

Schmidt, Martin; Sirvent, Mathias; Wollner, Winnifried

Article — Published Version

The cost of not knowing enough: mixed-integer optimization with implicit Lipschitz nonlinearities

Optimization Letters

Provided in Cooperation with:

Springer Nature

Suggested Citation: Schmidt, Martin; Sirvent, Mathias; Wollner, Winnifried (2021) : The cost of not knowing enough: mixed-integer optimization with implicit Lipschitz nonlinearities, Optimization Letters, ISSN 1862-4480, Springer, Berlin, Heidelberg, Vol. 16, Iss. 5, pp. 1355-1372,
<https://doi.org/10.1007/s11590-021-01827-9>

This Version is available at:

<https://hdl.handle.net/10419/287425>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



The cost of not knowing enough: mixed-integer optimization with implicit Lipschitz nonlinearities

Martin Schmidt¹ · Mathias Sirvent² · Winnifried Wollner³

Received: 18 May 2020 / Accepted: 3 November 2021 / Published online: 25 November 2021
© The Author(s) 2021

Abstract

It is folklore knowledge that nonconvex mixed-integer nonlinear optimization problems can be notoriously hard to solve in practice. In this paper we go one step further and drop analytical properties that are usually taken for granted in mixed-integer nonlinear optimization. First, we only assume Lipschitz continuity of the nonlinear functions and additionally consider multivariate implicit constraint functions that cannot be solved for any parameter analytically. For this class of mixed-integer problems we propose a novel algorithm based on an approximation of the feasible set in the domain of the nonlinear function—in contrast to an approximation of the graph of the function considered in prior work. This method is shown to compute approximate global optimal solutions in finite time and we also provide a worst-case iteration bound. In some first numerical experiments we show that the “cost of not knowing enough” is rather high by comparing our approach with the open-source global solver SCIP. This reveals that a lot of work is still to be done for this highly challenging class of problems and we thus finally propose some possible directions of future research.

Keywords Mixed-integer nonlinear optimization · Global optimization · Lipschitz optimization · Gas networks

✉ Martin Schmidt
martin.schmidt@uni-trier.de

Mathias Sirvent
mathias.sirvent@fau.de

Winnifried Wollner
wollner@mathematik.tu-darmstadt.de

- ¹ Department of Mathematics, Trier University, Universitätsring 15, 54296 Trier, Germany
- ² Discrete Optimization, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Cauerstr. 11, 91058 Erlangen, Germany
- ³ Fachbereich Mathematik, Technische Universität Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

1 Introduction

Mixed-integer nonlinear optimization problems (MINLPs) are one of the most important classes of models in mathematical optimization. This is due to its capability of modeling decisions via incorporating discrete aspects as well as the possibility of modeling nonlinear phenomena. However, the combination of these two aspects also makes these problems very hard to solve [1,2]. For a general overview about MINLP we refer to [3].

Both from a theoretic and algorithmic point of view it is important to classify MINLPs further. Probably the most important distinction is to be made between convex and nonconvex MINLPs. Since gradients yield valid cuts in the convex case, outer approximations of the feasible sets of the convex nonlinearities can be derived and exploited in algorithms; see, e.g., [4–8] and the references therein. In the nonconvex case this is not possible. Here, one typically needs to derive convex underestimators and concave overestimators that yield (piecewise) convex relaxations of the nonconvex nonlinearities [9–11]. These approaches and also most other algorithms for computing global optima of such nonconvex MINLPs are based on spatial branching. The method we propose is different in the sense that we generate an iteratively tightened but nonconvex outer approximation of the feasible set, which is algorithmically realized via constructing a tightened mixed-integer linear approximation of the MINLP. Moreover, spatial branching based on convex underestimators and concave overestimators usually exploit known analytical properties of the nonconvex nonlinear functions, which is obviously not possible if these properties, like, e.g., differentiability, are not known or even knowledge about the explicit representation is missing. In this case, one typically tries to resort to Lipschitz assumptions about the nonlinearities, which leads to the field of global Lipschitz optimization; see, e.g., [12–18] to name only a few. For a more detailed overview about this field see the textbook [19] and the references therein.

In this paper, we focus on a specific setting that can be observed in many applications (see below for some problem-specific references), namely that the Lipschitzian MINLP under consideration can be decomposed into a mixed-integer linear (MILP) part and a nonlinear part. Our working hypothesis in this and also the preceding works [20–23] is that the MILP part can be solved comparably fast and reliable whereas the nonlinearity really hampers the solution process—at least in combination with the MILP part of the problem. See also [24–27], where this working hypothesis is followed as well. At this point, we remark that the cited literature usually tries to get rid of the nonlinear functions by replacing or, so to say, re-modeling them using MILP-representable approximations. In this paper, we consider the case in which this re-modeling is not possible without adding additional unknowns and constraints to the problem.

One specific field of application that fits into the above discussion and that has been studied very successfully in recent years is mixed-integer nonlinear optimization of gas transport networks [22,28–31]; see the recent book [32] and the survey [33] for more references. This application will also be studied later in our numerical case study. MINLPs from gas transport optimization are defined on graphs that represent the transport network. Mass balances and simple physical as well as technical bounds yield linear constraints in this context and controllable elements are typically modeled

by mixed-integer (non)linear sets of variables and constraints. Here, the complicating nonlinearity is the system of differential equations modeling the gas flow through pipes. In its full beauty, these equations yield implicit and highly nonlinear constraints that can be evaluated only by simulation and that do not possess additional and desired analytical properties like convexity; see [34–39]. In almost all contributions of the literature cited above the authors make, depending on their special focus, tailored assumptions that allow to develop very effective solution approaches. In this paper, we follow the way paved by the paper [21], where general methods have been developed (and tested on gas transport problems) that only require Lipschitzian, 1-dimensional, and explicitly given nonlinearities. In this paper, we show that the approach can be extended to the case where the nonlinearity is only given implicitly. To this end, we relax the assumptions used in [21] as follows:

1. We consider constraints that are only implicitly given, i.e., constraints of type $F(x) = 0$ with $x \in \mathbb{R}^n$ such that a representation $F(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = x_i$ cannot be achieved analytically for any $i \in \{1, \dots, n\}$.
2. We consider multi-dimensional constraint functions $F(x) = 0$ with $x \in \mathbb{R}^n$ and $n \geq 2$.

Regarding (1) it is of course possible to re-model the implicit constraint as $F(x) = y$ and by adding additional linear constraints $y = 0$. We, however, refrain from such a re-modeling in this paper and consider the case were such a re-modeling, which also enlarges the problem, is not appropriate. It will turn out that both aspects drastically accentuate the computational hardness of the problem.

