Mathematical Programming (2019) 178:449–483
https://doi.org/10.1007/s10107-018-1309-x

FULL LENGTH PAPER

Series A

A decomposition method for MINLPs with Lipschitz continuous nonlinearities

Martin Schmidt[1,2] · **Mathias Sirvent**[1] · **Winnifried Wollner**[3]

Received: 24 July 2017 / Accepted: 11 June 2018 / Published online: 23 June 2018
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2018

Abstract

Many mixed-integer optimization problems are constrained by nonlinear functions that do not possess desirable analytical properties like convexity or factorability or cannot even be evaluated exactly. This is, e.g., the case for many problems constrained by differential equations or for models that rely on black-box simulation runs. For these problem classes, we present, analyze, and test algorithms that solve mixed-integer problems with Lipschitz continuous nonlinearities. Our theoretical results depend on the assumptions made on the (in)exactness of function evaluations and on the knowledge of Lipschitz constants. If Lipschitz constants are known, we prove finite termination at approximate globally optimal points both for the case of exact and inexact function evaluations. If only approximate Lipschitz constants are known, we prove finite termination and derive additional conditions under which infeasibility can be detected. A computational study for gas transport problems and an academic case study show the applicability of our algorithms to real-world problems and how different assumptions on the constraint functions up- or downgrade the practical performance of the methods.

Keywords Mixed-integer nonlinear optimization · Lipschitz optimization · Inexact function evaluations · Decomposition methods · Gas networks

✉ Martin Schmidt
mar.schmidt@fau.de

Mathias Sirvent
mathias.sirvent@fau.de

Winnifried Wollner
wollner@mathematik.tu-darmstadt.de

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

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

[3] Fachbereich Mathematik, Technische Universität Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

Mathematics Subject Classification 90-08 · 90C11 · 90C30 · 90C90

1 Introduction

Mixed-integer nonlinear optimization is an important tool for modeling real-world problems from a large variety of applications like, e.g., engineering, economics, or the natural sciences. The main reason for this is that this problem class combines both the capability of modeling nonlinearities and decision making modeled by discrete variables; cf. the survey [7] and the references therein for an overview of mixed-integer nonlinear optimization. However, this combination is also the reason why mixed-integer nonlinear optimization problems (MINLPs) are typically extremely hard to solve—both in theory [22,45] and practice [7, Chap. 2].

As it is often the case in optimization, convexity renders MINLPs much easier to solve. In particular, (extended) cutting plane [82,83] or outer approximation techniques [17,21] can be exploited, which is, in general, not possible for nonconvex MINLPs. Here, outer approximations (usually convex underestimators and concave overestimators) are only locally valid and one has to resort to spatial branching for achieving tight outer approximations that fulfill prescribed tolerances [2].

Unfortunately, many MINLPs neither exhibit convexity properties nor do they possess other desirable properties like separability or factorability that are frequently used for nonconvex optimization [55,74,75]. Examples include MINLPs with ordinary or partial differential equations, mixed-integer optimal control problems, or optimization problems based on "black-box" simulations.

In this paper, we present, analyze, and test algorithms for MINLPs with nonlinearities for which the only usable analytical property is Lipschitz continuity. Thus, these algorithms fit into the above mentioned frameworks—among others like parametric or bilevel optimization, where complicated nondifferentiable but Lipschitz continuous functions are frequently studied; cf., e.g., [14,16].

The field of global Lipschitz optimization without integer variables is well-studied. Such problems are tackled with suitable underestimators in the form of saw-tooth covers that were first presented in [19,65] for univariate cases. Extensions were made for higher dimensions [59–61] including branch-and-bound approaches [42,43]. Even more sophisticated branch-and-bound procedures have been studied in [41,62,78]. For an overview of the literature, see the books [38,44,64] and the references therein. An example for a software based on these concepts is the so-called Lipschitz global optimizer (LGO) [63]. Moreover, the use of relaxations instead of underestimators has also been considered; see [49]. Finally, MINLPs with strongly convex and Lipschitz continuous functions are studied in [4].

Further, in the context of optimal control problems one frequently has problems governed by a dynamical system involving a Lipschitz continuous right-hand side. In such a setting, the dynamical system determines a state for given control/decision variables. These control/decision variables often need to be determined by a mixed-integer linear program (MIP), cf. [32,39,40,68]. The mentioned methods utilize the knowledge that the nonlinearity in the problem stems from an evolutionary equation.

In contrast to this, we focus on generic Lipschitz continuous nonlinearities that, e.g., describe stationary states of such optimal control problems.

The contribution of this paper is the presentation and analysis of algorithms for nonconvex MINLPs with Lipschitz continuous constraints. All presented algorithms are decomposition methods that decouple the problem into a master and several subproblems. We show that the master problem can always be modeled as a MIP and that the subproblems contain the complicating nonlinear constraints. We study different variants of this general setup that vary in the aspects of (i) whether we have exact or only approximate knowledge about the Lipschitz constants and (ii) whether we are able to evaluate the constraint functions exactly or only approximately. If both global Lipschitz constants are known and exact function evaluations are available, we prove that our algorithm terminates finitely at approximate globally optimal points that satisfy prescribed tolerances. The same holds true for the case of known Lipschitz constants but only approximate evaluation of the constraint functions if we assume that we can a-priorily bound the error of the constraint evaluations. The case of only approximately known Lipschitz constants turns out to be more difficult. For instance, finite detection of infeasibility can only be achieved if algorithmic input parameters are chosen very carefully. Numerical results from the field of gas transport and an academic case study finally show the applicability of our approaches for solving real-world problems.

There are many related algorithms in the literature on MINLPs. One of our key algorithmic ideas is the relaxation of the Lipschitz continuous nonlinearities by using (piecewise) linear relaxations of special type. From a quite general standpoint, this is also the same for outer approximation [9,17,21] or the extended cutting plane method [82,83]. However, these algorithms are tailored for the convex case, whereas our algorithms require only Lipschitz continuity and thus allow for nonconvex or even non-differentiable constraints. Ideas for the nonconvex case that follow the same design rationale like over- and underestimating the given nonlinearity by simpler functions include, e.g., McCormick's inequalities [55], the piecewise linear relaxation approach described in [24,25,27], or, in general, spatial branching; cf., e.g., [75] and the references therein. Albeit this similarity, our approach significantly differs in how the piecewise linear outer approximations are constructed. Since we decompose our original MINLP into a master and several subproblems, our approach also has some similarities with generalized Benders decomposition [8,30]. However, the definition of the subproblems as well as the generation of additional constraints for the subsequent master problems clearly differ. Finally, the topic of inexact function evaluations in MINLP algorithms is only rarely addressed in the literature. For finite-dimensional MINLPs, "inexactness" mainly relates to the solution of NLP subproblems, cf., e.g., [10,52,53]. A particular source of such an inexactness occurs if the NLP involves the solution of an ordinary or partial differential equation since then relatively large (compared to machine accuracy) errors are unavoidable, and moreover, the error-estimations are typically true up to an unknown constant, up to data-oscillation terms, or require a sufficiently fine resolution of the problem; cf., e.g., [1,3,5,80,81].

The rest of the paper is structured as follows. In Sect. 2, we introduce the class of problems under consideration and define ε-feasibility. Afterward, the basic algorithmic framework is discussed in Sect. 3, where we also collect the set of assumptions made on the (in)exactness of constraint evaluations and on the knowledge of Lipschitz

constants. The algorithms for the cases with known Lipschitz constants—both with exact or approximate constraint evaluations—are then discussed in Sect. 4. A variant of these algorithms for only approximately known Lipschitz constants is presented in Sect. 5. Finally, a computational study regarding the application of our algorithms to gas transport problems and an academic case study are given in Sect. 6 before the paper closes with a conclusion and a brief discussion of possible future work in Sect. 7.

2 Problem setting

We are concerned with the problem

$$\begin{aligned} \min_x \quad & c^\top x && \text{(1a)} \\ \text{s.t.} \quad & Ax \geq b, \quad \underline{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \{0,1\}^m, && \text{(1b)} \\ & f_i(x_{i_1}) = x_{i_2}, \quad i \in [p], && \text{(1c)} \end{aligned}$$

where $c \in \mathbb{R}^{n+m}$, $A \in \mathbb{R}^{q \times (n+m)}$, $\underline{x} \in \mathbb{R}^n \times \{0\}^m$, $\bar{x} \in \mathbb{R}^n \times \{1\}^m$, and $b \in \mathbb{R}^q$ are given data and $[p] := 1, \ldots, p$. The feasible region of (1) is denoted by $\mathcal{F}$. The nonlinear functions $f_i : \mathbb{R} \to \mathbb{R}$ are (locally) Lipschitz continuous functions that couple certain real values x_{i_1} to certain other real values x_{i_2} for $i \in [p]$ and prohibit the use of standard MINLP solvers. The latter can, e.g., be the case if the f_i are outcome of the solution of an ordinary or partial differential equation or if f_i is only given by black-box simulation runs. In what follows, we also write $x_i = (x_{i_1}, x_{i_2}) \in \mathbb{R}^2$.

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

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

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

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b, \quad \underline{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \{0,1\}^m, \\ & |f_i(x_{i_1}) - x_{i_2}| \leq \varepsilon, \quad i \in [p], \end{aligned} \tag{3}$$

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.

Definition 1 (ε-*feasibility*) We call a point ε-feasible if it is feasible for Problem (3).

3 Basic algorithmic framework

In this section, we introduce the main ideas of our algorithms that will be presented in the two following sections. Moreover, we introduce some required notation and state the main assumptions on the knowledge about Lipschitz constants and on the (in)exactness of function evaluations.

The bounds $\underline{x}$ and $\bar{x}$ of (1) give rise to a-priorily known compact boxes $\Omega_i = [\underline{x}_{i_1}, \bar{x}_{i_1}] \times [\underline{x}_{i_2}, \bar{x}_{i_2}] \subset \mathbb{R}^2$ such that the graph of f_i over feasible points x_{i_1} is contained in Ω_i. As the details of the algorithms vary depending on certain assumptions we make regarding the available information on the functions f_i, we first provide a sketch of the generic algorithmic structure to be refined later.

The main idea is that the algorithm constructs a sequence of subsets $(\Omega_i^k)_k$ such that Ω_i^k converges to the graph of f_i for $k \to \infty$. In order to allow the use of standard MIP solvers, we assume that the relaxations $(\Omega_i^k)_k$ of the nonlinear constraints (1c) are finite unions of polytopes, i.e.,

$$\Omega_i^k = \bigcup_{j \in J_i^k} \Omega_i^k(j). \tag{4}$$

That is, $\Omega_i^k(j)$ are polytopes for all j in some finite index set J_i^k for every $i \in [p]$ and every iteration k. The precise definition of the sets $\Omega_i^k(j)$ will follow later. Up to now it is only important that Ω_i^k is a union of polytopes that forms a relaxation of the graph of the function $f_i(x_{i_1})$ for $\underline{x}_{i_1} \leq x_{i_1} \leq \bar{x}_{i_1}$. The algorithm then alternatingly solves master and subproblems. The master problem is defined over the relaxed sets Ω_i^k and the subproblems are used to effectively refine these relaxations to finally obtain an ε-feasible solution of the original problem. With these preparations, we are now in the position to state the kth master problem

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b, \quad \underline{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \{0,1\}^m, \\ & x_i \in \Omega_i^k, \quad i \in [p], \end{aligned} \tag{M(k)}$$

that we solve to global optimality providing a solution x^k. Note that the variable vector x in (M(k)) is the same as the one used in the original problem (1). However, when setting up a concrete MIP formulation for the constraints $x_i \in \Omega_i^k$, $i \in [p]$, we have to extend the original variable vector by additional auxiliary variables that are required to formulate mixed-integer linear models of unions of polytopes.

If the master problem's result is already ε-feasible for the original problem (1), we are done. If this is not the case, we need to improve our approximation of the graph of f_i. To this end, we consider the kth subproblem providing a new point on, or near, the graph of f_i that is, preferably, close to the solution x^k of the previous master problem (M(k)). This is achieved by restricting the subproblem to the polytope $\widetilde{\Omega}_i^k(j_i^k)$, where j_i^k denotes the polytope with $x_i^k \in \Omega_i^k(j_i^k)$ for all $i \in [p]$. With this at hand, the subproblem of the kth iteration reads

$$\min_{\widetilde{x}} \quad \|\widetilde{x} - x^k\|_2^2 \quad \text{s.t.} \quad f_i(\widetilde{x}_{i_1}) = \widetilde{x}_{i_2}, \quad \widetilde{x}_i \in \widetilde{\Omega}_i^k(j_i^k), \quad i \in [p]. \tag{S(k)}$$

In order to assert that the newly found point $\widetilde{x}^k$ is not too close to the corners of $\Omega_i^k(j)$, these subproblems work on an polyhedral inner-approximation $\widetilde{\Omega}_i^k(j_i^k) \subset \Omega_i^k(j_i^k)$.

3.1 Assumptions

For a precise construction of the sets $\Omega_i^k(j)$ and $\widetilde{\Omega}_i^k(j_i^k)$, we need to specify what information on f we are allowed to use. The easiest case—although quite unrealistic in practice—is the following.

