Table 5. Although BARON is faster than all other solvers on the largest number of model formulations, it turns out that the preferable model/solver combination does *not* use BARON. In order to determine the overall fastest combination, we again compare the values $\rho_s(0)$. It can be seen that the local solver CONOPT4 (applied to the alternative exact and approximate univariate reformulations) produces the shortest solution times for approximately 40 % of all feasible instances. Notice that both formulations solved with CONOPT4 are faster than all other combinations *if* an instance can be solved by CONOPT4 at all (the "horizontal" lines of the performance profiles of AEUR and AUR in Fig. 5 extend almost to the left boundary). This is to be expected since local solvers typically tend to be faster than global solvers. The global solver BARON applied to the other continuous reformulations yields significantly slower solution times and is the fastest combination for 15 % to 25 % of the feasible instances. Finally, BARON applied to the original MINLP model is distinctly the slowest combination of solver and model formulation.

We now turn to robustness. The most robust combination is defined as the one that solves the largest fraction of instances to optimality, i.e., the combination s

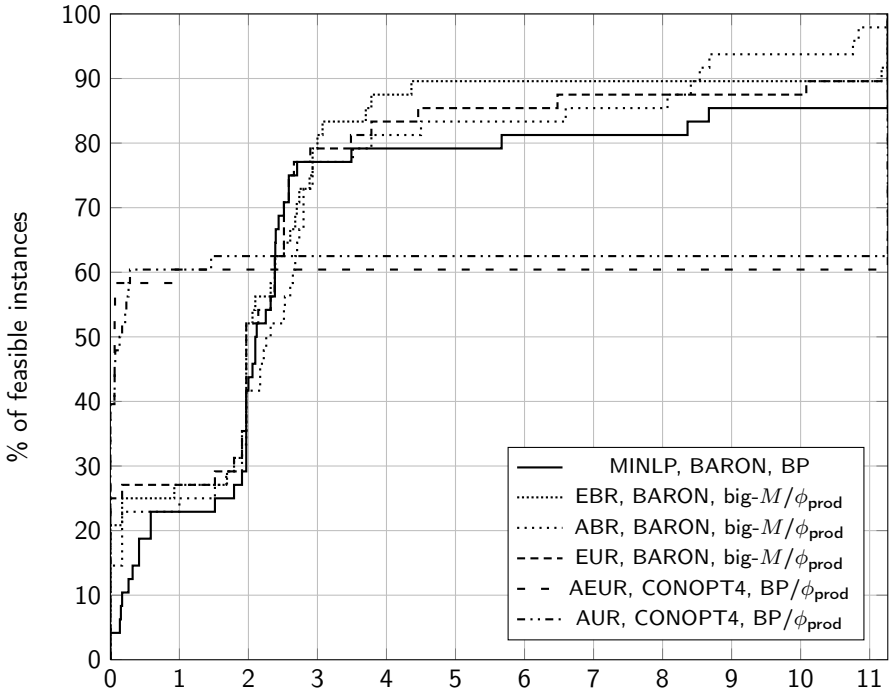

FIGURE 5. GasLib-582 test set: comparison of fastest combinations of indicator constraint type, pseudo NCP function, and solver for MINLP and continuous reformulations; cf. Table 5

TABLE 6. GasLib-582 test set: most robust solvers for every combination of MINLP model or continuous reformulation with indicator constraint type and pseudo NCP function (BP: bilinear product)

Model	big-M/ϕ_{FB}	big-M/ϕ_{prod}	BP/ϕ_{FB}	BP/ϕ_{prod}
MINLP	**BARON**		SCIP	
Exact bivar. reform.	**BARON**	BARON	BARON	BARON
Approx. bivar. reform.	BARON	**BARON**	BARON	BARON
Exact univar. reform.	**BARON**	BARON	BARON	BARON
Alt. exact univar. reform.	BARON	BARON	BARON	**BARON**
Approx. univar. reform.	BARON	BARON	**BARON**	BARON

with the largest value

$$\rho_s^* := \lim_{\tau \nearrow r_M} \rho_s(\tau).$$

As in the performance comparison, we list the most robust solver for every combination of model reformulation, indicator constraint type, and pseudo NCP function in Table 6. Here BARON appears even more often than in the case of computing times: It is always the most robust solver, except for the single case where SCIP is applied to the MINLP model with bilinear products as indicator constraints. In contrast to the case of computing times, the choice of indicator constraints does not seem to have a significant impact here.

