A Successive Linear Relaxation Method for MINLPs with Multivariate Lipschitz Continuous Nonlinearities

Julia Grübel, Richard Krug, Martin Schmidt, Winnifried Wollner

Abstract. We present a novel method for mixed-integer optimization problems with multivariate and Lipschitz continuous nonlinearities. In particular, we do not assume that the nonlinear constraints are explicitly given but that we can only evaluate them and that we know their global Lipschitz constants. The algorithm is a successive linear relaxation method in which we alternate between solving a master problem, which is a mixed-integer linear relaxation of the original problem, and a subproblem, which is designed to tighten the linear relaxation of the next master problem by using the Lipschitz information about the respective functions. By doing so, we follow the ideas of Schmidt et al. (2018, 2021) and improve the tackling of multivariate constraints. Although multivariate nonlinearities obviously increase modeling capabilities, their incorporation also significantly increases the computational burden of the proposed algorithm. We prove the correctness of our method and also derive a worst-case iteration bound. Finally, we show the generality of the addressed problem class and the proposed method by illustrating that both bilevel optimization problems with nonconvex and quadratic lower levels as well as nonlinear and mixed-integer models of gas transport can be tackled by our method. We provide the necessary theory for both applications and briefly illustrate the outcomes of the new method when applied to these two problems.

1. Introduction

Mixed-integer nonlinear optimization problems (MINLPs) form one of today’s most important classes of optimization models. The reason is, at least, twofold. First, the capability of modeling nonlinearities allows to include many sophisticated aspects of, e.g., physics, economics, engineering, or medicine. Second, the incorporation of integer variables makes it possible to model decision making such as turning on or off a machine or investing in a product or not. Of course, this modeling flexibility comes at the price of models that are hard to solve for realistically sized instances since MINLPs are NP-hard in general [17, 29]. Nevertheless, the theoretical and algorithmic advances of the last years and decades make it possible today to solve rather large-scale instances to global optimality in a reasonable amount of time [6]—in particular if problem-specific structural properties of the model can be exploited algorithmically; see, e.g., [7, 15, 16, 33, 34] for the convex as well as [1, 36, 49, 51] for the nonconvex case.

In addition, there has been a significant amount of research devoted to the cases in which only very few structural assumptions can be exploited. This is the framework considered in this paper since we assume that certain functions of the model are not given explicitly but can be evaluated and some analytical properties such as Lipschitz continuity is known. To illustrate why this is important, let us sketch three areas of
applications in which only few assumptions on the structure of the model (or on specific parts of the model) can be made. First, many mixed-integer optimization problems subject to ordinary or partial differential equations fit into this context. In many cases, approaches in this field are driven by incorporating the so-called control-to-state map into the optimization model to “eliminate” the differential equation from the model; see, e.g., [5, 8, 23, 52]. This mapping, however, cannot be stated explicitly in general and one thus has to resort to exploiting analytical properties such as Lipschitz continuity and the ability to evaluate the mapping, at least in an approximate way. Second, and rather related to the first example, optimization models incorporating constraints that rely on calls to expensive simulation software can be cast in the framework mentioned in this paper as well if enough analytic information is known about the input-output mapping of the simulation code; see, e.g., [2, 10]. Third and finally, bilevel optimization problems can also be interpreted as models in which a single constraint makes the problem much harder to solve [12–14]. In this case, it is the constraint that models the optimality of the decisions of the lower-level (or follower) problem given the decisions of the leader (which is the decision maker modeled in the upper-level problem). The best-reply function, which models the optimal response of the follower, can usually not be written in closed form but it can be evaluated (by solving the lower-level problem for a given feasible point of the upper level) and, under suitable assumptions [13], analytical properties such as Lipschitz continuity can be established.

This paper addresses a special class of MINLPs for which closed-form expressions for the nonlinearities are not available but Lipschitz continuity is guaranteed with known Lipschitz constants. To this end, all three areas of application discussed above can be addressed by the method proposed in this paper. One part of our contribution, indeed, is that the case studies presented later in Sections 4 and 5 explicitly show the applicability of our method to bilevel problems with nonconvex and quadratic lower-level problems (Section 4) and to problems on gas transport networks that are subject to differential equations (Section 5). Both are classes of problems that received a lot of attention during the last years; see, e.g., [3, 12, 31] and [32, 41]. Before we are able to tackle these problems, we first need to formally state the problem class under consideration, which is what we do in Section 2. Afterward, in Section 3, we describe the main rationale of the method, present it formally, and analyze it theoretically. The latter leads to a correctness theorem showing that the method finitely terminates with an approximate solution and we further derive a worst-case iteration bound.

The main contributions of our work are the following. We develop an algorithm that only requires very weak assumptions for being applied. Hence, the method can be applied to a very large set of problems that cannot be solved with other classic MINLP solvers, which require that all constraints of the problem are given in closed form. To illustrate the generality of the method, we present the application to nonlinear gas network optimization as well as bilevel problems with nonconvex quadratic lower-level problems. Our work clearly needs to be seen as a generalization of the works [47, 48]. In particular, we generalize [47] to the multidimensional case for which we present a more effective numerical scheme compared to [48] that uses different geometries for the outer approximation as compared to those used in [47]. Since our main workhorse is the Lipschitz continuity of the nonlinearities, we are still in line with the works [25, 26, 28, 42–44, 53], to name only a few. For a more detailed overview about this field see the textbook [27] and the references therein as well as [47, 48], where we discussed the positioning of the method in the literature in more detail.
2. Problem Definition

We consider the problem
\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b, \quad \bar{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \\
& \quad f_i(x_{I_i}) = x_{r_i}, \quad i \in [p],
\end{align*}
\]
for \( c \in \mathbb{R}^{n \times m}, A \in \mathbb{R}^{q \times (n \times m)}, b \in \mathbb{R}^q \) are given data, \( \bar{x} \in \mathbb{R}^n \times \mathbb{Z}^m \) and \( \bar{x} \in \mathbb{R}^n \times \mathbb{Z}^m \) are finite bounds, and \( [p] := \{1, \ldots, p\} \). Hence, we consider a linear objective (1a), linear mixed-integer constraints (1b), and nonlinear constraints defined by the functions \( f_i: \mathbb{R}^{l_i} \to \mathbb{R} \). All \( f_i, i \in [p] \), are Lipschitz continuous functions and \( l_i = |I_i| \) is the number of their arguments. Moreover, \( I_i \subseteq [n] \) is the index set of the variables on which the function \( f_i \) depends. Without loss of generality, we further assume that \( r_i \in [n] \) with \( r_i \notin I_i \) for all \( i \in [p] \). In what follows, we also write \( x_{I_i} = (x_{I_i}, x_{r_i}) \in \mathbb{R}^{l_i+1} \).

The main challenge when solving Problem (1) is that we assume that the nonlinear functions \( f_i \) are not given in closed form but that we can only evaluate them and that we know their Lipschitz constants.

The \( \varepsilon \)-relaxed version of the original problem (1) is given by
\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b, \quad \bar{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \\
& \quad |f_i(x_{I_i}) - x_{r_i}| \leq \varepsilon, \quad i \in [p],
\end{align*}
\]
where \( \varepsilon > 0 \) is a prescribed tolerance. A feasible point of (2) is called an \( \varepsilon \)-feasible point of Problem (1). Moreover, we call a global solution of (2) an approximate global optimal solution in what follows.

3. The Algorithm

In this section, we introduce an iterative procedure to solve Problem (1) to approximate global optimality. The main idea is to relax all nonlinearities by utilizing the Lipschitz continuity of these functions. In each iteration, the relaxed problem, which we will call the master problem, needs to be solved to global optimality. Subsequently, a subproblem is solved to tighten the relaxation for the next master problem. This procedure is then repeated until an \( \varepsilon \)-feasible solution is found or until it can be shown that the original problem is infeasible.

The master problem in iteration \( k \) reads
\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax \geq b, \quad \bar{x} \leq x \leq \bar{x}, \quad x \in \mathbb{R}^n \times \mathbb{Z}^m, \\
& \quad x_{I_i} \in \Omega^k_i, \quad i \in [p], \quad \quad (M(k))
\end{align*}
\]
where \( \Omega^k_i \) is a relaxation of the graph of the nonlinearity \( f_i \). This relaxation will be stated in terms of mixed-integer linear constraints; see below. The idea behind is to partition the domain of \( f_i \) into a set of boxes that are indexed using indices in \( J^k_i = \{1, \ldots, |J^k_i|\} \) and to linearly relax the graph over each box with the region \( \Omega^k_i(j), j \in J^k_i \), using the

\footnote{Note that we omit transpositions here and in what follows for the ease of better reading.}
Lipschitz continuity of $f_i$. Hence, we obtain
\[ \Omega^k_i = \bigcup_{j \in J^k_i} \Omega^k_i(j). \]  (3)

After solving the master problem, the boxes that contain the solution $x^k$ are identified and split in smaller boxes to get a finer relaxation for the next iteration. The main purpose of solving the subproblem afterward is to find good splitting points for these boxes. To this end, the subproblem determines a point of the graph of each nonlinearity and, at the same time, tries to minimize the distance to the solution of the last master problem. Hence, the subproblem is given by
\[
\min_{\tilde{x}} \| \tilde{x} - x^k \|^2_2 \quad \text{s.t.} \quad f_i(\tilde{x}_{I_i}) = \tilde{x}, \quad \tilde{x}_{I_i} \in \tilde{\Omega}^k_i(j^k_i), \quad i \in [p],
\]  (S(k))

where $j^k_i \in J^k_i$ denotes the box with $x^k_{I_i} \in \Omega^k_i(j^k_i)$ for all $i \in [p]$ and $\tilde{\Omega}^k_i(j^k_i)$ is a suitably chosen subregion of $\Omega^k_i(j^k_i)$. The reason for the use of these subregions will be discussed in detail in Section 3.2.

Figure 1 depicts the subproblem (S(k)) with the regions $\tilde{\Omega}^k_i(j^k_i)$ and $\Omega^k_i(j^k_i)$ (left) and the corresponding regions $\Omega^{k+1}_i(j)$ and $\Omega^{k+1}_i(j+1)$ of the master problem (M(k)) in the next iteration (right).

3.1. Construction of the Master Problem’s Feasible Region. We now describe in detail how we construct the relaxations $\Omega^k_i$. First, we define the box
\[ B(v, \bar{v}) := \{ x \in \mathbb{R}^d : v \leq x \leq \bar{v} \} \]
for $v, \bar{v} \in \mathbb{R}^d$, $v \leq \bar{v}$, and arbitrary dimension $d$.

For each $i \in [p]$, we assume that we are given vectors $\psi^k_i(j), \bar{\psi}^k_i(j) \in \mathbb{R}^{l_i}$ for $j \in J^k_i$ such that the boxes $B(\psi^k_i(j), \bar{\psi}^k_i(j))$ have pairwise disjoint interiors and cover the bounding box of $x_{I_i}$, i.e., we have
\[
B(x_{I_i}, \bar{x}_{I_i}) = \bigcup_{j \in J^k_i} B(\psi^k_i(j), \bar{\psi}^k_i(j)).
\]  (4)
Let $L_i$ be the Lipschitz constant of $f_i$ on $B(x_{I_i}, \bar{x}_{I_i}) \subset \mathbb{R}^{I_i}$ w.r.t. a given norm $\|\cdot\|$, where any (weighted) norm in $\mathbb{R}^{I_i}$ can be used. Let $m^k_i(j)$ be the center point of the box $B(v^k_i(j), \bar{v}^k_i(j))$, i.e.,

$$m^k_i(j) = \frac{1}{2} (v^k_i(j) + \bar{v}^k_i(j))$$

holds. Due to the Lipschitz continuity of $f_i$, we have

$$x_{r_i} \leq f_i(m^k_i(j)) + L_i \|x_{I_i} - m^k_i(j)\|,$$
$$x_{r_i} \geq f_i(m^k_i(j)) - L_i \|x_{I_i} - m^k_i(j)\|$$

for $x_{I_i} \in B(v^k_i(j), \bar{v}^k_i(j))$. Since $\|x_{I_i} - m^k_i(j)\|$ attains its maximum over $B(v^k_i(j), \bar{v}^k_i(j))$ in the vertices of the box, we can replace $x_{I_i}$ with $\bar{v}^k_i(j)$. It thus holds

$$x_{r_i} \leq f_i(m^k_i(j)) + L_i \|\bar{v}^k_i(j) - m^k_i(j)\| = f_i(m^k_i(j)) + \frac{L_i}{2} \|\bar{v}^k_i(j) - v^k_i(j)\|, \quad (5a)$$
$$x_{r_i} \geq f_i(m^k_i(j)) - L_i \|\bar{v}^k_i(j) - m^k_i(j)\| = f_i(m^k_i(j)) - \frac{L_i}{2} \|\bar{v}^k_i(j) - v^k_i(j)\| \quad (5b)$$

for $x_{I_i} \in B(v^k_i(j), \bar{v}^k_i(j))$. With this we can define the region $\Omega^k_i(j)$ for $j \in J^k_i$ as the box

$$\Omega^k_i(j) := \{(x_{I_i}, x_{r_i}) \in \mathbb{R}^{I_i+1} : v^k_i(j) \leq x_{I_i} \leq \bar{v}^k_i(j),$$
$$x_{r_i} \leq f_i(m^k_i(j)) + \frac{1}{2} L_i \|\bar{v}^k_i(j) - v^k_i(j)\|,$$
$$x_{r_i} \geq f_i(m^k_i(j)) - \frac{1}{2} L_i \|\bar{v}^k_i(j) - v^k_i(j)\| \}.$$  

The next step is to define the region $\Omega^k_i$ as the union of all $\Omega^k_i(j)$; see (3). Using (4), we obtain a covering of the bounding box of $x_{I_i}$ and via (5), we obtain bounds for $x_{r_i}$. Hence, it follows that $\Omega^k_i$ is a relaxation of the graph of the function $f_i$.

**Proposition 3.1.** It holds graph($f_i$) $\subseteq \Omega^k_i$ on $B(x_{I_i}, \bar{x}_{I_i})$ for all $i \in [p]$ and all $k$.