Assumption 1 We have an oracle that evaluates $f_i(x_{i_1})$ for all $i \in [p]$ and all f_i are globally Lipschitz continuous on $\underline{x}_{i_1} \leq x_{i_1} \leq \bar{x}_{i_1}$ with known global Lipschitz constant L_i.

More realistic is the setting, where the functions f_i cannot be evaluated exactly, but a reasonably good error bound $\varepsilon_i^f > 0$ is known.

Assumption 2 We have an oracle that provides an approximation $\hat{f}_i(x_{i_1})$ of $f_i(x_{i_1})$ such that

$$f_i(x_{i_1}) \in \left[\hat{f}_i(x_{i_1}) - \varepsilon_i^f(x_{i_1}),\, \hat{f}_i(x_{i_1}) + \varepsilon_i^f(x_{i_1})\right]$$

with known $\varepsilon_i^f(x_{i_1}) > 0$ for all $i \in [p]$. Further, all f_i are globally Lipschitz continuous on $\underline{x}_{i_1} \leq x_{i_1} \leq \bar{x}_{i_1}$ with known global Lipschitz constant L_i.

Even more realistic, and one of the novel aspects of this paper, is the setting in which we have Lipschitz continuous functions, but can only guess the Lipschitz constant, e.g., by evaluation of derivatives. This is formalized by the following assumption.

Assumption 3 We have an oracle that evaluates $f_i(x_{i_1})$ for all $i \in [p]$ and all f_i are locally Lipschitz continuous with known local Lipschitz constant $L_i(x_{i_1})$, i.e., there exists an unknown $\varepsilon_i^L > 0$ (independent of x_{i_1} for simplicity) such that f_i is Lipschitz continuous on the interval

$$\Theta_i(x_{i_1}) := \left[x_{i_1} - \varepsilon_i^L,\, x_{i_1} + \varepsilon_i^L\right] \tag{5}$$

with Lipschitz constant $L_i(x_{i_1})$.

A simple extension of this assumption is the case in which the local Lipschitz constant can only be obtained approximately, e.g., due to errors in the calculation of derivatives.

Assumption 4 We have an oracle that evaluates $f_i(x_{i_1})$ for all $i \in [p]$ and all f_i are locally Lipschitz continuous with known approximate local Lipschitz constant $L_i(x_{i_1})$,

i.e., there exists an unknown $\varepsilon_i^L > 0$ and a known $\delta_i^L \in \mathbb{R}$ (independent of x_{i_1}) such that f_i is Lipschitz continuous on the interval

$$\Theta_i(x_{i_1}) := \left[x_{i_1} - \varepsilon_i^L, x_{i_1} + \varepsilon_i^L\right]$$

with Lipschitz constant $L_i(x_{i_1}) + \delta_i^L$.

Depending on which assumption we use, our algorithmic approach differs. For example, if the global Lipschitz constant is known, then given two points on the graph of f, we can easily construct a quadrilateral such that the graph of f between the two points is contained in the quadrilateral. This allows the construction of sets $\Omega_i^k \supseteq \text{graph}(f_i)$ and, consequently, (M(k)) is a relaxation of (1). See Sects. 4 and 4.1 for the algorithms based on Assumptions 1 and 2.

In contrast, if the Lipschitz constant, although existent, is not known, then such a relaxation cannot be constructed. The algorithm corresponding to the Assumptions 3 and 4 is discussed in Sect. 5.

Remark 1 Before we start with the detailed discussion of the respective algorithms, we briefly discuss the rationale behind the assumptions above. Clearly, Assumption 1 is the generic best case, which has previously been discussed in the literature on global optimization; cf. the references discussed in Sect. 1. In many applications, however, this assumption is not justified. For instance, if complicated nonlinear functions, e.g., involving the solution of an ODE or PDE, need to be evaluated it is unreasonable to assume an exact evaluation to be possible. Nonetheless, in many such situations error estimates for the desired function value are available, see, e.g., [1,5,18,66]. A similar situation arises when natural phenomena need to be modeled and even an exact evaluation of the model does not give an exact prediction of the real phenomenon. This motivates Assumption 2 in which we assume that an upper bound ε_i^f for the error in the computed function value $\hat{f}_i$ is available. A second difficulty lies in the fact that in many applications the Lipschitz constant of the function is not available. In many such cases an evaluation of the derivative of f_i can provide an estimate for the Lipschitz constant. Indeed, if f_i' is continuous then by the mean-value theorem of differential calculus $2|f_i'(x)| + 1$ is a Lipschitz constant for the function f_i near x—however, with the drawback that without further knowledge of f_i it is not precisely clear which points are "near"; giving the setting of Assumption 3. The final Assumption 4 deals with the case when the derivative of f_i cannot be evaluated exactly. This again is the case if, e.g., f_i involves the solution of an ODE or PDE.

Remark 2 In order to illustrate the possible range of applications that we can tackle using the proposed methods, we present the relation to bilevel optimization problems. To this end, we consider the bilevel problem

$$\min_{x,y} \quad c^\top x + d^\top y \tag{6a}$$

$$\text{s.t.} \quad Ax + By \geq a, \tag{6b}$$

$$\underline{x} \leq x \leq \bar{x},\ x \in \mathbb{R}^n \times \{0,1\}^m, \tag{6c}$$

$$y \in \arg\min\{g(x,y)\colon h(x,y) \leq 0,\ y \in \mathbb{R}^\ell\}, \tag{6d}$$

where x are the upper level variables that are decided in the leader's problem

$$\min_{x,y} \quad c^\top x + d^\top y \quad \text{s.t.} \quad Ax + By \geq a,\ \underline{x} \leq x \leq \bar{x},\ x \in \mathbb{R}^n \times \{0,1\}^m$$

and y are the lower level variables that are optimized in the follower's problem

$$\min_{y} \quad g(x,y) \quad \text{s.t.} \quad h(x,y) \leq 0,\ y \in \mathbb{R}^\ell. \tag{7}$$

The latter problem can be seen as a parameterized optimization problem, where the parameters are the given upper level decisions x. We denote the optimal value function of the lower level by

$$\varphi(x) = \min_y\{g(x,y)\colon h(x,y) \leq 0,\ y \in \mathbb{R}^\ell\}$$

and the set of solutions is given by the point-to-set mapping

$$\Psi(x) = \arg\min_y\{g(x,y)\colon h(x,y) \leq 0,\ y \in \mathbb{R}^\ell\}.$$

We assume that $g(x,\cdot)$ and $h(x,\cdot)$ are convex functions. Then, the lower level problem is convex. Let us further assume that the feasible set of the lower level is non-empty and compact and that the Mangasarian–Fromowitz constraint qualification (MFCQ) holds at all points (x,y) with $y \in \Psi(x)$. Then, the optimal value function φ is locally Lipschitz continuous [15,46]. Depending on the feasible set of the upper level, φ can also be shown to be globally Lipschitz continuous. Moreover, the bilevel problem (6) can be replaced by

$$\min_{x,y,\eta} \quad c^\top x + d^\top y \tag{8a}$$

$$\text{s.t.} \quad Ax + By \geq a, \tag{8b}$$

$$\underline{x} \leq x \leq \bar{x},\ x \in \mathbb{R}^n \times \{0,1\}^m, \tag{8c}$$

$$h(x,y) \leq 0,\ y \in \mathbb{R}^l, \tag{8d}$$

$$g(x,y) \leq \eta,\ \eta = \varphi(x); \tag{8e}$$

cf., e.g., [16]. Under the additional assumptions of linear lower level functions $h(x,y) = Cx + Dy - b$ and $g(x,y) = f^\top x + e^\top y$, this so-called value function formulation exactly fits into the framework of Problem (1) if the leader's decision variable x is a scalar. This is, e.g., the case for many price setting problems modeled by bilevel models; cf. [51] for a survey. Many other applications of bilevel programming can be found in [16].

This framework can also be used to illustrate our hierarchy of assumptions. If the upper level model yields a compact feasible set for x, the optimal value function φ is globally Lipschitz continuous. In many applications on bilevel programming one can also prove (i) the uniqueness of the lower level for given upper level decisions, i.e., $\Psi(x)$ is a singleton, and (ii) the uniqueness of the dual variables of the lower level; cf., e.g., [33]. In this case, φ is even differentiable so that one can compute a global Lipschitz constant; cf. [23]. In this situation, Assumption 1 applies, and the evaluation of the subproblem corresponds to solving the lower level problem with an additional regularization term. Since the solution of the follower's problem is usually obtained by a numerical optimization algorithm, this subproblem evaluation is, in general, not exact. However, the corresponding error is usually bounded due to tolerances of the used solver. This situation is directly covered by Assumption 2. The remaining Assumptions 3 and 4 finally correspond to situations, where the optimal value function is locally Lipschitz but cannot be shown to be global Lipschitz or where global Lipschitz constants cannot be computed.

In summary, the proposed methods can be used to solve bilevel problems with mixed-integer linear upper level and convex lower level. However, additional upper level nonlinearities can also be considered if they fit into the frameworks of the mentioned assumptions.

Remark 3 Before we close this section, we finally give a brief overview about the available solvers that are able to solve the class of problems that we tackle with our methods. We give this overview with respect to the modeling language GAMS and the solvers available with GAMS. Currently, there are five solvers that are able to solve nonconvex mixed-integer nonlinear problems to global optimality: ANTIGONE [56], BARON [76,77], COUENNE [6], LINDOGLOBAL [54], and SCIP [31]. All of them require that the nonlinearities are given explicitly in algebraic form and are typically restricted to a given set of nonlinear expressions. Only a few of them allow for nonsmooth expressions (like the absolute function) that are typically reformulated using standard mixed-integer modeling techniques. The only solver included in GAMS that we are aware of that only uses Lipschitz continuity and thus allows for quite general nonsmooth constraints is LGO [63]. However, LGO is not able to deal with integer variables.

4 The case of known Lipschitz constants

In this section, we first assume that Assumption 1 holds. To construct the set Ω_i^k within the kth master problem (M(k)), we assume that we have given values $x_{i_1}^{k,j} \in \mathbb{R}$ for $j \in \{0\} \cup J_i^k = \{0\} \cup \{1, \ldots, |J_i^k|\}$ with

$$\underline{x}_{i_1} =: x_{i_1}^{k,0} < x_{i_1}^{k,1} < \cdots < x_{i_1}^{k,|J_i^k|} := \bar{x}_{i_1}.$$

Now, since we know the global Lipschitz constant L_i of f_i on $\Omega_i = [\underline{x}_{i_1}, \bar{x}_{i_1}] \times [\underline{x}_{i_2}, \bar{x}_{i_2}] \subset \mathbb{R}^2$, we can define the sets $\Omega_i^k(j)$ for $j \in J_i^k$ as

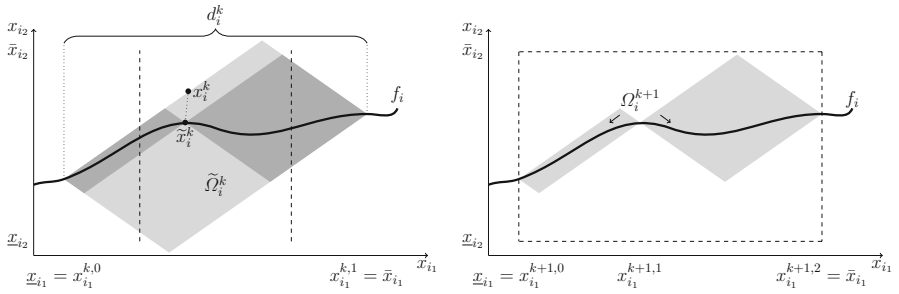

Fig. 1 Left: visualization of the subproblem (S(k)); see also Problem (11). Right: feasible set of the master problem in iteration $k+1$

$$\begin{aligned}\Omega_i^k(j) = \Big\{ (x_{i_1}, x_{i_2}) \in \mathbb{R}^2 :\ & x_{i_1}^{k,j-1} \le x_{i_1} \le x_{i_1}^{k,j}, \\ & x_{i_2} \le f_i\left(x_{i_1}^{k,j-1}\right) + L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right), \\ & x_{i_2} \ge f_i\left(x_{i_1}^{k,j-1}\right) - L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right), \\ & x_{i_2} \le f_i\left(x_{i_1}^{k,j}\right) + L_i\left(x_{i_1}^{k,j} - x_{i_1}\right), \\ & x_{i_2} \ge f_i\left(x_{i_1}^{k,j}\right) - L_i\left(x_{i_1}^{k,j} - x_{i_1}\right) \Big\}.\end{aligned} \tag{9}$$

The corresponding set $\Omega_i^k = \bigcup_{j \in J_i^k} \Omega_i^k(j)$ is depicted in Fig. 1 (right). For what follows, we abbreviate

$$\mathcal{X}_i^k := \left\{ x_{i_1}^{k,0}, x_{i_1}^{k,1}, \ldots, x_{i_1}^{k,|J_i^k|} \right\},$$

i.e., $\mathcal{X}_i^k$ is the set of sampling points that is used for the definition of Ω_i^k. In other words, the set Ω_i^k is uniquely defined by the set $\mathcal{X}_i^k$. By construction, we have the following proposition.