Again we compare the 6 performance profiles of the best combinations (bold) for each row of Table 6, see Fig. 6. Overall, the results look more "homogeneous" than in Fig. 5. All combinations produce comparatively good values ρ_s^* from 85 % to

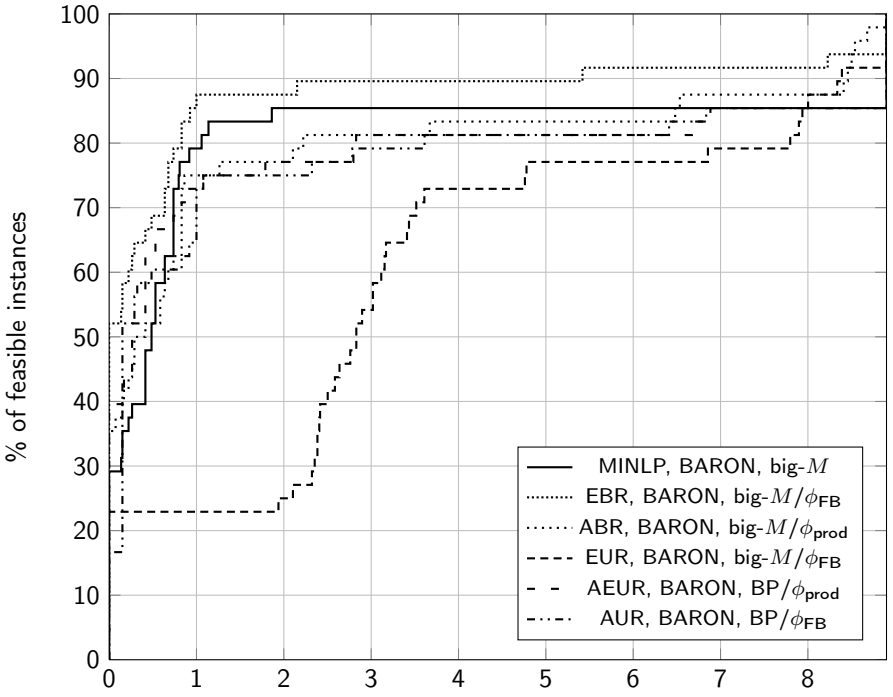

FIGURE 6. GasLib-582 test set: comparison of most robust combinations of indicator constraint type, pseudo NCP function, and solver for MINLP and continuous reformulations; cf. Table 6

98 % of solved instances. The two bivariate reformulations produce the best values (significantly above 90 %). Note further that these model formulations use big-M indicator constraints. The good robustness is probably caused by the convexity conserving property of the big-M formulation, which contrasts with the inherently nonconvex bilinear products.

5.4. **The HG Test Set.** The set $\mathcal{F}_2$ contains 41 instances of $\mathcal{T}_2$, hence 43 of the 84 instances are infeasible. The computing times are up to 67 s in cases where solutions are found and up to 45 s in cases where a solver detects infeasibility.

The minimal objective values (in €/s) now range from 0 to 0.069 505 and the maximal gaps from 0 to 0.022 102. The average maximal gap over all $p \in \mathcal{F}_2$ is significantly smaller than for $\mathcal{F}_1$ at approximately 0.000 677 8 and most of the individual gaps are again zero. Except for one case, the active states in the optimal solutions for a given set of boundary values are identical for all $s \in \mathcal{S}$ or consist of different sets of identical compressor units yielding the same objective value.[2] The exceptional case is exactly the one leading to the maximal gap. Excluding this case reduces the maximal gap over all instances by one order of magnitude.

In Table 7 we list the numbers of feasible instances for which a model could not be solved to optimality (i.e., infeasibility is reported or the solver fails). As it was the case for the GasLib-582 station, BARON yields the smallest number of false infeasibility reports (just one). Moreover, the choice of the class of solver (global vs. local) seems to be more crucial than the choice of the specific model formulation:

[2]The detection of mathematical symmetry in this situation could be used to reduce the complexity of the corresponding station model.