For what follows, we abbreviate

$$\mathcal{X}^k_i := \{B(v^k_i(1), \bar{v}^k_i(1)), \ldots, B(v^k_i(J^k_i)), \bar{v}^k_i(J^k_i))\},$$

i.e., $\mathcal{X}^k_i$ is the set of boxes that is used to define $\Omega^k_i$. In our algorithm, we use $\Omega^k_i$ to replace the nonlinear constraints $f_i(x_{I_i}) = x_{r_i}$ for all $i \in [p]$ in Problem (1) in order to obtain a relaxation.

**Lemma 3.2.** The master problem (M($k$)) can be modeled as mixed-integer linear problem.
Proof. We can write \((M(k))\) using the following Big-M formulation:

\[
\begin{align*}
\min_{\bar{x}} & \quad c^T \bar{x} \\
\text{s.t.} & \quad A\bar{x} \geq b, \quad \bar{x} \leq \bar{x}, \quad \bar{x} \in \mathbb{R}^n \times \mathbb{Z}^m, \\
& \quad x_i \geq v_i^k(j) - M(1 - z^k_i(j)), \quad i \in [p], \ j \in J^k, \\
& \quad x_i \leq v_i^k(j) + M(1 - z^k_i(j)), \quad i \in [p], \ j \in J^k, \\
& \quad x_r \leq f_i(m^k_i(j)) + \frac{1}{2}L_i\|v_i^k(j) - v_i^k(j)\| + M(1 - z^k_i(j)), \quad i \in [p], \ j \in J^k, \\
& \quad x_r \geq f_i(m^k_i(j)) - \frac{1}{2}L_i\|v_i^k(j) - v_i^k(j)\| - M(1 - z^k_i(j)), \quad i \in [p], \ j \in J^k, \\
& \quad \sum_{j \in J^k} z^k_i(j) = 1, \quad i \in [p], \\
& \quad z^k_i(j) \in \{0, 1\}, \quad i \in [p], \ j \in J^k.
\end{align*}
\]  

The rationale of this model is as follows. For each nonlinearity \(i \in [p]\) and each box \(j \in J^k\), we introduce a binary variable \(z^k_i(j)\) that indicates whether the solution lies in this box or not. If \(z^k_i(j) = 1\), the constraints \((6c)\)–\((6f)\) are equivalent to the definition of \(\Omega_i^k(j)\). If \(z^k_i(j) = 0\), then \((6c)\)–\((6f)\) are always fulfilled if the constant \(M\) is chosen sufficiently large. Constraint \((6g)\) finally ensures that for each nonlinearity \(i \in [p]\) exactly one box \(j \in J^k\) is chosen.

Let us remark that it is always possible in our setting to obtain finite and sufficiently large values \(M\) by using the finite bounds on the variables in \((1b)\).

3.2. Construction of the Subproblem. We now introduce the chosen subregion of \(\Omega_i^k(j)\) used in the subproblem \((S(k))\). We define this region as

\[
\tilde{\Omega}_i^k(j) = \Omega_i^k(j) \cap \Omega_i^k(j),
\]

using the further subregion

\[
\tilde{\Omega}_i^k(j) = \{(x_l, x_r) \in \mathbb{R}^{I_i+1}: (1 - \lambda)v_i^k(j) + \lambda v_i^k(j) \leq x_l \leq x_r \leq \lambda v_i^k(j) + (1 - \lambda)v_i^k(j), \forall \lambda \in (0, 1/2)\},
\]

for some \(\lambda \in (0, 1/2)\). This ensures that the solution of the subproblem \((S(k))\) cannot be arbitrarily close to the edges of the box by \(\lambda > 0\) is chosen appropriately.

Let us also note that for each iteration \(k \in \mathbb{N}\), the subproblem \((S(k))\) can be separated into \(p\) smaller problems under reasonable assumptions. If the index sets \(I_i = (I_i, r_i)\) are non-overlapping, i.e.,

\[
(I_i \cup \{r_i\}) \cap (I_j \cup \{r_j\}) = \emptyset \quad \text{for all} \quad i, j \in [p], \ i \neq j,
\]

these smaller problems can be solved in parallel. The above assumption can always be satisfied by introducing additional auxiliary variables.

Lemma 3.3. Suppose that the index sets \(I_i = (I_i, r_i)\) are non-overlapping, i.e., \((7)\) holds. Then, the subproblem \((S(k))\) is completely separable, i.e., we can solve the subproblem in iteration \(k\) by solving \(p\) smaller problems given by

\[
\begin{align*}
\min_{\bar{x}_I} & \quad \|\bar{x}_I - x_{r_i}^k\|_2^2 \\
\text{s.t.} & \quad f_i(\bar{x}_I) = \bar{x}_{r_i}, \quad \bar{x}_I \in \tilde{\Omega}_i^k(j_i^k).
\end{align*}
\]  

Proof. The constraints of \((S(k))\), i.e.,

\[
f_i(\bar{x}_I) = \bar{x}_{r_i}, \quad \bar{x}_I \in \tilde{\Omega}_i^k(j_i^k), \quad i \in [p],
\]
completely decouple along \( i \in [p] \) and so does the objective function
\[
\|\tilde{x} - x^k\|_2^2 = \sum_{i \in [p]} \|\tilde{x}_{I_i} - x^k_{I_i}\|_2^2.
\]
Therefore, the solution of the subproblem \((S(k))\) can be obtained by solving Problem (8) for all \( i \in [p] \).

Note that, in the formal sense of complexity theory, the subproblem can be as hard to solve as the originally given MINLP. However, the split into multiple and thus much smaller subproblems can make a huge difference in practice. Moreover, we completely split the mixed-integer aspects from the nonlinear aspects of the problem, which can also help a lot in solving the subproblems although they are still hard in the formal sense.

### 3.3. Formal Statement of the Algorithm

Before we can formally introduce the algorithm, we need the following notation. Let \( B(\bar{y}, \bar{v}) \subseteq \mathbb{R}^d \) be a box with an interior point \( x \in \text{int} B(\bar{y}, \bar{v}) \), i.e., \( \bar{y} < x < \bar{v} \). The point \( x \) splits the box into a set of boxes that we define as
\[
S(\bar{y}, \bar{v}, x) := \{ B(\bar{w}, \bar{w}) : (w_l = \bar{y}_l \land w_u = x_l) \lor (w_l = x_l \land w_u = \bar{v}_l) \text{ for all } l \in [d] \}.
\]

We can utilize this notation to get a finer relaxation of graph(\( f \)) by splitting an element \( X^k_i \) using the solution of the subproblem \((S(k))\) as the splitting point. This yields a set of smaller boxes that still fulfills Condition (4) as the following proposition states.

**Proposition 3.4.** For a given box \( B(\bar{y}, \bar{v}) \subseteq \mathbb{R}^d \) and an interior point \( x \in \text{int} B(\bar{y}, \bar{v}) \), the set \( S(\bar{y}, \bar{v}, x) \) contains \( 2^d \) smaller boxes that have pairwise disjoint interiors and that completely cover the box \( B(\bar{y}, \bar{v}) \), i.e.,
\[
B(\bar{y}, \bar{v}) = \bigcup_{b \in S(\bar{y}, \bar{v}, x)} b.
\]

We can now present the complete method, which is given in Algorithm 1. Before we prove its correctness, let us first discuss its basic functionality. After the master problem \((M(k))\) is solved in Step 3, it is checked in Step 8 if its solution is already \( \varepsilon \)-feasible for the original problem. To determine the boxes \( j^k_i \in J^k_i \) in Step 11 one can simply check the indicator variables \( z^k_i(j) \) of the MIP formulation (6). If the solution is not yet \( \varepsilon \)-feasible, then there are nonlinearities \( f_i \) that have feasibility violations larger than \( \varepsilon \). For these nonlinearities we refine the relaxation of the master problem in Step 15 and re-iterate.

Note that it is not necessary for the correctness of Algorithm 1 to solve the subproblem \((S(k))\) to global optimality. Our rationale, however, is that optimal solutions of the subproblems yield better splitting points that lead to faster convergence in practice. For the correctness of the algorithm it is sufficient to find feasible points of \((S(k))\) that are guaranteed to exist due to the following lemma.

**Lemma 3.5.** All subproblems \((S(k))\) are feasible if Property (7) is satisfied.

**Proof.** Because of Property (7), the nonlinear constraints \( f_i(\tilde{x}_{I_i}) = \tilde{x}_r \) in \((S(k))\) do not depend on each other. From Proposition 3.1 we know that the graph of \( f_i \) over the initial region \( B(\tilde{x}_{I_i}, \tilde{x}_{I_i}) \) lies entirely in \( \Omega^k_i \). Since the nonempty subregion \( \Omega^k_i(\bar{j}) \) restricts \( \Omega^k_i \) in all but the last dimension, the claim follows. 

Next, we prove that Algorithm 1 always terminates after finitely many iterations.
Algorithm 1 Iterative Method to Approximately Solve Problem (1)

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

Ensure: An approximate globally optimal and \( \varepsilon \)-feasible point for Problem (1) or an indication that Problem (1) is infeasible.

1: Set \( k \leftarrow 0 \) and initialize \( X_0^k = \{ B(x_{i,1}, \bar{x}_{i,1}) \} \) for all \( i \in [p] \).
2: for \( k = 0, 1, 2, \ldots \) do
3: Solve the master problem \((M(k))\) to global optimality.
4: if \((M(k))\) is infeasible then
5: \hspace{1cm} return "Problem (1) is infeasible."
6: end if
7: Let \( x^k \) denote the optimal solution of \((M(k))\).
8: if \( |f_i(x^k_i) - x^k_i| \leq \varepsilon \) for all \( i \in [p] \) then
9: \hspace{1cm} return \( x^k \).
10: end if
11: Determine the boxes \( j_i^k \in J_i^k \) for all \( i \in [p] \).
12: Solve the subproblem \((S(k))\) and let \( \bar{x}^k \) denote the optimal solution.
13: for \( i \in [p] \) do
14: if \( |f_i(x^k_i) - x^k_i| > \varepsilon \) then
15: \hspace{1cm} \( X^{k+1}_i \leftarrow \left( \{ x^k_i \} \cup B(v_i^k(j_i^k), \bar{v}_i^k(j_i^k)) \right) \) \( \cup S(v_i^k(j_i^k), \bar{v}_i^k(j_i^k), \bar{x}_i^k) \).
16: else \( \hspace{1cm} X^{k+1}_i \leftarrow X^k_i \).
17: end if
18: \end{for}
19: \end{for}

Theorem 3.6. There exists a constant \( K < \infty \) such that Algorithm 1 either terminates with an approximate globally optimal solution \( x^{k^*} \) or with the indication of infeasibility in an iteration \( k^* \leq K \).

Proof. The box \( \Omega_i^k(j_i) \) is bounded for each iteration \( k \) and all \( i \in [p] \) and \( j_i \in J_i^k \). For the \( x_{r_i} \)-coordinate, the bounding inequalities are given by

\[
|f_i(m_i^k(j_i)) - x_{r_i}^k| \leq L_i\|v_i^k(j_i) - \bar{v}_i^k(j_i)\| = x_{r_i} \leq f_i(m_i^k(j_i)) + \frac{1}{2} L_i\|v_i^k(j_i) - \bar{v}_i^k(j_i)\|.
\]

Therefore the corresponding side length of the box \( \Omega_i^k(j_i) \) in its \( x_{r_i} \)-coordinate is \( d_{r_i}^k(j_i) := L_i\|v_i^k(j_i) - \bar{v}_i^k(j_i)\|. \)

If \( d_{r_i}^k(j_i) \leq \varepsilon \) holds, then the inequality

\[
|f_i(x_{r_i}^k) - x_{r_i}^k| > \varepsilon
\]

in Step 14 of Algorithm 1 cannot be fulfilled. It follows that the box \( B(v_i^k(j_i^k), \bar{v}_i^k(j_i^k)) \) will not be split but remains the same in Step 17.

Next, we analyze how \( d_{r_i}^k(j_i) \) changes if a box is split into smaller boxes. Let \( B(v_i^{k+1}(j_i), \bar{v}_i^{k+1}(j_i)) \) be one of the smaller boxes that is added to \( X_i^k \) in Step 15. The side length \( d_{r_i}^{k+1}(j_i) \) of the corresponding box \( \Omega_i^{k+1}(j_i) \) can be bounded from above via

\[
d_{r_i}^{k+1}(j_i) = L_i\|v_i^{k+1}(j_i) - \bar{v}_i^{k+1}(j_i)\| \leq (1 - \lambda)L_i\|v_i^k(j_i) - \bar{v}_i^k(j_i)\| = (1 - \lambda)d_{r_i}^k(j_i).
\]
This means that $d_{r_i}^k(j)$ is decreased by at least a factor of $(1 - \lambda)$ if a box is split in Step 15. Since $((1 - \lambda)^k)_{k \in \mathbb{N}}$ is a geometric sequence with $|1 - \lambda| < 1$, it converges to zero, i.e., $(1 - \lambda)^k \to 0$ as $k \to \infty$. It follows that any box $B(\bar{x}_I^k, \bar{x}_I^k(j))$—including the first box $B(\bar{x}_I^0, \bar{x}_I^0)$—can only be split finitely many times in Step 15 before the side length $d_{r_i}^k(j)$ of $\Omega_{r_i}^k(j)$ fulfills $d_{r_i}^k(j) \leq \varepsilon$.

Since the index set $[p]$ is finite, there are only finitely many boxes in $\mathcal{X}_r^k$ for all $i \in [p]$ and all $k$. These boxes can only be split finitely many times. Hence, there exists an iteration $K < \infty$ in which no box will be split in Step 15. This, however, can only be the case if the if-condition in Step 14 does not hold for all $i \in [p]$. Thus, we have

$$|f_i(x_{I_i}^K) - x_{r_i}^K| \leq \varepsilon \text{ for all } i \in [p],$$

which is the if-condition in Step 8. This means that the algorithm terminates in Step 9 in an iteration $K$. Hence, it follows that there exists a $K < \infty$ such that Algorithm 1 terminates in Step 5 or 9 in an iteration $k^* \leq K$. \hfill \square

We close this section by stating and proving a result for the worst-case number of required iterations of Algorithm 1.

**Theorem 3.7.** Algorithm 1 terminates after at most

$$K = \sum_{i \in [p]} \sum_{k=0}^{S_i} 2^{k l_i}$$

iterations with

$$S_i = \left\lfloor \log_{(1 - \lambda)} \left( \frac{\varepsilon}{L_i \|\bar{x}_I - \bar{x}_I^k\|} \right) \right\rfloor$$

for $i \in [p]$.