Proposition 1 *We have*

$$\bigcup_{j \in J_i^k} \Omega_i^k(j) \supseteq \operatorname{graph}(f_i)$$

on $[\underline{x}_{i_1}, \bar{x}_{i_1}]$ *for all* $i \in [p]$ *and all* k. *Thus, the* k*th master problem* (M(k)) *is a relaxation of* (1).

The next lemma shows that we can rely on today's effective and reliable MIP software for solving the master problems.

Lemma 1 *The master problem* (M(k)) *can be modeled as mixed-integer linear problem.*

Proof The constraints $Ax \geq b$, $\underline{x} \leq x \leq \bar{x}$, and $x \in \mathbb{R}^n \times \{0,1\}^m$ are obviously mixed-integer linear constraints. Thus, it remains to prove that $x_i \in \Omega_i^k$, $i \in [p]$, can be formulated with mixed-integer linear constraints as well. The sets Ω_i^k are finite unions of polytopes; cf. (4) and (9). Such unions of polytopes can be modeled using big-M constraints, yielding the MIP-formulation

$$\min_{x,z} \; c^\top x \tag{10a}$$

$$\text{s.t. } Ax \geq b, \quad \underline{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \{0,1\}^m, \tag{10b}$$

$$-M\left(1 - z_i^{k,j}\right) + x_{i_1}^{k,j-1} \leq x_{i_1} \leq x_{i_1}^{k,j} + M\left(1 - z_i^{k,j}\right), \quad i \in [p],\ j \in J_i^k, \tag{10c}$$

$$x_{i_2} \leq f_i\left(x_{i_1}^{k,j-1}\right) + L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right) + M\left(1 - z_i^{k,j}\right), \quad i \in [p],\ j \in J_i^k, \tag{10d}$$

$$x_{i_2} \geq f_i\left(x_{i_1}^{k,j-1}\right) - L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right) - M\left(1 - z_i^{k,j}\right), \quad i \in [p],\ j \in J_i^k, \tag{10e}$$

$$x_{i_2} \leq f_i\left(x_{i_1}^{k,j}\right) + L_i\left(x_{i_1}^{k,j} - x_{i_1}\right) + M\left(1 - z_i^{k,j}\right), \quad i \in [p],\ j \in J_i^k, \tag{10f}$$

$$x_{i_2} \geq f_i\left(x_{i_1}^{k,j}\right) - L_i\left(x_{i_1}^{k,j} - x_{i_1}\right) - M\left(1 - z_i^{k,j}\right), \quad i \in [p],\ j \in J_i^k, \tag{10g}$$

$$\sum_{j \in J_i^k} z_i^{k,j} = 1, \quad i \in [p], \tag{10h}$$

$$z_i^{k,j} \in \{0,1\}, \quad i \in [p],\ j \in J_i^k. \tag{10i}$$

for the kth master problem. The additional binary variables z in (10i) (de)activate polytopes and their corresponding linear constraints (10c)–(10g). SOS-1 constraints (10h) ensure that only one polytope is activated. □

We note that a MIP like (M(k)) can be solved in finite time, i.e., a standard MIP solver can compute a global optimal solution of (M(k)) or prove infeasibility in finite time. Due to Proposition 1, we thus know that either (M(k)) is solved to global optimality in finite time or the original problem (1) is infeasible.

Given a solution of (M(k)), the subproblem (S(k)) is used to either determine that the solution found by (M(k)) is close enough to the original feasible set, or alternatively, to provide a new point on the x_{i_1}-axis to be added in the definition of $\mathcal{X}_i^{k+1}$. To assert that the newly found points do not accumulate at an already known value the subproblem is solved on a smaller set $\widetilde{\Omega}_i^k(j) \subset \Omega_i^k(j)$. For a given $j \in J_i^k$ this set is defined as

$$\widetilde{\Omega}_i^k(j) = \Omega_i^k(j) \cap \widehat{\Omega}_i^k(j)$$

with the subsets

$$\widehat{\Omega}_i^k(j) = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid \; x_{i_1}^{k,j-1} + 0.25 d_i^{k,j} \leq x_{i_1} \leq x_{i_1}^{k,j} - 0.25 d_i^{k,j}\right\},$$

where $d_i^{k,j} = x_{i_1}^{k,j} - x_{i_1}^{k,j-1}$ is the length of the corresponding subinterval, see also Fig. 1 (left) for an illustration. Note that the constant 0.25 can be replaced by any other constant in $(0, 1/2)$.

Before we formally describe the algorithm, we first show that the subproblems can all be solved in parallel in every iteration of the algorithm provided that the index pairs $i = (i_1, i_2)$ are non-overlapping. This means that $i_1 \neq j_1$ and $i_1 \neq j_2$ for all $i \neq j$. Note that this property can always be achieved by using additional slack variables.

Lemma 2 *Assume that the index pairs $i = (i_1, i_2)$ are non-overlapping. Then, the subproblems (S(k)) are completely separable, i.e., we can solve the kth subproblem by solving p two-dimensional problems*

$$\min_{\widetilde{x}_i} \quad \|\widetilde{x}_i - x_i^k\|_2^2 \quad s.t. \quad f_i(\widetilde{x}_{i_1}) = \widetilde{x}_{i_2}, \quad \widetilde{x}_i \in \widetilde{\Omega}_i^k(j_i^k). \tag{11}$$

Proof The constraint set

$$f_i(\widetilde{x}_{i_1}) = \widetilde{x}_{i_2}, \quad \widetilde{x}_i \in \widetilde{\Omega}_i^k(j_i^k), \quad i \in [p],$$

of the subproblem (S(k)) completely decouples along $i \in [p]$. The same holds for the objective function,

$$\|\widetilde{x} - x^k\|_2^2 = \sum_{i \in [p]} \|\widetilde{x}_i - x_i^k\|_2^2.$$

Thus, the subproblem's solution $(x_i)_{i \in [p]} \in \mathbb{R}^{2p}$ is made up of the solutions $x_i \in \mathbb{R}^2$ of (11) for all $i \in [p]$. □

We are now ready to present the entire algorithm. It is formally given in Algorithm 1. Before we present the convergence analysis of Algorithm 1, we first make some explanatory comments. After the master problem has been solved (Line 3), the algorithm checks whether we already found an ε-feasible point in Line 8. Note further that the determination of the indices j_i^k in Line 11 can easily be implemented using the binary variables $z_i^{k,j}$ of the MIP formulation (10). If an ε-feasible point is not yet found, there are subproblems with feasibility violation larger than ε. For these indices i, we then refine the relaxation by adding the corresponding subproblem's solution $\widetilde{x}_{i_1}^k$ to the set $\mathcal{X}_i^k$. Thus, it is immediately clear that $|J_i^k| \leq k$ in every iteration k and all $i \in [p]$ and that the growth of the master problem per iteration in terms of variables and constraints is $\mathcal{O}(p)$.

We further remark that we do not have to solve the subproblems to global optimality. However, we think that a local or even global optimal solutions of the subproblems yield new sampling points $\widetilde{x}_{i_1}^k$ of better quality that finally lead to faster convergence

Algorithm 1 Decomposition Method

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

Ensure: Returns an approximate globally optimal and ε-feasible point for Problem (1) or indication of infeasibility.

Set $k \leftarrow 1$ and initialize $\mathcal{X}_i^k = \{\underline{x}_{i_1}, \bar{x}_{i_1}\}$ for all $i \in [p]$.
while true **do**
  Solve the master problem (M(k)) to global optimality.
  if (M(k)) is infeasible **then**
    return "Problem (1) is infeasible".
  end if
  Let x^k denote the optimal solution of (M(k)).
  if $|f_i(x_{i_1}^k) - x_{i_2}^k| \leq \varepsilon$ for all $i \in [p]$ **then**
    return x^k.
  end if
  Determine the polytopes $j_i^k \in J_i^k$ for all $i \in [p]$.
  Solve the subproblems (S(k)), respectively (11), for all $i \in [p]$ with $|f_i(x_{i_1}^k) - x_{i_2}^k| > \varepsilon$ and let $\widetilde{x}_i^k$ denote the optimal solutions.
  for $i \in [p]$ **do**
    if $|f_i(x_{i_1}^k) - x_{i_2}^k| > \varepsilon$ **then**
      Set $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k \cup \{\widetilde{x}_{i_1}^k\}$.
    else
      Set $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k$.
    end if
  end for
  Increase $k \leftarrow k + 1$.
end while

of the algorithm. Nevertheless, for the correctness of the algorithm we only require the subproblems to return a feasible point, which is always possible due to the following lemma.

Lemma 3 *All subproblems* (S(k)) *are feasible.*

Proof The claim follows immediately by Lipschitz continuity and the definition of the sets $\widetilde{\Omega}_i^k(j_i^k)$. □

We now come to the proof of convergence for Algorithm 1. The proof will be based on the fact that the volume of the feasible region decreases by at least $\delta = \varepsilon^2/(16L)$ in each iteration, unless convergence has been achieved. This elementary result will be provided in the following Lemma 4 and yields the, then easy to show, Theorem 1. In addition, the calculated constant immediately provides an upper bound on the number of iterations needed for convergence. If this information is not desired, we note that Algorithm 1 under Assumption 1 is a special case of Assumption 2 and the shorter approach given in Theorem 2 can be utilized.

For the next lemma, we introduce some more notation. As already discussed, the set $\mathcal{X}_i^k$ defines the unions of polytopes Ω_i^k. If the former set is updated in Algorithm 1, we have $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k \cup \{\widetilde{x}_{i_1}^k\}$. This update corresponds to the refinement of one quadrilateral in Ω_i^k. To be more specific, the quadrilateral j_i^k is replaced by two smaller ones that we denote by $\Omega_i^k(j_1^k)$ and $\Omega_i^k(j_2^k)$. Thus, we have

$$\Omega_i^{k+1} = \Omega_i^k(j_1^k) \bigcup \Omega_i^k(j_2^k) \bigcup_{j \neq j_i^k \in J_i^k} \Omega_i^k(j).$$

Now we can state and prove the key lemma for the convergence theorem following afterward.

Lemma 4 *There exists a constant $\delta > 0$ depending on ε and L alone, such that as long as Algorithm 1 does not terminate in Line 5 or 9, there exists $\delta^k > \delta$ for every k such that*

$$\mathrm{Vol}\left(\Omega_i^k(j_1^k)\right) + \mathrm{Vol}\left(\Omega_i^k(j_2^k)\right) = \mathrm{Vol}\left(\Omega_i^k(j_i^k)\right) - \delta^k$$

holds.

Proof A simple calculation shows, that the corners (left, right, bottom, top) of $\Omega_i^k(j_i^k)$ are given in (x_{i_1}, x_{i_2})-coordinates as

$$\begin{aligned}
\mathbf{l} &= (a,\ f_i(a))^\top,\\
\mathbf{r} &= (b,\ f_i(b))^\top,\\
\mathbf{b} &= \left(\frac{f_i(a) - f_i(b) + L(a+b)}{2L},\ \frac{f_i(a) + f_i(b) + L(a-b)}{2}\right)^\top,\\
\mathbf{t} &= \left(\frac{f_i(b) - f_i(a) + L(a+b)}{2L},\ \frac{f_i(a) + f_i(b) + L(b-a)}{2}\right)^\top,
\end{aligned}$$

where $a = x_{i_1}^{k, j_i^k - 1}$ and $b = x_{i_2}^{k, j_i^k}$, cf. Fig. 2. Consequently, the volume of $\Omega_i^k(j_i^k)$ is given as

$$\begin{aligned}
\mathrm{Vol}\left(\Omega_i^k(j_i^k)\right) &= \left|\det\left(\mathbf{t} - \mathbf{l}\ \mathbf{b} - \mathbf{l}\right)\right|\\
&= \left|L\frac{(a-b)^2}{2} - \frac{(f_i(a) - f_i(b))^2}{2L}\right|\\
&= L\frac{(a-b)^2}{2} - \frac{(f_i(a) - f_i(b))^2}{2L}.
\end{aligned}$$