TABLE 7. HG test set: numbers of false infeasibility reports and solver failures (top: solver does not find a feasible solution with any model, bottom: no solver finds a feasible solution with any variant of given model)

BARON	SCIP	KNITRO	Ipopt	CONOPT4	
1	3	2	8	5	
MINLP	EBR	ABR	EUR	AEUR	AUR
3	2	2	2	2	4

TABLE 8. HG test set: fastest solvers for every combination of MINLP model or continuous reformulation with indicator constraint type and pseudo NCP function (BP: bilinear product)

Model	big-M/ϕ_{FB}	big-M/ϕ_{prod}	BP/ϕ_{FB}	BP/ϕ_{prod}
MINLP	KNITRO		**BARON**	
Exact bivar. reform.	BARON	**CONOPT4**	CONOPT4	BARON
Approx. bivar. reform.	BARON	**CONOPT4**	Ipopt	Ipopt
Exact univar. reform.	KNITRO	BARON	Ipopt	**Ipopt**
Alt. exact univar. reform.	Ipopt	SCIP	Ipopt	**CONOPT4**
Approx. univar. reform.	BARON	SCIP	Ipopt	**CONOPT4**

the MINLP solvers report false infeasibility in 1 to 3 cases whereas the local solvers have significantly larger numbers of failure of 5 and 8.

Turning to the performance investigation, we follow the same two-stage approach as before. Table 8 lists the fastest solvers for the MINLP model and all tested continuous reformulations. It can be seen that BARON is the fastest solver for the MINLP model (to which only global solvers can be applied) and that local solvers (CONOPT4 and Ipopt) are the fastest for all continuous reformulations, where both global and local solvers are applied. This is in line with our expectations since local solvers typically tend to be faster than global solvers. Moreover, it is noticeable that the pseudo NCP function ϕ_{prod} yields better results than the Fischer–Burmeister function ϕ_{FB}, and that global solvers outperform local solvers when applied to big-M indicator constraints while the converse behavior is observed for bilinear indicator constraints (except for the case of BARON applied to the exact bivariate reformulation using bilinear indicator constraints and $\phi = \phi_{\mathrm{prod}}$).

Figure 7 shows the performance profiles for every bold combination of Table 8. The approximate bivariate reformulation (ABR) solved with CONOPT4 is the overall fastest combination, performing best on more than 30 % of all instances. The approximate univariate (AUR), exact bivariate (EBR), and alternative exact univariate (AEUR) reformulations combined with CONOPT4 come next, each performing best on roughly 20 % of all instances. Since the approximate univariate formulation shows better results than EBR and AEUR for small values $\tau > 0$, we may summarize that the approximate reformulations (combined with $\phi = \phi_{\mathrm{prod}}$) tend to be the fastest combinations. A possible reason might be that for local solvers the enlarged feasible set is preferable to the feasible sets of lower dimension of the exact reformulations. In addition, the overall preferable approximate bivariate scheme is also the "most regular" formulation with respect to the LICQ (cf. Sect. 4.2). Finally we note that both the exact univariate reformulation and the MINLP model cannot compete with the other reformulations.