**Proof.** From the proof of Theorem 3.6 we know that a box $B(\bar{x}_I^k(j_i^k), \bar{x}_I^k(j_i^k))$ can only be split finitely many times before the side length $d_{r_i}^k(j)$ of $\Omega_{r_i}^k(j)$ satisfies $d_{r_i}^k(j) \leq \varepsilon$. We can give an upper bound for how many iterations this takes for the first box $B(\bar{x}_I^0, \bar{x}_I^0)$ by solving the equation

$$(1 - \lambda)^k L_i \|\bar{x}_I - \bar{x}_I^k\| = \varepsilon$$

for $k$, which yields

$$k = \log_{(1 - \lambda)} \left( \frac{\varepsilon}{L_i \|\bar{x}_I - \bar{x}_I^k\|} \right).$$

Since the box $B(\bar{x}_I^k(j_i^k), \bar{x}_I^k(j_i^k))$ will not be split anymore if $d_{r_i}^k(j) \leq \varepsilon$, we can round this value to obtain

$$S_i = \left\lfloor \log_{(1 - \lambda)} \left( \frac{\varepsilon}{L_i \|\bar{x}_I - \bar{x}_I^k\|} \right) \right\rfloor.$$

For each box that is split, there are $2^{k l_i}$ smaller boxes that are added to $\mathcal{X}_r^k$. Therefore, for each $i \in [p]$ the maximal number of iterations required until there are no boxes left in $\mathcal{X}_r^k$ that can be split is bounded from above by

$$\sum_{k=0}^{S_i} (2^{k l_i})^k = \sum_{k=0}^{S_i} 2^{k l_i}.$$  

Since it is possible that in each iteration, there is only a single nonlinearity $i \in [p]$ for which a box is split, we have to sum up (9) for each $i \in [p]$ to get

$$K = \sum_{i \in [p]} \sum_{k=0}^{S_i} 2^{k l_i}.$$
Remark 3.8. Theorem 3.7 states that choosing $\lambda = 0.5$ results in the lowest number of iterations in the worst-case. Then, no subproblem (S(k)) is needed as one can simply evaluate the nonlinearity $f_i$ in the center point $m_k^i(j_k^i)$ of the current box to receive the splitting point. However, in practice it can be better to choose a smaller parameter $\lambda$, which allows the splitting point to be closer to the master problem’s solution and which, thus, may result in a finer approximation of the nonlinearity near the optimal solution of Problem (1).

4. Application to Nonlinear Bilevel Problems

The method developed in the previous section can be applied to nonlinear bilevel problems with nonconvex lower-level models, which is an extremely challenging class of problems. To illustrate this, we consider optimistic MIQP-QP bilevel problems of the form

$$\begin{align*}
\min_{x,y} & \quad \frac{1}{2} x^T H_u x + c_u^T x + \frac{1}{2} y^T G_u y + d_u^T y \\
\text{s.t.} & \quad A x + B y \leq a, \quad x \leq \bar{x}, \quad x \in \mathbb{R}^{n_x}, \\
& \quad y \in \arg\min \left\{ \frac{1}{2} \tilde{y}^T G_1 \tilde{y} + d_1^T \tilde{y} : C x + D \tilde{y} \leq b, \quad y \leq \tilde{y}, \quad \tilde{y} \in \mathbb{R}^{n_y} \right\},
\end{align*}$$

(10)

where $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ denote the upper- and lower-level variables, which are finitely bounded by $\bar{x}$, $\bar{y}$, and $\tilde{y}$. Further, we have matrices $A \in \mathbb{R}^{m_a \times n_x}$, $B \in \mathbb{R}^{m_a \times n_y}$, $C \in \mathbb{R}^{m_c \times n_x}$, $D \in \mathbb{R}^{m_c \times n_y}$, as well as right-hand side vectors $a \in \mathbb{R}^{m_a}$ and $b \in \mathbb{R}^{m_c}$. In addition, we have $c_u \in \mathbb{R}^{n_u}$ and $d_u, d_1 \in \mathbb{R}^{n_u}$. Finally, $H_u \in \mathbb{R}^{n_x \times n_x}$, $G_u \in \mathbb{R}^{n_y \times n_y}$ are positive semidefinite and symmetric matrices, while $G_1 \in \mathbb{R}^{n_y \times n_y}$ is a possibly indefinite and symmetric matrix. Thus, the upper level is a convex-quadratic problem over linear constraints and the lower-level problem

$$\begin{align*}
\min_y & \quad \frac{1}{2} \tilde{y}^T G_1 \tilde{y} + d_1^T \tilde{y} \\
\text{s.t.} & \quad C x + D \tilde{y} \leq b, \quad y \leq \tilde{y}, \quad \tilde{y} \in \mathbb{R}^{n_y},
\end{align*}$$

(11)

is an $x$-parameterized and continuous but nonconvex quadratic problem. Let $\varphi(\cdot)$ be the optimal-value function of the lower level, i.e.,

$$\varphi(x) := \min_{\tilde{y}} \left\{ \frac{1}{2} \tilde{y}^T G_1 \tilde{y} + d_1^T \tilde{y} : C x + D \tilde{y} \leq b, \quad y \leq \tilde{y}, \quad \tilde{y} \in \mathbb{R}^{n_y} \right\}.$$

With this, we can rewrite Problem (10) equivalently as the single-level problem

$$\begin{align*}
\min_{x,y} & \quad \frac{1}{2} x^T H_u x + c_u^T x + \frac{1}{2} y^T G_u y + d_u^T y \\
\text{s.t.} & \quad A x + B y \leq a, \quad C x + D y \leq b, \quad x \leq \bar{x}, \quad y \leq \tilde{y}, \quad x \in \mathbb{R}^{n_x}, \quad y \in \mathbb{R}^{n_y}, \\
& \quad \frac{1}{2} y^T G_1 y + d_1^T y \leq \varphi(x),
\end{align*}$$

(12)

see, e.g., [13]. We now reformulate Problem (12) so that it fits into the framework introduced above. Therefore, we introduce the auxiliary variables $\eta_1$ and $\eta_2$ as well as the nonlinear function $f : \mathbb{R}^{n_y} \to \mathbb{R}$ with $f(y) = 1/2 y^T G_1 y + d_1^T y$. Based on this notation,
Problem (12) can be restated as
\[
\min_{x,y,\eta_1,\eta_2} \quad \frac{1}{2} x^\top H_a x + c_a^\top x + \frac{1}{2} y^\top G_a y + d_a^\top y \\
\text{s.t.} \quad Ax + By \leq a, \quad Cx + Dy \leq b, \tag{13a}
\]
\[
x \leq \bar{x}, \quad y \leq \bar{y}, \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^n, \tag{13b}
\]
\[
\eta_2 - \eta_1 \leq 0, \quad \varphi(x) = \eta_1, \quad f(y) = \eta_2, \quad \eta_1, \eta_2 \in \mathbb{R}. \tag{13c}
\]

Now, the method developed in Section 3 can be applied to (13) if (i) the nonconvex functions \( \varphi \) and \( f \) are Lipschitz continuous on the projections of the bilevel constraint region onto the decision space of the upper and lower level, respectively, i.e., on the domains
\[
F_x := \{ x \in [x, \bar{x}]: \exists y \in \mathbb{R}^n \text{ such that } Ax + By \leq a, Cx + Dy \leq b, y \leq \bar{y} \},
\]
\[
F_y := \{ y \in [y, \bar{y}]: \exists x \in \mathbb{R}^n \text{ such that } Ax + By \leq a, Cx + Dy \leq b, x \leq \bar{x} \},
\]
and if (ii) the associated Lipschitz constants are computable.

What makes things more complicated compared to the general setup described in Section 3 is that we can only evaluate the optimal-value function \( \varphi(x) \) but we cannot optimize over it. Thus, we cannot use subproblem \((S(k))\) directly to obtain a new splitting point. On the other hand, following the strategy to take the box center \( m \) as the new splitting point simplifies solving problem \((S(k))\) to evaluating \( \varphi(m) \). More precisely, using the box center corresponds to setting \( \lambda = 1/2 \). This, however, is only applicable if \( \varphi(m) \) is well-defined, which we ensure with the following assumption.

**Assumption 1.** The set \( T(x) := \{ y \in \mathbb{R}^n: Dy \leq b - Cx, y \leq \bar{y} \} \) is nonempty for all \( x \in B(x, \bar{x}) \).

This assumption implies that \(-\infty < \varphi(x) < +\infty\) for all \( x \in B(x, \bar{x}) \), i.e., a minimizer \( y \) of Problem (11) exists for all \( x \in B(x, \bar{x}) \) and, thus, for every possible box center \( m \).

Before we present the theoretical developments that are required to apply our method to the introduced class of bilevel problems, let us briefly discuss that interpreting approximate solutions of bilevel problems with nonconvex lower-level problems needs to be done with some care. In particular, it is shown in [4] that lower-level solutions that are only \( \varepsilon \)-feasible can lead to upper-level solutions that can be arbitrary far away from actual bilevel solutions. Since this also applies to our method, we later always present the difference of our solutions to the known optimal solutions of the bilevel problems in our test set.

### 4.1. Lipschitz Continuity Properties

To apply our method with the outlined modifications for the subproblem to Problem (13), it remains to show that the properties (i) and (ii) are fulfilled. We start with the nonconvex function \( f \). Since the relevant domain \( B(y, \bar{y}) \) of this function is compact, continuous differentiability of \( f \) implies global Lipschitz continuity of \( f \) on this set. Since \( B(y, \bar{y}) \) is convex and compact, the tightest Lipschitz constant can be computed by solving the optimization problem
\[
\max_{y \in B(y, \bar{y})} ||G_1 y + d_1||. \tag{14}
\]

Note that it would also be possible to compute the Lipschitz constant in (14) over the feasible set of the master problem, i.e., over the set \( F_y \). However, this involves solving an optimization problem not over a simple box but over a more complex polytope. In our computational study, we test both variants. We will denote the former method as the “fast” method and the latter as the “slow” method. In the absence of lower- and upper-level constraints except for simple variable bounds, both approaches coincide.
Next, we continue with the more difficult case of proving Lipschitz continuity of the optimal-value function $\varphi$. To this end, we exploit a variant of the Hoffman Lemma; for the original version see the main theorem in [24] or Lemma 5.8 in [50]. For the ease of presentation, we assume from now on that the finite bounds on $y$ are part of the lower-level inequality constraints and, thus, also part of the matrix $C$.

**Theorem 4.2.** Suppose Assumption 1 holds. Then, there exists $L_H > 0$ such that for any $x, \bar{x} \in B(x, \bar{x})$ it holds: For any $y \in T(x)$ we can find a point $\tilde{y} \in T(\bar{x})$ with

$$||\tilde{y} - y|| \leq L_H ||C(x - \bar{x})|| \leq L_H ||C|| ||\bar{x} - x||.$$  

The scalar $L_H$ is the so-called Hoffman constant. A sharp characterization of this constant and an algorithm to compute it can be found in [37, 40]. Based on the introduced variant of the Hoffman Lemma, we can now establish Lipschitz continuity of the optimal-value function $\varphi$ under Assumption 1. Our proof follows the idea of the proof of Corollary 5.2 in [50]. There, the Lipschitz continuity of the optimal-value function of a linear program with right-hand side perturbation is demonstrated. In contrast to this, we have a quadratic program with right-hand side perturbation.

**Theorem 4.2.** Suppose Assumption 1 holds. Then, there exists $L > 0$ such that for any $x, \bar{x} \in B(x, \bar{x})$ it holds

$$||\varphi(\bar{x}) - \varphi(x)|| \leq L ||\bar{x} - x||.$$  

**Proof.** Let any $x, \bar{x} \in B(x, \bar{x})$ be given. By Assumption 1, minimizers $y$ and $\tilde{y}$ of Problem (11) exist for $x$ and $\bar{x}$, respectively. By Lemma 4.1, for $y$ we can find a point $\tilde{y} \in T(\bar{x})$ such that

$$||\tilde{y} - y|| \leq L_H ||C|| ||\bar{x} - x||$$  

holds for some $L_H > 0$. Based on this, we can conclude

$$\varphi(\bar{x}) - \varphi(x)$$

$$\leq \frac{1}{2} \tilde{y}^\top G_1 \tilde{y} + d_1^\top \tilde{y} - \left( \frac{1}{2} y^\top G_1 y + d_1^\top y \right)$$

$$= \frac{1}{2} \tilde{y}^\top G_1 \tilde{y} - \frac{1}{2} y^\top G_1 y + d_1^\top (\tilde{y} - y)$$

$$= \frac{1}{2} \tilde{y}^\top G_1 \tilde{y} - \frac{1}{2} \tilde{y}^\top G_1 y + \frac{1}{2} \tilde{y}^\top G_1 y - \frac{1}{2} y^\top G_1 y - \frac{1}{2} y^\top G_1 \tilde{y} + d_1^\top (\tilde{y} - y)$$

$$= \frac{1}{2} \tilde{y}^\top G_1 (\tilde{y} - y) + \frac{1}{2} y^\top G_1 (\tilde{y} - y) + \frac{1}{2} y^\top G_1 y - \frac{1}{2} y^\top G_1 \tilde{y} + d_1^\top (\tilde{y} - y)$$

$$= \frac{1}{2} (\tilde{y}^\top G_1 + d_1^\top + y^\top G_1 + d_1^\top) (\tilde{y} - y) + \frac{1}{2} y^\top G_1 \tilde{y} - \frac{1}{2} y^\top G_1 \tilde{y}$$

$$\leq \frac{1}{2} \left( ||G_1 \tilde{y} + d_1|| + ||G_1 y + d_1|| \right) ||\tilde{y} - y||$$

$$\leq L_{G_1} ||\tilde{y} - y||$$

$$\leq L_H ||C|| ||\bar{x} - x||,$$

where we use the symmetry of $G_1$ and $L_{G_1} := \max \{ ||G_1 y + d_1|| : x \in B(x, \bar{x}), y \in T(x) \}$, which is well-defined due to Assumption 1.

