Solving Mixed-Integer Nonlinear Programs using Adaptively Refined Mixed-Integer Linear Programs

Robert Burlacu[1], Björn Geissler[1], and Lars Schewe[1]

Abstract. We propose a method for solving mixed-integer nonlinear programs (MINLPs) to global optimality by discretization of occuring nonlinearities. The main idea is based on using piecewise linear functions to construct mixed-integer linear program (MIP) relaxations of the underlying MINLP. In order to find a global optimum of the given MINLP we develope an iterative algorithm which solves MIP relaxations that are adaptively refined. We are able to give convergence results for a wide range of MINLPs requiring only continuous nonlinearities with bounded domains and an oracle computing maxima of the nonlinearities on their domain. Moreover, the practicalness of our approach is shown numerically by an application from the field of gas network optimization.

1. Introduction

A wide variety of algorithms to solve MINLPs exists. The main idea behind these is to augment standard branch-and-bound algorithms. We investigate one of the older ideas: Replacing nonlinearities with piecewise linear functions and solve the resulting problem as an MIP. The drawback of these methods is that a small approximation error leads to large MIPs, which become difficult to solve. Building on work by [11], we reduce the problem of solving an MINLP to solving a sequence of MIPs with gradually increasing accuracy.

We show convergence results and give tight bounds on the complexity of the MIPs required to achieve an a-priorily given accuracy. In order to obtain these results, our technique relies on combining a longest-edge bisection rule with MIP models for piecewise linear functions and analyze their interaction. The advantages of our approach are demonstrated on examples taken from gas transport optimization. We are able to provide near-optimal solutions to hard compressor cost optimization problems, which are difficult to solve for state-of-the-art MINLP solvers. More generally, our method can be used for arbitrary MINLP problems with bounded variables and requires only continuous functions for which global optima can be computed.

We highlight two differences from standard approaches for solving MINLPs: In contrast to spatial branch-and-bound, where a single branch-and-bound tree is used, we solve multiple MIPs and thus use multiple branch-and-bound trees. Moreover, unlike many methods that use only piecewise linear *approximations* to construct approximate solutions to MINLPs, we extend

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany

Key words and phrases. Mixed-Integer Nonlinear Programming, Piecewise Linear Approximation, Gas Transport Optimization, Global Optimization, Adaptivity.

such approximations to piecewise linear *relaxations* in our subproblems. As a consequence, we have both the opportunity to obtain global optimal solutions and prove infeasibility.

There are many possible ways to utilize piecewise linear approximations and relaxations for MINLPs. The main problem in all cases, however, is to construct good approximations of the nonlinear functions, the secondary problem is how to incorporate them into an MIP.

One way to construct such an approximation is to fix the error in advance and compute optimal linearizations for each function, see [29, 30] in case of approximations and [25] in case of one-dimensional relaxations. For up to three-dimensional functions, explicit approximation techniques for general nonlinear functions have been developed in [24]. Nevertheless, the main drawback of all these methods is that the number of simplices in the approximation grows exponentially with the dimension of the function. We mention the approach of Rovatti et al. [31] that circumvents this problem by dropping the requirement that the piecewise linear function needs to interpolate the original nonlinear function at the vertices of the triangulation. In our case, we base our relaxations on approximations that interpolate the function at the vertices. The relaxations, however, do not.

To integrate these methods into an MIP solver, we use the classical incremental method by Markowitz and Manne [22] as extended to relaxations in [11]. Originally developed for one-dimensional functions, a generalization to higher dimensions is described in [21, 38] and [9]. It yields a so-called locally ideal formulation of the resulting MIP; see [27, 28]. The main disadvantage of this (and related methods) is that the number of binary variables is linear in the number of simplices of the corresponding triangulation, and therefore, as mentioned above, grows exponentially in the dimension of the nonlinear function. This leads to a worst-case running time bound that is doubly exponential in the dimension of the nonlinear function. A way to, at least, get worst-case exponential and not doubly exponential bounds is the modeling technique proposed by Vielma and Nemhauser [37], which only needs a logarithmic number of binary variables.

To show that our technique is well-suited to solve structured MINLPs, we solve compressor energy minimization problems for gas transport networks. These have been studied in the literature, e.g., in [23] problems of this type have been solved using piecewise linear approximations. In [12] an algorithm that is similar to the one discussed here was applied to the problem, but no theoretical justification was given.

This article is organized as follows. In Section 2, we formally describe both the MINLP problems we consider and the algorithm to solve those. Additionally, some details on the construction of arising MIP relaxations are given. In Section 3, a convergence theory is developed, together with a complexity analysis for MIP relaxations occuring in our approach. Moreover, we show that the intuitive procedure of adding points with maximal linearization error while approximating a nonlinear function will not yield a convergent algorithm. Section 4 contains an exemplary application of our method, wherein in the context of gas transport optimization an MINLP model for the problem of compressor energy minimization is established.

After pointing out implementation issues in Section 5, we show numerical results in Section 6 demonstrating the advantages of our algorithm over state-of-the-art MINLP solvers.

2. Problem Statement and Algorithm

This section provides a formal description of the problems we deal with in the following, together with our solution algorithm.

We consider MINLP problems of type

$$\begin{aligned} \min\ & c^\top x \\ & g_i(x) = 0 && \text{for } i = 1,\dots,k_1, \\ & h_j(x) \le 0 && \text{for } j = 1,\dots,k_2, \\ & l \le x \le u, \\ & x \in \mathbb{R}^{d-p} \times \mathbb{Z}^p, \end{aligned} \tag{P}$$

where $k_1, k_2, d, p \in \mathbb{N}$, $g_i\colon \mathbb{R}^d \to \mathbb{R}$ for $i = 1,\dots,k_1$, and $h_j\colon \mathbb{R}^d \to \mathbb{R}$ for $j = 1,\dots,k_2$ are continuous real-valued functions and $l, u \in \mathbb{R}^d$ are the vectors of lower and upper bounds on the variables. Furthermore, with $\mathcal{F} := \{g_i\colon i = 1,\dots,k_1\} \cup \{h_j\colon j = 1,\dots,k_2\}$, $\mathcal{F}' \subset \mathcal{F}$ is the set of all nonlinear functions of $\mathcal{F}$ and $D_f \subset \mathbb{R}^d$ is the compact domain of the nonlinear function $f \in \mathcal{F}'$. Additionally, we assume that there is an oracle computing global optima, i.e., $\max_{x \in D_f} f(x)$, for any $f \in \mathcal{F}'$. Note that in this matter, for a considerable number of nonlinear functions, analytic formuals are given; see [9] for a more detailed discussion.

Our goal is to find an optimal solution for (P) such that no constraint of (P) is violated by more than an a-priorily (but arbitrarily) given error bound by applying techniques from mixed-integer linear programming. We therefore adaptively construct MIP relaxations of (P) and solve these to global optimality until no given error bound is violated. We indicate that in practice, the smallest attainable error bound is determined by the tolerance of the MIP solver.

Although not all constraints of (P) may be fulfilled exactly by our approach, we refer to it as global optimization for MINLP problems, since all a-priorily given error bounds are controllable and in fact all so-called global optimization algorithms, even for MIP or LP problems, deal with tolerances, e.g., for integrality, optimality, or constraint violations.

2.1. MIP Relaxations.

In order to construct MIP relaxations for (P), we extend the so-called Generalized Incremental Model representing piecewise linear approximations of the nonlinear functions contained in $\mathcal{F}'$; see [9] for more details and numerical experiments showing an advantage of the Generalized Incremental Model over other models.

Definition 2.1. The set $\mathcal{T} = \{S_1,\dots,S_n\}$ with full-dimensional simplices $S_i \subset \mathbb{R}^d$ for $i = 1,\dots,n$ is called a *triangulation* of D_f, if $D_f = \bigcup_{i=1}^n S_i$ and $\operatorname{int}(S_j) \cap \operatorname{int}(S_k) = \emptyset$ for every $j \neq k$.

In Definition 2.1, the notation $\operatorname{int}(S)$ is used to denote the relative interior of a set $S \subset \mathbb{R}^d$. From now on, we consider a simplex $S \subset \mathbb{R}^d$ to be full-dimensional and described by its extreme points $\mathcal{V}(S) = \{\bar{x}_0,\dots,\bar{x}_d\}$. Since

we are only interested in an optimal solution for (P) such that no constraint of (P) is violated by more than an a-priorily given error bound, for our purposes it suffices if a piecewise linear approximation ϕ_f of $f \in \mathcal{F}'$ is a set of continuous piecewise linear functions, which completely cover the domain D_f. Therefore, ϕ_f is not necessarily a function itself, but more generally described as a set of continuous affine functions $\phi_{f;S_i}(x)\colon S_i \to \mathbb{R}$, together with a triangulation $\mathcal{T}(\phi_f) = \{S_1, \dots, S_n\}$ of D_f and auxiliary binary variables $z_f \in \{0,1\}^n$ indicating which of the n functions $\phi_{f;S_i}$ is chosen. Herewith, the aim is to formulate a mixed-integer linear model in which $y = \phi_f(x, z_f)$ holds for $x \in D_f$.

There are two main ideas of the Generalized Incremental Model. At first, any point x^S inside a simplex S can be expressed either as a convex combination of its vertices or equivalently as $x^S = \bar{x}_0^S + \sum_{j=1}^d (\bar{x}_j^S - \bar{x}_0^S)\delta_j^S$ with $\sum_{j=1}^d \delta_j^S \le 1$ and $\delta_j^S \ge 0$ for $j = 1, \dots, d$.

The other main idea is that all simplices of a triangulation are ordered in such a way that the last vertex of any simplex is equal to the first vertex of the next one. In this way, we can construct a hamiltonian path and model the piecewise linear approximation along this path. It is now sufficient to show that an ordering of the simplices with the following properties is available:

(O1) The simplices in $\mathcal{T} = \{S_1, \dots, S_n\}$ are ordered such that $S_i \cap S_{i+1} \neq \emptyset$ for $i = 1, \dots, n-1$, and

(O2) for each simplex S_i its vertices $\bar{x}_0^{S_i}, \dots, \bar{x}_d^{S_i}$ can be labeled such that $\bar{x}_d^{S_i} = \bar{x}_0^{S_{i+1}}$ for $i = 1, \dots, n-1$.