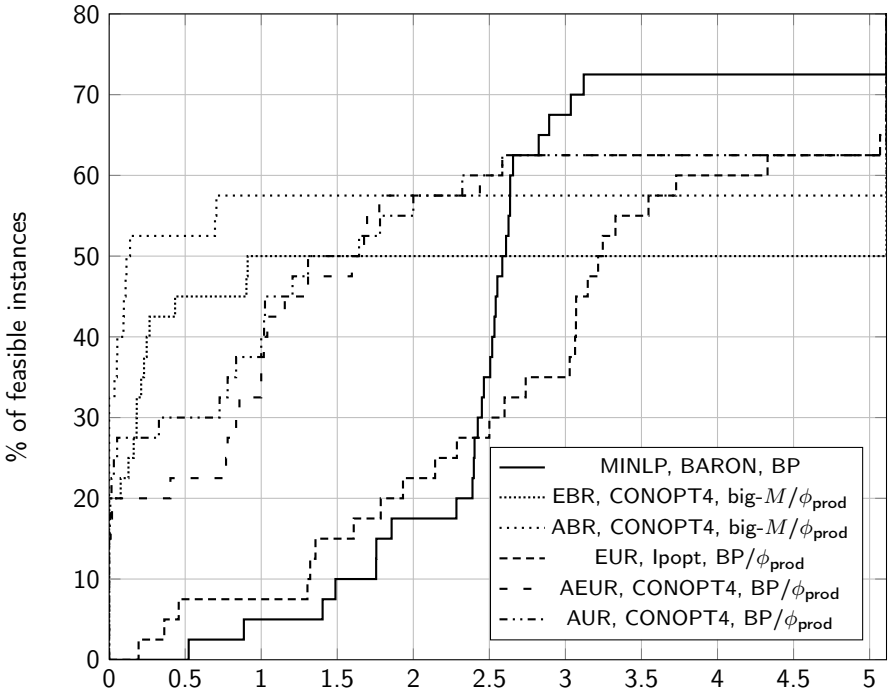

FIGURE 7. HG test set: comparison of fastest combinations of indicator constraint type, pseudo NCP function, and solver for MINLP and continuous reformulations; cf. Table 8

TABLE 9. HG test set: most robust solvers for every combination of MINLP model or continuous reformulation with indicator constraint type and pseudo NCP function (BP: bilinear product)

Model	big-M/ϕ_{FB}	big-M/ϕ_{prod}	BP/ϕ_{FB}	BP/ϕ_{prod}
MINLP	**KNITRO**		**SCIP**	
Exact bivar. reform.	BARON	BARON	BARON	**BARON**
Approx. bivar. reform.	BARON	BARON	**BARON**	BARON
Exact univar. reform.	BARON	SCIP	BARON	**BARON**
Alt. exact univar. reform.	BARON	SCIP	BARON	**BARON**
Approx. univar. reform.	BARON	SCIP	KNITRO	**BARON**

Regarding robustness, the situation changes completely. We list the most robust solver (the one with largest ρ_s^*) for every combination of model reformulation, indicator constraint type, and pseudo NCP function in Table 9. For the original MINLP model, KNITRO (with big-M indicator constraints) and SCIP (with bilinear indicator constraints) are equally robust. BARON is the most robust solver in 80 % of all continuous reformulations. The bilinear product indicator constraints always yield better results than the big-M ones. Except for KNITRO applied to the approximate univariate reformulation using bilinear indicator constraints and $\phi = \phi_{\text{FB}}$, no local solver is more robust than one of the global solvers.

The performance profiles of the most robust combinations of Table 9 are given in Fig. 8. All models yield comparable results, solving 85 % to 93 % of all instances. The highest percentages are obtained by the original MINLP and the exact bivariate

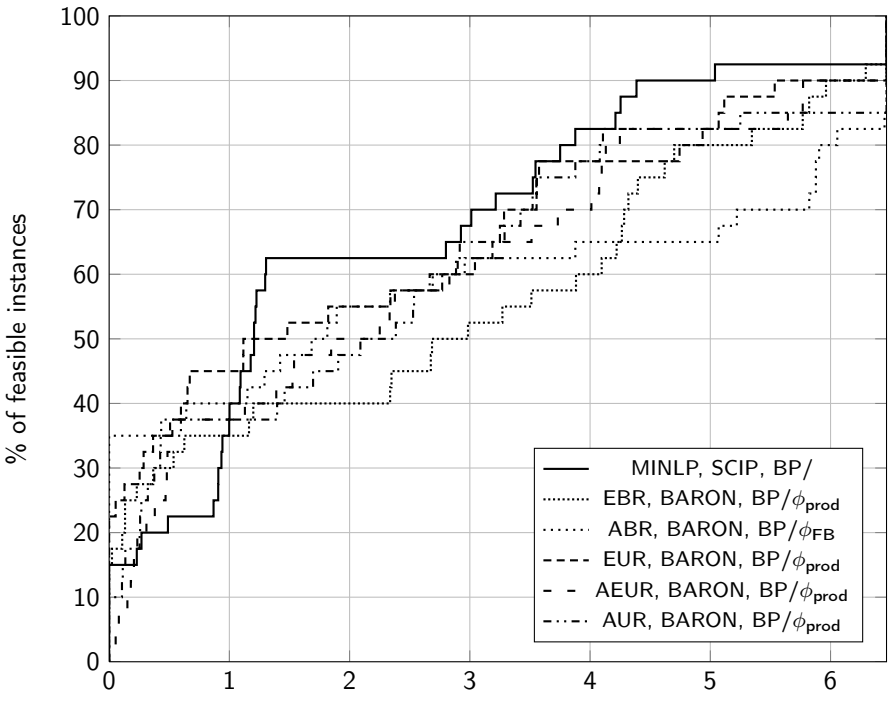

FIGURE 8. HG test set: comparison of most robust combinations of indicator constraint type, pseudo NCP function, and solver for MINLP and continuous reformulations; cf. Table 9