Analogously, by Lemma 4.1, for $\tilde{y}$ we can find a point $\hat{y} \in T(x)$ such that

$$||\hat{y} - \tilde{y}|| \leq L_H ||C|| ||x - \bar{x}||$$
holds for the same $L_H > 0$. With the same arguments as before, we obtain
\[ \varphi(x) - \varphi(\tilde{x}) \leq L_H ||C||L_G||\tilde{x} - x||. \]
Consequently,
\[ |\varphi(\tilde{x}) - \varphi(x)| \leq L_H ||C||L_G||\tilde{x} - x|| =: L||\tilde{x} - x|| \]
holds and the claim follows. \[\square\]

Let us finally note that the presented method is not restricted to nonconvex but quadratic problems in the lower level in general. If one has knowledge about the Lipschitz constant of a nonconvex lower-level problem with more general nonlinearities, the method can be applied as it is explained in this section.

4.2. Implementation Details. In this section, we discuss some implementation details to clarify how we modified and extended Algorithm 1 to get a more tailored method for the considered bilevel setup.

4.2.1. “Slow” and “Fast” Method for $\varphi$. Because $T(x) \subseteq B(y, \bar{y})$ holds for all $x \in B(\bar{x}, \tilde{x})$, we immediately get
\[ L_G = \max_{x \in B(\bar{x}, \tilde{x}), y \in T(x)} ||G_1y + d_1|| \leq \max_{y \in B(y, \bar{y})} ||G_1y + d_1||. \]
Since we have to compute $L_G$, for computing the Lipschitz constant of $\varphi$, we can distinguish between the “fast” and the “slow” method for $\varphi$ as well.

4.2.2. Additional Nonlinearities. Bilinear nonlinearities of the form $x_iy_j$ in the lower- or upper-level objective function can be easily reformulated to fit in our setup. If such nonlinearities occur, e.g., in the lower level, an additional variable $y_k$ is introduced in the lower level. Moreover, the constraint $y_k = x_i$ is added to the lower level, while the nonlinear objective term $x_iy_j$ is replaced by the product $y_ky_j$. The resulting bilevel problem then fits in our setup.

4.2.3. Box Filtering. Figure 2 illustrates the case that after splitting an initial bounding box, here $[0, 3]^2$, a few times, there might be boxes (such as $[2.25, 3]^2$) that do not include any point from the bilevel constraint region, which is colored in red in Figure 2. To avoid further investigation of these boxes, we can detect these boxes by checking if the intersection of the bilevel constraint region with newly created boxes is empty. However, this requires to solve an LP feasibility problem and thus creates some additional computational effort. The benefit, however, is that constraints (6c)–(6f) and the corresponding binary variables are not added to the master problem for the filtered boxes in the given application. However, as the bilevel constraint region is also accounted for in the master problem, these filtered boxes are not further split anyway, i.e., the constraints and variables that are not added to the master problem would be redundant if added. Hence, the additional computational effort of box filtering should only be undertaken if necessary.

Note that box filtering is not necessary if there are no lower-level and upper-level constraints except for simple variable bounds since the intersection of the bilevel constraint region and every possible box created by the algorithm can never be empty. In contrast, box filtering is necessary if the “slow” method is applied since, e.g., Problem (14) for computing the Lipschitz constant of the nonconvex function $f$ might be infeasible on newly created boxes. Thus, these boxes must be filtered after each box splitting and before Lipschitz constants are updated. This does not occur with the “fast” method and box filtering is therefore not necessary for this method.
This is not necessarily true for the case of general lower-level constraints, i.e., when
we can compute tighter Lipschitz constants for smaller sets.

4.2.6. \textit{Tighter Lipschitz Constants for Box-Constrained Lower Levels.} For instances with
simple variable bounds on the lower level that are not influenced by upper-level decisions,
we can compute tighter Lipschitz constants for \( \phi \). To this end, we now explicitly take
into account bilinear terms of the form \( x_iy_j \) in the lower-level objective function and do
not reformulate these terms as described in Section 4.2.2. Hence, the lower-level objective
function is given by

\[
\phi(\hat{x}) = \frac{1}{2} y^\top \begin{bmatrix} E & F^\top \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} e \\ \hat{x} \end{bmatrix}^\top \begin{bmatrix} y \\ x \end{bmatrix},
\]

where \( E \) and \( F \) are suitably chosen matrices and \( e \) is a suitably chosen vector. Now,
for given \( \hat{x}, \hat{x} \in B(x, \hat{x}) \), let \( \hat{y} \) and \( \tilde{y} \) be defined as

\[
\hat{y} := \arg\min_{y \in B(y, \hat{y})} f(y, \hat{x}), \quad \tilde{y} := \arg\min_{y \in B(y, \hat{y})} f(y, \hat{x}), \tag{15}
\]

where \( B(y, \hat{y}) \) is the lower level’s feasible set in this case. Note that \( B(y, \hat{y}) = T(x) \) holds
for all \( x \in B(x, \hat{x}) \) because \( \hat{x} \) only influences the lower-level objective function but not
the lower level’s feasible set. Thus, it holds

\[
\varphi(\hat{x}) - \varphi(\hat{x}) = f(\hat{y}, \hat{x}) - f(\hat{y}, \hat{x}) \leq f(\tilde{y}, \hat{x}) - f(\hat{y}, \hat{x})
\]

\[
= \tilde{y}^\top F^\top \hat{x} - \tilde{y}^\top F^\top \hat{x} \leq ||F\tilde{y}||||\hat{x} - \hat{x}||.
\]

Note that \( f(\hat{y}, \hat{x}) \leq f(\tilde{y}, \hat{x}) \) holds since \( \hat{y} \) minimizes \( f(y, \hat{x}) \) over \( B(y, \hat{y}) \) and \( \tilde{y} \in B(y, \hat{y}) \).
This is not necessarily true for the case of general lower-level constraints, i.e., when
optimizing in (15) over \( T(\hat{x}) \) and \( T(\hat{x}) \), respectively, since \( T(\hat{x}) \neq T(\hat{x}) \) and \( \tilde{y} \notin T(\hat{x}) \)
might hold. Analogously, we obtain \( \varphi(\hat{x}) - \varphi(\hat{x}) \leq ||F\tilde{y}||||\hat{x} - \hat{x}|| \) and, consequently,

\[
|\varphi(\hat{x}) - \varphi(\hat{x})| \leq L_F||\hat{x} - \hat{x}||
\]

holds with \( L_F := \max\{||Fy||: y \in B(y, \hat{y})\} \). Thus, \( L_F \) is a valid Lipschitz constant.

4.2.5. \textit{Big-Ms.} As a valid Big-M in the master problem, we use the maximum of \( ||\hat{y} - y||_{\infty}, ||\hat{x} - x||_{\infty} \), and

\[
\max_{y \in B(y, \hat{y})} \frac{1}{2} y^\top G_1 y + d_1 y - \min_{y \in B(y, \hat{y})} \frac{1}{2} y^\top G_1 y + d_1 y.
\]

4.2.6. \textit{Lipschitz Constant Updates.} The Lipschitz constants are updated after each box
splitting since it is to be expected that the constants get smaller if they are computed on
smaller sets.
Table 1. Overview of the considered BASBLib instances. The number of lower- and upper-level constraints do not contain simple variable bounds and do not contain the additional constraints obtained due to the reformulation discussed in Section 4.2.2. Contrarily, the additional variables due to this reformulation are counted.

| ID       | n_x | m_u | n_y | m_l | G_l \geq 0? |
|----------|-----|-----|-----|-----|------------|
| as_1981_01 | 4   | 1   | 4   | 4   | yes        |
| b_1998_02  | 2   | 0   | 3   | 0   | no         |
| b_1998_03  | 2   | 0   | 3   | 0   | no         |
| b_1998_04  | 1   | 0   | 2   | 0   | no         |
| b_1998_05  | 1   | 0   | 2   | 0   | no         |
| d_1978_01  | 2   | 0   | 4   | 0   | yes        |
| d_2000_01  | 2   | 0   | 2   | 2   | no         |
| fl_1995_01 | 2   | 0   | 4   | 0   | yes        |
| sa_1981_01 | 1   | 1   | 2   | 1   | yes        |
| sa_1981_02 | 2   | 2   | 4   | 0   | yes        |

4.3. Numerical Results. In our computational study below, we consider the QP-QP instances from the BASBLib library [38]. We first describe which instances need to be excluded because they do not fit into our setup. First, we exclude the three instances d_1992_01, b_1984_02, and dd_2012_02 because they contain nonlinear constraints. Next, the two instances y_1996_02 and lmp_1987_01 are excluded due to nonconvex upper-level objective functions, i.e., the matrix $H_u$ is not positive semidefinite for these instances. The instance sc_1998_01 is not considered because $C$ is the zero matrix and, thus, the resulting optimization problem is not a “true” bilevel problem because the lower-level problem is not constrained by upper-level variables. Such problems can easily be solved by backwards induction, i.e., an optimal solution to the lower-level problem can first be determined and, given this lower-level optimal solution, the upper-level problem can be solved to optimality. Finally, we have to exclude the following four instances because they violate Assumption 1:

- tmh_2007_01 (lower level is not feasible for $x = 9$),
- b_1988_01 (lower level is not feasible for $x = 9$),
- b_1998_07 (lower level is not feasible for $x = 9$), and
- cw_1990_02 (lower level is not feasible for $x = 7$).

In total, 10 instances (out of 20) remain; see Table 1. Note that, due to the applied reformulations as described in Section 4.2.2, the reported number of variables might differ from those reported in the BASBLib. Moreover, the reported number of constraints does not include additional constraints necessary due to these reformulations. Variable bounds are also not counted as constraints.

We implemented the algorithm in Python 3.7.9. All computations were conducted on a machine with an Intel(R) Core(TM) i7-8565U CPU with 4 cores, 1.8 GHz to 4.6 GHz, and 16 GB RAM. The master problem (M(k)) and the subproblem (S(k)) are modeled using Pyomo 5.7.2 [22] and solved with Gurobi 9.1.0 [21]. The Hoffman constant is computed with the algorithm described in [40] using the MATLAB code made publicly available by the authors of [40].\textsuperscript{2} We set $\varepsilon = 10^{-1}$ and use a time limit of 5 h. If the instance is solved\textsuperscript{2}\textsuperscript{2}The MATLAB code can be found at http://www.andrew.cmu.edu/user/jfp/hoffman.html.
Table 2. Computational results with “slow” method and box filtering. Runtimes are given in seconds.

| ID            | $\varepsilon$ | $k$ | runtime | final $\varepsilon$ | diff to opt. | diff to opt. value |
|---------------|---------------|-----|---------|---------------------|--------------|--------------------|
| as_1981_01   | $10^{-1}$     | 166 | 3862    | 0.092              | 0.000        | 0.000              |
| as_1981_01   | $10^{-2}$     | 580 | 18,000  | 0.015              | 0.000        | 0.000              |
| d_2000_01    | $10^{-1}$     | 37  | 117     | 0.075              | 0.031        | 0.000              |
| d_2000_01    | $10^{-2}$     | 64  | 209     | 0.010              | 0.024        | 0.000              |
| d_2000_01    | $10^{-3}$     | 95  | 377     | 0.001              | 0.000        | 0.000              |
| d_2000_01    | $10^{-4}$     | 123 | 513     | 0.000              | 0.000        | 0.000              |
| d_2000_01    | $10^{-5}$     | 151 | 692     | 0.000              | 0.000        | 0.000              |
| sa_1981_01   | $10^{-1}$     | 2092| 18,000  | 0.286              | 0.378        | 3.981              |
| sa_1981_02   | $10^{-1}$     | 464 | 18,000  | 14.862             | 4.550        | 89.157             |

within the time limit, we decrease $\varepsilon$ by subsequently dividing by ten until $\varepsilon = 10^{-5}$ is reached to see how much accuracy can be reached by our algorithm in the given time limit. The obtained results for the case of box filtering and determining the Lipschitz constant with the “slow” method are summarized in Table 2, while the results for the “fast” method without box filtering are given in Table 3. Note that Table 2 contains results for 4 instances while Table 3 contains results for 10 instances. The reason is that 6 instances only have simple variable bounds so that there is no difference between the “slow” and the “fast” method as well as between box filtering and no box filtering (except for the additional computational effort due to the LP feasibility problems solved in the former case, see Section 4.2.3). In these cases, we only list the respective instances in Table 3. The tables are organized as follows. The first column states the ID of the instance and the second one states the used $\varepsilon$ for the termination criterion. The number of required iterations is denoted by $k$, “runtime” states the runtimes in seconds, and the “final $\varepsilon$” column contains the tolerance that is actually reached. Finally, the columns “diff to opt.” and “diff to opt. value” contain the 2-norm distance of our solution to the one reported in the BASBLib and the respective difference in the objective value.

Before we discuss the results in detail, let us comment on two important aspects. First, it can be expected that our method performs rather bad if there are multiplicities. To get a finer relaxation, all boxes covering the multiple solutions have to be split at least once if $\varepsilon$-feasibility is not reached by the first split. Second, it is to be expected that our method performs rather good for instances with small variable ranges, e.g., ranges such as $[0, 1]^n_x \times [0, 1]^n_y$ instead of $[0, 1000]^n_x \times [0, 1000]^n_y$, as well as small lower-level objective function ranges. In particular, the range of the lower level’s objective function is small in the case of small Lipschitz constants. The Lipschitz constant derived here for the optimal-value function is valid for all quadratic programs with right-hand side perturbations that satisfy Assumption 1. However, for specific bilevel applications much tighter Lipschitz constants might be derived by exploiting problem-specific structural properties.

In what follows, we comment on the obtained computational results in the light of these two aspect. Multiplicities are reported for instance as_1981_01. As can be seen in Table 2, for $\varepsilon = 10^{-2}$, this instance is not solved to $\varepsilon$-feasibility within the time limit despite the fact that an optimal solution is already reached. This effect is even enhanced if box filtering is deactivated and the “fast” method is used; see Table 3. In
instances by increasing Lipschitz constants of the optimal-value function.

The smallest and the size of the variable ranges. To this end, we first order the considered
instances by increasing Lipschitz constants of the function and the size of the variable ranges.