More formally, the Generalized Incremental Model is described by:

$$x = \bar{x}_0^{S_1} + \sum_{i=1}^{n}\sum_{j=1}^{d} (\bar{x}_j^{S_i} - \bar{x}_0^{S_i})\delta_j^{S_i}, \tag{1a}$$

$$y = \bar{y}_0^{S_1} + \sum_{i=1}^{n}\sum_{j=1}^{d} (\bar{y}_j^{S_i} - \bar{y}_0^{S_i})\delta_j^{S_i}, \tag{1b}$$

$$1 \ge \sum_{j=1}^{d} \delta_j^{S_i} \qquad \text{for } i = 1, \dots, n, \tag{1c}$$

$$0 \le \delta_j^{S_i} \qquad \text{for } i = 1, \dots, n;\ j = 1, \dots, d, \tag{1d}$$

$$z_i \ge \sum_{j=1}^{d} \delta_j^{S_{i+1}} \qquad \text{for } i = 1, \dots, n-1, \tag{1e}$$

$$z_i \le \delta_d^{S_i} \qquad \text{for } i = 1, \dots, n-1, \tag{1f}$$

$$z_i \in \{0,1\} \qquad \text{for } i = 1, \dots, n-1. \tag{1g}$$

Constraints (1e)–(1g) ensure that the δ-variables satisfy the filling condition, which states that if for any simplex S_i a variable $\delta_j^{S_i}$ is positive, then $\delta_j^{S_{i-1}} = 1$ must hold. This means that $\delta_j^{S_i}$ can only be positive if for all previous simplices $k = 1, \dots, i-1$ the variables $\delta_d^{S_k}$ are equal to one.

With $\epsilon_u(f, S) := \max_{x \in S} f(x) - \phi_{f;S}$ as the maximum underestimation and $\epsilon_o(f, S) := \max_{x \in S} \phi_{f;S} - f(x)$ as the maximum overestimation of f by $\phi_{f;S}$,

an MIP relaxation for (P) can easily be obtained by replacing (1b) with

$$y = \bar{y}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} (\bar{y}_j^{S_i} - \bar{y}_0^{S_i}) \delta_j^{S_i} + e,$$

and adding the inequalities

$$\epsilon_u(f, S_1) + \sum_{i=1}^{n-1} z_i (\epsilon_u(f, S_{i+1}) - \epsilon_u(f, S_i)) \geq e,$$

$$-\epsilon_o(f, S_1) - \sum_{i=1}^{n-1} z_i (\epsilon_o(f, S_{i+1}) - \epsilon_o(f, S_i)) \leq e.$$

For results concerning the computation of the approximation errors $\epsilon_u(f, S)$ and $\epsilon_o(f, S)$, which is based on computing global optima for f, we again refer to [9].

2.2. **The Algorithm.** We start by solving a coarse MIP relaxation and check whether all error bounds are satisfied in its optimal solution. Then, we locally refine those piecewise linear approximations, where the error bounds are violated. Afterward, from the refined approximation, a new MIP relaxation is constructed and the procedure is started over with the new relaxation until all error bounds are satisfied. A more formal description of this approach is given in Algorithm 1.

In the following we denote the projection of the optimal solution x of an MIP relaxation on D_f by x_f for all $f \in \mathcal{F}'$. Moreover, the Euclidean norm is denoted by $\|\cdot\|$. The refinement procedure in Algorithm 1 is in our case a longest-edge bisection procedure as given in Algorithm 2.

Algorithm 2 Longest-edge bisection on a simplex S

Require: A simplex S and a scalar δ.

Ensure: If the longest edge of S is greater than δ, a set of two simplices S' and S'' with $S = S' \cup S''$ is returned. Otherwise no refinement is performed.

1: Set $e \leftarrow$ longest edge of S with $\mathcal{V}(e) = \{\bar{x}_a, \bar{x}_b\}$ and $\bar{x}_a, \bar{x}_b \in \mathcal{V}(S)$.
2: **if** $\|e\| \leq \delta$ **then**
3: **return** $\mathcal{V}(S)$.
4: **else**
5: Set $\hat{x} \leftarrow$ midpoint of the longest edge e.
6: Set $\mathcal{V}(S') \leftarrow (\mathcal{V}(S) \setminus \bar{x}_b) \cup \hat{x}$; $S' \leftarrow \text{conv}(\mathcal{V}(S'))$.
7: Set $\mathcal{V}(S'') \leftarrow (\mathcal{V}(S) \setminus \bar{x}_a) \cup \hat{x}$; $S'' \leftarrow \text{conv}(\mathcal{V}(S''))$.
8: **return** S', S''.
9: **end if**

Remark 2.2. We easily obtain a triangulation $\mathcal{T}(\phi_f^{i+1})$ of D_f by replacing S_f in $\mathcal{T}(\phi_f^i)$ with S'_f and S''_f, since $S_f = S'_f \cup S''_f$. Therefore, there is a piecewise linear approximation $\phi_f^{i+1}(\tilde{x}_f)$ for every $\tilde{x}_f \in D_f$.

Algorithm 1 Global optimization by adaptively refined MIP relaxations

Require: An MINLP problem P of type (P), upper bounds $\epsilon_f^0 > 0$ for the absolute linearization errors in the piecewise linear approximations used to construct the initial relaxation and the maximal absolute linearization errors $\epsilon_f > 0$ for all $f \in \mathcal{F}'$.

Ensure: If P is feasible, the algorithm returns an optimal solution x of an MIP relaxation Π of P with $|f(x_f) - y_f| \leq \epsilon_f$ for all $f \in \mathcal{F}$ and $c^T x \leq c^T x'$ for any feasible point x' of P. If no such MIP relaxation Π of P exists, this is reported by returning *infeasible*.

Compute an initial piecewise linear approximation ϕ_f^0 of $f \in \mathcal{F}'$ satisfying the upper bound ϵ_f^0 for all $f \in \mathcal{F}'$.
Set $i \leftarrow 0$.
repeat
  Construct an MIP relaxation Π^i of P from ϕ_f^i for all $f \in \mathcal{F}'$.
  Solve Π^i.
  if Π^i is *feasible* **then**
    Set $x^i \leftarrow$ optimal solution of Π^i.
  else
    return *infeasible*.
  end if
  Set stop $\leftarrow$ true.
  for all $f \in \mathcal{F}'$ **do**
    Set $x_f^i \leftarrow$ projection of x^i on D_f.
    Set $y_f^i \leftarrow$ value of the variable for the approximated function value of f.
    if $\left|f(x_f^i) - y_f^i\right| > \epsilon_f$ **then**
      Set $z_f \leftarrow$ values of the binary variables in x^i used to model ϕ_f^i.
      Set $S_f \leftarrow$ simplex in $\mathcal{T}(\phi_f^i)$, which has been selected according to z_f.
      Set $\mathcal{T}(\phi_f^{i+1}) \leftarrow$ refinement of the triangulation $\mathcal{T}(\phi_f^i)$ (see Algorithm 2).
      Set $\phi_f^{i+1} \leftarrow$ piecewise linear approximation according to $\mathcal{T}(\phi_f^{i+1})$.
      Set stop $\leftarrow$ false.
    else
      Set $\phi_f^{i+1} \leftarrow \phi_f^i$.
    end if
  end for
  Set $i \leftarrow i + 1$.
until stop.
return x^{i-1}.

3. Theoretical Results

We begin this section by proving the correctness and convergence of Algorithm 1. Subsequently, we show a result justifying the utilization of the longest-edge bisection. Finally, an analysis regarding the number of

refinement steps performed in Algorithm 1 and consequently the size of correspondig MIP relaxations is given.

3.1. **Convergence Results.** We first of all show that a piecewise linear approximations obtained by Algorithm 2 can be modeled by the Generalized Incremental Model.

Theorem 3.1. *Let $\mathcal{T}$ be a triangulation with properties (O1) and (O2). Then any triangulation $\mathcal{T}'$ obtained by applying Algorithm 2 to $\mathcal{T}$ has properties (O1) and (O2).*

Proof. With $\mathcal{T} = \{S_1, \dots, S_n\}$, let S_k be the simplex which has to be refined and $\bar{x}_0^{S_k}, \dots, \bar{x}_d^{S_k}$ its labeled vertices. One of the two simplices S'_k and S''_k (as in Algorithm 2) contains $\bar{x}_0^{S_k}$, whereas the other one contains $\bar{x}_d^{S_k}$, since $\mathcal{V}(S'_k)$ and $\mathcal{V}(S''_k)$ only differ in one vertex. Let $\bar{x}_0^{S_k}$ be contained in S'_k and $\bar{x}_d^{S_k}$ in S''_k. We order the simplices of $\mathcal{T}'$ as

$$(S_1, \dots, S_{k-1}, S'_k, S''_k, S_{k+1}, \dots, S_n) \tag{2}$$

and apply the following labeling:

$$\bar{x}_0^{S'_k} = \bar{x}_0^{S_k},\ \bar{x}_d^{S'_k} = \hat{x}; \qquad \bar{x}_0^{S''_k} = \hat{x},\ \bar{x}_d^{S''_k} = \bar{x}_d^{S_k}. \tag{3}$$

Furthermore, S_{k-1} and S'_k are linked by $\bar{x}_d^{S_{k-1}} = \bar{x}_0^{S'_k}$, and S''_k and S_{k+1} by $\bar{x}_d^{S''_k} = \bar{x}_0^{S_{k+1}}$. Therefore, the ordering (2) of the simplices of $\mathcal{T}'$ has properties (O1) and (O2), because $S'_k \cap S''_k \neq \emptyset$ trivially holds, $\bar{x}_d^{S'_k} = \bar{x}_0^{S''_k}$ as in (3) and the rest is inherited from $\mathcal{T}$. □

Note that in case no initial triangulation is given, we can simply choose a standard triangulation with vertices equal to the set of extreme points of D_f; see [26] on this topic. As shown in [9], such a triangulation has always an ordering satisfying (O1) and (O2). With higher dimension d of D_f, however, even the triangulation of D_f itself becomes intractable, because at least $\lceil 6^{\frac{d}{2}} d! / \left(2(d+1)^{\frac{d+1}{2}}\right) \rceil$ simplices are needed; see [32].

Definition 3.2. The refinement procedure in Algorithm 1 is called *δ-precise*, if for an arbitrary sequence $S^i \in \mathcal{T}_i$ of refined simplices with initial triangulation $\mathcal{T}_0$ of D_f and given $\delta > 0$, there exists an index $N \in \mathbb{N}$, such that

$$\operatorname{diam}(S^N) < \delta \tag{4}$$

holds, whereby $\operatorname{diam}(S^N) := \sup_{x', x'' \in S^N} \{\|x' - x''\|\}$.