reformulation (EBR). The fastest continuous reformulations, i.e., the approximate bivariate and the alternative approximate univariate reformulation, yield the smallest values at approximately 85 %. Thus, we have exactly the opposite situation as for the performance comparison.

5.5. **Summary.** The first main result is that local solvers applied to continuous reformulations tend be faster than global solvers applied to the mixed-integer model and its continuous reformulations. This was to be expected. Moreover, our numerical results suggest that this tendency becomes more evident for larger instances. When comparing only the continuous reformulations, the results on the larger test set indicate that the larger feasible sets of the approximate reformulations are favorable and that the regularity of the formulation has a strong connection to the performance of the solution process. (Recall that the "most regular" approximate bivariate reformulation leads to the fastest solution times for the large test set.) On the contrary, global solvers (especially BARON) lead to more robust results, independent of the choice of model formulation.

When the decision is made which solver should be used, this also indicates a good choice for the type of indicator constraints. Especially for larger instances, global solvers tend to be faster on big-M formulations and local solvers lead to better results when bilinear products are used. We think that the reason for this effect is that the behaviour of global solvers is stronger influenced by an increased amount of nonconvexity than it is the case for local solvers. This changes if the focus lies on robustness of the solution process, where the bilinear indicator constraint is clearly preferable to the big-M formulation.

The second result is about the type of pseudo NCP functions: The product formulation ϕ_{prod} distinctly outperforms Fischer–Burmeister functions. Although

the latter are quite prominent in the literature, the simple formulation using products usually leads to faster and more robust formulations in practice.

Finally, we can state that—at least for the models under consideration in this paper—there are no significant drawbacks concerning the quality of solutions when local solvers are used. However, this might change if the fixed boundary values are replaced by feasible ranges.

6. Conclusion

In this article we have presented MINLP and GDP models for cost optimization and feasibility testing of gas compressor stations. Moreover, we have considered different types of continuous reformulation techniques from the literature and applied them to the application problem. Our computational study shows that local solvers applied to continuous reformulations can be used to replace MINLP formulations that can only be tackled by global solvers. The continuous reformulations yield comparably robust results, optimal values of almost the same quality and tend to be solvable within shorter solution times. Together with the publications [25, 37, 41], this article provides a complete continuous reformulation of the discrete-continuous problem of stationary gas transport optimization.

However, some questions remain open. We have considered a stationary and isothermal variant of the problem of compressor station optimization. Since including gas temperature as a dynamic variable mainly leads to increased nonlinearity and nonconvexity in the model, we expect that continuous reformulations tend to be even more favorable for these models. In contrast, the consideration of the transient case will also increase the amount of discrete aspects so that it is unclear, which formulation will be favorable in this case.

Acknowledgement

This work has been supported by the German Federal Ministry of Economics and Technology owing to a decision of the German Bundestag. The responsibility for the content of this publication lies with the authors. This research has been conducted as part of the Energie Campus Nürnberg and supported by funding through the "Aufbruch Bayern (Bavaria on the move)" initiative of the state of Bavaria. Finally, we thank our former industry partner Open Grid Europe GmbH for their support.

References

[1] M. K. Banda and M. Herty. "Multiscale modeling for gas flow in pipe networks." In: *Mathematical Methods in the Applied Sciences* 31 (Aug. 2008), pp. 915–936. DOI: 10.1002/mma.948.

[2] M. K. Banda, M. Herty, and A. Klar. "Gas flow in pipeline networks." In: *Networks and Heterogeneous Media* 1.1 (Mar. 2006), pp. 41–56. DOI: 10.3934/nhm.2006.1.41.