Finally, let us discuss the results in dependence of the tightness of the Lipschitz constants and the size of the variable ranges. To this end, we first order the considered instances by increasing Lipschitz constants of the function $f$: $b_{1998.02}$, $b_{1998.03}$, $d_{2000.01}$, $d_{1978.01}$, $fl_{1995.01}$, $sa_{1981.02}$, $as_{1981.01}$, $sa_{1981.01}$, $b_{1998.04}$, and $b_{1998.05}$. Indeed, the three instances with lowest Lipschitz constants are solved for the smallest $\varepsilon$ within the time limit; see Tables 2 and 3. In addition to these 3 instances,

| ID          | $\varepsilon$ | $k$ | runtime | final $\varepsilon$ | diff to opt. | diff to opt. value |
|-------------|---------------|-----|---------|---------------------|-------------|-------------------|
| as_{1981.01}| 10^{-1}       | 560 | 18000   | 47.344             | 10.440      | 0.000             |
| b_{1998.02}| 10^{-1}       | 4   | 1       | 0.025              | 0.000       | 0.000             |
| b_{1998.02}| 10^{-2}       | 8   | 5       | 0.009              | 0.004       | -0.000            |
| b_{1998.02}| 10^{-3}       | 16  | 11      | 0.001              | 0.002       | -0.000            |
| b_{1998.02}| 10^{-4}       | 40  | 36      | 0.000              | 0.000       | 0.000             |
| b_{1998.02}| 10^{-5}       | 37  | 33      | 0.000              | 0.000       | 0.000             |
| b_{1998.03}| 10^{-1}       | 5   | 3       | 0.037              | 0.000       | 0.000             |
| b_{1998.03}| 10^{-2}       | 10  | 7       | 0.008              | 0.000       | 0.000             |
| b_{1998.03}| 10^{-3}       | 14  | 8       | 0.001              | 0.001       | 0.000             |
| b_{1998.03}| 10^{-4}       | 27  | 22      | 0.000              | 0.000       | 0.000             |
| b_{1998.03}| 10^{-5}       | 84  | 102     | 0.000              | 0.000       | 0.000             |
| b_{1998.04}| 10^{-1}       | 1180| 8038    | 0.061              | 0.021       | 0.178             |
| b_{1998.04}| 10^{-2}       | 1538| 18000   | 0.058              | 0.019       | 0.150             |
| b_{1998.05}| 10^{-1}       | 1485| 18000   | 0.477              | 0.002       | 0.002             |
| d_{1978.01}| 10^{-1}       | 100 | 237     | 0.059              | 0.491       | 0.356             |
| d_{1978.01}| 10^{-2}       | 507 | 18000   | 0.029              | 0.329       | 0.237             |
| d_{2000.01}| 10^{-1}       | 179 | 193     | 0.097              | 0.000       | 0.000             |
| d_{2000.01}| 10^{-2}       | 136 | 132     | 0.007              | 0.000       | 0.000             |
| d_{2000.01}| 10^{-3}       | 177 | 272     | 0.001              | 0.000       | 0.000             |
| d_{2000.01}| 10^{-4}       | 206 | 357     | 0.000              | 0.000       | 0.000             |
| d_{2000.01}| 10^{-5}       | 385 | 1078    | 0.000              | 0.000       | 0.000             |
| fl_{1995.01}| 10^{-1}      | 570 | 18000   | 0.123              | 0.414       | 0.855             |
| sa_{1981.01}| 10^{-1}     | 330 | 205     | 0.082              | 0.203       | 2.316             |
| sa_{1981.01}| 10^{-2}     | 830 | 1221   | 0.009              | 0.068       | 0.778             |
| sa_{1981.01}| 10^{-3}     | 3120| 18000  | 0.001              | 0.018       | 0.203             |
| sa_{1981.02}| 10^{-1}     | 725 | 18000  | 0.827              | 1.060       | 20.995            |

In case of identical Lipschitz constants, alphabetical sorting is used. The ordering of the considered instances by increasing Lipschitz constants of the optimal-value function $\varphi$ is slightly different: $b_{1998.02}$.
the “fast” method with no box filtering also solves the instances \texttt{d\_1978\_01}, \texttt{sa\_1981\_01}, and \texttt{b\_1998\_04} at least for the initial $\varepsilon$. Besides having one of the lowest Lipschitz constants, the instance \texttt{d\_1978\_01} has relatively low variable ranges. The other two instances \texttt{sa\_1981\_01} and \texttt{b\_1998\_04} have relatively low number of lower- and upper-level variables, which leads to reduced dimensions of the boxes and therefore of the worst-case number of iterations. Nevertheless, the disadvantage of the large Lipschitz constant of \texttt{b\_1998\_04} is reflected in the results as for $\varepsilon = 10^{-1}$ already over 2 h are needed to compute an $\varepsilon$-feasible solution.

In total, our method solves 7 out of 10 instances. Interestingly, 2 out of the 3 unsolved instances even have a convex lower-level problem and could thus be solved with specialized methods such as, e.g., \cite{30} that explicitly exploit this property. In addition, as our method only requires very weak assumptions and can be applied to a broad range of problems besides nonconvex bilevel problems, it cannot be expected that it outperforms specifically tailored methods like, e.g., the BASBL solver \cite{39}.

5. Application to Gas Network Optimization

In this section, we use Algorithm 1 to solve stationary gas network optimization problems. We start by modeling the gas network and state an implicit nonlinear pressure law function for gas flow in pipes. We cannot state this function explicitly but it is possible to evaluate it rather cheaply. Then, we analyze its derivatives to derive suitable Lipschitz constants. Finally, numerical results on test instances show the successful application of our method.

5.1. Modeling. We model the gas network as a directed and weakly connected graph $(V, A)$, where the arcs $A$ are composed of pipes $A_{pi}$, short pipes $A_{sp}$, valves $A_{va}$, compressor stations $A_{cs}$, and control valves $A_{cv}$, i.e.,

\[ A = A_{pi} \cup A_{sp} \cup A_{va} \cup A_{cs} \cup A_{cv}. \]

The two main variables that describe the state of the gas flowing through the network are the pressure $p$ and mass flow $q$. Each node $u \in V$ has a bounded pressure variable $p_u \in [\bar{p}_u, \bar{p}_u]$ and a given mass flow $q_u$ that is supplied to or withdrawn from the network. A node $u \in V$ is called an entry node if $q_u > 0$, an exit node if $q_u < 0$, and an inner node if $q_u = 0$. In addition, each arc $a \in A$ has a variable $q_a \in [\bar{q}_a, \bar{q}_a]$ that models the mass flow in the arc.

The balance equation

\[ q_u + \sum_{a \in \delta^\text{in}(u)} q_a - \sum_{a \in \delta^\text{out}(u)} q_a = 0 \quad \text{for all } u \in V \tag{16} \]

ensures that no gas is gained or lost. Here, $\delta^\text{in}(u)$ and $\delta^\text{out}(u)$ denote the sets of in- and outgoing arcs of node $u$.

A short pipe $a = (u, v) \in A_{sp}$ directly connects its nodes $u$ and $v$. Therefore, the related pressure values coincide:

\[ p_u = p_v \quad \text{for all } a = (u, v) \in A_{sp}. \tag{17} \]

A valve $a = (u, v) \in A_{va}$ is either open or closed. If it is open, it is modeled as a short pipe. If it is closed, the mass flow $q_a$ has to be zero and the related pressures are decoupled but the corresponding pressure difference between the nodes $u$ and $v$ cannot
exceed a given value $\Delta \bar{p}_a$. This can be modeled by introducing a binary variable $o_a$ that indicates if the valve $a$ is open ($o_a = 1$) or closed ($o_a = 0$). The valve model then reads
\begin{align}
q_a & \geq q_a o_a & \text{for all } a \in A_{va}, \\
q_a & \leq \bar{q}_a o_a & \text{for all } a \in A_{va}, \\
p_u - p_v & \leq \Delta \bar{p}_a (1 - o_a) & \text{for all } a = (u, v) \in A_{va}, \\
p_v - p_u & \leq \Delta \bar{p}_a (1 - o_a) & \text{for all } a = (u, v) \in A_{va}.
\end{align}

Compressor stations $a \in A_{cs}$ have a fixed flow direction, i.e., $\bar{q}_a \geq 0$ holds, and can increase the pressure of the gas. We model this pressure increase by introducing the variable $\Delta p_a \in [0, \Delta \bar{p}_a]$. Then, the compressor station model is given by
\begin{equation}
p_v = p_u + \Delta p_a \quad \text{for all } a = (u, v) \in A_{cs}.
\end{equation}

Note that this model is a significant simplification of how compressor stations in real gas networks operate. More complicated and realistic models can be found in, e.g., [41, 45].

Control valves $a \in A_{cv}$ are modeled similar to compressor stations but decrease the gas pressure instead of increasing it. We again have $q_a \geq 0$ and
\begin{equation}
p_v = p_u - \Delta p_a \quad \text{for all } a = (u, v) \in A_{cv}
\end{equation}
with $\Delta p_a \in [0, \Delta \bar{p}_a]$.

Until now, we have modeled all gas network components in a (mixed-integer) linear way. The only components that are still missing are pipes $a \in A_{pi}$. We will describe the pressure loss in a pipe in dependence of the inflow pressure and the mass flow using a nonlinear and Lipschitz continuous function
\begin{equation}
p_v = P_{v,a}(p_u, q_a) \quad \text{for all } a = (u, v) \in A_{pi},
\end{equation}
which we will analyze in the next section.

The goal is to minimize the overall activity of the compressor stations. Therefore, the full stationary gas network optimization problem is given by
\begin{align*}
\min & \quad \sum_{a \in A_{cs}} \Delta p_a \\
\text{s.t.} & \quad \text{mass balance (16),} \\
& \quad \text{short pipe model (17),} \\
& \quad \text{valve model (18),} \\
& \quad \text{compressor station model (19),} \\
& \quad \text{control valve model (20),} \\
& \quad \text{pipe model (21),} \\
& \quad p_u \in [\bar{p}_u, \bar{p}_u] \quad \text{for all } u \in V, \\
& \quad q_a \in [\bar{q}_a, \bar{q}_a] \quad \text{for all } a \in A, \\
& \quad \Delta p_a \in [0, \Delta \bar{p}_a] \quad \text{for all } a \in A_{cs} \cup A_{cv}, \\
& \quad o_a \in \{0, 1\} \quad \text{for all } a \in A_{va}.
\end{align*}

This model has the form of Problem (1). The pipe equations (21) constitute the Lipschitz nonlinearities (1c) while the other constraints fit (1b) as they are linear. Hence, we can use Algorithm 1 to solve it.

In contrast to the gas network model considered in [47], we do not restrict ourselves to tree-structured networks. This has the consequence that the mass flows in the network cannot be pre-computed. Therefore, the nonlinearity on each arc is multivariate as it...
depends on the pressure and the mass flow. This means that Algorithm 1 can be applied to a much broader class of gas transport models.

5.2. Lipschitz Continuity of the Gas Flow Equation. In this section, we derive and analyze the nonlinear pipe model (21). For the sake of readability, we henceforth omit the subscript \( a \) that indicates the pipe \( a \in A_\text{pi} \).

Gas flow along a pipe can be modeled by the stationary momentum equation. This ordinary differential equation (ODE) reads

\[
\partial_x \left( p + \frac{\chi^2}{\rho} \right) = -\frac{1}{2} \theta |\chi| \frac{\rho}{p}, \quad \chi = \rho v, \quad \theta = \frac{\lambda}{D}
\]

where \( p, v, \chi, \) and \( \rho \) model the pressure, velocity, mass flux, and density of the gas and \( \lambda \) as well as \( D \) denote the pipe’s friction coefficient and the diameter of the pipe; see, e.g., [19]. The relation between the mass flow \( q \) and mass flux \( \chi \) is given by \( q = A \chi \) where \( A = \pi D^2/4 \) is the cross-sectional area of the pipe.

The pressure \( p \) and density \( \rho \) are coupled by the equation of state

\[
p = R_s T z \rho
\]

for real gas, where \( R_s \) denotes the specific gas constant. The compressibility factor \( z \) can be computed by the so-called AGA formula

\[
z = 1 + \alpha p, \quad \alpha = 0.257 \frac{1}{p_c} - 0.533 \frac{T_c}{p_c T} < 0
\]

with pseudocritical pressure \( p_c \), pseudocritical temperature \( T_c \), and temperature \( T \) that we assume to be constant; see, e.g., [35]. We only consider a positive compressibility factor, which is equivalent to

\[
p < \frac{1}{|\alpha|}.
\]

We further introduce the speed of sound \( c \) which is defined via

\[
\frac{1}{c^2} = \frac{\partial \rho}{\partial p}
\]

and the squared mach number

\[
\eta = \frac{v^2}{c^2}.
\]

Lemma 5.1. It holds

\[
\eta = R_s T \frac{\chi^2}{p^2}.
\]

Proof. We solve the equation of state (22) for \( \rho \) and obtain

\[
\rho = \frac{p}{R_s T z}.
\]

We can use this and the definition of the speed of sound to get

\[
\eta = v^2 \frac{1}{c^2} = \left( \frac{\chi}{\rho} \right)^2 \frac{\partial \rho}{\partial p} = \left( \frac{\chi R_s T z}{p} \right)^2 \frac{R_s T z - p R_s T \alpha}{(R_s T z)^2} = \left( \frac{\chi R_s T z}{p} \right)^2 \frac{1}{R_s T z^2} = R_s T \frac{\chi^2}{p^2}.
\]

\( \square \)
We assume that the velocity of the gas is subsonic, i.e., \( \eta < 1 \) holds, as it is the case for real-world gas networks. This is equivalent to
\[ p > |\chi| \sqrt{R_s T}. \] (24)
In what follows, we only consider pressures within the interval \((|\chi| \sqrt{R_s T}, 1/|\alpha|)\). Therefore, we have
\[ p^2 - \chi^2 R_s T > 0 \quad \text{and} \quad 1 + \alpha p > 0, \]
which we will use many times throughout this section.

For a pipe \((u,v)\) with length \(L\) and pressure \(p_u\) at node \(u\), the pressure function reads
\[ p(x, p_u, \chi) = F^{-1}\left( F(p_u) - \frac{1}{2} R_s T |\chi| \theta x \right), \] (25a)
\[ F(p) = \frac{1}{\alpha} p + \left( \chi^2 R_s T - \frac{1}{\alpha^2} \right) \ln(|1 + \alpha p|) - \chi^2 R_s T \ln(p), \] (25b)
for \(x \in [0, L]\); see [19, 20]. From [20], we further know the following properties of \(F\).