The last equality holds because of Lipschitz continuity. Further, the vertical distance between top and bottom corner is

$$0 \leq \mathbf{t}_2 - \mathbf{b}_2 = \frac{L}{2}(2b - 2a) = L(b-a).$$

Since by construction $f_i(x_{i_1})$ and x_{i_2} are between these extreme values it holds

$$|f_i(x_{i_1}) - x_{i_2}| \leq L(b-a).$$

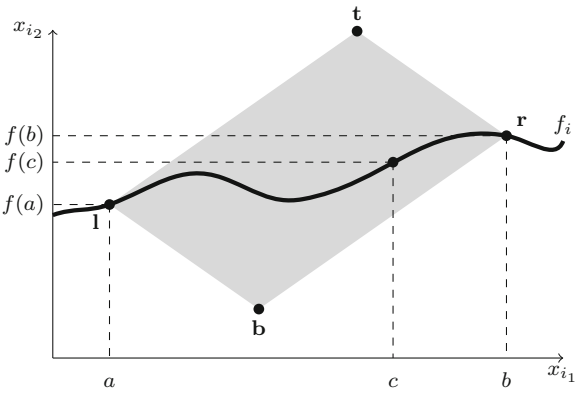

Fig. 2 Illustration of the proof of Lemma 4

By assumption, Algorithm 1 does not terminate in Line 9 and consequently,

$$\varepsilon/L < |f_i(x_{i_1}) - x_{i_2}|/L \leq |a - b|.$$

Since $\Omega_i^k(j_i^k)$ is either a line, and then $f(x_{i_1}) = x_{i_2}$, or a proper quadrilateral, the bound $\varepsilon/L < |f_i(x_{i_1}) - x_{i_2}|/L$ implies that $\Omega_i^k(j_i^k)$ is a proper quadrilateral and thus $\mathrm{Vol}(\Omega_i^k(j_i^k)) > 0$, i.e., $|f_i(a) - f_i(b)| < L|a - b|$. Moreover, the maximal x_2-distance over $\Omega_i^k(j_i^k)$ is attained over the point b (or below t) and is given as

$$-|f_i(a) - f_i(b)| + L(b - a) > \varepsilon,$$

where the last inequality is due to the fact that the condition for entering Line 9 of the algorithm is not satisfied. Thus, we obtain

$$|f_i(a) - f_i(b)| < L|a - b| - \varepsilon. \tag{12}$$

Let now $d = b - a$, then the new intersection point $c = \widetilde{x}_{i_1}^k$ satisfies

$$c \in [a + 0.25d, b - 0.25d].$$

Analogous calculations give the volumes of the new quadrilaterals as

$$\mathrm{Vol}\left(\Omega_i^k\left(j_1^k\right)\right) = L\frac{(a-c)^2}{2} - \frac{(f_i(a) - f_i(c))^2}{2L},$$
$$\mathrm{Vol}\left(\Omega_i^k(j_2^k)\right) = L\frac{(c-b)^2}{2} - \frac{(f_i(c) - f_i(b))^2}{2L}.$$

A straightforward calculation gives

$$\begin{aligned}
&\mathrm{Vol}\left(\Omega_i^k\left(j_i^k\right)\right)\\
&= L\frac{(a-c+c-b)^2}{2} - \frac{(f_i(a)-f_i(c)+f_i(c)-f_i(b))^2}{2L}\\
&= L\frac{(a-c)^2+2(a-c)(c-b)+(c-b)^2}{2}\\
&\quad - \frac{(f_i(a)-f_i(c))^2+2(f_i(a)-f_i(c))(f_i(c)-f_i(b))+(f_i(c)-f_i(b))^2}{2L}\\
&= \mathrm{Vol}\left(\Omega_i^k\left(j_1^k\right)\right)+\mathrm{Vol}\left(\Omega_i^k\left(j_2^k\right)\right)\\
&\quad + L(a-c)(c-b) - \frac{1}{L}(f_i(a)-f_i(c))(f_i(c)-f_i(b)).
\end{aligned}$$

Consequently, the definition

$$\delta^k := L(a-c)(c-b) - \frac{1}{L}(f_i(a)-f_i(c))(f_i(c)-f_i(b))$$

is given by the statement of the Lemma. It remains to show $\delta^k \geq \delta > 0$.

First, assume that $(f_i(a)-f_i(c))(f_i(c)-f_i(b)) \leq 0$. Then by our previous considerations

$$\delta^k \geq L(a-c)(c-b) \geq \frac{L}{16}d^2 = \frac{L}{16}|a-b|^2 > \frac{\varepsilon^2}{16L} =: \delta > 0.$$

holds true.

Second, otherwise the two factors $f_i(a)-f_i(c)$ and $f_i(c)-f_i(b)$ have equal sign and are not zero. Thus, either $f_i(a) < f_i(c) < f_i(b)$ or $f_i(a) > f_i(c) > f_i(b)$ holds. We assume that the first is the case, i.e., $f_i(a) < f_i(c) < f_i(b)$. The other case can be treated analogously. Now, since

$$\begin{aligned}
|f_i(a)-f_i(c)| &= f_i(c)-f_i(a) \leq L(c-a) \quad \text{and}\\
|f_i(b)-f_i(c)| &= f_i(b)-f_i(c) \leq L(b-c),
\end{aligned}$$

we can deduce from (12) that at least one of the inequalities

$$\begin{aligned}
f_i(c)-f_i(a) &\leq L(c-a) - \frac{\varepsilon}{2} \quad \text{or}\\
|f_i(b)-f_i(c)| &= f_i(b)-f_i(c) \leq L(b-c) - \frac{\varepsilon}{2}
\end{aligned}$$

is true. We assume that

$$|f_i(b)-f_i(c)| = f_i(b)-f_i(c) \leq L(b-c) - \frac{\varepsilon}{2} \tag{13}$$

holds. (The other case can again be treated analogously.) We can now estimate the term δ^k. First, by Lipschitz continuity, we obtain

$$-\frac{1}{L}(f_i(a) - f_i(c)) = \frac{1}{L}(f_i(c) - f_i(a)) \leq c - a.$$

Then, noting that $f_i(c) - f_i(b) < 0$, we assert

$$\begin{aligned}\delta^k &\geq L(a-c)(c-b) + (c-a)(f_i(c) - f_i(b)) \\ &= (c-a)\left(L(b-c) + f_i(c) - f_i(b)\right).\end{aligned}$$

By the sharpened estimate (13), the second factor is bounded from below by $\varepsilon/2$ and the first factor by $d/4$, and we conclude

$$\delta^k \geq \frac{d\varepsilon}{8} > \frac{\varepsilon^2}{8L} > \frac{\varepsilon^2}{16L} = \delta > 0. \qquad \square$$

We are now ready to prove the main convergence theorem for Algorithm 1.

Theorem 1 *There exists a $K < \infty$ such that Algorithm 1 either terminates with an approximate globally optimal point x^k or with the indication of infeasibility in an iteration $k \leq K$.*

Proof Assume that the algorithm does not terminate after a finite number of iterations. Then, there exists a sequence $(x^k)_k$ such that all master problems (M(k)) are feasible and such that there exists at least one $i \in [p]$ with $|f_i(x^k_{i_1}) - x^k_{i_2}| > \varepsilon$ for all k. Since there are only finitely many possible values in $[p]$ there is at least one $i \in [p]$ for which $|f_i(x^k_{i_1}) - x^k_{i_2}| > \varepsilon$ infinitely many times. We denote the corresponding subsequence by k_i. By Lemma 4, we know that $\mathrm{Vol}(\Omega_i^{k_i}) = \mathrm{Vol}(\Omega_i^{k_i-1}) - \delta^{k_i}$ with $\delta^{k_i} > \delta$ for some $\delta > 0$. As a consequence for the selected i it is

$$0 \leq \mathrm{Vol}\left(\Omega_i^{k_i}\right) \to -\infty$$

contradicting our assumption that there are infinitely many iterations. $\square$

Note that, in contrast to many other approaches for global mixed-integer nonlinear optimization, our algorithm does not rely on cutting off the visited ε-infeasible points from the feasible set for all subsequent iterations. Moreover, Lemma 4 also allows, in principle, for an analysis of the speed of convergence of Algorithm 1.

4.1 Inexact evaluation of nonlinear functions

In case of Assumption 2, in which only inexact evaluations of f are available, it is easy to modify the decomposition method of Algorithm 1. In order to modify the algorithm,

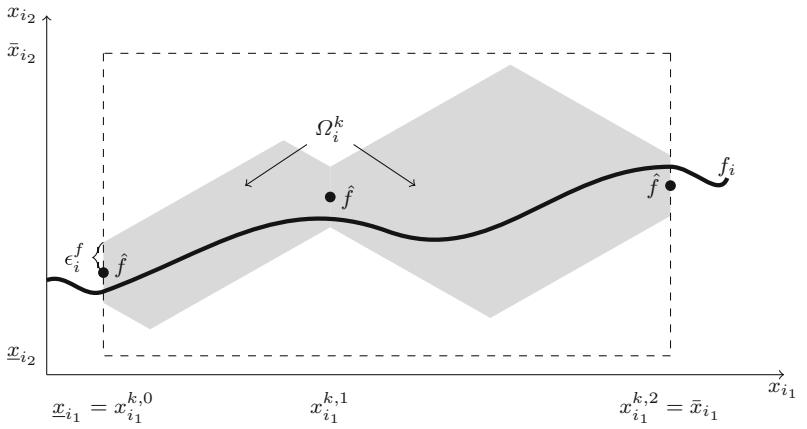

Fig. 3 Visualization of the feasible set of the master problem for a single $i \in [p]$ with $|J_i^k| = 2$ and inexact function evaluation

we need to assert $\Omega_i^k \supseteq \text{graph } f_i$. Given that $\varepsilon_i^f(x_{i_1})$ is known this can be done by the following modification of the set $\Omega_i^k(j)$:

$$\begin{aligned}\Omega_i^k(j) = \Big\{ & (x_{i_1}, x_{i_2}) \in \mathbb{R}^2 : x_{i_1}^{k,j-1} \le x_{i_1} \le x_{i_1}^{k,j}, \\ & x_{i_2} \le \hat{f}_i\left(x_{i_1}^{k,j-1}\right) + \varepsilon_i^f\left(x_{i_1}^{k,j-1}\right) + L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right), \\ & x_{i_2} \ge \hat{f}_i\left(x_{i_1}^{k,j-1}\right) - \varepsilon_i^f\left(x_{i_1}^{k,j-1}\right) - L_i\left(x_{i_1} - x_{i_1}^{k,j-1}\right), \\ & x_{i_2} \le \hat{f}_i\left(x_{i_1}^{k,j}\right) + \varepsilon_i^f\left(x_{i_1}^{k,j}\right) + L_i(x_{i_1}^{k,j} - x_{i_1}), \\ & x_{i_2} \ge \hat{f}_i\left(x_{i_1}^{k,j}\right) - \varepsilon_i^f\left(x_{i_1}^{k,j}\right) - L_i\left(x_{i_1}^{k,j} - x_{i_1}\right)\Big\};\end{aligned} \tag{14}$$

see Fig. 3. Moreover, we have to replace the exact function evaluation $f_i(x_{i_1})$ by its approximation $\hat{f}_i(x_{i_1})$ in Line 8, 12, and 14 of Algorithm 1. Finally, the ε-checks "$|f_i(x_{i_1}^k) - x_{i_2}^k| \le \varepsilon$" in the same lines have to be replaced by "$|\hat{f}_i(x_{i_1}^k) - x_{i_2}^k| \le \varepsilon - \varepsilon_i^f$"; of course assuming that $\varepsilon - \varepsilon_i^f > 0$.

With these minor changes, Algorithm 1 can also be applied in case of Assumption 2. However, the analysis is not identical. This is due to the fact that a volume reduction as in Lemma 4 no longer needs to happen because each of the domains $\Omega_i^k(j)$ contain the quadrilaterals

$$\text{conv}\left\{\left(x_{i_1}^{k,j-1}, f_i\left(x_{i_1}^{k,j-1}\right) \pm \varepsilon_i^f\left(x_{i_1}^{k,j-1}\right)\right), \left(x_{i_1}^{k,j}, f_i\left(x_{i_1}^{k,j}\right) \pm \varepsilon_i^f\left(x_{i_1}^{k,j}\right)\right)\right\}.$$

The volume of these quadrilaterals is not reduced by subdivisions, if the two volumes of the resulting quadrilaterals are summed. As a consequence, we cannot rely on a volume reduction for the convergence proof. Nonetheless, convergence of the algorithm is still true provided that ε is chosen large enough compared to ε_i^f.

Theorem 2 *Suppose that Assumption 2 holds and that Algorithm 1 is applied with the modified sets Ω_i^k defined in (14) and*

$$\varepsilon > \varepsilon^f := 2 \max_{i \in [p]} \sup_{x \in [\underline{x}_{i_1}, \bar{x}_{i_1}]} \varepsilon_i^f(x).$$

Then there exists a $K < \infty$ such that Algorithm 1 either terminates with an approximate globally optimal point x^k or with the indication of infeasibility in an iteration $k \leq K$.

Proof It is clear, that by Line 15 of Algorithm 1 a refined interval is split into two new subintervals. By construction of $\widetilde{x}^k$ each of these subintervals has a length of at most 3/4 of the original interval.