[3] B. T. Baumrucker, J. G. Renfro, and L. T. Biegler. "MPEC problem formulations and solution strategies with chemical engineering applications." In: *Computers & Chemical Engineering* (2008), pp. 2903–2913. DOI: 10.1016/j.compchemeng.2008.02.010.

[4] H. G. Bock, E. Kostina, H. X. Phu, and R. Rannacher, eds. *Modeling, Simulation and Optimization of Complex Processes*. Berlin: Springer, 2005.

[5] J. Brouwer, I. Gasser, and M. Herty. "Gas Pipeline Models Revisited: Model Hierarchies, Nonisothermal Models, And Simulations of Networks." In: *Multiscale Model. Simul.* 9.2 (2011), pp. 601–623.

[6] R. H. Byrd, J. Nocedal, and R. A. Waltz. "KNITRO: An integrated package for nonlinear optimization." In: *Large Scale Nonlinear Optimization, 35–59, 2006*. Springer Verlag, 2006, pp. 35–59.

[7] R. G. Carter. "Compressor Station Optimization: Computational Accuracy and Speed." In: *28th Annual Meeting*. Paper 9605. Pipeline Simulation Interest Group. 1996.

[8] R. G. Carter. "Pipeline Optimization: Dynamic Programming after 30 Years." In: *30th Annual Meeting*. Paper 9803. Pipeline Simulation Interest Group. 1998.

[9] R. G. Carter, D. W. Schroeder, and T. D. Harbick. *Some Causes and Effects of Discontinuities in Modeling and Optimizing Gas Transmission Networks*. Tech. rep. Carlisle, PA, USA: Stoner Associates, Apr. 1994.

[10] D. Cobos-Zaleta and R. Z. Ríos-Mercado. "A MINLP Model for a Problem of Minimizing Fuel Consumption on Natural Gas Pipeline Networks." In: *Proc. XI Latin-Ibero-American Conference on Operations Research*. Paper A48-01. 2002, pp. 1–9.

[11] E. D. Dolan and J. J. Moré. "Benchmarking Optimization Software with Performance Profiles." In: *Math. Program.* 91 (2 2002), pp. 201–213. DOI: 10.1007/s101070100263.

[12] P. Domschke, B. Geißler, O. Kolb, J. Lang, A. Martin, and A. Morsi. "Combination of Nonlinear and Linear Optimization of Transient Gas Networks." In: *INFORMS Journal of Computing* 23.4 (2011), pp. 605–617. DOI: DOI10.1287/ijoc.1100.0429.

[13] A. S. Drud. "CONOPT – A Large-Scale GRG Code." In: *INFORMS J. Comput.* 6.2 (1994), pp. 207–216. DOI: 10.1287/ijoc.6.2.207.

[14] A. S. Drud. *CONOPT: A System for Large Scale Nonlinear Optimization, Reference Manual for CONOPT Subroutine Library*. Tech. rep. ARKI Consulting and Development A/S, Bagsvaerd, Denmark, 1996.

[15] A. S. Drud. *CONOPT: A System for Large Scale Nonlinear Optimization, Tutorial for CONOPT Subroutine Library*. Tech. rep. ARKI Consulting and Development A/S, Bagsvaerd, Denmark, 1995.

[16] K. Ehrhardt and M. C. Steinbach. "KKT Systems in Operative Planning for Gas Distribution Networks." In: *Proc. Appl. Math. Mech.* 4.1 (2004), pp. 606–607.

[17] K. Ehrhardt and M. C. Steinbach. "Nonlinear Optimization in Gas Networks." In: *Modeling, Simulation and Optimization of Complex Processes*. Ed. by H. G. Bock, E. Kostina, H. X. Phu, and R. Rannacher. Berlin: Springer, 2005, pp. 139–148.

[18] Eurostat. *Gross inland energy consumption, by fuel*. Downloaded on April 3, 2013. 2013. URL: http://epp.eurostat.ec.europa.eu/tgm/refreshTableAction.do?tab=table&plugin=1&pcode=tsdcc320&language=en.

[19] A. Fischer. "A special Newton-type optimization method." In: *Optimization* 24.3-4 (1992), pp. 269–284. DOI: 10.1080/02331939208843795.

[20] *GasLib – a library of gas network instances*. http://gaslib.zib.de. 2015.