**Lemma 5.2.** The function \(F\) as defined in (25b) is differentiable for \(p \in (|\chi| \sqrt{R_s T}, 1/|\alpha|)\) with
\[ F'(p) = \frac{p^2 - \chi^2 R_s T}{p(1 + \alpha p)} > 0. \] (26)
The second derivative fulfills
\[ F''(p) = \frac{p^2 + \chi^2 R_s T(1 + 2 \alpha p)}{p^2(1 + \alpha p)^2} > 0. \]

The property in (26) implies that \(F\) is strictly increasing. Therefore, the inverse in (25a) is well-defined. To evaluate the pressure function \(p\) in (25a), the equation
\[ F(p) = F(p_u) - \frac{1}{2} R_s T |\chi| \theta x \] (27)
needs to be solved. This can be done numerically using Newton’s method since \(F\) is strictly increasing and convex; see [20]. In the same way, (27) can be solved for \(p_u\) if \(p\) and \(\chi\) are given.

We are interested in the pressure at the end of the pipe, i.e., for \(x = L\), and need to find a Lipschitz constant for
\[ p_v = p_v(p_u, \chi) := p(L, p_u, \chi). \]
To this end, we make the following assumption.

**Assumption 2.** We assume that the variables \(\chi\) and \(p_u\) of each pipe \((u,v)\) are bounded by
\[ \chi \in [\chi, \bar{\chi}] = [q/A, q/A], \quad p_u \in [p_u, \bar{p}_u] \subset \left(|\chi| \sqrt{R_s T}, \frac{1}{|\alpha|}\right) \]
with
\[ p_v(p_u, \chi) \in \left(|\chi| \sqrt{R_s T}, \frac{1}{|\alpha|}\right) \quad \text{for all} \ (p_u, \chi) \in \mathcal{F} := [p_u, \bar{p}_u] \times [\chi, \bar{\chi}] \]
The following lemma guarantees the Lipschitz continuity of \(p_v(p_u, \chi)\) on \(\mathcal{F}\).
Lemma 5.3. Let $f : F \rightarrow \mathbb{R}$ be a partially differentiable function on a compact and convex subset $F \subset \mathbb{R}^d$ with $d \in \mathbb{N}$. Then, $f$ is Lipschitz continuous on $F$ with Lipschitz constant $L = 1$ w.r.t. the weighted $1$-norm
\[
\|\tilde{x}\|_w := \sum_{i=1}^{d} |\tilde{x}_i| w_i
\]
with positive weights $w \in \mathbb{R}^d$ and $w_i \geq \max_{x \in F} |\partial_x f(x)|$ for all $i \in [d]$.

Proof. Let $\tilde{x}, \tilde{y} \in F$. We define the auxiliary function $g : [0, 1] \rightarrow \mathbb{R}$ as $g(\lambda) := f(\lambda \tilde{x} + (1 - \lambda) \tilde{y})$. Now, we use the fundamental theorem of calculus to prove the claim:
\[
|f(\tilde{x}) - f(\tilde{y})| = |g(1) - g(0)|
= \left| \int_{0}^{1} g'(\lambda) \, d\lambda \right|
= \left| \int_{0}^{1} (\tilde{x} - \tilde{y})^T \nabla f(\lambda \tilde{x} + (1 - \lambda) \tilde{y}) \, d\lambda \right|
= \left| \int_{0}^{1} \sum_{i=1}^{d} (\tilde{x}_i - \tilde{y}_i) \partial_{x_i} f(\lambda \tilde{x} + (1 - \lambda) \tilde{y}) \, d\lambda \right|
\leq \int_{0}^{1} \sum_{i=1}^{d} |(\tilde{x}_i - \tilde{y}_i) \partial_{x_i} f(\lambda \tilde{x} + (1 - \lambda) \tilde{y})| \, d\lambda
\leq \int_{0}^{1} \sum_{i=1}^{d} |\tilde{x}_i - \tilde{y}_i| \max_{x \in F} |\partial_{x_i} f(x)| \, d\lambda
\leq \|\tilde{x} - \tilde{y}\|_w. \quad \square
\]

Using this weighted $1$-norm allows to get tighter bounds in (5) compared to the usual $1$-norm. To actually compute this weighted norm one could solve $\max_{x \in F} |\partial_x f(x)|$, which is an NLP for each $i \in [d]$. In the case of Algorithm 1, the set $F$ will always be a box. For the function $p_v(p_u, \chi)$, we give the optimal solution or at least a suitable upper bound of these NLPs in the following two theorems.

Theorem 5.4. If Assumption 2 holds, the derivative $\partial_{p_u} p_v(p_u, \chi) > 0$ attains its maximum on a given box $\emptyset \neq [p_u^{\min}, p_u^{\max}] \times [\chi, \chi] \subseteq [p_u, \bar{p}_u] \times [\chi, \bar{\chi}]$ in $(p_u^{\min}, \chi^+)$ if $\chi^+ \geq 0$, or in $(p_u^{\max}, \chi^+) \text{ else.}$

Theorem 5.5. Let $F = [p_u^{\min}, p_u^{\max}] \times [\chi^-, \chi^+] \subseteq [p_u, \bar{p}_u] \times [\chi, \bar{\chi}]$ be a given box with $F \neq \emptyset$. If Assumption 2 holds, the derivative $\partial_{\chi} p_v(p_u, \chi) \leq 0$ has the following properties:
\[
\min_{(p_u, \chi) \in F \cap \mathbb{R} \times \mathbb{R} \geq 0} \partial_{\chi} p_v(p_u, \chi) = \partial_{\chi} p_v(p_u^{\min}, \chi^+)
\quad \text{if } \chi^+ \geq 0,
\min_{(p_u, \chi) \in F \cap \mathbb{R} \times \mathbb{R} \leq 0} \partial_{\chi} p_v(p_u, \chi) > \partial_{\chi} p_v(p_u^{\max}, \chi^-), -\chi^-)
\quad \text{if } \chi^- < 0.
\]

The proofs of these two theorems can be found in the appendix of this paper as they are rather long and technical.

Theorem 5.4 and Theorem 5.5 can be used to not only compute the bounds in (5) for the initial box $[p_u, \bar{p}_u] \times [\chi, \bar{\chi}]$ but also for every new box that is created in Step 15 of Algorithm 1. This can significantly tighten the bounds in (5) as the iteration proceeds.
5.3. Numerical Results. Now, we apply Algorithm 1 to two test problems from the GasLib library \[46\], which contains stationary gas network benchmark instances. To this end, we implemented our method in Python 3.8.10. The computations were done on a machine with an Intel(R) Core(TM) i7-8550U CPU with 4 cores, 1.8 GHz to 4.0 GHz, and 16 GB RAM. The master problems (M(\(k\))) and the subproblems (S(\(k\))) have been modeled using GAMS 36.2.0 \[9\]. We used the solver CPLEX 20.1.0.1 \[11\] to solve the master problems, which are MIPs, and the solver SNOPT 7.7.7 \[18\] for the subproblems, which are NLPs.

To improve the performance of our method, we detect boxes \([p_u^-, p_u^+] \times [\chi^-, \chi^+]\) in \(\mathcal{X}_k\) that lie outside the feasible set of a pipe. From \((29)\) and \((30)\) we know that this is the case if \(\chi^- \geq 0\) and \(p_v(p_u^+, \chi^-) < p_v\) holds or if \(\chi^+ \leq 0\) and \(p_v(p_u^-, \chi^+) > p_v\) holds. Additionally, we can fix the flow in pipes that are not part of a cycle. This allows us to reduce the dimension of the corresponding nonlinearities by one. To get well-scaled problems, we model all pressure values in bar instead of Pa and exclusively use mass flow values (in kg s\(^{-1}\)) instead of mass flux values. As a result, we have to scale all values of \(\partial_\chi p_v(p_u, \chi)\) that are used to compute the bounds in \((5)\) by a factor of \(10^{-5}/A\).

To compute a sufficiently large value \(M\) for the master problem constraints \((6c)-(6f)\) we use the difference between the values that can occur on the right-hand sides of the inequalities and the bounds of the variables on the left-hand side. This leads to the formula
\[
M := \max \left\{ \left(\bar{p}_u - p_u\right), \left(\bar{q} - \bar{q} \right), \left(\frac{1}{\alpha} - \bar{p}_v\right), \left(\bar{p}_v - \max\left\{\bar{q}, |\bar{q}|\right\} \sqrt{\bar{R}\bar{v}T}\right) \right\}.
\]

The first instance we solve is GasLib-11, which is shown in Figure 3. It contains 11 nodes including 3 entries and 3 exits, 8 pipes, 2 compressor stations, and a single valve. Because the network has a cycle, not all flows on all arcs are known a priori.

The second instance we solve is GasLib-24; see Figure 4. It consists of 24 nodes including 3 entries and 5 exits, 19 pipes, 3 compressor stations, a single control valve, a single short pipe, and a single resistor, which we replace by a short pipe. There are two cycles in the network.
Table 4. The runtimes and numbers of iterations of Algorithm 1 for different parameters $\lambda$. If no overall time is given, the time limit of 1000 s was reached before an $\varepsilon$-feasible solution could be found.

| Instance | $\lambda$ | # iterations | Mean iteration time (s) | Overall time (s) |
|----------|-----------|---------------|------------------------|------------------|
| GasLib-11 | 0.125      | 182           | 1.37                   | 248.76          |
| GasLib-11 | 0.25       | 89            | 0.78                   | 69.45           |
| GasLib-11 | 0.375      | 74            | 0.64                   | 47.41           |
| GasLib-11 | 0.5        | 123           | 0.85                   | 104.24          |
| GasLib-24 | 0.125      | 218           | 4.60                   | –               |
| GasLib-24 | 0.25       | 95            | 1.67                   | 158.43          |
| GasLib-24 | 0.375      | 198           | 4.53                   | 896.17          |
| GasLib-24 | 0.5        | 216           | 4.64                   | –               |

For both instances we tested the values 0.125, 0.25, 0.375, and 0.5 for the parameter $\lambda$. We chose the value $\varepsilon = 0.1$ (in bar) for the termination criterion. The resulting runtimes and numbers of iterations are listed in Table 4. In both cases, $\lambda = 0.25$ and $\lambda = 0.375$ yield better results than $\lambda = 0.125$ or $\lambda = 0.5$. For $\lambda = 0.125$ this can be explained by the fact that the boxes do not shrink fast enough when split because the splitting point can be quite close to the edge of its box. Choosing $\lambda = 0.5$, on the other hand, removes the possibility for the splitting points to be closer to the master problem solution and therefore potentially to the optimal solution of Problem (1).

For GasLib-11 the best result was achieved for $\lambda = 0.375$ for which an $\varepsilon$-feasible solution was found in 47.41 s. The method terminated after 74 iterations with a mean iteration time of 0.64 s. The mean time to solve the master problem was 0.35 s and the mean time to solve the subproblems was 0.14 s. The remaining 0.16 s were used to (re-)build the model in each iteration.

For GasLib-24 the best result was achieved for $\lambda = 0.25$ for which an $\varepsilon$-feasible solution was found in 158.43 s. With 95 iterations, the mean iteration time was 1.67 s. This mean includes 1.05 s to solve the master problem, 0.23 s to solve the subproblems, and 0.39 s to (re-)build the model in each iteration.
Figure 5 shows the progress of the maximal error over the course of the iterations for the best-case of both test instances. One can see that it falls rapidly in the first iterations. After that, it fluctuates strongly with only a slight downward trend until the threshold $\max_{i \in [p]} |f_i(x_{I_i}) - x_{r_i}| \leq \varepsilon$ is reached. This behavior is typical as the approximation progress from splitting a box in Step 15 of Algorithm 1 is reduced as the boxes get smaller. The fluctuations can be explained by the master problem solution switching to a bigger box $J_i^k \in J_i^k$ after the previous box has been refined a number of times in Step 15.

6. Conclusion

In this paper we developed a successive linear relaxation method for solving MINLPs with nonlinearities that are not given in closed form. Instead, we only assume that these multivariate nonlinearities can be evaluated and that we know their global Lipschitz constants. We illustrate the flexibility of this class of models and of our method by showing that it can be applied to bilevel optimization models with nonlinear and nonconvex lower-level problems as well as to nonconvex MINLPs for gas transport problems that are constrained by differential equations. Moreover, we proved finite termination of the method (at approximate global optimal solutions) and derived a worst-case iteration bound.

Finally, let us sketch three important topics for future research that are out of scope of this paper. First, one can weaken the assumptions made in this paper as it is done in [47]. In particular, the assumption of exact function evaluations is rather strong in some applications such as, e.g., in the case of PDE constraints. Second, one could also incorporate general-purpose local solvers to compute feasible points that lead to upper bounds. Together with lower bounds that we directly get from solving the master problem in every iteration, this could be used to obtain a gap and thus a further termination criterion besides the one based on the approximation accuracy that we used in this paper. Third, it is obvious (and also not expected) that the proposed method is not competitive with MINLP solution methods that explicitly use the structural properties of the nonlinearities that are given in closed form. However, there are many possible ways of improving the general method proposed in this paper. For instance, the incorporation of
presolving and dimension-reduction techniques would help a lot to improve the performance of the method.

Acknowledgments

The first author gratefully acknowledges the use of the services and facilities of the Energie Campus Nürnberg and financial support of the state of Bavaria. All authors thank the DFG for their support within projects A05, B08, and C08 in CRC TRR 154.

Appendix A. Proofs of Theorem 5.4 and Theorem 5.5

In this appendix, we present the proof for Theorem 5.4 and Theorem 5.5. To this end, we first need to analyze the first- and second-order derivatives of the pressure function $p(x, p_u, \chi)$. The first-order derivatives are considered in Lemma A.1, the second-order derivatives $\partial^2_p p$ and $\partial_\chi \partial_p p$ are examined in Lemma A.2 and A.3, and, finally, the second-order derivative $\partial^2_\chi p$ is investigated in Lemma A.4, A.5, and A.6.

We start with the following properties of the first-order derivatives from [20].