Our contribution is the following. We formalize the problem class in Sect. 2 and the corresponding assumptions sketched above (Sect. 3). Based on this, we state an algorithm and prove its correctness, i.e., that it computes approximate global optimal solutions (or proves the infeasibility of the problem) in finite time. Moreover, we prove a worst-case iteration bound for the algorithm. Finally, we present a numerical case study in Sect. 4, where we apply the method to nonconvex problems from gas transport. This case study reveals that the “cost of not knowing enough” is rather high: Only small gas transport networks can be solved in reasonable time under the stated very weak assumptions. Moreover, a comparison with the open-source global solver SCIP shows that the latter is outperforming our approach if additional structure of the problem is at hand that is exploited by SCIP but not by our approach. This is why we consider the addressed problem class as an open computational challenge for which we state possible directions of future research in Sect. 5.

2 Problem definition

We consider problems of the form

$$\min_x h^\top x \tag{1a}$$

$$\text{s.t. } Ax \geq b, \quad x \in [\underline{x}, \bar{x}], \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \tag{1b}$$

$$F_i(x_{c(i)}) = 0, \quad i \in [p], \tag{1c}$$

where $A \in \mathbb{R}^{\ell \times (n+m)}$, $b \in \mathbb{R}^{\ell}$, $h \in \mathbb{R}^{n+m}$, and $[p] := \{1, \dots, p\}$. The p constraints $F_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$, $i \in [p]$, comprise all nonlinearities of the problem. Here, and in what follows, we use the splitting $x = (x_c, x_d)$ of the entire variable vector with $x_c \in \mathbb{R}^n$ and $x_d \in \mathbb{Z}^m$. That is, x_c are all continuous and x_d are all discrete variables of the problem and the nonlinear functions F_i , $i \in [p]$, only depend on the continuous variables $x_{c(i)}$, i.e., $x_{c(i)}$ is a sub-vector of the vector x_c of all continuous variables. We remark that all variables are finitely bounded by $-\infty < \underline{x} \leq \bar{x} < \infty$. Moreover, we note that the assumption that the nonlinearities only depend on the continuous variables is only made to simplify the technical discussions later on—it can always be formally satisfied by introducing auxiliary continuous variables.

Instead of optimizing the objective of (1) over the feasible set \mathcal{F} given by (1b, 1c) we replace \mathcal{F} by an approximating sequence $\mathcal{F}_k \approx \mathcal{F}$ and globally optimize the sequence of problems

$$\min \{h^\top x : x \in \mathcal{F}_k\}. \tag{2}$$

The iteration can then be stopped once a solution x^k of (2) is close enough to the original feasible set \mathcal{F} . To this end let

$$\begin{aligned} \min_x \quad & h^\top x \\ \text{s.t.} \quad & Ax \geq b, \quad x \in [\underline{x}, \bar{x}], \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \\ & |F_i(x_{c(i)})| \leq \varepsilon, \quad i \in [p], \end{aligned} \tag{3}$$

be the ε -relaxed version of the original problem (1). Note that we only relax the nonlinearities whereas all other constraints stay as they are. The precise choice of the approximate sets \mathcal{F}_k will be detailed later in Sect. 3.

Definition 2.1 (ε -feasibility) We call a point ε -feasible if it is feasible for Problem (3).

3 Algorithm

The main idea of our algorithm for solving Problem (1) to approximate global optimality is to split the problem into its mixed-integer linear and its nonlinear part. The mixed-integer linear part is solved in the so-called master problem, which additionally contains a successively tightened approximation of the zero sets of the nonlinearities F_i , $i \in [p]$. Let x^k denote the master problem’s solution in iteration k . Then, for every master problem solution, we check the feasibility of the solution w.r.t. the nonlinearities F_i . In case of feasibility, we have found a global optimal solution of the original problem; and in case of infeasibility, we construct a tighter approximation of the zero sets for the next master problem based on the information obtained by evaluating the nonlinearities F_i at the current master solution x^k . For the latter, we need (i) the function values $F_i(x_{c(i)}^k)$, $i \in [p]$, as well as (ii) the global Lipschitz constants L_i of the F_i w.r.t. the norm $\|\cdot\|_\infty$ on \mathbb{R}^{n_i} , i.e., it holds

$$|F_i(x_1) - F_i(x_2)| \leq L_i \|x_1 - x_2\|_\infty \quad \text{for all } x_1, x_2 \in \mathbb{R}^{n_i} \cap [x_{c(i)}, \bar{x}_{c(i)}].$$

Assumption 1 We have an oracle that evaluates $F_i(x_{c(i)})$ for all $i \in [p]$ and all F_i are globally Lipschitz continuous on $x_{c(i)} \in [\underline{x}_{c(i)}, \bar{x}_{c(i)}]$ with known global Lipschitz constant L_i .

Remark 3.1 Let us note that for a continuously differentiable function F_i and arbitrary $x_1, x_2 \in \mathbb{R}^{n_i} \cap [\underline{x}_{c(i)}, \bar{x}_{c(i)}]$, the estimate

$$|F_i(x_1) - F_i(x_2)| = \left| \int_0^1 \nabla F_i(x_2 + s(x_1 - x_2))(x_1 - x_2) ds \right| \leq \max_{x \in \mathbb{R}^{n_i} \cap [\underline{x}_{c(i)}, \bar{x}_{c(i)}]} \|\nabla F_i(x)\|_1 \|x_1 - x_2\|_\infty$$

holds by the fundamental theorem of calculus. Hence one obtains an estimate of the needed Lipschitz constant by setting

$$L_i = \max_{x \in \mathbb{R}^{n_i} \cap [\underline{x}_{c(i)}, \bar{x}_{c(i)}]} \|\nabla F_i(x)\|_1.$$

We now show how to construct the successively tightened approximations of the zero sets in case of Assumption 1. Let Ω_i^k denote this approximation of the zero set of function F_i in iteration k of the algorithm. Initially, we start with the box $\Omega_i^0 := [\underline{x}_{c(i)}, \bar{x}_{c(i)}]$ defined by the original bounds of the problem (or after presolving; see Sect. 4). Assume now that the evaluation of the nonlinearity yields $|F_i(x_{c(i)}^k)| \geq \varepsilon$ for a prescribed tolerance $\varepsilon > 0$. We then exclude the box

$$B_i^k := \left\{ x \in \mathbb{R}^{n_i} : \|x - x_{c(i)}^k\|_\infty < \frac{|F_i(x_{c(i)}^k)|}{L_i} \right\} \tag{4}$$

in the next iteration. If $|F_i(x_{c(i)}^k)| < \varepsilon$ holds, we set $B_i^k = \emptyset$. Putting these boxes together, we obtain the nonconvex outer approximation

$$\Omega_i^k = [\underline{x}_{c(i)}, \bar{x}_{c(i)}] \setminus \bigcup_{j=[k-1]} B_i^j, \quad k = 1, 2, \dots \tag{5}$$

With this at hand, we can now formulate the master problem:

$$\min_x h^\top x \tag{6a}$$

$$\text{s.t. } Ax \geq b, \quad x \in [\underline{x}, \bar{x}], \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \tag{6b}$$

$$x_{c(i)} \in \Omega_i^k, \quad i \in [p]. \tag{6c}$$

The main goal of this section is to prove the correctness of the algorithm, i.e., finite termination of the algorithm at approximate global optimal points. However, before we do this, we first state and prove some properties of the master problem that will be used later and formally state the algorithm.