On the other hand, the maximal vertical difference on an interval $(x_{i_1}^{k,j-1}, x_{i_1}^{k,j})$, for any $j = 1, \ldots, |J_i^k|$ and $i \in [p]$, is bounded by

$$\varepsilon^f + L_i d_i^{k,j} \geq 2 \max \left\{ \varepsilon_i^f \left(x_{i_1}^{k,j-1} \right), \varepsilon_i^f \left(x_{i_1}^{k,j} \right) \right\} + L_i \left(x_{i_1}^{k,j} - x_{i_1}^{k,j-1} \right).$$

As a consequence, if $d_i^{k,j} \leq (\varepsilon - \varepsilon^f)/L_i$ it necessarily holds

$$\left| f_i \left(x_{i_1}^k \right) - x_{i_2}^k \right| \leq \varepsilon^f + L_i d_i^{k,j} \leq \varepsilon$$

and thus an interval with $d_i^{k,j} \leq (\varepsilon - \varepsilon^f)/L_i$ will never be refined. Since there are only finitely many iterations possible before all $d_i^{k,j}$ are smaller than this bound the algorithm terminates after $K < \infty$ iterations. □

5 The case of unknown Lipschitz constants

We now consider the more realistic case in which the functions are globally Lipschitz, but knowledge of the global Lipschitz constant is not available. In order to proceed algorithmically, we assume that Assumption 3 is satisfied, i.e., for a given point x_{i_1}, we can obtain a local Lipschitz constant $L_i(x_{i_1})$ of the corresponding interval, i.e., f_i is Lipschitz continuous on

$$\Theta_i(x_{i_1}) = \left[x_{i_1} - \varepsilon_i^L, x_{i_1} + \varepsilon_i^L \right]$$

with constant $L_i(x_{i_1})$, but we do not know ε_i^L. This assumption is indeed meaningful. For instance, if $f_i \in C^2$ and f_i' can be evaluated at a given point x_{i_1}, then

$$L_i(x_{i_1}) = 2 \left| f_i'(x_{i_1}) \right| + 1 \tag{15}$$

is a correct local Lipschitz estimate for which we used the factor 2 to overestimate the local change rate of the function and where the offset 1 is used to handle the situation

$f_i'(x_{i_1}) = 0$. We proceed, similarly to Sect. 4, by successively creating a subdivision of Ω_i^k using the sampling points

$$\mathcal{X}_i^k = \left\{ x_{i_1}^{k,0}, x_{i_1}^{k,1}, \ldots, x_{i_1}^{k,|J_i^k|} \right\}$$

with

$$\underline{x}_{i_1} =: x_{i_1}^{k,0} < x_{i_1}^{k,1} < \cdots < x_{i_1}^{k,|J_i^k|} := \bar{x}_{i_1}$$

and abbreviate

$$h_i^k := \max_{j=1,\ldots,|J_i^k|} \left\{ x_{i_1}^{k,j} - x_{i_1}^{k,j-1} \right\}.$$

For this subdivision, we can compute an estimate for the global Lipschitz constant via

$$\widetilde{L}_i = \max \left\{ \max_{j=0,\ldots,|J_i^k|} L_i \left(x_{i_1}^{k,j} \right), \max_{j=1,\ldots,|J_i^k|} \frac{\left| f_i \left(x_{i_1}^{k,j} \right) - f_i \left(x_{i_1}^{k,j-1} \right) \right|}{x_{i_1}^{k,j} - x_{i_1}^{k,j-1}} \right\}. \quad (16)$$

With this, we now define the sets Ω_i^k for the kth master problem (M(k)) in analogy to (9) by

$$\begin{aligned} \Omega_i^k(j) = \Big\{ (x_{i_1}, x_{i_2}) \in \mathbb{R}^2 : \; & x_{i_1}^{k,j-1} \leq x_{i_1} \leq x_{i_1}^{k,j}, \\ & x_{i_2} \leq f_i \left(x_{i_1}^{k,j-1} \right) + \widetilde{L}_i \left(x_{i_1} - x_{i_1}^{k,j-1} \right), \\ & x_{i_2} \geq f_i \left(x_{i_1}^{k,j-1} \right) - \widetilde{L}_i \left(x_{i_1} - x_{i_1}^{k,j-1} \right), \\ & x_{i_2} \leq f_i \left(x_{i_1}^{k,j} \right) + \widetilde{L}_i \left(x_{i_1}^{k,j} - x_{i_1} \right), \\ & x_{i_2} \geq f_i \left(x_{i_1}^{k,j} \right) - \widetilde{L}_i \left(x_{i_1}^{k,j} - x_{i_1} \right) \Big\} \end{aligned}$$

and set, as before,

$$\Omega_i^k = \bigcup_{j \in J_i^k} \Omega_i^k(j).$$

Although the construction is analogous to (9) the sets $\Omega_i^k(j)$ constructed here no longer give an outer approximation of the graph of f_i, i.e., the analogue of Proposition 1 is no longer true. However, clearly, Lemma 1 remains valid. For the definition of the subproblems w.r.t. a given $j \in J_i^k$, we define the sets

$$\widetilde{\Omega}_i^k(j) = \widehat{\Omega}_i^k(j)$$

with the subsets

$$\widehat{\Omega}_i^k(j) = \left\{(x_1, x_2) \in \mathbb{R}^2 : x_{i_1}^{k,j-1} + 0.25 d_i^{k,j} \leq x_{i_1} \leq x_{i_1}^{k,j} - 0.25 d_i^{k,j}\right\}. \quad (17)$$

Notice that, in contrast to Sect. 4, we do not include the bounds on the x_{i_2}-variable in the latter definition, i.e., we do not consider the intersection with Ω_i^k. This is due to the fact that since $\widetilde{L}_i$ is not necessarily the true Lipschitz constant, it can happen that $\text{graph}(f_i) \cap \widehat{\Omega}_i^k(j) \cap \Omega_i^k(j) = \emptyset$ for some $j \in J_i^k$ and, consequently, the subproblems (S(k)) on the subdomain $\widehat{\Omega}_i^k(j) \cap \Omega_i^k(j)$ could be infeasible.

For the same reason, $\text{graph}(f_i) \not\subseteq \Omega_i^k$, it can happen that the master problem (M(k)) is infeasible if the estimate $\widetilde{L}_i$ is not good enough, although (1) has feasible points. Thus, infeasibility of (M(k)) can be due to two reasons, either $\widetilde{L}_i$ is too small, or because (1) is truly infeasible. In order to obtain a better estimate for $\widetilde{L}_i$, it is useful to add additional points to the decomposition $\mathcal{X}_i^k$ at which $\widetilde{L}_i$ is sampled. Unfortunately, just continuously refining $\mathcal{X}_i^k$ does not terminate if (1) is infeasible. Since we cannot decide whether this is the case, we must stop if a maximal fineness of the sampling has been reached indicating that either the Lipschitz constant is varying very rapidly or that (1) is indeed infeasible. If this situation is encountered, we call the problem *potentially infeasible* in the following.

The extended method for only approximately known Lipschitz constants is stated in Algorithm 2. For the above depicted refinement of $\mathcal{X}_i^k$ in case of infeasibility of the master problem, we add a minimal refinement that yields finite termination of the algorithm; cf. Line 5. However, since solving (M(k)) can potentially be expensive, more aggressive strategies, e.g., refinement of all subintervals exceeding length μ may be advantageous from a computational point of view.

Theorem 3 *Let Assumption 3 be given, then Algorithm 2 terminates after finitely many iterations.*

Proof By construction, if the master problem is infeasible, we either terminate or the largest subinterval is halved in size. Since all variables are bounded below and above, the latter can occur only finitely many times. As this needs one subinterval to be larger than μ, infeasibility of (M(k)) can only occur finitely many times before termination.

In the other case, we solve the subproblems. However, the Lipschitz constant is only updated if the subproblem solution $\widetilde{x}_i^k \notin \Omega_i^k$ and thereby proves that our Lipschitz constant estimate was wrong. This can happen only if the corresponding subinterval is larger than the assumed lower bound on the size of $\Theta_i(\widetilde{x}_{i_1})$; cf. (5). If such a subinterval is selected its length is reduced, at least, by a factor $3/4$. Hence such an update can happen only finitely many times until the length of the subinterval reaches the (unknown) lower bound; and thus the estimate of the Lipschitz constant coincides with the Lipschitz constant. If no updates to the Lipschitz constant are made the algorithm coincides with Algorithm 1 and thus Theorem 1 shows the assertion. □

Next, we briefly discuss the meaning of the results of Algorithm 2 and potential computational improvements. First, we consider the case that Algorithm 2 terminates in Line 9 returning the indication that the problem is potentially infeasible.

Algorithm 2 Infeasible Decomposition Method

Require: Problem (1) and $\varepsilon, \mu > 0$.
Ensure: Returns an ε-feasible point for Problem (1), or an indication of potential infeasibility.

Set $k \leftarrow 1$
while true **do**
  Solve the master problem (M(k)) to global optimality.
  if (M(k)) is infeasible **then**
    if $\max_{i\in[p]} h_i^k > \mu$ **then**
      Choose an arbitrary $i' \in \arg\max_{i\in[p]} h_i^k$ and compute $\mathcal{X}_{i'}^{k+1}$ by refining the largest subinterval in the set $\mathcal{X}_{i'}^k$ (via bisection).
      Update the Lipschitz constant estimate by (16).
    else
      return "Problem (1) is potentially infeasible".
    end if
  else
    Let x^k denote the optimal solution of (M(k)).
    if $|f_i(x_{i_1}^k) - x_{i_2}^k| \leq \varepsilon$ for all $i \in [p]$ **then**
      return x^k.
    end if
    Determine the polytopes $j_i^k \in J_i^k$ for all $i \in [p]$.
    Solve the subproblems (S(k)), (11) respectively, with $\widetilde{\Omega}_i^k(j)$ as defined in (17) for all $i \in [p]$ with $|f_i(x_{i_1}^k) - x_{i_2}^k| > \varepsilon$ and let $\widetilde{x}_i^k$ denote the optimal solutions.
    for $i \in [p]$ **do**
      if $|f_i(x_{i_1}^k) - x_{i_2}^k| > \varepsilon$ and $\widetilde{x}_i^k \notin \Omega_i^k$ **then**
        Set $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k \cup \{\widetilde{x}_{i_1}^k\}$.
        Update the Lipschitz constant estimate by (16).
      else if $|f_i(x_{i_1}^k) - x_{i_2}^k| > \varepsilon$ and $\widetilde{x}_i^k \in \Omega_i^k$ **then**
        Set $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k \cup \{\widetilde{x}_{i_1}^k\}$.
      else
        Set $\mathcal{X}_i^{k+1} \leftarrow \mathcal{X}_i^k$.
      end if
    end for
  end if
  Increase $k \leftarrow k + 1$.
end while

Corollary 1 *Suppose that Algorithm 2 terminates with an indication that the problem is potentially infeasible. If $\mu < \varepsilon_i^L$ for all $i \in [p]$, then Problem (1) is infeasible.*

Proof By definition of the if/else decision the only case when Algorithm 2 terminates with the result "potentially infeasible" is when all subintervals have length less than μ. By definition, the Lipschitz estimates $\widetilde{L}_i$ are valid on an interval of diameter $2\varepsilon_i^L$ around the evaluation points. Thus, $L_i \leq \widetilde{L}_i$ and graph $f_i \subset \Omega_i^k$. □

Remark 4 Finally, we would like to note that the difficulty in the above approach is that an unknown Lipschitz constant hampers the possibility to obtain an outer approximation of graph(f_i) and that updates of $\widetilde{L}_i$, in particular due to infeasibility of (M(k)), are quite costly. Hence it might be useful to consider a preparatory phase in which L_i is sampled over $[\underline{x}_{i_1}, \bar{x}_{i_1}]$ to get a good initial guess. This is in particular true as sampling is much cheaper than solving (M(k)). Since, the Lipschitz estimate still might be

proven wrong in the course of the algorithm it is still necessary to consider Algorithm 2 and not Algorithm 1. Note that these sampling points need not be included in the set $\mathcal{X}_i^0$ as long as Line 7 and 21 assert that $\widetilde{L}_i$ is monotone non-decreasing.

Remark 5 To verify potential infeasibility or even infeasibility with Corollary 1, possibly many additional refinements are necessary until the maximum of all h_i^k is equal or smaller than μ. This is to be expected as it is shown in [57,79] that optimizing a single Lipschitz continuous function over the unit cube needs $(L/2\varepsilon)^n$ function evaluations in the worst case.

Remark 6 The case of Assumption 4 is easily addressed. We can simply modify $\Omega_i^k(j)$ in a similar way as in Sect. 4.1 and consider

$$\widetilde{L}_i = \max_{j=0,\ldots,|J_i^k|} \left\{ L\left(x_{i_1}^k\right) + \delta_i^L \right\}$$

for computing the Lipschitz constant estimates. The convergence analysis of Theorem 3 remains the same with the obvious changes as in Theorem 2.

6 Computational study

In this section, we present and discuss computational results for Algorithm 1 and Algorithm 2. First, we discuss Algorithm 1 for a detailed mixed-integer nonlinear model of steady-state gas flow in Sect. 6.1. Afterward, we show the applicability of Algorithm 2 on the basis of an exemplary case study in Sect. 6.2.

6.1 Algorithm 1: stationary gas flow in networks

The optimization of gas transport networks is currently a highly active field of research of applied optimization. For an overview of the literature see the recent book [47] and the survey article [67], the references therein, and, e.g., the recent papers [28,29,58]. Most parts of the presentation of the model in this section is motivated by the model studied in [36].