[21] B. Geißler. "Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization." PhD thesis. Friedrich-Alexander Universität Erlangen-Nürnberg, Germany, 2011.

[22] B. Geißler, A. Morsi, and L. Schewe. "A New Algorithm for MINLP Applied to Gas Transport Energy Cost Minimization." In: *Facets of Combinatorial Optimization*. Ed. by M. Jünger and G. Reinelt. Springer Berlin Heidelberg, 2013, pp. 321–353. DOI: 10.1007/978-3-642-38189-8_14.

[23] I. E. Grossmann. "Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques." In: *Optim. Eng.* 3.3 (2002). Special issue on mixed-integer programming and its applications to engineering, pp. 227–252. DOI: 10.1023/A:1021039126272.

[24] T. Jeníček and J. Králik. "Optimized Control of Generalized Compressor Station." In: *PSIG Annual Meeting*. 1995.

[25] T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, eds. *Evaluating Gas Network Capacities*. SIAM-MOS series on Optimization. SIAM, Mar. 2015. xvi + 376. Forthcoming.

[26] K. Kraemer, S. Kossack, and W. Marquardt. "An efficient solution method for the MINLP optimization of chemical processes." In: *17th European Symposium on Computer Aided Process Engineering*. Ed. by V. Plesu and P. S. Agachi. Vol. 24. Computer Aided Chemical Engineering. Elsevier, 2007, pp. 105–110. DOI: http://dx.doi.org/10.1016/S1570-7946(07)80041-1.

[27] K. Kraemer and W. Marquardt. "Continuous Reformulation of MINLP Problems." In: *Recent Advances in Optimization and its Applications in Engineering*. Ed. by M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels. Springer Berlin Heidelberg, 2010, pp. 83–92. DOI: 10.1007/978-3-642-12598-0_8.

[28] J. Králik. "Compressor Stations in SIMONE." In: *Proceedings of 2nd International Workshop SIMONE on Innovative Approaches to Modeling and Optimal Control of Large Scale Pipeline Networks*. Prague, 1993, pp. 93–117.

[29] *LaMaTTO++: A Framework for Modeling and Solving Mixed-Integer Nonlinear Programming Problems on Networks*. http://www.mso.math.fau.de/edom/projects/lamatto.html. 2015.

[30] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

[31] A. Martin, D. Mahlke, and S. Moritz. "A Simulated Annealing Algorithm for Transient Optimization in Gas Networks." In: *Math. Methods Oper. Res.* 66.1 (2007), pp. 99–115. DOI: 10.1007/s00186-006-0142-9.

[32] A. Martin and M. Möller. "Cutting Planes for the Optimization of Gas Networks." In: *Modeling, Simulation and Optimization of Complex Processes*. Ed. by H. G. Bock, E. Kostina, H. X. Phu, and R. Rannacher. Berlin: Springer, 2005, pp. 307–329.

[33] A. Martin, M. Möller, and S. Moritz. "Mixed Integer Models for the Stationary Case of Gas Network Optimization." In: *Math. Program.* 105.2-3, Ser. B (2006), pp. 563–582. DOI: 10.1007/s10107-005-0665-5.

[34] B. A. McCarl. *GAMS User Guide*. VERSION 23.0. 2009.

[35] F. M. Odom and G. L. Muster. *Tutorial on Modeling of Gas Turbine Driven Centrifugal Compressors*. Tech. rep. 09A4. Pipeline Simulation Interest Group, 2009.

[36] A. Osiadacz. "Nonlinear Programming Applied to the Optimum Control of a Gas Compressor Station." In: *Int. J. Numer. Methods Eng.* 15.9 (1980), pp. 1287–1301. DOI: 10.1002/nme.1620150902.

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

[38] *PSIG 30th Annual Meeting, Denver, Colorado*. Pipeline Simulation Interest Group. 1998.

[39] R. Raman and I. E. Grossmann. "Modeling and Computational Techniques for Logic Based Integer Programming." In: *Comput. Chem. Eng.* 18.7 (1994), pp. 563–578.

[40] M. Schmidt. "A Generic Interior-Point Framework for Nonsmooth and Complementarity Constrained Nonlinear Optimization." PhD thesis. Gottfried Wilhelm Leibniz Universität Hannover, 2013.