Proposition 3.2 *Assume that Assumption 1 holds. Then, it holds*

$$\Omega_i^k \supseteq (\ker(F_i) \cap [\underline{x}_{c(i)}, \bar{x}_{c(i)}])$$

for all $i \in [p]$ and all k . Thus, the master problem (6) is a relaxation of (1) for all k .

Proof The proposition follows directly from the construction (4) and (5) and the Lipschitz continuity of F_i . □

The next lemma shows that we can use state-of-the-art MILP software for solving the master problems.

Lemma 3.3 *The nonconvex master problem (6) can be modeled as a mixed-integer linear problem. The number of additional variables and constraints required to formulate (6c) for an $i \in [p]$ is $\mathcal{O}(k |c(i)|)$.*

Proof The constraints $Ax \geq b$, $x \in [\underline{x}, \bar{x}]$, and $x \in \mathbb{R}^n \times \mathbb{Z}^m$ are obviously mixed-integer linear constraints. Thus, it remains to prove that $x_{c(i)} \in \Omega_i^k$, $i \in [p]$, can be formulated with mixed-integer linear constraints as well. We define the index set

$$\mathcal{I}_i^k := \{j \in [k - 1] : |F_i(x_{c(i)}^j)| \geq \varepsilon\}, \quad i \in [p],$$

for the boxes to be excluded. Moreover, we define

$$\underline{x}_{c(i)}^j := x_{c(i)}^j - 1 \frac{|F_i(x_{c(i)}^j)|}{L_i} \quad \text{and} \quad \bar{x}_{c(i)}^j := x_{c(i)}^j + 1 \frac{|F_i(x_{c(i)}^j)|}{L_i} \tag{7}$$

for all $i \in [p]$ and $j \in \mathcal{I}_i^k$; see Fig. 1 for an illustration. In (7) we use the notation $\mathbf{1}$ for the vector of ones in appropriate dimension.

To model the gray area in Fig. 1, we have to exclude the white rectangles B_i^j for all $j \in \mathcal{I}_i^k$. This can be done in the following way:

$$\Omega_i^k = \left\{ x_{c(i)} : z_l^{j,1}, z_l^{j,2} \in \{0, 1\} \right. \quad \text{for all } l \in c(i), j \in \mathcal{I}_i^k, \tag{8a}$$

$$x_l \leq \bar{x}_l^j + z_l^{j,1}(\bar{x}_l - \bar{x}_l^j) \quad \text{for all } l \in c(i), j \in \mathcal{I}_i^k, \tag{8b}$$

$$x_l \geq \underline{x}_l + z_l^{j,1}(\bar{x}_l^j - \underline{x}_l) \quad \text{for all } l \in c(i), j \in \mathcal{I}_i^k, \tag{8c}$$

$$x_l \leq \underline{x}_l^j + (1 - z_l^{j,2})(\bar{x}_l - \underline{x}_l^j) \quad \text{for all } l \in c(i), j \in \mathcal{I}_i^k, \tag{8d}$$

$$x_l \geq \underline{x}_l + (1 - z_l^{j,2})(\underline{x}_l^j - \underline{x}_l) \quad \text{for all } l \in c(i), j \in \mathcal{I}_i^k, \tag{8e}$$

$$\left. \sum_{l \in c(i)} (z_l^{j,1} + z_l^{j,2}) \geq 1 \right\} \quad \text{for all } j \in \mathcal{I}_i^k \tag{8f}$$

We remark that (8b, 8c) check if $x_l < \bar{x}_l^j$. If this holds true, then $z_l^{j,1} = 0$. On the other hand, (8d, 8e) check if $x_l > \underline{x}_l^j$. If this holds true, then $z_l^{j,2} = 0$. As we want to

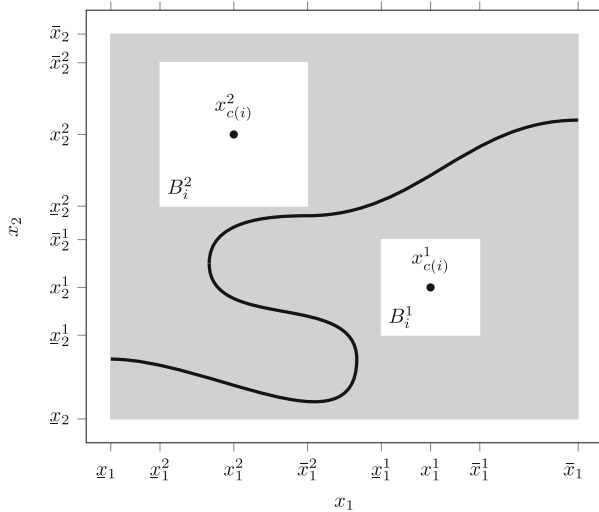


Fig. 1 Set Ω_i^2 with $c(i) = \{1, 2\}$

Algorithm 1 Global Optimization of MINLPs with Implicit Nonlinearities

Require: Problem (1) and $\varepsilon > 0$.

Ensure: Returns an approximate global optimal point for Problem (1) or an indication of infeasibility.

- 1: Set $k \leftarrow 0$ and initialize $\Omega_i^0 \leftarrow [x_{c(i)}, \bar{x}_{c(i)}]$ for all $i \in [p]$.
 - 2: **while** true **do**
 - 3: Solve the master problem (6) to global optimality.
 - 4: **if** (6) is infeasible **then return** "Problem (1) is infeasible".
 - 5: Let x^k denote the optimal solution of (6).
 - 6: Evaluate $F_i(x_{c(i)}^k)$ for all $i \in [p]$.
 - 7: **if** $|F_i(x_{c(i)}^k)| \leq \varepsilon$ for all $i \in [p]$ **then return** x^k .
 - 8: **for** $i \in [p]$ **do**
 - 9: **if** $|F_i(x_{c(i)}^k)| > \varepsilon$ **then** set $\Omega_i^{k+1} \leftarrow \Omega_i^k \setminus B_i^k$ **else** set $\Omega_i^{k+1} \leftarrow \Omega_i^k$.
 - 10: **end for**
 - 11: Increase $k \leftarrow k + 1$.
 - 12: **end while**
-

ensure that B_i^j is excluded, we have to add the inequality (8f). One can easily see that System (8) requires $\mathcal{O}(k |c(i)|)$ additional variables and constraints for every $i \in [p]$. □

We additionally remark that after having solved the master problem, all F_i can be evaluated in parallel in every iteration. The overall algorithm is formally stated in Algorithm 1.