One of the main tasks in gas transport is to transport a so-called nomination—prescribed supply and discharge flows together with additional restrictions like bounds for gas pressures etc.—at minimum costs. Gas mainly flows from higher to lower pressures. Thus, in order to transport gas over large distances through pipeline systems it is required to increase the gas pressure. This is done by compressors that can be, among other network devices, controlled by the dispatcher and thus add discrete aspects to the problem. In combination with nonlinear gas physics, entire gas transport models are mixed-integer nonlinear problems governed by differential equations for modeling gas physics on a graph.

Here, we consider the stationary case, present a mixed-integer model with ODEs, and show how it can be tackled with the algorithm presented above. 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 from the network, and the set of inner nodes V_0. The set of arcs consist of pipes A_{pi}, control valves A_{cv}, and compressor machines A_{cm}, which all are described in detail below.

Stationary gas flow in networks is mainly described by mass flow q and the three gas state quantities pressure p, temperature T, and density ρ. These state quantities are coupled by an equation of state. We choose the thermodynamical standard equation for real gases $\rho R_{\mathrm{s}} z(p) T = p$, where R_{s} is the specific gas constant and z is the compressibility factor that we model using the formula $z(p) = 1 + \alpha p$ of the American Gas Association [48]. We further assume isothermal gas flow, i.e., the gas temperature T is fixed at a suitable constant value. We associate positive gas flow on arcs $a = (u, v)$ with 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. The sets $\delta^{\mathrm{in}}(u) := \{a \in A : a = (v, u)\}$ and $\delta^{\mathrm{out}}(u) := \{a \in A : a = (u, v)\}$ are the sets of in- and outgoing arcs for node $u \in V$.

For each node $u \in V$, we assume lower and upper values $\underline{p}_u$ and $\bar{p}_u$ to be given that restrict the corresponding pressure variable p_u, i.e.,

$$p_u \in [\underline{p}_u, \bar{p}_u] \quad \text{for all } u \in V. \tag{18}$$

In addition, we model mass conservation by

$$\sum_{a \in \delta^{\mathrm{out}}(u)} q_a - \sum_{a \in \delta^{\mathrm{in}}(u)} q_a = q_u \begin{cases} \geq 0, & u \in V_+, \\ \leq 0, & u \in V_-, \\ = 0, & u \in V_0, \end{cases} \quad \text{for all } u \in V. \tag{19}$$

For every arc $a \in A$ there is a mass flow variable q_a that is bounded from below and above, i.e.,

$$q_a \in [\underline{q}_a, \bar{q}_a] \quad \text{for all } a \in A. \tag{20}$$

The remaining model of an arc depends on the specific type of the arc.

Pipes $a \in A_{\mathrm{pi}}$ are used to transport the gas through the network. They typically outnumber all other network elements. A pipe is specified by its length L_a, its diameter D_a, and its friction factor λ_a, which we model using the empirical formula of Prandtl–Colebrook; see, e.g., [12] or [69, Chap. 9]. We assume each pipe to be cylindrically shaped and horizontal. In this situation, isothermal gas flow through a pipe is described by a system of partial differential equations—the Euler equations for compressible fluids [20]—consisting of the continuity and the momentum equation that form a quasilinear system of hyperbolic balance laws. In what follows, we only consider the stationary case in which the continuity equation asserts constant mass flow for every pipe, justifying the choice for a single flow variable q_a for all arcs $a \in A$. Thus, we are left with the stationary variant

$$\partial_x \left(p_a + \frac{\tilde{q}_a^2}{\rho_a} \right) = -\frac{1}{2} \theta_a \frac{\tilde{q}_a |\tilde{q}_a|}{\rho_a}, \quad \tilde{q}_a = q_a / A_a, \quad \theta_a = \frac{\lambda_a}{D_a}, \quad a \in A_{\mathrm{pi}}, \tag{21}$$

of the momentum equation, coupling density $\rho_a = \rho_a(x)$ and pressure $p_a = p_a(x)$ with mass flow q_a along the arc, i.e., $x \in [0, L_a]$. The momentum equation (21) describes the pressure loss in a pipe due to ram pressure and frictional forces. Note that the density ρ in (21) can be eliminated using the equation of state. The coupling of the pressure solution of (21) with node pressure variables of (18) is given by

$$p_u = p_a(0), \quad p_v = p_a(L_a) \quad \text{for all } a = (u, v) \in A_{\text{pi}}. \tag{22}$$

Control valves $a \in A_{\text{cv}}$ are used to decrease gas pressure. This is mainly required at transition points between large transport pipelines and regional substructures that are not able to handle high pressure levels. These elements involve discrete aspects since they can be operated in different modes: They can be active, in bypass mode, or closed. Closed control valves simply block the gas flow ($q_a = 0$) and thus decouple the in- and outflow pressure. If they are open, control valves can operate in bypass mode, yielding equal pressures $p_u = p_v$. Finally, if activated, control valves are able to decrease the inflow pressure by a controllable amount $\Delta_a \in [\underline{\Delta}_a, \bar{\Delta}_a]$. In summary, the complete description reads

$$\begin{aligned} a \text{ is active} &\implies p_v = p_u - \Delta_a, \quad \Delta_a \in [\underline{\Delta}_a, \bar{\Delta}_a], \\ a \text{ is in bypass mode} &\implies p_v = p_u, \\ a \text{ is closed} &\implies q_a = 0. \end{aligned} \tag{23}$$

More detailed information about control valves and specific MIP models can be found in [26].

Finally, we describe our model of compressor machines $a \in A_{\text{cm}}$. They are used to increase the inflow gas pressure to a higher outflow pressure in order to transport gas over large distances. In general, a compressor machine can be in the same three modes as control valves, cf. (23). However, the active state is much more complicated. We only consider so-called turbo compressors that are typically modeled by characteristic diagrams; cf., e.g., [71,72] for a detailed description of turbo compressor models. It turns out that the model of a turbo compressor is highly nonlinear and nonconvex. Since our focus here does not lie on detailed compressor modeling, we use known mixed-integer linear outer approximations

$$c_a(p_u, p_v, q_a, P_a, y_a) \geq 0 \quad \text{for all } a = (u, v) \in A_{\text{cm}} \tag{24}$$

of the operating ranges. In (24), P_a stands for the power required for compression and the variables y_a are additional auxiliary variables required to formulate the specific outer approximation model; see [26] for the details.

We now collect all component models and obtain the entire optimization problem

$$\min \sum_{a \in A_{\text{cm}}} P_a \quad \text{s.t.} \quad (18)-(24).$$

This model is a nonconvex mixed-integer optimization problem that contains the ODE (21) for every pipe. Typically, the ODE is discretized or replaced by an approx-

imation of its solution in order to obtain a finite-dimensional problem; cf. [71–73] and the references therein. We, however, follow a different approach motivated by [36] and show that the model can be written as an optimization problem of type (1) with nonlinearities f_i that arise due to the ODE solutions on the pipes of the network. In our setting, the functions f_i correspond to the relation between in- and outflow pressures of the pipes. Under the assumption of subsonic flow, i.e., for the squared mach number η satisfying

$$\eta_a(x) := \frac{\tilde{q}_a^2 \hat{T}}{p_a(x)^2} < 1, \quad x \in [0, L_a], \quad \hat{T} := R_s T, \tag{25}$$

and a positive compressibility factor

$$z(p_a(x)) > 0 \quad \text{for all } x \in [0, L_a],$$

it is shown in [36] that the pressure loss along pipe $a = (u, v) \in A_{\text{pi}}$ for given inflow pressure $p_u = p_a(0)$ and mass flow $\tilde{q}_a$ reads

$$p_a(x; p_u, \tilde{q}_a) = F_a^{-1}\left(F_a(p_u) - \frac{1}{2}\hat{T}\tilde{q}_a|\tilde{q}_a|\theta_a x\right), \quad x \in [0, L_a], \tag{26}$$

where

$$F_a(p_a) := \frac{1}{\alpha}p_a + \left(\tilde{q}_a^2\hat{T} - \frac{1}{\alpha^2}\right)\ln(|1 + \alpha p_a|) - \tilde{q}_a^2\hat{T}\ln(p_a)$$

holds; cf. [37] for the case of real gas and [35] for the case of ideal gas. Finally note that (26) is a function of type (1c) if the flow $\tilde{q}_a$ is known.

6.1.1 Results

Our real-world test instance is the Greek natural gas transport network that is made up of 134 nodes (3 entries, 45 exits, 86 inner nodes) and 133 arcs (86 pipes, 45 short pipes[1], 1 control valve, 1 compressor). We note that the network is a tree. Hence, all gas flows are known a-priorily and (26) fits into our framework. The Greek natural gas transmission system operator DESFA provides nomination data on its website [13]. After discarding imbalanced nominations, the data set ranging from 11/01/2011 to 02/17/2016 yields daily instances for 1234 days. We manually double all nominated flows to increase the overall nonlinearity and hardness of the instances. We remark that all network and nomination data of this instance are publicly available as the GasLib-134 instance; see [70]. Detailed information about the Greek compressor are not publicly available. Hence we use the data of the compressor `compressor_1` of the publicly available instance GasLib-135.

Algorithms 1 and 2 iteratively solve a MIP and multiple NLP models. We implemented the algorithms using the C++ software framework LaMaTTO++ [50] and solve

[1] Short pipes are pipes with very short length such that the pressure loss is negligible.

Table 1 Overview of the results of Algorithm 1 grouped by infeasible instances ("inf."), optimal instances with deactivated compressor ("opt."), and optimal instances with activated compressor ("opt. (compr.)")

ε^f	Status	#	∅ k	∅ Total	∅ Master	∅ Sub	∅ Sub (Id.)
0	inf.	603	1.60	0.38	0.01	0.36	0.05
	opt.	151	9.30	2.14	0.40	1.74	0.31
	opt. (compr.)	480	17.75	4.13	1.10	3.02	0.62
0.001	inf.	597	1.61	0.37	0.01	0.35	0.04
	opt.	155	10.66	2.54	0.56	1.98	0.43
	opt. (compr.)	482	19.60	4.79	1.43	3.36	0.69
0.002	inf.	593	1.61	0.38	0.01	0.36	0.04
	opt.	159	13.26	3.22	0.91	2.31	0.53
	opt. (compr.)	482	23.06	7.41	3.35	4.07	0.96

Number of instances ("#") and averages of iterations ("∅ k"), total runtimes ("∅ Total"), master problem runtimes ("∅ Master"), subproblem runtimes ("∅ Sub"), and subproblem runtimes in case of parallelization ("∅ Sub (Id.)"). All runtimes are given in seconds

the MIPs with Gurobi 6.5.0 [34] and the NLPs with Ipopt 24.1.3 [84]. All computations have been performed on an Intel© Core™i5-3360M CPU with 4 cores of 2.8 GHz each and 4 GB RAM. We choose $\varepsilon = 1 \times 10^{-2}$ (scaled to denote error in bar for the pressure loss on pipes) as the tolerance of both Algorithms 1 and 2. Moreover, we test $\varepsilon^f = 0$ (yielding Assumption 1) and $\varepsilon^f = 1 \times 10^{-3}$ as well as $\varepsilon^f = 2 \times 10^{-3}$ (both yielding Assumption 2). Obviously, larger ε^f or smaller ε increase the expected number of required iterations. Before we discuss the numerical results let us finally note that we, for the ease of implementation, did not implement that all subproblems are solved in parallel.

First, Table 1 gives a basic overview of the results in terms of average numbers. All instances have been solved by our algorithm: In dependence of the parameter ε^f, approximately 48–49% of the instances are proven to be infeasible ("inf."). On the other hand, $\sim$ 12–13% of the instances are solved to global optimality with deactivated compressor ("opt.") and $\sim$ 39% with activated compressor ("opt. (compr.)"). It can be seen that infeasibility is detected with significantly less iterations than global optimality is proven. In particular, many infeasible instances are shown to be infeasible by the master problem of the first iteration, which indicates that the reason for infeasibility is solely located in the linear part of the model. The iteration numbers for computing global optimal solutions in case of feasibility differ depending on whether the compressor is activated or deactivated ($\sim$ 9–13 vs. $\sim$ 18–23 iterations respectively). Iteration numbers directly translate to runtimes that are also given in Table 1. "∅ Total" denotes the overall runtime of the algorithm and "∅ Master" and "∅ Sub" denote the average runtimes of the master and subproblems. Lastly, "∅ Sub (Id.)" stands for the artificial runtimes for the subproblems if they would have been solved in parallel for every iteration.

Next, we discuss the complexity of the method, which is determined by the complexity of the master and subproblems. The amount of subproblem NLPs that have to be solved in every iteration is bounded above by the number of pipes in the network.