**Lemma A.1.** The function $p$ as defined in (25a) is differentiable for $p \in (|\chi|\sqrt{R_sT}, 1/|\alpha|)$ with

$$\frac{\partial p}{\partial p}(x, p_u, \chi) = \frac{F'(p_u)}{F'(p(x, p_u, \chi))} > 0,$$

$$\frac{\partial p}{\partial \chi}(x, p_u, \chi) = \frac{2\chi R_sT \ln \left\{ \frac{(1+\alpha p_u)\chi}{\chi \delta x} \right\} - R_sT |\chi|\theta x}{F'(p(x, p_u, \chi))}$$

for $\chi = 0$,

$$< 0,$$  
< 0,  
for $\chi \neq 0$,

(30)

and

$$\text{sign} \left( \frac{\partial p}{\partial x}(x, p_u, \chi) \right) = - \text{sign}(\chi).$$

(31)

The sign condition in (31) implies

$p_u > p(x, p_u, \chi)$ if $\chi > 0$,

$p_u < p(x, p_u, \chi)$ if $\chi < 0$,

$p_u = p(x, p_u, \chi)$ if $\chi = 0$

for $x \in (0, L]$. This can be written as

$$\text{sign}(\chi) = \text{sign}(p_u - p(x, p_u, \chi)).$$

From [19] we know that the second derivative w.r.t. $p_u$ of $p$ is given by

$$\frac{\partial^2 p}{\partial p_u^2} = \frac{(p_u^2 + R_sT \chi^2(1 + 2\alpha p_u))(p^2 - R_sT \chi^2)^2}{F'(p)p_u^2(1 + \alpha p_u)^2(p^2 - \chi^2 R_sT)^2}$$

$$- \frac{(p_u^2 + R_sT \chi^2(1 + 2\alpha p_u))(p_u^2 - R_sT \chi^2)^2}{F'(p)p_u^2(1 + \alpha p_u)^2(p^2 - \chi^2 R_sT)^2}.$$

(32)

The sign of this derivative is given by the following lemma.

**Lemma A.2.** It holds

$$\text{sign} \left( \frac{\partial^2 p}{\partial p_u^2} \right) = - \text{sign}(\chi)$$

for $p \in (|\chi|\sqrt{R_sT}, 1/|\alpha|)$ and $x \in (0, L].$
Proof. Because of (23), (24), and (26), the denominator in (32) is positive. Thus, only the numerator
\[(p_u^2 + R_s T \chi^2 (1 + 2 \alpha p_u)) (p^2 - R_s T \chi^2)^2 - (p^2 + R_s T \chi^2(1 + 2 \alpha p)) (p_u^2 - R_s T \chi^2)^2\] (33)
determines the sign of \(\frac{d^2 p}{dp_u^2}\). We note that \((p_u^2 + R_s T \chi^2 (1 + 2 \alpha p_u))\) as well as \((p^2 + R_s T \chi^2(1 + 2 \alpha p))\) are positive:
\[p_u^2 + R_s T \chi^2 (1 + 2 \alpha p_u) = (p_u^2 - R_s T \chi^2) + 2 R_s T \chi^2 (1 + \alpha p_u) > 0,\] (34a)
\[p^2 + R_s T \chi^2(1 + 2 \alpha p) = (p^2 - R_s T \chi^2) + 2 R_s T \chi^2 (1 + \alpha p) > 0.\] (34b)
It follows that both the first and the second term in (33) are positive. We rewrite them as
\[\frac{p_u^2}{p^2} (p_u^2 + R_s T \chi^2 (1 + 2 \alpha p_u)) (p^2 - R_s T \chi^2)^2 = p_u^2 \left(1 + R_s T \chi^2 \frac{1 + 2 \alpha p_u}{p_u^2}\right) (1 - \eta)^2,\]
\[\frac{p^2}{p_u^2} (p^2 + R_s T \chi^2(1 + 2 \alpha p)) (p_u^2 - R_s T \chi^2)^2 = p_u^2 \left(1 + R_s T \chi^2 \frac{1 + 2 \alpha p_u}{p_u^2}\right) (1 - \eta_u)^2\]
using the squared mach numbers
\[\eta = R_s T \chi^2 p, \quad \eta_u = R_s T \chi^2 p_u.\]
Now, if the sign of \(\chi\) is given, all terms are easily comparable except for
\[1 + R_s T \chi^2 \frac{1 + 2 \alpha p_u}{p_u^2}, \quad \frac{1 + R_s T \chi^2}{p^2}.\]
We take the derivative w.r.t. the pressure and get
\[\frac{\partial}{\partial p} \left(1 + R_s T \chi^2 \frac{1 + 2 \alpha p_u}{p_u^2}\right) = R_s T \chi^2 \frac{\partial}{\partial p} \left(\frac{1 + 2 \alpha p_u}{p_u^2}\right) = R_s T \chi^2 \frac{-2 p - 2 \alpha p^2}{p^4}\]
\[= -2 R_s T \chi^2 \frac{p(1 + \alpha p)}{p^4} < 0.\]
We can now determine the sign of \(\frac{d^2 p}{dp_u^2}\) in dependence of \(\chi\). If \(\chi = 0\), then \(p = p_u\) and, therefore,
\[\frac{d^2 p}{dp_u^2} = 0\]
holds. If \(\chi > 0\), then \(p_u > p\) and \(\eta > \eta_u\). We thus have
\[\frac{d^2 p}{dp_u^2} < 0.\]
If \(\chi < 0\), we analogously get
\[\frac{d^2 p}{dp_u^2} > 0.\]
\[\square\]
Next, we analyze the mixed second-order derivative of \(p\).

Lemma A.3. It holds
\[\frac{\partial^2 p}{\partial \chi \partial p_u} \begin{cases} 0, \quad \text{for } \chi = 0, \\ > 0, \quad \text{for } \chi \neq 0. \end{cases}\]
for \(p \in ([\chi] \sqrt{R_s T}, 1/|\alpha|)\) and \(x \in (0, L)\).
Proof. It holds
\[
\frac{\partial^2 p}{\partial \chi \partial p_u} = \frac{\partial}{\partial \chi} \left( \frac{F'(p_u)}{F'(p)} \right) = \frac{\partial}{\partial \chi} \left( \frac{(p_u^2 - \chi^2 R_s T)p(1 + \alpha p)}{(p^2 - \chi^2 R_s T)p_u(1 + \alpha p_u)} \right)
\]
\[
= \left( -2\chi R_s T p(1 + \alpha p) + (p_u^2 - \chi^2 R_s T) \left((1 + \alpha p) \frac{\partial p}{\partial \chi} + \alpha p \frac{\partial p}{\partial \chi} \right) \right) \frac{(p^2 - \chi^2 R_s T)p_u(1 + \alpha p_u)}{(p^2 - \chi^2 R_s T)^2 p_u^2(1 + \alpha p_u)^2}
\]
\[
- \frac{(p_u^2 - \chi^2 R_s T)p(1 + \alpha p) \left( 2p \frac{\partial p}{\partial \chi} - 2\chi R_s T \right) p_u(1 + \alpha p_u)}{(p^2 - \chi^2 R_s T)^2 p_u^2(1 + \alpha p_u)^2}
\]
\[
= \frac{-2\chi R_s T p(1 + \alpha p) \left((p_u^2 - \chi^2 R_s T) - (p^2 - \chi^2 R_s T) \right)}{(p^2 - \chi^2 R_s T)^2 p_u(1 + \alpha p_u)}
\]
\[
+ \frac{(p_u^2 - \chi^2 R_s T) \left((1 + 2\alpha p)(p^2 - \chi^2 R_s T) - 2p^2(1 + \alpha p) \right) \frac{\partial p}{\partial \chi}}{(p^2 - \chi^2 R_s T)^2 p_u(1 + \alpha p_u)}
\]
\[
= \frac{2\chi R_s T p(1 + \alpha p) \left((p_u^2 - \chi^2 R_s T) - (p^2 - \chi^2 R_s T) \right)}{(p^2 - \chi^2 R_s T)^2 p_u(1 + \alpha p_u)}
\]
\[
+ \frac{(p_u^2 - \chi^2 R_s T) \left((1 + 2\alpha p)(p^2 - \chi^2 R_s T) - 2p^2(1 + \alpha p) \right) \frac{\partial p}{\partial \chi}}{(p^2 - \chi^2 R_s T)^2 p_u(1 + \alpha p_u)}
\]
\[
= \frac{2\chi R_s T p(1 + \alpha p)(p_u^2 - p^2) - (p_u^2 - \chi^2 R_s T)(p^2 + \chi^2 R_s T(1 + 2\alpha p)) \frac{\partial p}{\partial \chi}}{(p^2 - \chi^2 R_s T)^2 p_u(1 + \alpha p_u)}
\]

We note that \((p^2 + \chi^2 R_s T(1 + 2\alpha p)) > 0\) holds because of (34b). Now, if \(\chi = 0\), then \(p_u = p\) and \(\frac{\partial p}{\partial \chi} = 0\) holds. It follows
\[
\frac{\partial^2 p}{\partial \chi \partial p_u} = 0.
\]

If \(\chi \neq 0\), then \(\chi(p_u^2 - p^2) > 0\) as well as \(\frac{\partial p}{\partial \chi} < 0\) holds and we get
\[
\frac{\partial^2 p}{\partial \chi \partial p_u} > 0. \quad \square
\]

From \(\frac{\partial p}{\partial \chi} = 0\) for \(\chi = 0\) and by using Schwarz’s theorem for \(\chi \neq 0\), it follows
\[
\frac{\partial^2 p}{\partial p_u \partial \chi} = \frac{\partial^2 p}{\partial \chi \partial p_u} \begin{cases} = 0, & \text{for } \chi = 0, \\ > 0, & \text{for } \chi \neq 0. \end{cases}
\]

(35)

For the second-order derivative w.r.t. \(\chi\) of \(p\), we can only give the sign for positive mass flux \(\chi\).

Lemma A.4. It holds
\[
\frac{\partial^2 p}{\partial \chi^2} < 0
\]
for \(p \in (|\chi|/\sqrt{R_s T}, 1/|\alpha|), x \in (0, L], \text{ and } \chi > 0.\)

Proof. We introduce the auxiliary function
\[
h := h(x, p_u, \chi) := R_s T \ln \left( \frac{(1 + \alpha p_u)p_u}{(1 + \alpha p_u)p} \right) + \frac{1}{2} R_s T \operatorname{sign}(\chi) \theta x
\]

(36)
to rewrite the derivative of $p$ w.r.t. $\chi$ as

$$\frac{\partial p}{\partial \chi} = -2\chi hp(1 + \alpha p) \over p^2 - \chi^2 R_s T.$$  

It holds

$$\text{sign}(h) = \text{sign}(p_u - p) = \text{sign}(\chi).$$

We can take the derivative of $h$ w.r.t. $\chi$ for $\chi \neq 0$ and get

$$\frac{\partial h}{\partial \chi}(x, p_u, \chi) = R_s T(1 + \alpha p_u)p \alpha p_u \frac{\partial p_u}{\partial \chi}(1 + \alpha p_u)p - (1 + \alpha p_u)p(1 + \alpha p_u)\frac{\partial p}{\partial \chi} \over (1 + \alpha p_u)p^2.$$  

$$= -R_s T \frac{\partial p}{(1 + \alpha p_u)\partial \chi} = 2\chi R_s Th(x, p_u, \chi) \over p^2 - \chi^2 R_s T > 0.$$  

Now, we can take the second derivative of $p$ w.r.t. $\chi$ and get

$$\frac{\partial^2 p}{\partial \chi^2} = \frac{\partial}{\partial \chi} \left(-2\chi hp(1 + \alpha p) \over (p^2 - \chi^2 R_s T)^2 \right)$$  

$$= -2 \left(hp(1 + \alpha p) + \chi p(1 + \alpha p)\frac{\partial p}{\partial \chi} + \chi h(1 + 2\alpha p)\frac{\partial p}{\partial \chi}\right) \over (p^2 - \chi^2 R_s T)^2$$  

$$+ 2\chi hp(1 + \alpha p)(2p\frac{\partial p}{\partial \chi} - 2\chi R_s T) \over (p^2 - \chi^2 R_s T)^2$$  

$$= -2h \left((p(1 + \alpha p) + \chi(1 + 2\alpha p)\frac{\partial p}{\partial \chi})(p^2 - \chi^2 R_s T) - \chi p(1 + \alpha p)(2p\frac{\partial p}{\partial \chi} - 2\chi R_s T)\right) \over (p^2 - \chi^2 R_s T)^2$$  

$$- 2\chi p(1 + \alpha p)\frac{\partial h}{\partial \chi} \over (p^2 - \chi^2 R_s T)^2.$$  

It holds $\text{sign}(h) = \text{sign}(\chi)$ and we have $(p^2 + \chi^2 R_s T(1 + 2\alpha p)) > 0$ because of (34b). For $\chi > 0$, we get $\frac{\partial p}{\partial \chi} < 0$ and $\frac{\partial h}{\partial \chi} > 0$. The claim then follows. $\square$

To compensate for the fact that the last result only holds for $\chi > 0$, we estimate the value of $\frac{\partial h}{\partial \chi}$ for $\chi < 0$ with values for $\chi > 0$.

**Lemma A.5.** It holds

$$\frac{\partial p}{\partial \chi}(x, p_u, \chi) > \frac{\partial p}{\partial \chi}(x, p(x, p_u, \chi), -\chi)$$

for $p \in (|\chi|\sqrt{R_s T}, 1/|\alpha|)$, $x \in (0, L]$, and $\chi < 0$.

**Proof.** Equation (27), i.e.,

$$F(p) = F(p_u) - \frac{1}{2} R_s T \chi |\theta x,$$

is equivalent to

$$F(p_u) = F(p) - \frac{1}{2} R_s T(-\chi) |\theta x.$$
Therefore, if \((p_u, \chi, p)\) fulfills (27), then \((p, -\chi, p_u)\) fulfills (27) as well. This means, we can write the pressure \(p_u\) as a function of \(p\) and \(\chi\):

\[
p_u(x, p, \chi) = p(x, p, -\chi).
\]

Now, we have

\[
\frac{\partial p}{\partial \chi}(x, p_u, \chi) = \frac{2\chi R_s T \ln \left( \frac{(1 + \alpha p_u)p_u}{(1 + \alpha p_u)p_u} \right) - R_s T' |\chi| \theta x}{F'(p)} = 2(-\chi) R_s T \ln \left( \frac{(1 + \alpha p_u)p_u}{(1 + \alpha p_u)p_u} \right) - R_s T' |\chi| \theta x
\]

\[
= \frac{F'(p_u) \frac{\partial p_u}{\partial \chi}(x, p, -\chi)}{F'(p)} > \frac{\partial p_u}{\partial \chi}(x, p, -\chi)
\]

because \(p_u < p\) holds for \(\chi < 0\), \(F'\) is strictly increasing in \(p\), and \(\frac{\partial p_u}{\partial \chi}(x, p, -\chi) < 0\) for \(\chi \neq 0\).