We now prove correctness of the algorithm, i.e., that the algorithm terminates after a finite number of iterations with an approximate global optimal solution or with an indication of infeasibility of the original problem. Here, an approximate global optimum is an optimal point that does not violate any constraint more than a given $\varepsilon > 0$, i.e., an approximate global optimum is a solution of Problem (3).

Theorem 3.4 *Suppose that Assumption 1 holds. Then, Algorithm 1 terminates after a finite number of iterations at an approximate globally optimal point x^k of Problem (1) or with an indication that Problem (1) is infeasible.*

Proof We assume that the algorithm does not terminate after a finite number of iterations. That is, there exists an $i \in [p]$ and a corresponding subsequence (indexed by l) of iterates with

$$|F_i(x_{c(i)}^l)| > \varepsilon. \tag{9}$$

We investigate the master problem’s solutions $x_{c(i)}^l$. Since all variables are bounded, the subsequence $(x_{c(i)}^l)$ has a convergent subsequence $(x_{c(i)}^\mu)$. Thus, we can write

$$\|x_{c(i)}^\alpha - x_{c(i)}^\beta\|_\infty < \delta \tag{10}$$

for all sufficiently large indices α and β of the μ -subsequence and arbitrarily small $\delta > 0$. However, all iterates $x_{c(i)}^l$ are excluded from the subsequent feasible set, see (4), and together with (9) we can write

$$\|x_{c(i)}^\alpha - x_{c(i)}^\beta\|_\infty \geq \frac{|F_i(x_{c(i)}^k)|}{L_i} > \frac{\varepsilon}{L_i},$$

for all α, β . This contradicts (10). □

Next, we establish a worst-case bound for required number of iterations.

Theorem 3.5 *For given $i \in [p]$, let $n_i = |c(i)|$, i.e., $x_{c(i)} \in \mathbb{R}^{n_i}$. Furthermore, let $\sigma^i := \bar{x}_{c(i)} - \underline{x}_{c(i)} \in \mathbb{R}^{n_i}$. Then, Algorithm 1 terminates after a maximum number of*

$$\sum_{i \in [p]} \prod_{j=1}^{n_i} \left(\left\lfloor \sigma_j^i \frac{L_i}{\varepsilon} + 1 \right\rfloor \right)$$

iterations.

Proof We show that for each constraint $F_i(x_{c(i)}) = 0, i \in [p]$, there are at most

$$\prod_{j=1}^{n_i} \left(\left\lfloor \sigma_j^i \frac{L_i}{\varepsilon} + 1 \right\rfloor \right)$$

iterations k for which $\Omega_i^{k+1} \neq \Omega_i^k$ holds. Since in each iteration at least one of the sets Ω_i^k needs to be changed this proves the assertion.

To see the claim, we notice, that the hypercube $[x_{c(i)}, \bar{x}_{c(i)}] \subset \mathbb{R}^{n_i}$ can be covered by

$$N = \prod_{j=1}^{n_i} \left(\left\lfloor \sigma_j^i \frac{L_i}{\varepsilon} + 1 \right\rfloor \right)$$

hypercubes H_ε^k , $k \in [N]$, with side-length ε/L_i . To see this, we note that each of the intervals $[x_j, \bar{x}_j]$ for $j \in c(i)$ can be decomposed as

$$x_j = x_j + 0 \cdot \frac{\varepsilon}{L_i} < x_j + 1 \cdot \frac{\varepsilon}{L_i} < \dots < x_j + \left\lfloor \sigma_j^i \frac{L_i}{\varepsilon} \right\rfloor \frac{\varepsilon}{L_i} \leq \bar{x}_j$$

into $\lfloor \sigma_j^i L_i / \varepsilon + 1 \rfloor$ subintervals of length lower or equal ε/L_i . Taking the product of these intervals gives the desired cover of the hypercube.

Now, by definition, in an iteration k during which Ω_i^k is changed, a point $x_{c(i)}$ together with its neighborhood B_i^k is excluded from Ω_i^k . In order for this to happen,

$$\frac{|F_i(x_{c(i)})|}{L_i} > \frac{\varepsilon}{L_i}$$

needs to hold. Consequently, the neighborhood B_i^k contains a hypercube with side-length $2\varepsilon/L_i$. As a consequence, if $x_{c(i)}$ is contained in H_ε^k , then no future iterate can be placed in this H_ε^k . This proves the claimed bound on the iterations. \square

The last theorem shows that the maximum number of required iterations is bounded above by $\mathcal{O}((L^{\max}/\varepsilon)^{\bar{n}})$, where $\bar{n} = \max\{c(i) : i \in [p]\}$ is the maximum number of arguments of the nonlinear functions F_i and $L^{\max} = \max\{L_i : i \in [p]\}$ is the maximum Lipschitz constant. Moreover, the asymptotic bounds derived in [40,41] indicate that, in general, no better worst-case iteration bound can be expected.

Remark 3.6 During the course of the algorithm, the excluded boxes B_i^k are determined based on the master problem’s solution. To tighten the feasible set of the initial master problem, we first equidistantly sample points from the initial box Ω_i^0 . Afterward, we sort these sampling points by the excluded box volume, see (4), in descending order and add them to the initial master problem if the sampling points are not excluded by the box of a previous sampling point. By doing so, we obtain a tighter feasible region from the beginning at the cost of a larger MILP formulation.

4 Numerical case study

In this sect., we present computational results of Algorithm 1 applied to stationary gas transport optimization. We briefly describe the model in Sect. 4.1 and discuss the results in Sect. 4.2.

4.1 Stationary gas transport optimization

One central task in the gas industry is to transport prescribed supplied and discharged flows at minimum costs. Gas mainly flows from higher to lower pressures and in order to transport gas over large distances through pipeline systems, it is required to increase the gas pressure. This is done by compressors. Usually, these compressors add discrete aspects to the problem. In contrast to that, for this case study we decide to choose the easiest and, thus, continuous modeling of a compressor so that we can purely focus on the main aspects of the proposed algorithm, i.e., the iteratively tightened outer approximation of the nonconvex feasible set induced by the nonlinearities of the model.