[41] M. Schmidt, M. C. Steinbach, and B. M. Willert. "A Primal Heuristic for Nonsmooth Mixed Integer Nonlinear Optimization." In: *Facets of Combinatorial Optimization*. Ed. by M. Jünger and G. Reinelt. Berlin, Heidelberg: Springer, 2013, pp. 295–320. DOI: 10.1007/978-3-642-38189-8_13.

[42] M. Schmidt, M. C. Steinbach, and B. M. Willert. "An MPEC based heuristic." In: *Evaluating Gas Network Capacities*. Ed. by T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. SIAM-MOS series on Optimization. SIAM, Mar. 2015. Chap. 9. Forthcoming.

[43] M. Schmidt, M. C. Steinbach, and B. M. Willert. "High detail stationary optimization models for gas networks." In: *Optimization and Engineering* (2014), pp. 1–34. DOI: 10.1007/s11081-014-9246-x.

[44] M. Schmidt, M. C. Steinbach, and B. M. Willert. *High Detail Stationary Optimization Models for Gas Networks: Validation and Results.* Tech. rep. Submitted. Friedrich-Alexander-Universität Erlangen-Nürnberg, Discrete Optimization; Leibniz Universität Hannover, Institute for Applied Mathematics, Oct. 2014. URL: http://www.optimization-online.org/DB_HTML/2014/10/4602.html.

[45] O. Stein, J. Oldenburg, and W. Marquardt. "Continuous Reformulations of Discrete-Continuous Optimization Problems." In: *Comput. Chem. Eng.* 28.10 (2004), pp. 1951–1966.

[46] M. C. Steinbach. "On PDE Solution in Transient Optimization of Gas Networks." In: *J. Comput. Appl. Math.* 203.2 (2007), pp. 345–361. DOI: 10.1016/j.cam.2006.04.018.

[47] D. Sun and L. Qi. "On NCP-Functions." In: *Comput. Optim. Appl.* 13.1-3 (1999). Computational Optimization - a Tribute to Olvi Mangasarian, Part II, pp. 201–220. DOI: 10.1023/A:1008669226453.

[48] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.

[49] M. Tawarmalani and N. V. Sahinidis. "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study." In: *Math. Program.* 99.3 (2004), pp. 563–591.

[50] M. Tawarmalani and N. V. Sahinidis. "A polyhedral branch-and-cut approach to global optimization." In: *Math. Program.* 103.2 (2005), pp. 225–249. DOI: 10.1007/s10107-005-0581-8.

[51] SCIP*: Solving Constraint Integer Programs.* http://scip.zib.de/. 2015.

[52] S. Vigerske. "Decomposition in Multistage Stochastic Programming and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming." PhD thesis. Humboldt-Universität zu Berlin, 2012.

[53] A. Wächter and L. T. Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming." In: *Math. Program.* 106.1 (2006), pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[54] P. J. Wong and R. E. Larson. "Optimization of Natural-Gas Pipeline Systems Via Dynamic Programming." In: *IEEE Trans. Automat. Contr.* 13 (Oct. 1968), pp. 475–481. DOI: 10.1109/TAC.1968.1098990.

[55] P. J. Wong and R. E. Larson. "Optimization of Tree-Structured Natural-Gas Transmission Networks." In: *J. Math. Anal. Appl.* 24 (1968), pp. 613–626.

[56] S. Wright, M. Somani, and C. Ditzel. "Compressor Station Optimization." In: *30th Annual Meeting*. Paper 9805. Pipeline Simulation Interest Group. 1998.

[57] J. J. Ye and D. L. Zhu. "Optimality conditions for bilevel programming problems." In: *Optimization* 33 (1995), pp. 9–27.

[1]Daniel Rose, Marc C. Steinbach, Leibniz Universität Hannover, Institute for Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, [2]Martin Schmidt, Friedrich-Alexander-Universität Erlangen-Nürnberg, Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany; (b) Energie Campus Nürnberg, Fürther Str. 250, 90429 Nürnberg, Germany

E-mail address: [1]{rose,mcs}@ifam.uni-hannover.de

E-mail address: [2]mar.schmidt@fau.de

URL: [1]www.ifam.uni-hannover.de/~{rose,mcs}

URL: [2]http://www.mso.math.fau.de/edom/team/schmidt-martin