Outer Approximation for Global Optimization of Mixed-Integer Quadratic Bilevel Problems

Thomas Kleinert, Veronika Grimm, Martin Schmidt

Abstract. Bilevel optimization problems have received a lot of attention in the last years and decades. Besides numerous theoretical developments there also evolved novel solution algorithms for mixed-integer linear bilevel problems and the most recent algorithms use branch-and-cut techniques from mixed-integer programming that are especially tailored for the bilevel context. In this paper, we consider MIQP-QP bilevel problems, i.e., models with a mixed-integer convex-quadratic upper level and a continuous convex-quadratic lower level. This setting allows for a strong-duality-based transformation of the lower level which yields, in general, an equivalent nonconvex single-level reformulation of the original bilevel problem. Under reasonable assumptions, we can derive both a multi- and a single-tree outer-approximation-based cutting-plane algorithm. We show finite termination and correctness of both methods and present extensive numerical results that illustrate the applicability of the approaches. It turns out that the proposed methods are capable of solving bilevel instances with several thousand variables and constraints and significantly outperform classical solution approaches.

1. Introduction

Bilevel optimization problems are used in various applications, e.g., in energy markets [10, 17, 26, 28, 31, 36], in critical infrastructure defense [9, 13], or in pricing problems [12, 37] to model hierarchical decision processes. As such, they embed one optimization problem, the so-called lower-level problem into the constraints of a so-called upper-level problem. This leads to inherent nonconvexities, which render already linear bilevel problems with a linear upper- and lower-level problem NP-hard; see, e.g., [11, 29, 32].

In this work, we study mixed-integer quadratic bilevel problems of the form

\[
\begin{align*}
\min_{x,y} & \quad q_u(x,y) = \frac{1}{2} x^\top H_u x + c_u^\top x + \frac{1}{2} y^\top G_u y + d_u^\top y \\
\text{s.t.} & \quad Ax + By \geq a, \\
& \quad x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \quad \text{for all } i \in I := \{1, \ldots, |I|\}, \\
& \quad x_i \in \mathbb{R} \quad \text{for all } i \in R := \{|I| + 1, \ldots, n_x\}, \\
& \quad y \in \arg\min_y \left\{ q_l(\bar{y}) = \frac{1}{2} y^\top G_l \bar{y} + d_l^\top \bar{y} : Cx_1 + D\bar{y} \geq b, \ \bar{y} \in \mathbb{R}^{n_y} \right\},
\end{align*}
\]

where \(H_u \in \mathbb{R}^{n_x \times n_x}\), \(G_u \in \mathbb{R}^{n_y \times n_y}\), and \(G_l \in \mathbb{R}^{n_y \times n_y}\) are symmetric and positive semidefinite matrices. Furthermore, we have vectors \(c_u \in \mathbb{R}^{n_x}\), \(d_u \in \mathbb{R}^{n_y}\), \(d_l \in \mathbb{R}^{n_y}\), matrices \(A \in \mathbb{R}^{m_u \times n_x}\), \(B \in \mathbb{R}^{m_u \times n_y}\), \(C \in \mathbb{R}^{m_l \times n_x}\), \(D \in \mathbb{R}^{m_l \times n_y}\), as well as right-hand side vectors \(a \in \mathbb{R}^{m_u}\) and \(b \in \mathbb{R}^{m_l}\). The variables \(x = (x_I, x_R)\) denote the integer \((x_I)\) and continuous \((x_R)\) upper-level variables and \(y\) denotes the (continuous) lower-level variables.
variables. Note that we w.l.o.g. ordered the integer and continuous upper-level variables for the ease of presentation. In this setup, the upper-level problem is a convex mixed-integer quadratic problem (MIQP) and for fixed integer upper-level variables $x_I$, the lower level is a convex quadratic problem (QP), i.e., it is a parametric convex QP. In total, we are facing an MIQP-QP bilevel problem with the following key properties:

(i) All upper-level integer variables $x_I$ are bounded.
(ii) All linking variables, i.e., upper-level variables that appear in the lower-level constraints, are integer.

We note that in Problem (1), we implicitly assume that all integer upper-level variables are linking variables. However, this formulation also contains the more general case of integer upper-level variables that do not appear in the lower-level problem by setting some columns in the matrix $C$ to zero.

The main motivation of this work is to exploit the two properties above to develop a multi- and a single-tree solution approach for Problem (1) based on outer approximation for convex mixed-integer nonlinear problems (MINLP) [15, 22]. The above-mentioned two properties are required by many other state-of-the-art algorithms for linear bilevel problems with purely integer (IP-IP) or mixed-integer upper- and lower-level problems (MIP-MIP); see, e.g., [18, 20, 40, 47, 51]. These methods successfully use bilevel-tailored branch-and-bound or branch-and-cut methods to solve quite large instances of hundreds or thousands of variables and constraints. However, they cannot directly deal with continuous lower-level problems and/or have not yet been extended to the convex-quadratic case—if this is possible at all. An extension of [51] to the general class of mixed-integer nonlinear bilevel problems with integer linking variables is proposed in [38] but the computational study therein only covers mixed-integer linear bilevel problems.