Now, for \(\frac{\partial p}{\partial \chi}(x, p(x, p_u, \chi), -\chi)\), we can determine the sign of its derivative w.r.t. \(\chi\) for \(\chi < 0\).

**Lemma A.6.** It holds

\[
\frac{\partial^2 p}{\partial \chi^2}(x, p(x, p_u, \chi), -\chi) > 0
\]

for \(p \in (|\chi|\sqrt{R_s T}, 1/|\alpha|)\), \(x \in (0, L]\), and \(\chi < 0\).

**Proof.** We know that \(p_u = p(x, p(x, p_u, \chi), -\chi)\) holds. Therefore, we can write

\[
\frac{\partial p}{\partial \chi}(x, p(x, p_u, \chi), -\chi) = \frac{2(-\chi) R_s T \ln \left( \frac{(1 + \alpha p(x, p_u, \chi))p_u}{(1 + \alpha p(x, p_u, \chi))p_u} \right) - R_s T' |\chi| \theta x}{F'(p_u)}
\]

by using the definition (36) of \(h\). Then, the second-order derivative is given by

\[
\frac{\partial^2 p}{\partial \chi^2}(x, p(x, p_u, \chi), -\chi) = p_u(1 + \alpha p_u) \frac{-2h \frac{\partial h}{\partial \chi} (p_u^2 - \chi^2 R_s T^2) - (4\chi^2 h R_s T^2)}{(p_u^2 - \chi^2 R_s T^2)^2}.
\]

Because of \(\text{sign}(h) = \text{sign}(\chi) = -1\) and \(\frac{\partial h}{\partial \chi} > 0\), the claim follows.

Now, we gathered all tools that are needed to find the optimal solution or at least a suitable upper bound for the NLPs \(\max_{x \in F} |\partial_{p_u} p_v|\) and \(\max_{x \in F} |\partial_{\chi} p_v|\) that are needed to compute the weighted norm (28) for (5).

**Proof of Theorem 5.4.** The claim is a direct consequence of Lemma A.2 and Lemma A.3. □
Proof of Theorem 5.5. The first property is a direct consequence of (35) and Lemma A.4. Now, we prove the second property. From (35) it follows that $p_u = p_u^-$ must hold in the minimum. We know from Lemma A.5 that
\[
\partial_\chi p_v(p_u^-, \chi) > \partial_\chi p_v(p_v(p_u^-, \chi), -\chi)
\]
holds for all $\chi \in [\chi^-, \chi^+] \cap \mathbb{R}_{\leq 0}$. Since we are dealing with a compact set, this also holds for the minima:
\[
\min_{\chi \in [\chi^-, \chi^+] \cap \mathbb{R}_{\leq 0}} \partial_\chi p_v(p_u^-, \chi) > \min_{\chi \in [\chi^-, \chi^+] \cap \mathbb{R}_{\leq 0}} \partial_\chi p_v(p_v(p_u^-, \chi), -\chi).
\]
Hence, the claim follows from Lemma A.6. □

References

[1] F. A. Al-Khayyal and H. D. Sherali. “On Finitely Terminating Branch-and-Bound Algorithms for Some Global Optimization Problems.” In: SIAM Journal on Optimization 10.4 (2000), pp. 1049–1057. DOI: 10.1137/S105262349935178X.

[2] I. Bajaj, A. Arora, and M. M. Hasan. “Black-Box Optimization: Methods and Applications.” In: Black Box Optimization, Machine Learning, and No-Free Lunch Theorems. Springer, 2021, pp. 35–65. DOI: 10.1007/978-3-030-66515-9_2.

[3] Y. Beck, I. Ljubić, and M. Schmidt. A Survey on Bilevel Optimization Under Uncertainty. Tech. rep. 2022. URL: http://www.optimization-online.org/DB_HTML/2022/06/3963.html.

[4] Y. Beck, M. Schmidt, J. Thürauf, and D. Bienstock. On a Computationally Ill-Behaved Bilevel Problem with a Continuous and Nonconvex Lower Level. Tech. rep. 2022. URL: https://arxiv.org/abs/2202.01033.

[5] R. Becker, D. Meidner, and B. Vexler. “Efficient numerical solution of parabolic optimization problems by finite element methods.” In: Optimization Methods and Software 22.5 (2007), pp. 813–833. DOI: 10.1080/10556780701228532.

[6] P. Belotti, C. Kirches, S. Leyffer, J. Linder, J. Luettke, and A. Mahajan. “Mixed-integer nonlinear optimization.” In: Acta Numerica 22 (2013), pp. 1–131. DOI: 10.1017/S0962492913000032.

[7] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. “An algorithmic framework for convex mixed integer nonlinear programs.” In: Discrete Optimization 15 (2008), pp. 186–204. DOI: 10.1016/j.disopt.2006.10.011.

[8] C. Buchheim, R. Kuhlmann, and C. Meyer. “Combinatorial optimal control of semilinear elliptic PDEs.” In: Computational Optimization and Applications 70.3 (2018), pp. 641–675. DOI: 10.1007/s10589-018-9993-2.

[9] M. R. Bussieck and A. Meeraus. “General algebraic modeling system (GAMS).” In: Modeling languages in mathematical optimization. Springer, 2004, pp. 137–157. DOI: 10.1007/978-1-4613-0215-5_8.

[10] A. R. Conn, K. Scheinberg, and L. N. Vicente. Introduction to derivative-free optimization. MOS-SIAM Series on Optimization. SIAM, 2009. DOI: 10.1137/1.9780898718768.

[11] CPLEX, IBM ILOG. V12. 1: User’s Manual for CPLEX. 53. 2009, p. 157.

[12] S. Dempe. “Bilevel Optimization: Theory, Algorithms, Applications and a Bibliography.” In: Bilevel Optimization: Advances and Next Challenges. Ed. by S. Dempe and A. Zemkoho. Springer International Publishing, 2020, pp. 581–672. DOI: 10.1007/978-3-030-52119-6_20.
[13] S. Dempe. *Foundations of bilevel programming*. Springer, 2002. DOI: 10.1007/b101970.

[14] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova. *Bilevel Programming Problems*. Springer, 2015. DOI: 10.1007/978-3-662-45827-3.

[15] M. A. Duran and I. E. Grossmann. “An outer-approximation algorithm for a class of mixed-integer nonlinear programs.” In: *Mathematical Programming* 36.3 (1986), pp. 307–339. DOI: 10.1007/BF02592064.

[16] R. Fletcher and S. Leyffer. “Solving mixed integer nonlinear programs by outer approximation.” In: *Mathematical Programming* 66.1 (1994), pp. 327–349. DOI: 10.1007/BF01581153.

[17] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[18] P. E. Gill, W. Murray, and M. A. Saunders. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization.” In: *SIAM Review* 47.1 (2005), pp. 99–131. DOI: 10.1137/S0036144504446096.

[19] M. Gugat, G. Leugering, A. Martin, M. Schmidt, M. Sirvent, and D. Wintergerst. “Towards Simulation Based Mixed-Integer Optimization with Differential Equations.” In: *Networks* 72.1 (2018), pp. 60–83. DOI: 10.1002/net.21812.

[20] M. Gugat, R. Schultz, and D. Wintergerst. “Networks of pipelines for gas with non-constant compressibility factor: stationary states.” In: *Computational and Applied Mathematics* 37.2 (2018), pp. 1066–1097. DOI: 10.1007/s40314-016-0383-z.

[21] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2022. URL: http://www.gurobi.com.

[22] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Sirola. *Pyomo—Optimization Modeling in Python*. Vol. 67. Springer, 2017. DOI: 10.1007/978-3-319-68821-6.

[23] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Vol. 23. Mathematical Modelling; Theory and Applications. Springer, 2009. DOI: 10.1007/978-1-4020-8839-1.

[24] A. J. Hoffman. “On approximate solutions of systems of linear inequalities.” In: *Journal of Research of the National Bureau of Standards* 49.4 (1952), pp. 263–265.

[25] R. Horst. “Deterministic global optimization with partition sets whose feasibility is not known: Application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization.” In: *Journal of Optimization Theory and Applications* 58.1 (1988), pp. 11–37. DOI: 10.1007/BF00939768.

[26] R. Horst and N. V. Thoai. “Branch-and-bound methods for solving systems of Lipschitzian equations and inequalities.” In: *Journal of Optimization Theory and Applications* 58.1 (1988), pp. 139–145. DOI: 10.1007/BF00939776.

[27] R. Horst and H. Tuy. *Global Optimization*. 2nd. Berlin: Springer Verlag, 1996. DOI: 10.1007/978-3-662-03199-5.

[28] R. Horst and H. Tuy. “On the convergence of global methods in multiextremal optimization.” In: *Journal of Optimization Theory and Applications* 54.2 (1987), pp. 253–271. DOI: 10.1007/BF00939434.

[29] R. Kannan and C. L. Monma. “On the Computational Complexity of Integer Programming Problems.” In: *Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn, October 2–8, 1977*. Ed. by R. Henn, B. Korte, and W. Oettli. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 161–172. DOI: 10.1007/978-3-642-95322-4_17.
[30] T. Kleinert, V. Grimm, and M. Schmidt. “Outer Approximation for Global Optimization of Mixed-Integer Quadratic Bilevel Problems.” In: Mathematical Programming (Series B) (2021). DOI: 10.1007/s10107-020-01601-2.

[31] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt. “A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization.” In: EURO Journal on Computational Optimization 9 (2021), p. 100007. DOI: 10.1016/j.ejco.2021.100007.

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

[33] J. Kronqvist, D. Bernal, A. Lundell, and T. Westerlund. “A center-cut algorithm for quickly obtaining feasible solutions and solving convex MINLP problems.” In: Computers & Chemical Engineering 122 (2019). 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27), pp. 105–113. DOI: 10.1016/j.compchemeng.2018.06.019.

[34] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann. “A review and comparison of solvers for convex MINLP.” In: Optimization and Engineering 2 (2019), pp. 397–455. DOI: 10.1017/s11081-018-9411-8.

[35] S. Leyffer, A. Sartenaer, and E. Wannfelle. Branch-and-refine for mixed-integer nonconvex global optimization. Tech. rep. Preprint ANL/MCS-P1547-0908, Mathematics and Computer Science Division, Argonne National Laboratory. 2008. URL: https://wiki.mcs.anl.gov/leyffer/images/1/15/SOS-OA-ANL.pdf.

[36] W. Li. “The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program.” In: Linear Algebra and its Applications 187 (1993), pp. 15–40. DOI: 10.1016/0024-3795(93)90125-8.

[37] R. Paulavicius and C. S. Adjiman. “BASLib - a library of bilevel test problems.” Version v2.2. In: Zenodo (2017). DOI: 10.5281/zenodo.897966.

[38] R. Paulavičius, J. Gao, P.-M. Kleniati, and C. S. Adjiman. “BASBL: Branch-And-Sandwich BiLevel solver. Implementation and computational study with the BASBL test set.” In: Computers & Chemical Engineering 132 (2020). DOI: 10.1016/j.compchemeng.2019.106609.

[39] J. Peña, J. Vera, and L. F. Zuluaga. An algorithm to compute the Hoffman constant of a system of linear constraints. 2018. arXiv: 1804.08418 [math.OC].

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

[41] J. Pintér. “Branch- and bound algorithms for solving global optimization problems with Lipschitzian structure.” In: Optimization 19.1 (1988), pp. 101–110. DOI: 10.1080/02331938808843322.

[42] J. Pintér. “Branch- and bound algorithms for solving global optimization problems with Lipschitzian structure.” In: Optimization 19.1 (1988), pp. 101–110. DOI: 10.1080/02331938808843322.

[43] J. Pintér. “Globally convergent methods for n-dimensional multiextremal optimization.” In: Optimization 17.2 (1986), pp. 187–202. DOI: 10.1080/02331938608843118.

[44] J. D. Pintér. Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). Springer Verlag, 1996. DOI: 10.1007/978-1-4757-2502-5.
[45] D. Rose, M. Schmidt, M. C. Steinbach, and B. M. Willert. “Computational optimization of gas compressor stations: MINLP models versus continuous reformulations.” In: Mathematical Methods of Operations Research 83.3 (2016), pp. 409–444. DOI: 10.1007/s00186-016-0533-5.

[46] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent. “GasLib—A Library of Gas Network Instances.” In: Data 2.4 (2017). DOI: 10.3390/data2040040.

[47] M. Schmidt, M. Sirvent, and W. Wollner. “A Decomposition Method for MINLPs with Lipschitz Continuous Nonlinearities.” In: Mathematical Programming 178.1 (2019), pp. 449–483. DOI: 10.1007/s10107-018-1309-x.

[48] M. Schmidt, M. Sirvent, and W. Wollner. “The Cost of Not Knowing Enough: Mixed-Integer Optimization with Implicit Lipschitz Nonlinearities.” In: Optimization Letters 16.5 (2022), pp. 1355–1372. DOI: 10.1007/s11590-021-01827-9.

[49] E. M. B. Smith and C. C. Pantelides. “Global optimisation of nonconvex MINLPs.” In: Computers & Chemical Engineering 21 (1997), S791–S796. DOI: 10.1016/S0098-1354(97)87599-0.

[50] G. Still. Lectures on Parametric Optimization: An Introduction. University of Twente, The Netherlands, 2018. URL: http://www.optimization-online.org/DB_HTML/2018/04/6587.html.

[51] M. Tawarmalani and N. V. Sahinidis. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming. Theory, Algorithms, Software, and Applications. Vol. 65. Springer Science & Business Media, 2002. DOI: 10.1007/978-1-4757-3532-1.

[52] F. Tröltzsch. Optimal control of partial differential equations. Vol. 112. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2010. DOI: 10.1090/gsm/112.

[53] H. Tuy and R. Horst. “Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and D.C. Optimization problems.” In: Mathematical Programming 41.1 (1988), pp. 161–183. DOI: 10.1007/BF01580762.