We now show that Algorithm 2 is δ-precise for every $\delta > 0$. Let $\widetilde{\mathcal{T}}_k$ be the refined triangulation of an initial triangulation $\mathcal{T}_0$ obtained by applying Algorithm 2 in such a way that in every iteration $i \leq k$ all simplices of $\widetilde{\mathcal{T}}_{i-1}$ are refined.

Lemma 3.3. *Let $S \in \mathbb{R}^d$ be a simplex of $\mathcal{T}_0$ and e the longest edge of S. Then the longest edge of any simplex of $\widetilde{\mathcal{T}}_{l\binom{d+1}{2}}$ contained in S is bounded by $\left(\frac{\sqrt{3}}{2}\right)^l \|e\|$ with $l \in \mathbb{N}$.*

Proof. Let $\tilde{e}$ be one of the $d-1$ edges that are added in the first refinement step. Since $\tilde{e}$ can also be considered as the median of the triangle T described by $\bar{x}_a, \bar{x}_b$, where $\operatorname{conv}(\{\bar{x}_a, \bar{x}_b\}) = e$ (as in Algorithm 2), and a vertex $\bar{x}_c$ of the remaining $d-1$ vertices of S, with Apollonius' theorem it follows that

$$\|\tilde{e}\| \leq \frac{\sqrt{3}}{2}\|e\|. \tag{5}$$

Since a longest edge is halved in a refinement step and there are at most $\binom{d+1}{2}$ longest edges in S, the longest edge of any simplex of $\widetilde{\mathcal{T}}_{\binom{d+1}{2}}$ contained in S is bounded by (5). Applying this argument recursively, we see that after $l\binom{d+1}{2}$ refinement steps any edge of a simplex of $\widetilde{\mathcal{T}}_{l\binom{d+1}{2}}$ contained in S is bounded by $\left(\frac{\sqrt{3}}{2}\right)^l \|e\|$. □

Theorem 3.4. *Let $\delta > 0$, then there is an $\tilde{N} \in \mathbb{N}$, such that $\widetilde{\mathcal{T}}_{\tilde{N}}$ is a refinement of every triangulation obtained by Algorithm 2 with δ as input parameter.*

Proof. We focus only on $d \geq 2$, because in the case $d = 1$ the proposition obviously holds. With $l = \max\left\{0, \left\lceil \ln\left(\frac{\|e_0\|}{\delta}\right) / \ln\left(\frac{2}{\sqrt{3}}\right)\right\rceil\right\}$, applying Lemma 3.3 we conclude that after at most

$$\tilde{N} := \binom{d+1}{2} \max\left\{0, \left\lceil \frac{\ln\left(\frac{\|e_0\|}{\delta}\right)}{\ln\left(\frac{2}{\sqrt{3}}\right)} \right\rceil\right\}$$

refinement steps the longest edge of any simplex of $\widetilde{\mathcal{T}}_{\tilde{N}}$ is bounded by δ, where $\delta > 0$ and e_0 is the longest edge of all simplices of $\mathcal{T}_0$. Since Algorithm 2 only refines simplices with a longest edge larger than δ and no simplex in $\widetilde{\mathcal{T}}_{\tilde{N}}$ has an edge longer than δ, it follows by the pigeonhole principle that no refinement of $\mathcal{T}_0$ obtained by Algorithm 2 can be finer than $\widetilde{\mathcal{T}}_{\tilde{N}}$. □

Theorem 3.5. *Algorithm 2 is δ-precise for every $\delta > 0$. With $\tilde{N}$ as in Theorem 3.4, the number of refinement steps N (as in Definition 3.2) is bounded by*

$$N := m(2^{\tilde{N}} - 1) + 1, \tag{6}$$

where m is the number of simplices contained in $\mathcal{T}_0$.

Proof. Counting every single simplex that has to be refined in order to obtain the triangulation $\widetilde{\mathcal{T}}_{\tilde{N}}$ from $\mathcal{T}_0$, we get

$$m(1 + 2 + 4 + \ldots + 2^{\tilde{N}-1}) = m(2^{\tilde{N}} - 1)$$

refinements in total. Again, by the pigeonhole principle, it follows that every sequence $S^i \in \mathcal{T}_i$ of simplices has an element S^k with index $k \leq m(2^{\tilde{N}} - 1) + 1$, such that $S^k \in \widetilde{\mathcal{T}}_{\tilde{N}}$, since $\widetilde{\mathcal{T}}_{\tilde{N}}$ is a refinement of every triangulation obtained by Algorithm 2 with parameter δ. Therefore, simplex S^k has property (4), as no simplex in $\widetilde{\mathcal{T}}_{\tilde{N}}$ has an edge longer than δ. □

Theorem 3.6. *If the refinement procedure in Algorithm 1 is δ-precise for every $\delta > 0$, then Algorithm 1 is correct and terminates after a finite number of steps.*

Proof. We first show that Algorithm 1 terminates after a finite number of steps and prove correctness afterward.

Let x_f^i be the sequence of (projected) optimal solutions of Π^i for $f \in \mathcal{F}$. Since the refinement procedure is δ-precise for every $\delta > 0$, there is an index $N \in \mathbb{N}$ with $\|\bar{x} - \bar{x}_i\| < \delta$ for every $\delta > 0$, $\bar{x} \in S_f^N$ and $\bar{x}_i \in \mathcal{V}(S_f^N)$. Furthermore, because D_f is compact, f is uniformly continuous on D_f and therefore, there is a $\delta_f > 0$ such that $|f(\bar{x}) - \phi_f^N(\bar{x})| < \epsilon_f/2$. From the construction of the MIP relaxation it follows that $y_f^N \in [\phi_f^N(x_f^N) - (\epsilon_f/2), \phi_f^N(x_f^N) + (\epsilon_f/2)]$. Thus, we have

$$|f(x_f^N) - y_f^N| \leq |f(x_f^N) - \phi_f^N(x_f^N)| + \frac{\epsilon_f}{2} \leq \epsilon_f$$

and Algorithm 1 terminates after a finite number of steps.

If P is feasible, we can assure that Algorithm 1 does not return *infeasible* in any iteration. Since the algorithm terminates after a finite number of steps, there must be an MIP relaxation Π^N which has an optimal solution x^N with $|f(x_f^N) - y_f^N| \leq \epsilon_f$ for all $f \in \mathcal{F}$. Moreover, Π^N is a relaxation of P so that $c^T x^N \leq c^T x'$ for all feasible points x' of P.

If, on the other hand, there exists no MIP relaxation Π of P which has a feasible point $\tilde{x}$ satisfying $|f(\tilde{x}_f) - y_f| \leq \epsilon_f$ for all $f \in \mathcal{F}$, we can easily conclude that Algorithm 1 must return *infeasible*, since it terminates after a finite number of steps. □

Corollary 3.7. *Algorithm 1 together with* Algorithm 2 *as refinement procedure is correct and terminates after a finite number of steps.*

Proof. The corollary follows directly from Theorem 3.5 and Theorem 3.6. □

We point out, that for a continuous function $f \in \mathcal{F}'$, in general, its corresponding δ_f as in the proof of Theorem 3.6 might not be computable. Since such a $\delta_f > 0$ exists for every continuous $f \in \mathcal{F}'$, it suffices to set $\delta_f := 0$ as input parameter for Algorithm 2. In practice, however, the lower bound of an attainable δ is determined by the MIP solver.

3.2. **Limited Accuracy by Adding Points with Maximal Error.** Considering the refinement procedure in Algorithm 1, we are not restricted to using a longest-edge bisection as in Algorithm 2. Another, more intuitive approach is the following: If a simplex S_f of triangulation $\mathcal{T}(\mathcal{V}_f)$ is chosen for refinement, we add a point with maximal approximation error on S_f to the set $\mathcal{V}_f$ of linearization points obtaining a new triangulation $\mathcal{T}(\mathcal{V}_f \cup \{v\})$. Unfortunately, an approximation with arbitrary accuracy is not always attainable by such a procedure.

Indeed, let ϕ_f be a piecewise linear approximation of a nonlinear function $f \in \mathcal{F}'$ on a domain D_f with some triangulation $\mathcal{T}(\mathcal{V}_f)$ corresponding to a set $\mathcal{V}_f$ of linearization points obtained by successively adding solely points with maximal approximation error on a simplex. We show that there are nonlinear functions f, which cannot be approximated by ϕ_f with arbitrary accuracy.

Theorem 3.8. *Let $g\colon \mathbb{R} \to \mathbb{R}$ be a continuous nonlinear function with at least three roots x_0, x_1, and x_2. Additionally, let $g(x) > 0$ for $x \in (x_0, x_1)$*

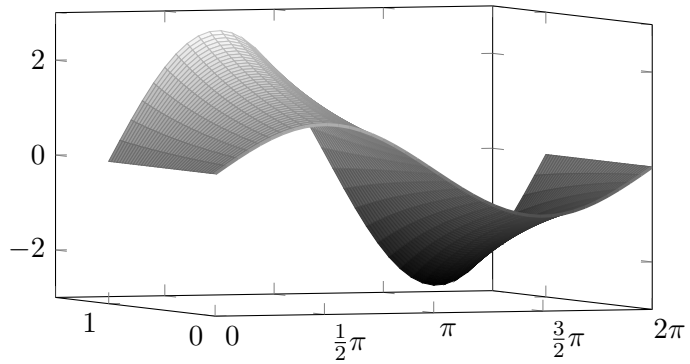

FIGURE 1. Extension of $g(x) = \sin(x)$ and $0 \leq x \leq 2\pi$ by multiplying with $h(y) = e^y$.

and $g(x) < 0$ for $x \in (x_1, x_2)$ or vice versa. Then g can be extended to a two-dimensional function f with domain D_f, such that there exists a fixed $\epsilon > 0$ with $|\phi_f(x,y) - f(x,y)| > \epsilon$ for any ϕ_f and some $(x,y) \in D_f$ depending on ϕ_f.