We model a gas network as a directed graph $G = (V, A)$ with node set V and arc set A . The set of nodes is partitioned into the set of entry nodes V_+ , where gas is supplied, the set of exit nodes V_- , where gas is discharged, and the set of inner nodes V_0 . The set of arcs consists of pipes A_{pi} and compressors A_{cm} . We associate positive gas flow on arcs $a = (u, v)$ with mass flow in arc direction, i.e., $q_a > 0$ if gas flows from u to v and $q_a < 0$ if gas flows from v to u . Moreover, mass flow variables q_a are bounded from below and above, i.e., $q_a \in [q_a, \bar{q}_a]$. The sets $\delta^{\text{in}}(u) := \{a \in A : a = (v, u)\}$ and $\delta^{\text{out}}(u) := \{a \in A : a = (u, v)\}$ are the sets of in- and outgoing arcs for node $u \in V$. Thus, we model mass conservation by

$$\sum_{a \in \delta^{\text{out}}(u)} q_a - \sum_{a \in \delta^{\text{in}}(u)} q_a = q_u \begin{cases} \geq 0, & u \in V_+, \\ \leq 0, & u \in V_-, \\ = 0, & u \in V_0, \end{cases} \quad \text{for all } u \in V, \quad (11)$$

where q_u denotes the supplied or discharged flows. In addition, we need bounded pressure variables $p_u \in [\underline{p}_u, \bar{p}_u]$ for each node $u \in V$.

Pipes $a \in A_{\text{pi}}$ are used to transport the gas through the network. We consider their length L_a , their diameter D_a , their cross-sectional area A_a , their slope s_a , and their friction factor λ_a , which we model using the formula of Nikuradse; see, e.g., [36]. Gas flow in networks is described by a system of partial differential equations—the Euler equations for compressible fluids [42]. In what follows, we consider the stationary case and assume small velocities, constant temperature \hat{T} , and constant compressibility factor \hat{z} . This leads to the so-called Weymouth equation, see [24,36], that describes the relation between pressure and mass flow on a pipe via

$$p_v^2 - p_u^2 + \Lambda_a |q_a| q_a = 0 \quad \text{for all } a = (u, v) \in A_{\text{pi}}, \quad \Lambda_a = \frac{L_a \lambda_a c^2}{A_a^2 D_a}, \quad (12)$$

where c denotes the speed of sound in natural gas.

Finally, we describe our model of compressors $a = (p_u, p_v) \in A_{\text{cm}}$. They are used to increase the inflow gas pressure to a higher outflow pressure, which we model via

$$p_v = p_u + \Delta_a, \quad 0 \leq \Delta_a^- \leq \Delta_a \leq \Delta_a^+, \quad (13)$$

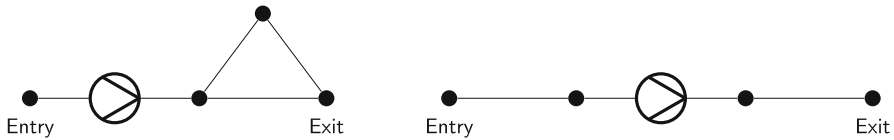


Fig. 2 GasLib-4 (left) and GasLib-4-Tree (right)

for given lower and upper compression bounds Δ_a^- and Δ_a^+ , respectively. For more complicated compressor models, see, e.g., [35,37]. We can, in principle, also include these more complex and mixed-integer nonlinear models but refrain from these complicated models to be able to focus on the main algorithmic aspects of the proposed approach.

We now collect all component models and obtain the mixed-integer optimization problem

$$\begin{aligned} \min \quad & \sum_{a \in A_{\text{cm}}} P_a \\ \text{s.t.} \quad & \text{pressure and flow bounds,} \quad \text{mass conservation: (11),} \\ & \text{pipe model: (12),} \quad \text{compressor model: (13).} \end{aligned}$$

Note that the only nonlinearity of the model is given by the pipe model (12). Consequently, we compute the global Lipschitz constant analytically and obtain (4) by evaluating (12).

4.2 Results

Our test instances are the networks GasLib-4 and GasLib-4-Tree (see Fig. 2 left and right) as well as the networks GasLib-11 and GasLib-24; see [43]. The two latter instances are publicly available at <http://gaslib.zib.de> and the two first ones are available at <https://github.com/m-schmidt-math-opt/cost-of-not-knowing-enough>.

To simplify the presentation and to focus on the main aspects of the proposed method, we modified the networks GasLib-11 and GasLib-24. For the GasLib-11, we removed the valve since this element is not covered in the model discussed in this paper. This makes the resulting network a tree. Moreover, we increased the lower pressure bound at the exits exit02 and exit03 to provide a higher compression. For the GasLib-24 network, we deleted the resistor, the short pipe, and the control valve. The resulting network still contains cycles. Moreover, we again increased the lower pressure bounds of the exits exit02, exit03, and exit04.

The Lipschitz constants are computed by applying Remark 3.1 to (12). For all cyclic networks, we fixed the mass flows where possible (usually on the arcs that connect entries or exits with cyclic parts of the network) to reduce the dimension of the nonlinear constraint (12) by one. The modified networks together with the preprocessed data are publicly available as input data in the GitHub repository at <https://github.com/m-schmidt-math-opt/cost-of-not-knowing-enough>, where also all implementations can be found.

For our algorithm we use the following relative termination criterion. Let x^k be the k th iterate that contains the values p_u^k , p_v^k , and q_a^k for the inflow pressure, the outflow pressure, and for the mass flow on pipe a . Denote further \tilde{p}_u^k , \tilde{p}_v^k , and \tilde{q}_a^k the values that one gets by solving Equation (12) for the respective value and by plugging in the two other values from the current iteration. For instance,

$$\tilde{p}_u^k = \sqrt{(p_v^k)^2 + \Lambda_a |q_a^k| q_a^k}$$

holds. We terminate our algorithm if the maximum relative error

$$\max \left\{ \frac{|\tilde{p}_u^k - p_u^k|}{\tilde{p}_u^k}, \frac{|\tilde{p}_v^k - p_v^k|}{\tilde{p}_v^k}, \frac{|\tilde{q}_a^k - q_a^k|}{\tilde{q}_a^k} \right\}$$

is below 1% for all pipes of the network.

Our algorithm is implemented in Python 3.8.8 using the corresponding Python interface of Gurobi. We solve all MILPs with Gurobi 9.1.1 using 2 physical cores, 4 logical processors, and up to 4 threads; see [44]. We also implemented the nonconvex model using Pyomo 5.7.3 [45,46] and solved it to global optimality with the open-source global solver SCIP 7.0.2 [47] to obtain a benchmark for our method. The time limit is set to 15 min for all tests. All computations were performed on an Intel® Core™ i5-3360M CPU with 4 cores and 2.8 GHz each and 4 GB RAM.

The first part of Table 1 displays the results for the GasLib-4-Tree network. This network is a tree so that we never sample the flow space (see the “—” entries in the second column). This instance is always solved in less than 1 s, independent of how coarse or fine the sampling is done before the main algorithm starts. SCIP (taken as a benchmark) solves the problem in 0.04 s and terminates with a global optimal objective function value of 6.74. Note that the objective function values in the first part of Table 1 are always slightly smaller, which is consistent to that we compute approximate global optimal points and that this approximation always corresponds to a relaxation. Finally, we see that the maximum relative error is always below 1% as desired. We see that the number of required iterations is monotonically decreasing in dependence of the granularity of the sampling process that we apply before the main algorithm starts. This is to be expected since the more sampling points we use, the better our initial outer approximation of the zero set of the nonlinearities usually will be.