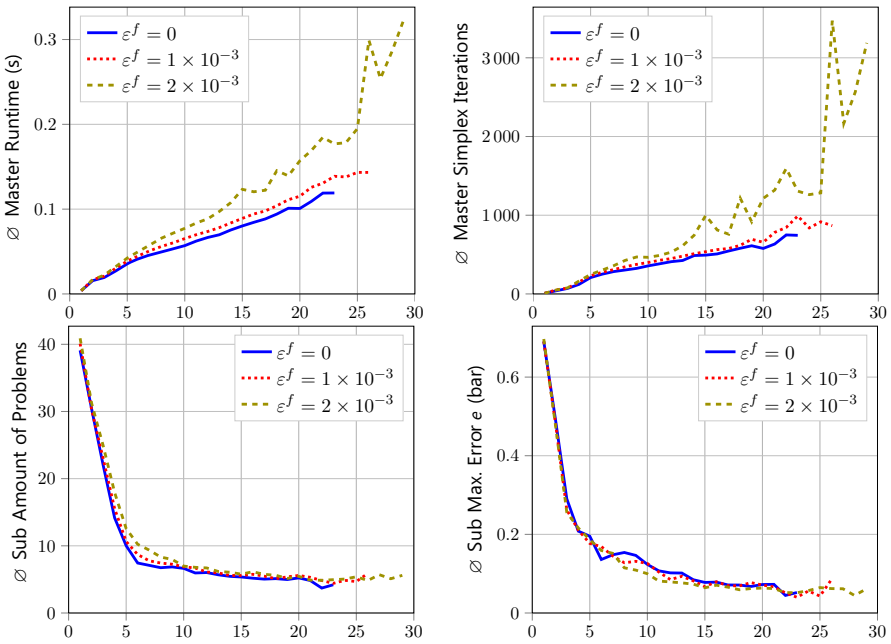

Fig. 4 Top: average runtimes (left) and simplex iterations (right) required to solve the master problem MIPs for Algorithm 1. The average is only taken for iterations (x-axis) for which data of more than 100 instances exist. Bottom: average number of solved subproblems (left) and maximal master problem error w.r.t. the nonlinearities for Algorithm 1. Again, the average is only taken for iterations (x-axis) for which data of more than 100 instances exist

The size of these NLPs in terms of variables and constraints stays constant over the course of the iterations. Since all subproblems can be solved in parallel (cf. Lemma 2), the computational effort for solving all subproblems in every iteration of the overall algorithm is, in principle, given by a single subproblem NLP. The change of complexity of the MIPs of the master problem over the course of the iterations is different. As additional linearization points yield additional binary variables and constraints, the size of the MIPs grows linearly over the course of the iterations. Consequently, the growing master problem runtimes and required simplex iterations grow as it can be seen in Fig. 4 (top). The linear growth in MIP model size translates into a worst-case exponential growth of runtimes, which cannot be avoided when using state-of-the-art MIP solver technology for the master problems. Fortunately, Fig. 4 (top) shows a more linear than an exponential behavior, which is most probably explained by the overall small number of required iterations. However, exponential behavior is expected for instances that require more iterations.

The number of subproblems that have to be solved and thus the related additional linearization points decrease over the course of the iterations as it can be seen in Fig. 4 (bottom left). The same holds for the maximal error

$$e^k := \max_{i \in [p]} \left| f_i\left(x_{i_1}^k\right) - x_{i_2}^k \right|$$

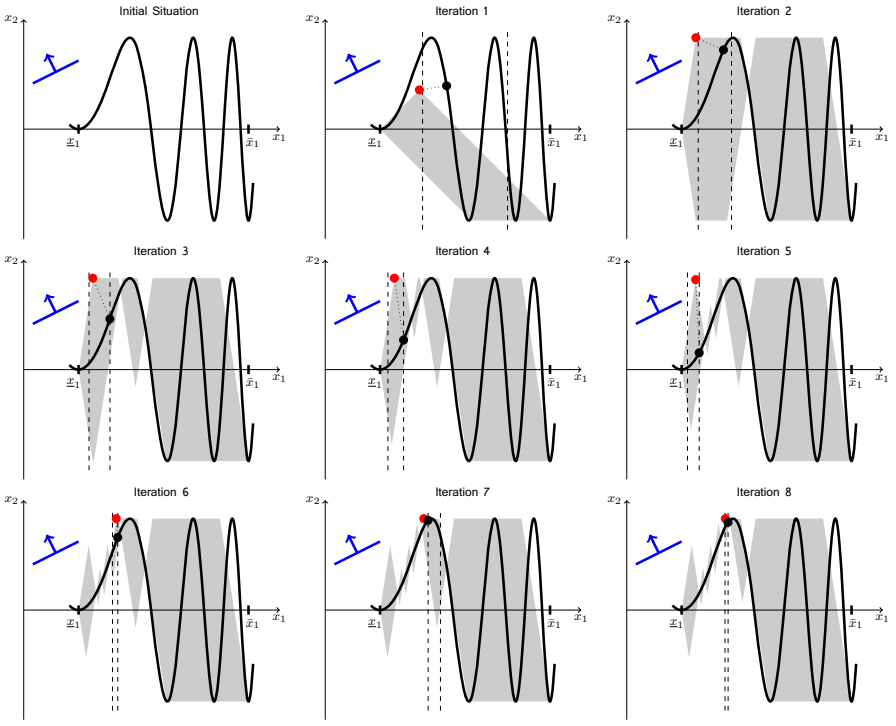

Fig. 5 Graphical visualization of Algorithm 2 solving (27) with quadrilaterals (gray), optimization direction (blue), master problem solution (red), subproblem solution (black) (color figure online)

of the master problem's solutions with respect to the nonlinearities f_i; see Fig. 4 (bottom right) for an illustration.

Finally, we discuss the effects of different values for the bound of function evaluation inexactness ε^f. Note that $\varepsilon^f = 0$ corresponds to Assumption 1, whereas $\varepsilon^f > 0$ corresponds to Assumption 2. Table 1 shows that the larger ε^f, the less infeasible instances we detect. This is to be expected since the feasible set grows with larger ε^f; see Fig. 3. Moreover, larger ε^f yield more added linearization points; see Fig. 4 (bottom left)—especially from iteration 5–10. As already discussed and illustrated in Fig. 4 (top), these linearization points yield larger master problem MIPs. Accordingly, the corresponding instances are harder to solve in general as it can be seen in Table 1 again.

Summing up, the observed computational behavior of Algorithm 1 is as expected. The algorithm is particularly strong for instances that can be solved within a small number of iterations in which the exponential growth of MIP complexity does not become apparent.

Although nonconvex mixed-integer nonlinear optimization problems from gas transport are known to be hard to solve, Algorithm 1 (both in the case of Assumptions 1 and 2) solves all instances on a real-world network in less than 10 s on average on a desktop computer, which shows the applicability of our approach.

6.2 Algorithm 2: exemplary case study

In this section, we study the behavior of Algorithm 2 in the case of Assumption 3. To this end, we consider the academic problem

$$\min_{x_1,x_2} \quad x_1 - 2x_2 \quad \text{s.t.} \quad x_2 = \sin\left(kx_1^2\right), \quad x_1 \in \left[0, \sqrt{\pi\ 11/10}\right], \tag{27}$$

cf. Fig. 5 (top left) for the case of $k = 5$. The initial sampling points are the bounds of x_1. We choose $\varepsilon = 1 \times 10^{-2}$, the choice of μ is irrelevant because the problem is feasible. Algorithm 2 applied to (27) terminates after 23 iterations with the global optimal solution. In Fig. 5, we study the first 8 iterations in detail. Note that the goal of Algorithm 2 is to compute an ε-feasible solution, which is a master problem's solution that satisfies the constraints of (27) with an ε-relaxation in the nonlinear constraints.

The quadrilateral in iteration 1 is determined by the initial sampling points $\underline{x}_1$ and $\bar{x}_1$. The derivatives at $\underline{x}_1$ and $\bar{x}_1$ are zero with the result that the initial Lipschitz constant estimates are $\widetilde{L} = 1$; see (15). Thus, the first quadrilateral does not contain graph(f_i) as $\widetilde{L}$ is too small. In iteration 2, $\widetilde{L}$ is updated such that the quadrilaterals are amplified. From iteration 3 to 5 the solutions of the master problem stay in a similar region, while additional sampling points tighten the approximation of graph(f_i) in this region. This tightening is crucial for the result of iteration 6 in which an important step is made towards ε-feasibility obtaining a master problem's solution close to feasibility. From iteration 7 on, the master problem's solutions finally take the last small steps to ε-feasibility.

Summing up, the decisive steps are made in iteration 2 and 6. The former significantly improves the Lipschitz constant estimate, while the latter provides a solution close to ε-feasibility. Finally, it can be clearly seen that the algorithm adaptively corrects and refines the approximation of graph(f_i) in a reasonable way: Only the region around the solution is considered, whereas areas far away from the solution are not refined; and even in latter iterations no outer approximation to graph(f_i) is obtained.

Due to the reasons given in Remark 3, we refrain from presenting a numerical comparison on large test sets with general-purpose solvers. However, we compare our method with a standard approach for globally solving nonconvex mixed-integer nonlinear problems. To this end, we choose the piecewise linear relaxation approach of [25]. The reasons why we choose this method for our comparison are the following. First, the general idea of using piecewise linear approximations for nonlinear functions is standard. Second, the extension to piecewise linear relaxation given in [25] is comparable to our method, since it also uses relaxations of the nonlinearity. In the following we compare the complexity of the obtained models in terms of the number of binary variables. To this end, we compare the number of binary variables in our final master problem with the number of binaries in the piecewise linear relaxation model, where we computed linearization points in advance so that the same accuracy is guaranteed a-priorily as for our algorithm. The results are plotted in Fig. 6 for Problem (27) for different parameters k (x-axis), which controls the oscillations of the constraint.

It can be seen that for stronger oscillations, the piecewise linear relaxation approach requires more binaries and that the number of binaries linearly corresponds to the

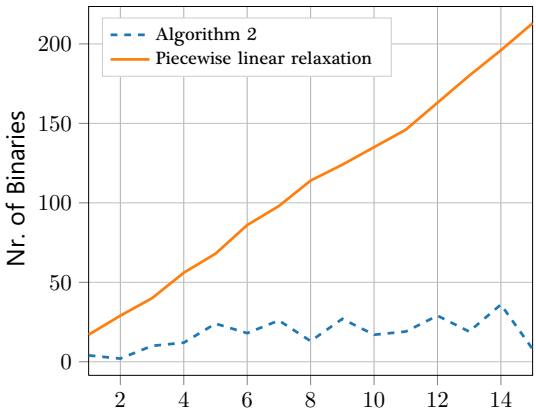

Fig. 6 Number of binary variables in the final master problem of Algorithm 2 (dashed) and a piecewise linear relaxation with an a-priori error bound (solid) for Problem (27) with different k (x-axis)

increase of k. In contrast, the number of binaries in the last master problem that we solve in our algorithm stays almost constant (and approximately below 25). Hence, especially for strong oscillations, our method seems to work well in this particular example. To obtain a fair comparison, we need to highlight two more points. First, 25 binary variables in the final master problem also corresponds to 24 master problems with $\ell = 1, 2, \ldots, 24$ binaries before we obtain the final master problem. This means that the overall effort of our algorithm is not only given by the final master problem. Nevertheless, the amount of MIPs to solve stays almost constant, whereas the complexity of the piecewise linear relaxation models linearly increases. On the other hand, there is also some effort to set up the piecewise linear relaxation model that a-priorily satisfies the claimed approximation accuracy. Second, the constant complexity of our approach is mainly explained by its adaptivity. Only the region around the solution is refined. The same adaptivity can also be exploited for the piecewise linear relaxation approach; cf. [11,24]. However, a more detailed numerical comparison is out of scope of this paper but will be part of our future research.

7 Conclusion

In this paper, we developed algorithms for MINLPs with Lipschitz continuous nonlinearities as they arise, e.g., in the context of mixed-integer optimization problems that are constrained by ordinary or partial differential equations as well as in the context of bilevel optimization problems. The concrete specification of our methods depend on the specific assumptions made on the (in)exactness of constraint function evaluations and on the knowledge about the Lipschitz constants. Under our strongest assumptions, i.e., the case of known Lipschitz constants and exact function evaluations, we can show that our algorithm finitely terminates at approximate globally optimal points or proves infeasibility. The same holds true, with some more technical assumptions, for the case of inexact function evaluations with a-priorily bounded evaluation errors. The more complicated case of unknown Lipschitz constants only allows for proving

finite termination. Moreover, we show which additional assumptions are necessary to detect infeasibility of the original problem.

Despite these results, there are still open questions to answer. One example is the case of inexact function evaluations without a-priorily known error bounds—a situation that frequently arises, e.g., in mixed-integer optimal control with partial differential equations. Finally, the extension of our methods (i) to the case of only implicitly stated constraints (like $c(x) = 0$) instead of the explicit case ($f_i(x_{i_1}) = x_{i_2}$) discussed in this paper and (ii) to higher dimensions is part of our future work.

Acknowledgements This research has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. We thank the DFG for their support within Project A05, A08, and B08 in CRC TRR 154.