Proof. Let $h\colon \mathbb{R} \to \mathbb{R}$ be a continuous and strictly increasing function with $h(0) > 0$ and $f\colon D_f \to \mathbb{R}, (x,y) \mapsto g(x)h(y)$ with

$$D_f = \{(x,y) \in \mathbb{R}^2 \,:\, x_0 \leq x \leq x_2,\ 0 \leq y \leq \bar{y}\}$$

and arbitrary $\bar{y} > 0$. See Figure 1 for an illustration of such a function. We now prove that

$$\tilde{\epsilon} := \min\left\{ \max_{x \in (x_0, x_1)} |g(x)|, \max_{x \in (x_1, x_2)} |g(x)| \right\} h(0)$$

is a lower bound for the approximation error of any piecewise linear approximation ϕ_f defined as above.

Let ϕ_f be such a piecewise linear approximation. Due to the construction of f, if $S_e \in \mathcal{T}(\mathcal{V}_f)$ is the simplex containing the edge $e = \operatorname{conv}\{(x_0, 0), (x_2, 0)\}$, then for all $(x,y) \in S_e$ either $\phi_f(x,y) \geq 0$ or $\phi_f(x,y) \leq 0$ holds. We now choose X to be one of the two intervals (x_0, x_1) and (x_1, x_2), such that with $x \in X$ either $g(x) > 0$ and $\phi_f(x,y) \leq 0$ or $g(x) < 0$ and $\phi_f(x,y) \geq 0$ holds. Then for any $x \in X$ it follows that

$$|0 - g(x)h(0)| = |\phi_f(x,0) - f(x,0)| < |\phi_f(x,y) - f(x,y)| \tag{7}$$

for all $y > 0$, as $h(y)$ is strictly increasing and that a point with maximal approximation error can never be contained on edge e. Thus, there is a simplex S_e with edge e in any approximation ϕ_f, because only points with maximal approximation error are added to the triangulation. Furthermore, considering that $\phi_f(x,0) = 0$ in any ϕ_f, with (7) it follows, that the maximal approximation error on S_e, and therefore on any piecewise linear approximation ϕ_f, is larger than $\tilde{\epsilon}$. □

Note that although a refinement strategy adding solely points with maximal approximation error on a simplex often works well in practice according to [9], [12], it is due to Theorem 3.8 not necessarily δ-precise for every $\delta > 0$. Moreover, there are many functions $g(x)$ as in Theorem 3.8, e.g., polynomials of kind $(x-a_1)(x-a_2)(x-a_3)$, and therefore many corresponding

functions $f(x, y)$. Hence, in terms of global optimization and controlling the approximation error, the refinement strategy adding linearization points on the longest edge of a simplex, such as the longest-edge bisection, is naturally motivated by Theorem 3.8.

3.3. **Complexity Analysis.** Finally, we analyze the complexity of the resulting MIP relaxations in Algorithm 1 in terms of their sizes. In our context, complexity is narrowed down to the maximal number of continuous and binary variables.

As we have seen in the previous section and due to Theorem 3.5, using the Generalized Incremental Method the number of both continuous and binary variables is of size $\mathcal{O}\big(2^{\tilde{N}}\big)$ or, with $\tilde{N} = \mathcal{O}(d^2|\ln(\operatorname{diam}(D_f)/\delta_f)|)$, of size $\mathcal{O}\big((\operatorname{diam}(D_f)/\delta_f)^{d^2}\big)$. In order to give more exact bounds we assume for the rest of this section that $f \in \mathcal{F}'$ is Lipschitz-continuous with Lipschitz-constant L_f. This is a common assumption in terms of practically interesting MINLP problems. Concerning convergence of Algorithm 1, we get the following result.

Theorem 3.9. *Let $f \in \mathcal{F}'$ be Lipschitz-continuous with Lipschitz-constant L_f. Then it suffices to set $\delta_f := \epsilon_f/(2L_f)$ as input parameter in Algorithm 2, so that Algorithm 1 together with Algorithm 2 as refinement procedure is correct and terminates after a finite number of steps.*

Proof. The proof works almost analogous to the proof of Theorem 3.6. With $\delta_f = \epsilon_f/(2L_f)$, now $\big|f(\bar{x}) - \phi_f^N(\bar{x})\big| < \epsilon_f/2$ holds by applying Lipschitz-continuity of f. □

Regarding the number of refinements performed in Algorithm 1, we now have

$$\tilde{N} = \mathcal{O}\left(d^2\left|\ln\left(\frac{L_f \operatorname{diam}(D_f)}{\epsilon_f}\right)\right|\right) \tag{8}$$

and a maximal number of variables of size $\mathcal{O}\big((L_f \operatorname{diam}(D_f)/\epsilon_f)^{d^2}\big)$. Compared to the classical result of $\Theta\big((1/\epsilon_f)^d\big)$ on the complexity of computing approximations of f to within ϵ_f, see [34] for more details, we see that a longest-edge bisection might not yield an approximation of f with minimal cost, since every binary variable corresponds to a refinement step and therefore an evaluation of f. However, due to the adaptivity of Algorithm 1 and because we only need locally fine approximations, a longest-edge bisection is very convenient in our case.

As shown by Vielma and Nemhauser in [36] and [37], a modeling of the MIP relaxations obtained by Algorithm 1 with a logarithmic number of binary variables is possible. Since their so-called Logarithmic Disaggregated Convex Combination Model is only based on a triangulation of D_f as in Definition 2.1 and does not impose any other special requirements, with Remark 2.2 we can easily use the model for our MIP relaxations. Therefore, concerning the number of binary variables we get $\mathcal{O}(\tilde{N})$ with $\tilde{N}$ as in (8), whereas the number of continuous variables equals the one of the Generalized Incremental Method and is therefore of size $\mathcal{O}(2^{\tilde{N}})$. However, we point out that in practice the Generalized Incremental Method might be superior in terms of runtime; see [9] and [4] for more detailed discussions.

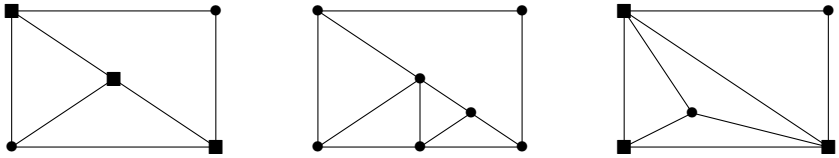

FIGURE 2. Triangulations of a two-dimensional box for which only a 3-way IB scheme (left and right), and a 2-way IB scheme (middle) exist. In case of 3-way IB schemes a minimal infeasible set of size three is marked by squares.

The Logarithmic Branching Convex Combination Model described by Vielma and Nemhauser requires a smaller amount of continuous variables than the Logarithmic Disaggregated Convex Combination Model using branching dichotomies deciding if a variable corresponding to a vertex of the triangulation is set to zero or not. In this way, only one continuous variable is needed for each vertex of a triangulation, instead of $d+1$ variables for each d-dimensional simplex of a triangulation as in the Logarithmic Disaggregated Convex Combination Model. Very recently, the approach of Vielma and Nemhauser has been generalized in the preprint [16] introducing the notion of k-way independent branching (IB) schemes, whereby a 2-way IB scheme for a triangulation is needed in order to utilize the Logarithmic Branching Convex Combination Model. Furthermore, the existence of k-way IB schemes has been characterized as follows. With $\mathcal{V}$ as the set of all vertices of the triangulation, a subset $V \in \mathcal{V}$ is called infeasible, if V is not contained in the set of all vertices describing some simplex $S \in \mathcal{S}$ of the triangulation, i.e., $V \not\subseteq \mathcal{V}(S)$ for all $S \in \mathcal{S}$.

Proposition 3.10 (Theorem 1, [16])**.** *A k-way IB scheme exists if and only if each minimal infeasible set $V \in \mathcal{V}$ has $|V| \leq k$.*

Unfortunately, this model is unsuitable for our purpose since 2-way IB schemes for triangulations occuring in Algorithm 1 not always exist, as we can see in Figure 2. Note that there is no direct impact of hanging nodes of a triangulation to the existence of 2-way IB schemes.

4. Our Application: Compressor Energy Minimization

In this section we cover an essential problem in gas transport networks optimization: Consider a gas transport network and a nomination, i.e., a pre-announced amount of gas for each so-called entry, where gas can be fed into the network, and each so-called exit, where gas can be withdrawn from the network. Find an admissible configuration of the controllable elements satisfying all physical and technical constraints. Furthermore, due to friction induced pressure drop on long pipes, compressor machines increasing certain gas pressures have to be taken into account. This leads to an objective minimizing the compressor energy. As gas physics is described via nonlinear functions, considering the discrete characteristic of the controllable elements, the problem can be modeled as an MINLP.

We base our model on the model used in [10]; for details and a thorough review of the literature we refer to this chapter and the book [19] in general.

The compressor model used here is more detailed than the one used in [10]; we refer to the discussion in [7] and [13].

4.1. **The MINLP Model.** We use a directed finite graph $G = (V, A)$ to model the gas network. The set V of nodes consists of the set V_+ of entry nodes (supplying gas), the set V_- of exit nodes (discharging gas) and the set V_0 of inner nodes (neither supplying nor discharging gas). The set A of arcs is composed of the set A_{P} of pipes, the set A_{V} of valves, the set A_{CV} of control valves, the set A_{S} of shortcuts, the set A_{R} of resistors and the set A_{CM} of compressor machines. Furthermore, the gas mass flow on an arc $a = (u, v) \in A$ is indicated by q_a and the pressure at a node $v \in V$ is indicated by p_v. A gas mass flow from node u to v will result in a positive gas mass flow $q_a \geq 0$, whereas a gas mass flow from node v to u corresponds to a negative gas mass flow $q_a < 0$. In the following, whenever we speak about gas flow, we refer to gas mass flow.

Considering a specific nomination, for each node $v \in V$, certain parameters are given: lower and upper bounds p_v^- and p_v^+ of the pressure variable p_v and the demand d_v in terms of gas flow. Thereby, we model a node by implementing lower and upper bounds on the pressure variable:

$$p_v^- \leq p_v \leq p_v^+. \tag{9}$$

Moreover, in terms of mass flow conservation, the constraint

$$\sum_{a \in \delta^-(v)} q_a - \sum_{a \in \delta^+(v)} q_a = d_v \tag{10}$$

is added. The demand d_v is non-negative for exit nodes, non-positive for entry nodes and zero for inner nodes.