However, this is not always the case as can be seen in the middle part of the table that displays the results for the GasLib-4 network, which contains a cycle. SCIP solves this model in 0.64 s and delivers the optimal objective function value of 9.57. The algorithm inside SCIP is using the explicitly given algebraic representation of the constraints to set up a spatial branching process that is based on concave overestimators and convex underestimators of the nonlinearities. This significantly outperforms our method, which is *not* using the explicit representation of the constraint. Nevertheless, we can still solve the instance in approximately 150 s. Interestingly, less sampling in the preprocessing phase seems to be favorable since, otherwise, already the initial MILP is getting too large so that every iteration (in which we have to solve MILPs

Table 1 The results of Algorithm 1 for the GasLib-4-Tree network (top), the GasLib-4 network (middle), and the GasLib-11 network (bottom). The first two columns state the equidistant step sizes for sampling the feasible pressure intervals (σ_p) and for the feasible mass flow intervals (σ_q), k is the number of iterations required by Algorithm 1, “ t (sampling)” and “ t (total)” is the time spent in the sampling routine and the overall solution time (in seconds). “Obj.” is the approximate global optimal objective function value and “Rel. Err.” is the achieved maximum relative error (in percent)

σ_p (in bar)	σ_q (in kg s ⁻¹)	k	t (sampling) (in s)	t (total) (in s)	Obj.	Rel. err. (in %)
—	—	14	—	0.10	6.58	0.66
10.0	—	12	0.00	0.09	6.65	0.61
5.0	—	13	0.00	0.09	6.62	0.88
2.5	—	12	0.00	0.09	6.58	0.98
1.0	—	7	0.00	0.08	6.56	0.62
0.5	—	4	0.01	0.13	6.56	0.97
0.1	—	2	0.33	0.55	6.62	0.78
—	—	150	—	148.03	9.46	0.85
10.0	100.0	154	0.00	147.34	9.45	0.85
5.0	50.0	167	0.00	166.07	9.46	0.80
5.0	10.0	172	0.00	367.07	9.48	0.98
5.0	5.0	153	0.01	440.38	9.48	0.96
1.0	50.0	168	0.02	508.19	9.46	0.82
1.0	10.0	122	0.26	595.17	9.45	0.73
1.0	5.0	—	—	—	—	—
1.0	1.0	—	—	—	—	—
—	—	55	—	16.27	12.64	0.78
10.0	—	51	0.00	20.84	12.57	0.90
5.0	—	52	0.01	30.01	12.64	0.83
2.5	—	47	0.02	29.38	12.58	0.98
1.0	—	37	0.17	37.05	12.52	0.95
0.5	—	27	0.86	57.58	12.62	0.96
0.1	—	22	49.81	336.67	12.53	0.98

of increasing size) is getting too costly. The extreme case is the very fine sampling shown in the last two rows of the middle part of the table, which leads to the case that our approach hits the time limit.

Finally, the bottom part of the table shows the results for the GasLib-11 network. This network is larger than the two other ones but is again a tree. This leads to the fact that all mass flows can be fixed in the preprocessing and that the dimensions of the nonlinear constraints on the arcs are reduced by one. Again, less sampling leads to better results. We can solve the instance in 16 s if we do not apply any sampling. However, SCIP is much faster and solves the instance in 0.12 s, yielding an optimal objective function value of 12.93. For the GasLib-24 network, our method always reached the time limit, whereas SCIP solves the instance in 1.13 s with an optimal objective function value of 27.76.

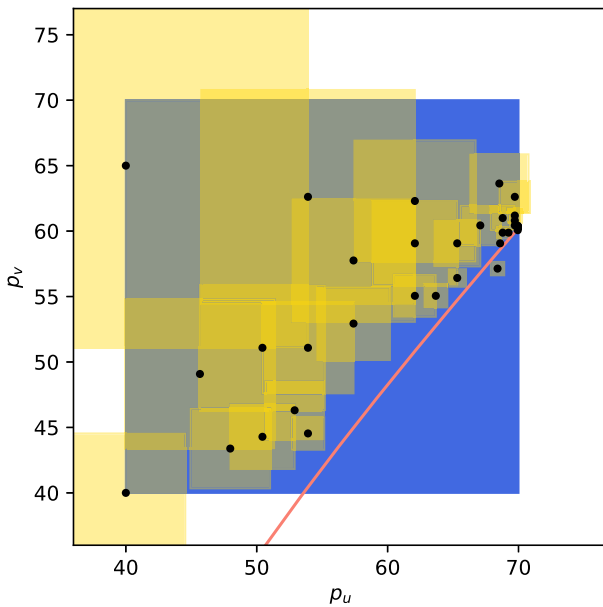


Fig. 3 Iterates in the (p_u, p_v) space of an exemplary pipe of the GasLib-11 network

To sum up, we see that the “cost of not knowing” is rather high since the benchmark (SCIP) is solving the problems much faster. However, one has to remark that we compare a hand-crafted implementation of our method with a highly evolved and large-scale software package. SCIP is (of course, on purpose) exploiting all the given insights based on the explicit algebraic representation of the constraints to construct tight convex envelopes. We, in contrast to that, do not exploit (again, on purpose) all this information. The strength of our method lies in the fact that the required assumptions are very weak. However, for such instances, a comparison with a global solver such as SCIP does not make sense since such a global solver would simply not be able to solve the problem.

Let us finally take a closer look into the behavior of the proposed method by analyzing the iterates in the (p_u, p_v) space of an exemplary pipe during the solution process of the GasLib-11 network; see Fig. 3.

First, we see that the nonlinearity (the red curve) is rather moderate in this example, which is something that SCIP explicitly exploits since the constructed convex envelopes are rather tight in this case. Our method, on the other hand, is blind to such structural characteristics of the model. The blue box is the initial feasible box that our algorithm starts with. Black points correspond to iterates and the respective yellow boxes represent the excluded boxes. Since the Lipschitz constant stays the same over the course of the iteration, the size of the boxes is monotonic in the infeasibility of the corresponding iterate. One can nicely see the automatic adaptivity of the method: The nonlinearity is approximated much more accurately close to the optimal point. Moreover, the objective function of the problem solved in each iteration yields iterates mostly above the feasible curve.