References

1. Ainsworth, M., Oden, J.T.: A Posteriori Error Estimation in Finite Element Analysis. Pure and Applied Mathematics. Wiley, New York (2000). https://doi.org/10.1002/9781118032824
2. 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
3. Babuška, I., Strouboulis, T.: The Finite Element Method and Its Reliability. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2001)
4. Baes, M., Del Pia, A., Nesterov, Y., Onn, S., Weismantel, R.: Minimizing lipschitz-continuous strongly convex functions over integer points in polytopes. Math. Progr. **134**(1), 305–322 (2012). https://doi.org/10.1007/s10107-012-0545-8
5. Becker, R., Rannacher, R.: An optimal control approach to a posteriori error estimation. In: Iserles, A. (ed.) Acta Numerica, pp. 1–102. Cambridge University Press, Cambridge (2001). https://doi.org/10.1017/S0962492901000010
6. Belotti, P.: COUENNE: A User's Manual. https://projects.coin-or.org/Couenne. Accessed 12 Feb 2018
7. 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
8. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numer. Math. **4**(1), 238–252 (1962). https://doi.org/10.1007/BF01386316
9. 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. Discrete Optim. **15**, 186–204 (2008). https://doi.org/10.1016/j.disopt.2006.10.011
10. Borchers, B., Mitchell, J.E.: An improved branch and bound algorithm for mixed integer nonlinear programs. Comput. Oper. Res. **21**(4), 359–367 (1994). https://doi.org/10.1016/0305-0548(94)90024-8
11. Burlacu, R., Geissler, B., Schewe, L.: Solving mixed-integer nonlinear programs using adaptively refined mixed-integer linear programs. Technical Report., Friedrich-Alexander Universität Erlangen-Nürnberg (2017). https://opus4.kobv.de/opus4-trr154/frontdoor/index/index/docId/151
12. Colebrook, C.F.: Turbulent flow in pipes with particular reference to the transition region between smooth and rough pipe laws. J. Inst. Civ. Eng. **11**(4), 133–156 (1939). https://doi.org/10.1680/ijoti.1939.13150
13. DESFA. http://www.desfa.gr (2016)
14. Dempe, S.: Foundations of Bi-level Programming. Springer, Berlin (2002). https://doi.org/10.1007/b101970
15. Dempe, S.: Foundations of Bi-level Programming. Springer, Berlin (2002). https://doi.org/10.1007/b101970
16. Dempe, S., Kalashnikov, V., Pérez-Valdés, G.A., Kalashnykova, N.: Bilevel Programming Problems. Springer, Berlin (2015). https://doi.org/10.1007/978-3-662-45827-3

17. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Progr. **36**(3), 307–339 (1986). https://doi.org/10.1007/BF02592064
18. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to adaptive methods for differential equations. In: Iserles, A. (ed.) Acta Numerica, pp. 105–158. Cambridge University Press, Cambridge (1995)
19. Evtuschenko, Y.G.: Numerical methods for finding global extrema. USSR Comput. Math. Math. Phys. **11**, 38–54 (1971). https://doi.org/10.1016/0041-5553(71)90065-6
20. Feistauer, M.M.: Mathematical Methods in Fluid Dynamics. Pitman Monographs and Surveys in Pure and Applied Mathematics. Longman Scientific & Technical, New York (1993)
21. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Progr. **66**(1), 327–349 (1994). https://doi.org/10.1007/BF01581153
22. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., San Francisco (1979)
23. Gauvin, J., Dubeau, F.: Differential properties of the marginal function in mathematical programming. In: Guignard, M. (ed.) Optimality and Stability in Mathematical Programming, pp. 101–119. Springer, Berlin (1982). https://doi.org/10.1007/BFb0120984
24. Geissler, B.: Towards globally optimal solutions for MINLPs by discretization techniques with applications in gas network optimization. Ph.D. Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2011)
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. Geissler, B., Martin, A., Morsi, A., Schewe, L.: The MILP-relaxation approach. In: Evaluating Gas Network Capacities [49], chap. 6, pp. 103–122. https://doi.org/10.1137/1.9781611973693.ch6
27. Geissler, B., Morsi, A., Schewe, L.: A new algorithm for MINLP applied to gas transport energy cost minimization. In: Jünger, M., Reinelt, G. (eds.) Facets of Combinatorial Optimization: Festschrift for Martin Grötschel, pp. 321–353. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-38189-8_14
28. 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
29. Geissler, B., Morsi, A., Schewe, L., Schmidt, M.: Solving highly detailed gas transport MINLPs: block separability and penalty alternating direction methods. INFORMS J. Comput. (2017). https://doi.org/10.1287/ijoc.2017.0780. (in press)
30. Geoffrion, A.M.: Generalized benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972). https://doi.org/10.1007/BF00934810
31. Gleixner, A., Eifler, L., Gally, T., Gamrath, G., Gemander, P., Gottwald, R.L., Hendel, G., Hojny, C., Koch, T., Miltenberger, M., Müller, B., Pfetsch, M.E., Puchert, C., Rehfeldt, D., Schlösser, F., Serrano, F., Shinano, Y., Viernickel, J.M., Vigerske, S., Weninger, D., Witt, J.T., Witzig, J.: The SCIP optimization suite 5.0. Technical Report 17-61, ZIB, Takustr.7, 14195 Berlin (2017)
32. Göttlich, S., Potschka, A., Ziegler, U.: Partial outer convexification for traffic light optimization in road networks. SIAM J. Sci. Comput. **39**(1), B53–B75 (2017). https://doi.org/10.1137/15M1048197
33. Grimm, V., Schewe, L., Schmidt, M., Zöttl, G.: Uniqueness of market equilibrium on a network: a peak-load pricing approach. Eur. J. Oper. Res. **261**(3), 971–983 (2017). https://doi.org/10.1016/j.ejor.2017.03.036
34. Gu, Z., Rothberg, E., Bixby, R.: Gurobi Optimizer Reference Manual, Version 6.5.0 (2015)
35. Gugat, M., Hante, F.M., Hirsch-Dick, M., Leugering, G.: Stationary states in gas networks. Netw. Heterog. Media **10**(2), 295–320 (2015). https://doi.org/10.3934/nhm.2015.10.295
36. Gugat, M., Leugering, G., Martin, A., Schmidt, M., Sirvent, M., Wintergerst, D.: Towards simulation based mixed-integer optimization with differential equations. Networks (2018). https://doi.org/10.1002/net.21812 (in press)
37. Gugat, M., Schultz, R., Wintergerst, D.: Networks of pipelines for gas with non-constant compressibility factor: stationary states. In: Computational and Applied Mathematics (2016). https://doi.org/10.1007/s40314-016-0383-z
38. Hansen, P., Jaumard, B.: Lipschitz optimization. In: Horst, R., Pardalos, P.M. (eds.) Handbook of Global Optimization, pp. 407–493. Springer, New York (1995). https://doi.org/10.1007/978-1-4615-2025-2_9

39. Hante, F.M.: Relaxation methods for hyperbolic pde mixed-integer optimal control problems. Optim. Control Appl. Methods **38**(6), 1103–1110 (2017). https://doi.org/10.1002/oca.2315
40. Hante, F.M., Sager, S.: Relaxation methods for mixed-integer optimal control of partial differential equations. Comput. Optim. Appl. **55**(1), 197–225 (2013). https://doi.org/10.1007/s10589-012-9518-3
41. Horst, R.: Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, dc-programming, and lipschitzian optimization. J. Optim. Theory Appl. **58**(1), 11–37 (1988). https://doi.org/10.1007/BF00939768
42. 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
43. 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
44. Horst, R., Tuy, H.: Global Optimization, 2nd edn. Springer, Berlin (1996). https://doi.org/10.1007/978-3-662-03199-5
45. 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 (1977). https://doi.org/10.1007/978-3-642-95322-4_17
46. Klatte, D., Kummer, B.: Stability properties of infima and optimal solutions of parametric optimization problems. In: Demyanov, V.F., Pallaschke, D. (eds.) Nondifferentiable Optimization: Motivations and Applications, pp. 215–229. Springer, Berlin (1985). https://doi.org/10.1007/978-3-662-12603-5_20
47. Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L.: Evaluating Gas Network Capacities. SIAM-MOS Series on Optimization. SIAM, Bangkok (2015). https://doi.org/10.1137/1.9781611973693
48. Králik, J., Stiegler, P., Vostrý, Z., Záworka, J.: Dynamic Modeling of Large-Scale Networks with Application to Gas Distribution, Studies in Automation and Control, vol. 6. Elsevier, Amsterdam (1988)
49. Kvasov, D.E., Sergeyev, Y.D.: Lipschitz global optimization methods in control problems. Autom. Remote Control **74**(9), 1435–1448 (2013). https://doi.org/10.1134/S0005117913090014
50. LaMaTTO++: A Framework for Modeling and Solving Mixed-Integer Nonlinear Programming Problems on Networks. http://www.mso.math.fau.de/edom/projects/lamatto.html (2017)
51. Labbé, M., Violin, A.: Bilevel programming and price setting problems. 4OR **11**(1), 1–30 (2013). https://doi.org/10.1007/s10288-012-0213-0
52. Leyffer, S.: Deterministic methods for mixed integer nonlinear programming. Ph.D. Thesis, University of Dundee, Dundee, Scotland, UK (1993)
53. Li, M., Vicente, L.N.: Inexact solution of NLP subproblems in MINLP. J. Glob. Optim. **55**(4), 877–899 (2013). https://doi.org/10.1007/s10898-012-0010-5
54. Lindo Systems Inc. https://www.lindo.com. Accessed on 12 Feb 2018
55. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part i—convex underestimating problems. Math. Progr. **10**(1), 147–175 (1976). https://doi.org/10.1007/BF01580665
56. Misener, R., Floudas, C.A.: ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. J. Glob. Optim. **59**(2), 503–526 (2014). https://doi.org/10.1007/s10898-014-0166-2
57. Nemirovsky, A.S., Yudin, D.B.: Problem Complexity and Method Effciency in Optimization. Wiley, New York (1983)
58. 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
59. Pintér, J.: Extended univariate algorithms for n-dimensional global optimization. Computing **36**(1), 91–103 (1986). https://doi.org/10.1007/BF02238195
60. Pintér, J.: Global optimization on convex sets. Oper. Res. Spektrum **8**(4), 197–202 (1986). https://doi.org/10.1007/BF01721128
61. Pintér, J.: Globally convergent methods for n-dimensional multiextremal optimization. Optimization **17**(2), 187–202 (1986). https://doi.org/10.1080/02331938608843118
62. 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

63. Pintér, J.: Nonlinear optimization with gams/lgo. J. Glob. Optim. **38**(1), 79–101 (2007). https://doi.org/10.1007/s10898-006-9084-2
64. Pintér, J.D.: Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). Springer, Berlin (1996). https://doi.org/10.1007/978-1-4757-2502-5
65. Piyavskii, S.A.: An algorithm for finding the absolute extremum of a function. USSR Comput. Math. Math. Phys. **12**, 57–67 (1972). https://doi.org/10.1016/0041-5553(72)90115-2
66. Rannacher, R., Vexler, B., Wollner, W.: A posteriori error estimation in PDE-constrained optimization with point wise inequality constraints. In: Constrained Optimization and Optimal Control for Partial Differential Equations, International Series of Numerical Mathematics, vol. 160, Springer, pp. 349–373 (2012). https://doi.org/10.1007/978-3-0348-0133-1_19
67. 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
68. Sager, S., Jung, M., Kirches, C.: Combinatorial integral approximation. Math. Methods Oper. Res. **73**(3), 363–380 (2011). https://doi.org/10.1007/s00186-011-0355-4
69. Saleh, J.: Fluid Flow Handbook. McGraw-Hill Handbooks. McGraw-Hill, New York (2002)
70. 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 **2**(4), 40 (2017). https://doi.org/10.3390/data2040040
71. 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
72. Schmidt, M., Steinbach, M.C., Willert, B.M.: The precise NLP model. In: Koch, T. et al. [49] Chap. 10, pp. 181–210. https://doi.org/10.1137/1.9781611973693.ch10
73. 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
74. 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
75. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming, vol. 65. Springer, Berlin (2002). https://doi.org/10.1007/978-1-4757-3532-1
76. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: a theoretical and computational study. Math. Progr. **99**(3), 563–591 (2004). https://doi.org/10.1007/s10107-003-0467-6
77. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Progr. **103**(2), 225–249 (2005). https://doi.org/10.1007/s10107-005-0581-8
78. Tuy, H., Horst, R.: Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and d.c. optimization problems. Math. Progr. **41**(1), 161–183 (1988). https://doi.org/10.1007/BF01580762
79. Vavasis, S.A.: Nonlinear Optimization: Complexity Issues. Oxford University Press Inc, Oxford (1991)
80. Verfürth, R.: A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley, New York (1996)
81. Verfürth, R.: A Posteriori Error Estimation Techniques for Finite Element Methods. Oxford University Press, Oxford (2013). https://doi.org/10.1093/acprof:oso/9780199679423.001.0001
82. Westerlund, T., Lundqvist, K.: Alpha-ECP, version 5.01: an interactive MINLP-solver based on the extended cutting plane method. Technical Report 01-178-A (2001)
83. Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. **19**, 131–136 (1995). https://doi.org/10.1016/0098-1354(95)87027-X
84. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Progr. **106**(1), 25–57 (2006). https://doi.org/10.1007/s10107-004-0559-y