4.1.1. *Pipes.* The most frequent components in our gas network are pipes. Every pipe $a = (u, v) \in A_{\mathrm{P}}$ is determined by its length L_a, diameter D_a, and roughness k_a. Gas flow within a pipe can be mainly described by mass flow q, pressure p, temperature T, density ρ, and the system of Euler equations together with a specific equation of state. For our purpose, the equation of state is given by the thermodynamical standard equation for real gases

$$\rho R_s z T = p,$$

where R_s is the specific gas constant and z is the compressibility factor. The latter can be described by the formula

$$z(p, T) = 1 + \alpha p, \quad \alpha = 0.257 \frac{1}{p_c} - 0.533 \frac{T_c}{p_c T} \tag{11}$$

of the American Gas Association. Furthermore, the pseudocritical pressure p_c and the pseudocritical temperature T_c, as well as the temperature T are assumed to be constant. Through certain simplifications, the pressure drop on a pipe can be given in analytic form using the well-known Weymouth equation

$$p_v^2 - p_u^2 = \frac{L_a \lambda_a R_s z_a T}{A_a^2 D_a} |q_a| q_a. \tag{12}$$

The friction factor λ_a is computed by Nikuradse's formula involving the diameter and roughness of the pipe. Moreover, with the diameter of the pipe, the cross-sectional area $A_a = D_a^2 \pi / 2$ can be computed. Regarding the

compressibility factor z_a, we use an averaged pressure on the pipe derived by the pressure bounds resulting in

$$z_a = z\Big(\frac{\max\{p_u^-, p_v^-\} + \min\{p_u^+, p_v^+\}}{2}, T\Big). \tag{13}$$

4.1.2. *(Control) Valves.* A valve $a = (u, v) \in A_{\mathrm{V}}$ is a controllable element in a gas network, which can either be closed or open. A closed valve impedes gas from passing, which leads to decoupled pressures at nodes u and v. On the contrary, for open valves, we have $p_u = p_v$ and no pressure drop. Valves are modeled with the help of binary switching variables $s_a \in \{0, 1\}$, whereby s_a is equal to one, if and only if the valve is open and vice versa:

$$q_a^- s_a \le q_a \le q_a^+ s_a, \tag{14a}$$

$$(p_v^+ - p_u^-)s_a + p_v - p_u \le p_v^+ - p_u^-, \tag{14b}$$

$$(p_u^+ - p_v^-)s_a + p_u - p_v \le p_u^+ - p_v^-. \tag{14c}$$

Like a valve, a control valve $a = (u, v) \in A_{\mathrm{CV}}$ can either be closed or open. Again, a closed control valve impedes gas from passing, resulting in decoupled pressures at nodes u and v. An open control valve, however, can reduce pressure within a given range $[\Delta_a^-, \Delta_a^+]$. To model a control valve, we use (14a) from above and replace equations (14b) and (14c) by

$$(p_v^+ - p_u^- + \Delta_a^-)s_a + p_v - p_u \le p_v^+ - p_u^-, \tag{15a}$$

$$(p_u^+ - p_v^- - \Delta_a^+)s_a + p_u - p_v \le p_u^+ - p_v^-. \tag{15b}$$

We point out that a control valve is a unidirectional element, i.e., $q_a^- \ge 0$ and that pressure can only be reduced in flow direction. Furthermore, a control valve is always located in a so-called control valve station, which additionally provides the possibility of a bypass, leading to identical pressures at nodes u and v.

4.1.3. *Shortcuts.* Shortcuts are elements, which are solely used for modeling purposes and do not exist in reality. They are often used in practice, e.g., to model hybrid points, such as gas storages. Modeling shortcuts can simply be done by adding the constraint

$$p_u = p_v \quad \text{for all } a = (u, v) \in A_{\mathrm{S}}. \tag{16}$$

4.1.4. *Resistors.* To model gas network elements causing pressure drop, e.g., measuring stations and filtration plants, we use resistors. Unlike control valves, resistors operate in both directions, which can be modeled with a binary variable $d_a \in \{0, 1\}$, where d_a equals to zero, if and only if gas flows in direction of the arc $a \in A_{\mathrm{R}}$, i.e., $q_a \ge 0$ and vice versa:

$$q_a^- d_a \le q_a \le q_a^+(1 - d_a). \tag{17}$$

We split the set of resistors $A_{\mathrm{R}} = A_{\mathrm{R_v}} \cup A_{\mathrm{R_c}}$ differentiating between two types of resistors: resistors $a \in A_{\mathrm{R_v}}$ with variable pressure drop and resistors $a \in A_{\mathrm{R_c}}$ with constant pressure drop.

A resistor $a \in A_{\mathrm{R_v}}$ with variable pressure drop is determined by its drag factor ξ_a and diameter D_a. Together with the equation for variable pressure drop by a resistor and using, like in the case of pipes, a constant temperature T

and an approximated compressibility factor z_a as in (13), we can model a resistor $a = (u, v) \in A_{\mathrm{R_v}}$ by

$$p_u^2 - p_v^2 + |\Delta_a|\Delta_a = w_a|q_a|q_a, \quad \Delta_a = p_u - p_v, \tag{18a}$$

$$(p_v^+ - p_u^-)d_a \le p_u - p_v \le (p_u^+ - p_v^-)(1 - d_a), \tag{18b}$$

The pressure drop is now given by the variable Δ_a and w_a is a resistor-specific constant.

In case of a resistor $a = (u, v) \in A_{\mathrm{R_c}}$ with constant pressure drop $\overline{\Delta}_a$, we simply model the resistor by adding

$$p_u - p_v + 2\overline{\Delta}_a d_a = \overline{\Delta}_a. \tag{19}$$

4.1.5. *Compressor Machines.* In order to compensate for pressure drop on large gas networks, compressor machines increasing the inlet pressure to a higher outlet pressure, are installed. Each compressor machine $a \in A_{\mathrm{CM}}$ is specified by its adiabatic efficiency $\eta_{\mathrm{ad},a}$ and its energy cost coefficient c_a. Again, we introduce a binary switching variable s_a, where $s_a = 1$ means that the machine is active and vice versa.

Furthermore, the operating range of a compressor machine is described by its non-convex characteristic diagram. We subdivide the set $A_{\mathrm{CM}} = A_{\mathrm{TC}} \cup A_{\mathrm{PC}}$ distinguishing between turbo compressors $a \in A_{\mathrm{TC}}$ (usually driven by gas turbines) and piston compressors $a \in A_{\mathrm{PC}}$ (typically shipped with electric or gas driven motors); see Figure 3 for their corresponding characteristic diagrams. The gray area describes the feasible operating range of the machine; see [7] for more details.

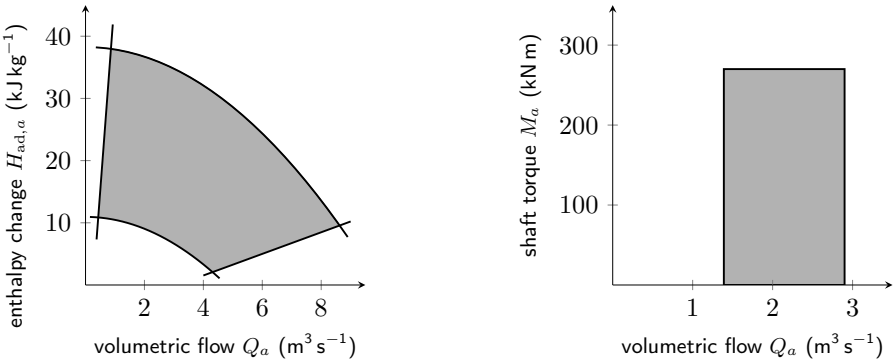

FIGURE 3. Characteristic diagrams for turbo compressor machines (left) and piston compressor machines (right). The feasible operating ranges are marked gray.

Since the description of the operating range requires a volumetric flow Q_a and we restricted ourselves on mass flow q_a, we first have to add

$$Q_a = \frac{q_a}{\rho_0} \quad \text{for all } a \in A_{\mathrm{CM}}, \tag{20}$$

where ρ_0 is the gas density under normal conditions, to the model. Furthermore, the adiabatic process of compression leads to a change in the specific

enthalpy $H_{\mathrm{ad},a}$. This can be modeled by the constraint

$$H_{\mathrm{ad},a} = \frac{\kappa}{\kappa - 1} R_s z_a T \left(\left(\frac{p_v}{p_u} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right) \quad \text{for all } a = (u, v) \in A_{\mathrm{CM}},$$

where κ is the adiabatic exponent which is assumed to be constant. Now, using (11) and a constant temperature T, the compressibility factor is approximated by

$$z_a = z\left(\frac{p_u^- + p_u^+}{2}, T \right) \quad \text{for all } a = (u, v) \in A_{\mathrm{CM}}.$$

In this way, the power P_a the compressor $a = (u, v) \in A_{\mathrm{CM}}$ consumes to increase the inlet pressure p_u to a higher outlet pressure p_v considering a certain mass flow q_a is modeled via

$$P_a = \frac{H_{\mathrm{ad},a}}{\eta_{\mathrm{ad},a}} q_a. \tag{21}$$

Additionally, to complete the modeling of the different states of a compressor machine, we add the constraints

$$s_a q_a^- \leq q_a \leq s_a q_a^+, \tag{22a}$$

$$s_a P_a^- \leq P_a \leq s_a P_a^+, \tag{22b}$$

$$\Delta_a^- s_a + (p_v^- - p_u^+)(1 - s_a) \leq p_v - p_u \leq \Delta_a^+ s_a + (p_v^+ - p_u^-)(1 - s_a), \tag{22c}$$

$$r_a^- p_u - (1 - s_a)(r_a^- p_u^+ + p_v^-) \leq p_v \leq r_a^+ p_u - (1 - s_a)(r_a^+ p_u^- + p_v^+), \tag{22d}$$

where P_a^- and P_a^+ are the bounds for the power consumption, Δ_a^- and Δ_a^+ are the bounds for the pressure increase and the compression ratio is bounded by $r_a^- \leq p_v/p_u \leq r_a^+$. We remark, that whenever a compressor is open, power consumption occurs while increasing the pressure of gas flowing in arc direction, i.e., $p_v \geq p_u$ and $q_a \geq 0$, and therefore, flow against the arc direction is impossible. Moreover, a closed compressor $a = (u, v) \in A_{\mathrm{CM}}$ results in decoupled pressures at nodes u and v.

For the characteristic diagrams, we use the convex outer-approximation approach described in [19] in case of a turbo compressor. In case of a piston compressors $a = (u, v) \in A_{\mathrm{PC}}$, a relaxation of the operating range is obtained by the constraints (22c)–(22d), together with

$$\frac{Q_a^-}{R_s z_a T} p_u \leq q_a \leq \frac{Q_a^+}{R_s z_a T} p_u, \tag{23}$$

where Q_a^- and Q_a^+ are given lower and upper bounds of the volumetric flow rate. Furthermore, the shaft torque M_a and $H_{\mathrm{ad},a}$ are connected via

$$M_a = \frac{V_o \rho_0}{2\pi \eta_{\mathrm{ad},a}} H_{\mathrm{ad},a},$$

where V_o is the operating volume of the piston compressor.