Both aspects make clear that a more sophisticated sampling method in the preprocessing phase could lead to much better result than those presented for the equidistant sampling used in our numerical results. Finally, our method is, in general, very well suited for warmstarting since the MILPs do not differ very much from iteration to iteration. However, our method right now does not use this possibility since a direct use of the warmstarting capabilities of Gurobi cannot be used. The reason is that the variable space changes from iteration to iteration and that the old point is not feasible anymore since it is explicitly excluded from the feasible set by the last added box. Exploiting the potential of warmstarting would, in our opinion, improve the performance of the algorithm since it will speed up the running time in every iteration of the method.

5 Conclusion

Without any doubt, mixed-integer nonlinear optimization is hard. Fortunately, in many practically relevant situations effective solution approaches exist. In this paper, we considered especially hard cases of mixed-integer nonlinear optimization with implicit constraints that have the adverse property that evaluating the constraint does not give a feasible point and that they may be multi-dimensional. The numerical results are somehow sobering since we show that the “cost of not knowing enough” about the problem’s structure is rather high. On the other hand, the class of considered problems is of high importance in many fields of applications.

Thus, there is need for computational improvement for this class of problems if they are tackled using MILP technology as a working horse. At this point, we suggest three possible directions of future research.

- 1 Is it possible to exploit local Lipschitz information to increase the volume of excluded infeasible regions during the course of the iteration?
- 2 Is it possible to create iterates that both yield larger excluded regions for the next iteration but that are also near the feasible region of the original problem?
- 3 We showed that a naive sampling does not necessarily lead to improved computation times. However, more involved sampling strategies together with suitable presolve strategies to decrease the initial box sizes has good chances of improving the computational performance and is part of our future work.

Acknowledgements The research of the first author has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. All authors thank the DFG for their support within projects A05, A08, and B08 in CRC TRR 154. Finally, we greatly thank our colleagues Alexander Martin and Günter Leugering. A few years ago, their research vision pointed out the direction of study that we follow here and lead to the first ideas and results that paved the way for this paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted

by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Kannan, R., Monma, C. L.: On the Computational Complexity of Integer Programming Problems. In: Henn, R., Korte, B., Oettli, W. (eds.) *Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn*, pp. 161–172. Springer Berlin Heidelberg, Berlin (1978) https://doi.org/10.1007/978-3-642-95322-4_17
2. Garey, M.R., Johnson, D.S.: *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co., New York (1979)
3. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. *Acta Numer.* **22**, 1–131 (2013). <https://doi.org/10.1017/S0962492913000032>
4. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36**(3), 307–339 (1986). <https://doi.org/10.1007/BF02592064>
5. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. *Math. Program.* **66**(1), 327–349 (1994). <https://doi.org/10.1007/BF01581153>
6. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. *Discret. Optim.* **15**, 186–204 (2008). <https://doi.org/10.1016/j.disopt.2006.10.011>
7. Kronqvist, J., Bernal, D.E., Lundell, A., Westerlund, T.: A centercut algorithm for quickly obtaining feasible solutions and solving convex MINLP problems. In: *Computers & Chemical Engineering 122 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27)*, pp. 105–113. (2019) <https://doi.org/10.1016/j.compchemeng.2018.06.019>
8. Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E.: A review and comparison of solvers for convex MINLP. *Optim. Eng.* **2**, 397–455 (2019). <https://doi.org/10.1007/s11081-018-9411-8>
9. Al-Khayyal, F.A., Sherali, H.D.: On finitely terminating branch-and-bound algorithms for some global optimization problems. *SIAM J. Optim.* **10**(4), 1049–1057 (2000). <https://doi.org/10.1137/S105262349935178X>
10. Smith, E.M.B., Pantelides, C.C.: Global optimisation of nonconvex MINLPs. *Comput. Chem. Eng.* **21**, S791–S796 (1997). [https://doi.org/10.1016/S0098-1354\(97\)87599-0](https://doi.org/10.1016/S0098-1354(97)87599-0)
11. Tawarmalani, M., Sahinidis, N. V.: *Convexification and global optimization in continuous and mixed integer nonlinear programming. Theory, algorithms, software, and applications*. Vol. 65. Kluwer Academic Publishers, Dordrecht (2002). <https://doi.org/10.1007/978-1-4757-3532-1>
12. Pintér, J.D.: *Global optimization in action (continuous and lipschitz optimization: algorithms, implementations and applications)*. Springer Verlag (1996). <https://doi.org/10.1007/978-1-4757-2502-5>
13. Pintér, J.: Globally convergent methods for n-dimensional multiextremal optimization. *Optimization* **17**(2), 187–202 (1986). <https://doi.org/10.1080/02331938608843118>
14. Horst, R., Thoai, N.V.: Branch-and-bound methods for solving systems of Lipschitzian equations and inequalities. *J. Optim. Theory Appl.* **58**(1), 139–145 (1988). <https://doi.org/10.1007/BF00939776>
15. Horst, R., Tuy, H.: On the convergence of global methods in multiextremal optimization. *J. Optim. Theory Appl.* **54**(2), 253–271 (1987). <https://doi.org/10.1007/BF00939434>
16. Horst, R.: Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, DCprogramming, and lipschitzian optimization. *J. Optim. Theory Appl.* **58**(1), 11–37 (1988). <https://doi.org/10.1007/BF00939768>
17. Pintér, J.: Branch- and bound algorithms for solving global optimization problems with Lipschitzian structure. *Optimization* **19**(1), 101–110 (1988). <https://doi.org/10.1080/02331938808843322>
18. Tuy, H., Horst, R.: Convergence and restart in branch-and-bound algorithms for global optimization Application to concave minimization and DC optimization problems. *Math. Program.* **41**, 161–183 (1988). <https://doi.org/10.1007/BF01580762>
19. Horst, R., Tuy, H.: *Global optimization*, 2nd edn. Springer Verlag, Berlin (1996). <https://doi.org/10.1007/978-3-662-03199-5>
20. Gugat, M., Leugering, G., Martin, A., Schmidt, M., Sirvent, M.M., Wintergerst, D.: Towards simulation based mixed-integer optimization with differential equations. *Networks* (2018). <https://doi.org/10.1002/net.21812>