Similar to control valves, (usually multiple) compressor machines are located in a so-called compressor station allowing for backward flow through a bypass valve.

4.2. **Summary of the MINLP Problem.** Considering the different elements of a gas network described in the previous subsections and recalling the minimization of the compressor's power consumption with energy cost coefficients c_a, we get the following MINLP problem:

$$\min \sum_{a \in A_{\mathrm{CM}}} c_a P_a \tag{24a}$$

$$\text{s.t. pressure bounds (9)}, \tag{24b}$$

$$\text{mass flow conservation (10)}, \tag{24c}$$

$$\text{pipes constraints (12) and (13)}, \tag{24d}$$

$$\text{(control) valves constraints (14) and (15)}, \tag{24e}$$

$$\text{shortcuts constraints (16)}, \tag{24f}$$

$$\text{resistors constraints (17)–(19)}, \tag{24g}$$

$$\text{compressor machines constraints (20)–(23)}. \tag{24h}$$

5. Implementation Issues

The implementation details discussed in this section can be thought of as either preprocessing techniques or parameter tuning, based on empirical values, for Algorithm 1, concerning MINLP problems in the fashion of (24).

Since Algorithm 1 iteratively solves MIP relaxations of the MINLP (24), the runtime of the algorithm mainly depends on the sizes of the MIP relaxations. Based on (8) from Section 3.3, the complexity of a relaxation of a nonlinear function drastically increases with the function's dimension. Hence, we reformulate the MINLP (24) by constructing so-called expression graphs in order to reduce the dimension of the nonlinear functions; see [9] and [1] for details. Moreover, the prepocessing methods described in [10] both tightening the bounds of the flow and pressure variables and reducing the amount of nonlinear functions taken into account for relaxation, are performed.

Unlike the formal description of Algorithm 1, in which in every refinement step, every nonlinear function $f \in \mathcal{F}'$ with an approximation error larger than ϵ_f is refined, we only refine a certain amount of all nonlinear functions, which are worst regarding to a specific score. We build this score upon the set

$$M^i := \left\{ \frac{\dim(f)^2 \left| f\left(x_f^i\right) - y_f^i \right|}{\epsilon_f} : f \in \mathcal{F}' \text{ and } \left| f\left(x_f^i\right) - y_f^i \right| > \epsilon_f \right\}, \tag{25}$$

in every iteration of Algorithm 1. In order to obtain the elements of M^i with corresponding nonlinear functions that have to be refined, we pursue marking startegies applied in adaptive finite elements methods; see [35] and [6]. With η^i as the maximum score contained in M^i, we refine any $f \in \mathcal{F}'$, which has a score larger than $\theta\eta^i$. Typically, $\theta = 0.5$ is chosen. Because, however, the runtime of an MIP grows exponentially with its size, we set $\theta = 0.75$ in our case. In this way, we can speed up the algorithm by solving many smaller-sized MIP relaxations instead of few bigger-sized ones. Since $\left| f\left(x_f^i\right) - y_f^i \right| \leq \epsilon_f$ holds (according to Section 3) for any $f \in \mathcal{F}'$ after a finite number of refinement steps, it follows, that this approach still is convergent. Note that the square of the dimension of f in (25) corresponds to (8).

Beyond that, whenever a feasible solution of an MIP relaxation is found, we fix all discrete variables of the underlying MINLP problem according to the MIP solution obtaining an NLP problem, which we solve to local optimality. With this relatively cheap primal heurisitic we are often able to find feasible solutions for the MINLP problem quite rapidly.

Finally, since we are only interested in a single globally optimal MIP solution satisfying some given error bounds ϵ_f for all $f \in \mathcal{F}'$, it is not necessary to solve every MIP relaxation to global optimality. Instead, it sufficies to solve only every kth MIP relaxation (and the first and last one, respectively) to global optimality and to use bigger relative MIP gaps otherwise. We chose $k = 50$ in our case. The relative MIP gaps depend on whether feasible solutions for the MINLP problems are found or not. With u as upper bound corresponding to the incumbent solution found by the local NLP solver and l as lower bound obtained by MIP relaxations as in Algorithm 1, the relative MIP gap is set to $(u - l)/(2u)$, if an upper bound is available. Otherwise, a relative MIP gap of 0.10 is chosen.

6. Computational results

In this section we present computational results for the GasLib-582 gas network consisting of 582 nodes, 278 pipes, 5 compressor stations, 23 control valves, 8 resistors, 26 valves and 269 shortcuts; see [17]. The following computations are solely based on a subset of all scaled (95 % of flow amount) nominations provided by GasLib-582 consisting of more than 4000 instances. We show both a detailed exemplary computation and a comparison with the state-of-the-art MINLP solvers Baron (version 17.1.2), see [33], and SCIP (version 3.2) [8].

All instances are solved using the C++ software framework LaMaTTO++, see [20], on a cluster using 12 cores of a machine with two Xeon 5650 "Westmere" chips (12 cores + SMT) running at 2.66 GHz with 12 MB Shared Cache per chip and 24 GB of RAM. Furthermore, we utilize Gurobi (version 6.0.4) as MIP solver; see [14]. As stated above, Baron and SCIP are used as state-of-the-art MINLP solvers, and CONOPT3, which in our case performs better than CONOPT4, as local NLP solver, all within GAMS (version 24.8.3); see [3].

Concerning the error bounds in Algorithm 1, we choose 20 bar for the pressure loss equations and 20 MW for the equations describing the power consumption as initial approximation errors. Due to the complexity of the underlying MINLP problems, final approximation errors for the exemplary computation are set to 2.0 bar and 0.2 MW, respectively. For the rest, we choose 1.0 bar and 0.1 MW as final approximation errors for practical reasons, because even in this case all computations run into time limit.

6.1. **Exemplary Computation.**

We start with an exemplary instance showing the practical performance of Algorithm 1 and that even with coarse final approximation errors, good solutions can be computed.

The result presented in Table 1 is obtained by solving a randomly chosen nomination of GasLib-582 (with ID “nomination_warm_95_2051”), which could be solved within a time limit of 10 hours. The first column indicates the iteration number of Algorithm 1. The next three columns contain the

size of the corresponding MIP relaxation, split into the number of continuous and binary variables and the number of constraints. Subsequently, the best lower bound l and upper bound u are given, followed by the runtime of the actual MIP relaxation and the NLPs that are solved to local optimality. The next column contains the relative gap computed by $(u - l)/u$. In the last two columns, the number of violated constraints and the number of constraints which are chosen for refinement are indicated. Finally, the last row gives an overview of the total runtime and number of constraints chosen for refinement.

TABLE 1. Exemplary computation of the GasLib-582 instance with ID "nomination_warm_95_2051" by Algorithm 1 combined with CONOPT3 as local NLP solver.

iter	cont	bin	cons	lower	upper	t_{MIP}	t_{NLP}	gap	viol	ref
0	2927	593	5714	114.40	416.02	1.85	3.77	0.725	77	1
1	2928	594	5716	114.40	316.80	1.12	2.62	0.639	74	4
2	2932	598	5724	114.40	316.80	1.49	2.04	0.639	74	1
3	2933	599	5726	114.40	316.80	1.37	1.53	0.639	74	1
4	2934	600	5728	114.40	316.80	1.37	1.59	0.639	78	2
5	2937	602	5733	114.40	316.80	1.43	3.06	0.639	75	1
25	3002	653	5849	135.59	316.80	5.88	8.57	0.572	79	5
49	3097	722	6013	218.59	316.80	1.91	5.60	0.310	77	2
50	3101	725	6020	243.67	316.80	15.64	7.84	0.231	72	1
51	3103	726	6023	243.67	316.80	6.44	3.17	0.231	73	1
75	3224	826	6244	265.82	316.80	10.93	24.34	0.161	65	2
99	3313	896	6403	294.40	316.80	3.08	10.68	0.071	61	2
100	3315	898	6407	301.61	316.80	284.05	20.19	0.048	54	1
101	3317	899	6410	301.61	316.80	14.28	16.92	0.048	50	2
125	3390	966	6550	301.61	316.80	3.97	12.00	0.048	40	4
149	3477	1051	6722	301.61	316.80	13.34	5.44	0.048	14	2
150	3479	1053	6726	305.89	316.80	233.16	1.55	0.034	14	1
151	3480	1054	6728	305.89	316.80	23.72	11.07	0.034	19	7
175	3549	1115	6858	305.89	316.80	5.77	1.96	0.034	6	1
190	3587	1149	6930	305.89	316.80	4.48	1.26	0.034	4	1
191	3588	1150	6932	305.89	316.80	21.30	1.52	0.034	5	1
192	3589	1151	6934	305.89	316.80	24.28	2.53	0.034	1	1
193	3591	1153	6938	305.89	316.80	16.79	2.10	0.034	4	1
194	3592	1154	6940	305.89	316.80	771.75	3.89	0.034	4	1
195	3592	1154	6940	306.88	316.80	126.78	4.93	0.031	4	1
total						4274.53	1156.66			537

After about 1.5 hours, Algorithm 1 is able to find an optimal solution for the MINLP problem such that no constraint is violated by more than 2.0 bar and 0.2 MW, respectively. Combined with CONOPT3, even a feasible solution for the MINLP problem could be found, which is optimal up to a relative gap of almost 0.03. Note that the final MIP relaxation consists of 3592 continuous and 1154 binary variables, and 6940 constraints only, whereas an MIP relaxation constructed by piecewise linear approximations satisfying the final approximation errors everywhere, consists of 39 193 continuous, 21 735 binary variables, and 61 709 constraints. The first feasible solution

for this MIP is found after a runtime of almost 8 hours, whereas after a total runtime of 10 hours lower and upper bounds are 131.35 and 594.50, respectively, resulting in a relative MIP gap of more than 0.77. We can see that although final approximation errors are relatively high, adaptivity in Algorithm 1 is crucial for a reasonable overall runtime. Since the runtime to solve an MIP problem exponentially increases with the size of the MIP, adaptivity becomes even more important if final approximation errors are tighter; see [9] for a more detailed discussion.

Since the upper bound in Table 1 remains unchanged after the first iteration, we conclude that with our approach even very coarse initial approximations can lead to solutions that are feasible for the MINLP and optimal within a relative gap of almost 0.03, in a couple of seconds only.

We finish the exemplary presentation of our approach with Figure 4, showing an iteration log of a nonlinear function $f\colon \mathbb{R}^2 \to \mathbb{R}, (x, y) \mapsto xy$ with domain $D_f = [1.028, 1.206] \times [30.350, 1139.280]$ and Lipschitz-constant $L_f = \operatorname{diam}(D_f)$. This function corresponds to the power consumption of a compressor; see (21). The final approximation error for f is set to 1.265. The maximal approximation error on the simplex containing the incumbent solution of the corresponding MIP relaxation is plotted by red lines, whereas the approximation error of the solution itself is given by blue lines. Moreover, whenever the function is marked for refinement a black dash is drawn.

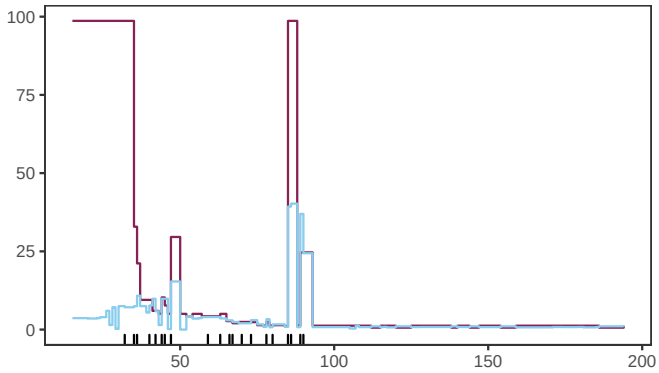

FIGURE 4. Iteration log for a nonlinear function $f = xy$ showing the maximal approximation error on the simplex containing the incumbent solution of the corresponding MIP relaxation (red lines), the approximation error of the solution (blue lines) and the iterations in which the approximation is refined (black dash).

As expected, the error of the MIP solution tends to a value smaller than 1.265 as more refinement steps are performed on f. The staircase-shaped descending of the error is characteristic for Algorithm 1. Often, solutions of consecutive MIP relaxations have a local affinity, i.e., the projections of the solutions on the domain D_f are close to each other. Hence, depending on the specific triangulation of the piecewise linear approximation, in some cases we need more than one refinement per nonlinear function in order to scale the error down to the next level. We point out, that exploiting this phenomenon,

e.g., by refining not only the simplex containing the MIP solution, but also adjacent simplices, may lead to further improvements.

With 20 refinements from iteration 0 to 90 and 2 refinements performed on f for the initial approximation, in total, 22 refinements and 195 iterations are enough to obtain an optimal MIP solution satisfying the final error bounds both for f and any other nonlinear function occuring in the MINLP. Regarding the required number of refinement steps, compared to the worst case estimation of $2^{\tilde{N}}$ as in (6) where $\tilde{N} = 3\left\lceil \ln\left(\frac{2\,\mathrm{diam}(D_f)\,\mathrm{diam}(D_f)}{1.265}\right) / \ln\left(\frac{2}{\sqrt{3}}\right)\right\rceil = 303$, a far less amount is needed in this case.

Finally, we remark that although the total runtime of 1.5 hours of Algorithm 1 and CONOPT3 appears to be long in order to obtain an MINLP solution, which is optimal within a relative gap of 0.03, it is quite short compared to other MINLP solvers.

6.2. **Comparison with State-of-the-Art MINLP Solvers.** The advantage of Algorithm 1 combined with the local NLP solver CONOPT3 over the state-of-the-art global MINLP solvers Baron and SCIP is demonstrated in Figure 5 by comparing relative gaps computed by $(u - l)/u$, with the aid of so-called performance profiles; see [5]. With $g_{p,s}$ as the best gap obtained by solver s after a certain time limit and the performance ratio $r_{p,s} = g_{p,s} / \min_s g_{p,s}$, the performance profile $\rho_s(\tau)$ is the probability for solver s that the ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio.

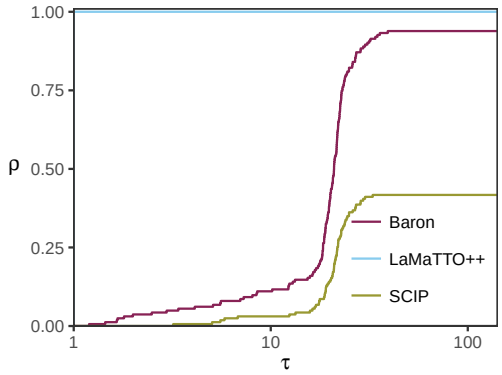

FIGURE 5. Performance profiles for Algorithm 1 combined with CONOPT3 (LaMaTTO++), Baron, and SCIP comparing relative gaps obtained after a time limit of 4 hours.

In order to calculate those profiles, 200 out of roughly 4000 nominations provided by GasLib-582 are randomly chosen; see Table 3 in Appendix A for their IDs. Considering only those nominations for which at least one solver was able to compute a feasible solution within the total time limit of 4 hours, 163 nominations remain for the performance profiles. Moreover, due to the complexity of the problems, each of the 163 nominations ran into timeout.

As we can see in Figure 5, our approach is clearly preferable to Baron and SCIP applied to gas network optimization problems described in GasLib-582. In all cases Algorithm 1 combined with CONOPT3 computes the smallest

gap, while Baron and SCIP in no case were capable of computing the smallest gap. Additionally, in most cases the relative gaps obtained by our approach are smaller than 0.10 and differ from the relative gaps computed by Baron and SCIP by a magnitude of almost 10, which can be deduced from Figure 5.

In order to gain an in-depth look at Figure 5 both lower bounds l and upper bounds u are compared in Figure 6. The upper bounds computed by CONOPT3 after fixing all discrete variables corresponding to an MIP solution obtained by Algorithm 1 are slightly tighter than the ones attained by Baron and clearly tighter than the ones computed by SCIP. The main benefit of our approach, however, derives from the fact that MIP relaxations obtained by Algorithm 1 yield notably tighter lower bounds than the lower bounds computed by Baron and SCIP and that the MIP relaxations can be solved both reliably and fast utilizing mature MIP technology. Note that Baron mainly struggles to deliver reasonable lower bounds, whereas in this regard, SCIP seems to be more balanced.

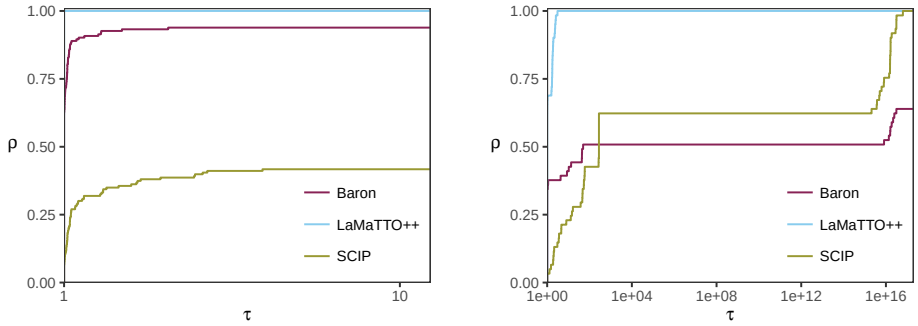

FIGURE 6. Performance profiles for Algorithm 1 combined with CONOPT3 (LaMaTTO++), Baron, and SCIP comparing upper bounds u (left) and lower bounds l (right) obtained after a time limit of 4 hours.

We summarize the comparison presenting the different solution statuses given by every solver and their frequency in Table 2. The first column indicates nominations with both an upper bound u and a lower bound l greater than zero, whereas in the second and third column only those nominations are taken into account, for which only an upper bound u, and a lower bound l, respectively, is available. The next column contains the number of nominations that are detected as infeasible, followed by a column in which nominations without any solution status are considered. The last column gives the number of nominations that are declared as infeasible, and for which at least one of the other solvers is able to compute a feasible solution for the corresponding nomination. Algorithm 1 combined with CONOPT3 is able to find feasible solutions for all 163 nominations with additional non-trivial lower bounds. As already pointed out, Baron struggles with computing lower bounds, since Baron is able to find feasible solutions for 153 nominations, but only in 49 cases a non-trivial lower bound as well. In this respect, SCIP is clearly inferior. Moreover, in 5 cases SCIP detects infeasiblity of the MINLP, which presumably is false, whereas in this matter, our approach and Baron are almost coherent. For further results concerning the detection of infeasibility we refer to [18] and [15].

TABLE 2. Frequency of different solution statuses given by Baron, Algorithm 1 combined with CONOPT3 (LaMaTTO++), and SCIP for a set of 200 randomly chosen nominations provided by GasLib-582.

	l / u	u	l	infeasible	none	error
Baron	49	104	0	35	12	0
LaMaTTO++	163	0	0	34	3	0
SCIP	13	55	0	19	108	5

We believe that the advantage of Algorithm 1 combined with CONOPT3 relies on the major discrete nature of the problems, since Algorithm 1 is both able to compute discrete decisions which are feasible for the MINLP very fast and tight lower bounds. In fact, for 161 out of 163 nominations, Algorithm 1 is able to compute discrete decisions proven to be feasible for the corresponding MINLP by the local NLP solver CONOPT3 after a total time limit of just 10 minutes. Furthermore, as we can see in Table 1, Algorithm 1 delivers reasonable lower bounds after only a few iterations, even with coarse initial approximations.

Finally, we point out that although a primal algorithm, which in our case is constituted by CONOPT3, is necessary to compute valid upper bounds, it is not necessary for Algorithm 1 to terminate. In fact, with tight final error bounds we can skip the primal algorithm entirely, because in this case Algorithm 1 eventually yields an optimal MIP solution, which can be considered as an optimal solution for the corresponding MINLP.

7. CONCLUSION

In this paper we presented a mixed-integer linear programming-based method for solving mixed-integer nonlinear programs to global optimality and developed a convergence theory. From a theoretical point of view, we are able to solve arbitrary MINLPs with continuous objective, continuous nonlinear constraints, and bounded variables, whereas in practice the dimension of the nonlinearities is a crucial factor. The capability of our approach is demonstrated in the context of gas network optimization containing a notably discrete part. Regarding other MINLP problems with considerable discrete aspects, we are confident that similar results would be achieved. In this matter, more empirical studies will give a deeper insight into MINLP classes our method is suitable for. The next natural step is to build an MINLP solver based on our approach and solve various MINLP problems, e.g., instances provided by the MINLPLib [2].