21. Schmidt, M., Sirvent, M., Wollner, W.: A decomposition method for MINLPs with Lipschitz continuous nonlinearities. *Math. Program.* **178**(1–2), 449–483 (2019). <https://doi.org/10.1007/s10107-018-1309-x>
22. Gugat, M., Leugering, G., Martin, A., Schmidt, M., Sirvent, M., Wintergerst, D.: MIP-based instantaneous control of mixed-integer PDE-constrained gas transport problems. *Comput. Optim. Appl.* **70**, 267–294 (2018). <https://doi.org/10.1007/s10589-017-9970-1>
23. Sirvent, M.: Incorporating differential equations into mixed-integer programming for gas transport optimization. PhD thesis. FAU Erlangen-Nürnberg, (2018)
24. Geißler, B., Martin, A., Morsi, A., Schewe, L.: The MILP-relaxation approach. In: *Evaluating gas network capacities. SIAM-MOS series on Optimization*, Vol 21, pp. 103–122. SIAM, Philadelphia, PA (2015). <https://doi.org/10.1137/1.9781611973693.ch6>
25. Geissler, B., Martin, A., Morsi, A., Schewe, L.: Using piecewise linear functions for solving MINLPs. In: Lee, J., Leyffer, S., (eds.) *Mixed Integer Nonlinear Programming. The IMA Volumes in Mathematics and its Applications*. Vol. 154, pp. 287–314 (2012) https://doi.org/10.1007/978-1-4614-1927-3_10
26. Burlacu, R., Geissler, B., Schewe, L.: Solving mixed-integer nonlinear programs using adaptively refined mixed-integer linear programs. Tech. Rep. FAU Erlangen-Nürnberg, 2017. <https://opus4.kobv.de/opus4-trr154/frontdoor/index/index/docId/151>
27. Sirvent, M., Kanelakis, N., Geissler, B., Biskas, P.: Linearized model for optimization of coupled electricity and natural gas systems. *J. Mod. Power Syst. Clean Energy* **5**(3), 364–374 (2017). <https://doi.org/10.1007/s40565-017-0275-2>
28. Geissler, B., Morsi, A., Schewe, L., Schmidt, M.: Solving highly detailed gas transport MINLPs: block separability and penalty alternating direction methods. *INFORMS J. Comput.* **30**(2), 1–15 (2018). <https://doi.org/10.1287/ijoc.2017.0780>
29. Geissler, B., Morsi, A., Schewe, L., Schmidt, M.: Solving power-constrained gas transportation problems using an MIP-based alternating direction method. *Comput. Chem. Eng.* **82**, 303–317 (2015). <https://doi.org/10.1016/j.compchemeng.2015.07.005>
30. Pfetsch, M.E., Fügenschuh, A., Geissler, B., Geissler, N., Gollmer, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., Morsi, A., Rövekamp, J., Schewe, L., Schmidt, M., Schultz, R., Schwarz, R., Schweiger, J., Stangl, C., Steinbach, M.C., Vigerske, S., Willert, B.M.: Validation of nominations in gas network optimization: models, methods, and solutions. *Optim. Methods Softw.* **30**(1), 15–53 (2015). <https://doi.org/10.1080/10556788.2014.888426>
31. Fügenschuh, A., Geissler, B., Gollmer, R., Hayn, C., Henrion, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., Mirkov, R., Rövekamp, J., Schewe, L., Schmidt, M., Schultz, R., Schwarz, R., Schweiger, J., Stangl, C., Steinbach, M.C., Willert, B.M.: Mathematical optimization for challenging network planning problems in unbundled liberalized gas markets. *Energy. Syst.* **5**(3), 449–473 (2014). <https://doi.org/10.1007/s12667-013-0099-8>
32. Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L.: Evaluating Gas Network Capacities. *SIAM-MOS series on Optimization*. SIAM (2015). <https://doi.org/10.1137/1.9781611973693>
33. Ríos-Mercado, R.Z., Borraz-Sánchez, C.: Optimization problems in natural gas transportation systems: a state-of-the-art review. *Appl. Energy* **147**, 536–555 (2015). <https://doi.org/10.1016/j.apenergy.2015.03.017>
34. Schmidt, M., Steinbach, M.C., Willert, B.M.: High detail stationary optimization models for gas networks: validation and results. *Optim. Eng.* **17**(2), 437–472 (2016). <https://doi.org/10.1007/s11081-015-9300-3>
35. Schmidt, M., Steinbach, M.C., Willert, B.M.: High detail stationary optimization models for gas networks. *Optim. Eng.* **16**(1), 131–164 (2015). <https://doi.org/10.1007/s11081-014-9246-x>
36. Fügenschuh, A., Geissler, B., Gollmer, R., Morsi, A., Pfetsch, M. E., Rövekamp, J., Schmidt, M., Spreckelsen, K., Steinbach, M. C.: 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*. Chap. 2, pp. 17–44. SIAM, Philadelphia, PA (2015) <https://doi.org/10.1137/1.9781611973693.ch2>
37. Schmidt, M., Steinbach, M. C., Willert, B. M.: The precise NLP model. In: Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L. (eds.) *Evaluating Gas Network Capacities. SIAM-MOS series on Optimization*. Chap. 10, pp. 181–210. SIAM, Philadelphia, PA (2015). <https://doi.org/10.1137/1.9781611973693.ch10>
38. Joormann, I., Schmidt, M., Steinbach, M. C., Willert, B. M.: What does feasible mean? In: Koch, T., Hiller, B., Pfetsch, M. E., Schewe, L. (eds.) *Evaluating Gas Network Capacities. SIAM-MOS series*

- on Optimization. Chap. 11, pp. 211–232. SIAM, Philadelphia, PA (2015). <https://doi.org/10.1137/1.9781611973693.ch11>
39. Mehrmann, V., Schmidt, M., Stolwijk, J.J.: Model and discretization error adaptivity within stationary gas transport optimization. *Vietnam J. Math.* **46**(4), 779–801 (2018). <https://doi.org/10.1007/s10013-018-0303-1>
 40. Nemirovsky, A.S., Yudin, D.B.: Problem complexity and method efficiency in optimization. Wiley-interscience series in discrete mathematics. John Wiley & Sons, New York (1983)
 41. Vavasis, S.A.: Nonlinear optimization: complexity issues. Oxford University Press Inc, New York (1991)
 42. Feistauer, M.M.: Mathematical methods in fluid dynamics Pitman monographs and surveys in pure and applied mathematics. Longman Scientific & Technical New York, Harlow (1993)
 43. Schmidt, M., Assmann, D., Burlacu, R., Humpola, J., Joormann, I., Kanelakis, N., Koch, T., Oucherif, D., Pfetsch, M.E., Schewe, L., Schwarz, R., Sirvent, M.: GasLib-a library of gas network instances. *Data* (2017). <https://doi.org/10.3390/data2040040>
 44. Gurobi. Gurobi optimization inc. <http://www.gurobi.com>
 45. Hart, W.E., Watson, J.P., Woodruff, D.L.: Pyomo: modeling and solving mathematical programs in Python. *Math. Program. Comput.* **3**, 219–260 (2011). <https://doi.org/10.1007/s12532-011-0026-8>
 46. Hart, W. E., Laird, C., Watson, J.P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D.: *Pyomo - optimization modeling in Python*. Springer, Boston, MA (2017). <https://doi.org/10.1007/978-1-4614-3226-5>
 47. Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S.J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M.E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D., Witzig, J.: *The SCIP Optimization Suite 7.0*. Technical Report. Optimization Online, (2020). http://www.optimization-online.org/DB_HTML/2020/03/7705.html

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.