Moreover, there is a lot of potential for improvement. The most promising idea, however, is to refine the nonlinearities such that warm starting procedures implemented in modern MIP solvers can be exploited.

Appendix A. IDs of GasLib-582 nominations

Table 3. IDs of all 200 GasLib-582 nominations that are chosen randomly in order to calculate the corresponding performance profiles in Section 6.

nomination_cold_95_									
101	1048	1089	1184	123	1321	1326	1642	1660	1729
1768	1911	2008	21	2120	2121	2158	2164	2187	229
2381	2393	2503	268	2720	3007	3111	33	3375	3539
3769	3970	4054	4127	4194	4209	468	676	82	954
957	971								
nomination_cool_95_									
1012	1032	1119	1130	1143	1153	1181	1291	1423	145
1576	1592	1665	1957	201	2040	2050	2168	2178	2196
2275	2284	2321	2364	2398	2453	25	2674	2682	2721
2770	2774	2794	2854	295	3103	312	3148	316	3201
3283	344	3509	3518	3567	3578	3642	3656	3658	3683
3756	378	379	3791	3858	3885	4072	4223	604	663
81	830	979							
nomination_freezing_95_									
1001	1035	1193	1243	1268	1328	1371	1434	1525	1558
1591	160	1889	1962	2308	2385	2408	2596	2685	2877
2964	3045	3068	3205	3260	3263	3291	3362	3417	3483
3550	3597	3635	3886	3911	3994	400	4008	4091	4211
426	448	454	466	52	56	639	645	741	818
836	881								
nomination_mild_95_									
1433	1455	1479	1661	1750	1922	1942	2052	2081	2450
2502	2568	2581	2612	2668	2817	3239	3309	3524	3541
3781	3795	4078	4178	4184	679	746	865		
nomination_warm_95_									
1095	1316	1385	1691	2051	2247	2314	2718	2824	3020
3579	360	450	637	786					

Acknowledgements

The authors thank the Deutsche Forschungsgemeinschaft for their support within Project B07 of the Sonderforschungsbereich/Transregio 154 "Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks". We would also like to show our gratitude to Martin Schmidt for many constructive comments on various issues. Furthermore, we are very thankful for his help on preparing the GasLib-582 data. Finally, we thank Alexander Martin and Mathias Sirvent for many fruitful discussion on the topics of this paper.

References

[1] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. "Mixed-integer nonlinear optimization". In: *Acta Numerica* 22 (2013), pp. 1–131. DOI: 10.1017/S0962492913000032.

[2] M. R. Bussieck, A. S. Drud, and A. Meeraus. "MINLPLib – A Collection of Test Models for Mixed-Integer Nonlinear Programming". In: *INFORMS Journal on Computing* 15.1 (2003), pp. 114–119.

[3] G. D. Corporation. *General Algebraic Modeling System (GAMS) Release 24.8.3*. Washington, DC, USA. 2017. URL: http://www.gams.com/.

[4] C. M. Correa-Posada and P. Sánchez-Martín. "Gas Network Optimization: A comparison of Piecewise Linear Models". Oct. 2014. URL: http://www.optimization-online.org/DB_HTML/2014/10/4580.html.

[5] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Mathematical Programming* 91.2 (2002), pp. 201–213. DOI: 10.1007/s101070100263.

[6] W. Dörfler. "A Convergent Adaptive Algorithm for Poisson's Equation". In: *SIAM Journal on Numerical Analysis* 33.3 (1996), pp. 1106–1124. DOI: 10.1137/0733054.

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

[8] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger, M. Winkler, J. T. Witt, and J. Witzig. *The SCIP Optimization Suite 3.2*. eng. Tech. rep. 15-60. Takustr.7, 14195 Berlin: ZIB, 2016.

[9] B. Geißler. "Towards Globally Optimal Solutions of MINLPs by Discretization Techniques with Applications in Gas Network Optimization". PhD thesis. FAU Erlangen-Nürnberg, 2011.

[10] B. Geißler, A. Martin, A. Morsi, and L. Schewe. "The MILP-relaxation approach". In: *Evaluating Gas Network Capacities*. Ed. by T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. SIAM-MOS series on Optimization. SIAM, 2015. Chap. 6, pp. 103–122. DOI: 10.1137/1.9781611973693.ch6.

[11] B. Geißler, A. Martin, A. Morsi, and L. Schewe. "Using Piecewise Linear Functions for Solving MINLPs". In: *Mixed Integer Nonlinear Programming*. Ed. by J. Lee and S. Leyffer. Vol. 154. The IMA Volumes in Mathematics and its Applications. Springer New York, 2012, pp. 287–314. DOI: 10.1007/978-1-4614-1927-3_10.

[12] 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. Berlin,

Heidelberg: Springer, 2013, pp. 321–353. DOI: 10.1007/978-3-642-38189-8_14.

[13] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. *Solving Highly Detailed Gas Transport MINLPs: Block Separability and Penalty Alternating Direction Methods*. 2016. URL: http://www.optimization-online.org/DB_HTML/2016/06/5523.html. Submitted.

[14] Z. Gu, E. Rothberg, and R. Bixby. *Gurobi Optimizer Reference Manual, Version 6.0.4*. Houston, Texas, USA: Gurobi Optimization Inc., 2015.

[15] B. Hiller, J. Humpola, T. Lehmann, R. Lenz, A. Morsi, M. E. Pfetsch, L. Schewe, M. Schmidt, R. Schwarz, J. Schweiger, C. Stangl, and B. M. Willert. "Computational results for validation of nominations". In: *Evaluating Gas Network Capacities*. Ed. by T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. SIAM-MOS series on Optimization. SIAM, 2015. Chap. 12, pp. 233–270. DOI: 10.1137/1.9781611973693.ch12.

[16] J. Huchette and J. P. Vielma. "Small independent branching formulations for unions of V-polyhedra". July 2016. URL: http://www.optimization-online.org/DB_HTML/2016/05/5454.html.

[17] J. Humpola, I. Joormann, D. Oucherif, M. E. Pfetsch, L. Schewe, M. Schmidt, and R. Schwarz. *GasLib – A Library of Gas Network Instances*. Nov. 2015. URL: http://www.optimization-online.org/DB_HTML/2015/11/5216.html.

[18] I. Joormann, M. Schmidt, M. C. Steinbach, and B. M. Willert. "What does "feasible" mean?" In: *Evaluating Gas Network Capacities*. Ed. by T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. SIAM-MOS series on Optimization. SIAM, 2015. Chap. 11, pp. 211–232. DOI: 10.1137/1.9781611973693.ch11.

[19] T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, eds. *Evaluating Gas Network Capacities*. SIAM-MOS series on Optimization. SIAM, 2015. xvii + 364. DOI: 10.1137/1.9781611973693.

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

[21] J. Lee and D. Wilson. "Polyhedral methods for piecewise-linear functions. I. The lambda method". In: *Discrete Appl. Math.* 108.3 (2001), pp. 269–285. DOI: 10.1016/S0166-218X(00)00216-X.

[22] H. M. Markowitz and A. S. Manne. "On the Solution of Discrete Programming Problems". In: *Econometrica* 25.1 (1957), pp. 84–110.

[23] A. Martin, M. Möller, and S. Moritz. "Mixed Integer Models for the Stationary Case of Gas Network Optimization". In: *Mathematical Programming, Series B* 105 (2006), pp. 563–582. DOI: 10.1007/s10107-005-0665-5.

[24] R. Misener and C. A. Floudas. "Piecewise-linear approximations of multidimensional functions". In: *J. Optim. Theory Appl.* 145.1 (2010), pp. 120–147. DOI: 10.1007/s10957-009-9626-0.

[25] A. Morsi. "Solving MINLPs on Loosely-Coupled Networks with Applications in Water and Gas Network Optimization". PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2013.

[26] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998. DOI: 10.1017/CBO9780511804120.

[27] M. Padberg and M. P. Rijal. *Location, Scheduling, Design and Integer Programming*. Kluwer Academic Publishers, Boston, 1996. DOI: 10.1007/978-1-4613-1379-3.

[28] M. Padberg. "Approximating separable nonlinear functions via mixed zero-one programs". In: *Operations Research Letters* 27.1 (2000), pp. 1–5. DOI: 10.1016/S0167-6377(00)00028-6.

[29] S. Rebennack and J. Kallrath. "Continuous piecewise linear delta-approximations for bivariate and multivariate functions". In: *J. Optim. Theory Appl.* 167.1 (2015), pp. 102–117. DOI: 10.1007/s10957-014-0688-2.

[30] S. Rebennack and J. Kallrath. "Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems". In: *J. Optim. Theory Appl.* 167.2 (2015), pp. 617–643. DOI: 10.1007/s10957-014-0687-3.

[31] R. Rovatti, C. D'Ambrosio, A. Lodi, and S. Martello. "Optimistic MILP modeling of non-linear optimization problems". In: *European J. Oper. Res.* 239.1 (2014), pp. 32–45. DOI: 10.1016/j.ejor.2014.03.020.

[32] W. D. Smith. "A Lower Bound for the Simplexity of the n-cube via Hyperbolic Volumes". In: *European Journal of Combinatorics* 21.1 (2000), pp. 131–137. DOI: 10.1006/eujc.1999.0327.

[33] M. Tawarmalani and N. V. Sahinidis. "A polyhedral branch-and-cut approach to global optimization". In: *Mathematical Programming* 103 (2 2005), pp. 225–249. DOI: 10.1007/s10107-005-0581-8.

[34] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, eds. *Information-Based Complexity*. Boston, San Diego, and New York: Academic Press, 1988.

[35] R. Verfürth. *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. John Wiley & Sons Inc and B. G. Teubner Publishers, 1996.

[36] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. "Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions". In: *Operations Research* 58.2 (2010), pp. 303–315. URL: http://www.jstor.org/stable/40605918.

[37] J. P. Vielma and G. L. Nemhauser. "Modeling disjunctive constraints with a logarithmic number of binary variables and constraints". In: *Mathematical Programming* 128.1-2 (2011), 49–72. DOI: 10.1007/s10107-009-0295-4.

[38] D. Wilson. "Polyhedral methods for piecewise-linear functions". Ph.D. thesis in Discrete Mathematics. University of Kentucky, 1998.