Aside from mixed-integer linear bilevel problems, algorithms for various classes of continuous convex-quadratic bilevel problems have been proposed. However, in contrast to algorithms for (M)IP-(M)IP bilevel problems, reported numerical results for continuous convex-quadratic bilevel problems seem to cover only rather small instances. A branch-and-bound algorithm for bilevel problems with a convex upper-level problem and a strictly convex lower-level problem is proposed in [4]. The author demonstrates the effectiveness of his method for problems with up to 15 variables and 20 constraints on each level. In [5], a convex-quadratic lower-level problem is replaced by its Karush–Kuhn–Tucker (KKT) conditions and then a branching on the complementarity constraints is applied. The authors report results for problems with up to 60 upper-level and 40 lower-level variables. This approach is generalized from linear upper-level to convex upper-level problems in [16]. Two different descent algorithms for bilevel problems with a strictly convex lower level and a concave or convex upper level are proposed in [50]. However, the authors do not provide computational results. Recently, also neural networks are used to tackle continuous convex-quadratic bilevel problems; see [30, 39].

To the best of our knowledge, tailored algorithms for mixed-integer quadratic bilevel problems of the form (1) are neither reported nor has their efficiency been demonstrated in a comprehensive computational study. In fact, there exist no code packages that can be used as a benchmark for our proposed solution techniques. Thus, we use well-known single-level reformulations based on KKT conditions and strong duality as a benchmark in our computational study in Section 5.

Our contribution is the following. We consider bilevel problems with a mixed-integer convex-quadratic upper level and a convex-quadratic lower level. For this nonconvex problem class, we provide an equivalent reformulation to a convex MINLP that uses strong duality of the lower level; see Section 2. Further, in Section 3
we propose a multi- and a single-tree solution approach that are both inspired by outer-approximation techniques for convex MINLPs. We prove the correctness of the methods and discuss further extensions. We are not aware of any other work that applies outer-approximation techniques from the area of convex MINLP to mixed-integer bilevel programming. In Section 4, we give details on the implementation of the proposed approaches and in Section 5, we evaluate their effectiveness in an extensive numerical study, in which we solve instances with up to several thousand variables and constraints. We conclude in Section 6.

2. A Convex Single-Level Reformulation

Most solution techniques for bilevel problems rely on a reformulation of the bilevel problem to a single-level problem [11]. For problems with a convex lower level, the lower-level problem can be replaced by its nonconvex KKT conditions. Especially for problems with linear lower-level constraints, this approach is very popular, because it allows for a mixed-integer linear reformulation of the KKT complementarity conditions using additional binary variables and big-M values; see, e.g., [23]. With this approach the bilevel problem (1) can be equivalently transformed to the following mixed-integer linear single-level problem

\[
\min_{x, y, u, \lambda} \quad q_u(x, y)
\]

s.t. \(Ax + By \geq 0\),

\(u \in \{0, 1\}^m\),

\(0 \leq Cx_I + Dy - b \leq M_1u\),

\(0 \leq \lambda \leq M_2(1 - u)\),

\(G_y + d_l = D^T\lambda\)

where \(M_1\) and \(M_2\) are sufficiently large numbers. Similarly, a strong-duality-based reformulation replaces the lower level by primal and dual feasibility conditions and a strong-duality equation. This approach is significantly less used in practice because even for linear-linear bilevel problems one obtains nonconvex bilinear terms due to products of primal upper-level and dual lower-level variables. These terms can only be linearized if all linking variables are integer. Recently, in [52], a numerical study is provided that compares the KKT approach with the strong-duality approach for linear bilevel problems with integer linking variables and linear lower-level problems. The authors conclude that the strong-duality reformulation works significantly better than the KKT reformulation for problems with large lower-level problems. Other contributions that successfully apply the strong-duality reformulation to linear bilevel problems include, e.g., [2, 24, 25, 27, 37, 41]. Except from Bard, who briefly sketches the idea in [4], we are not aware of any works that use strong duality for bilevel problems with quadratic lower levels. One reason might be that the resulting strong-duality equation is quadratic. Opposed to the linearized KKT reformulation, the strong-duality-based reformulation thus yields a quadratically constrained program. Despite this drawback, we derive such a strong-duality-based single-level reformulation after introducing some notation.

2.1. General Notation. The bilevel constraint region is denoted by

\[ P := \{(x, y) : Ax + By \geq a, \quad Cx_I + Dy \geq b, \quad x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \text{ for all } i \in I\}. \]

Throughout this paper, we assume that \(P\) is bounded. This set corresponds to the set obtained by relaxing the optimality of the lower-level problem. Its projection onto the decision space of the upper level is given by

\[ P_u := \{x : \exists y \text{ such that } (x, y) \in P\}. \]
For fixed \( x \in P_u \), the lower-level feasible region is given by
\[
P(x_I) = \{ y : Dy \leq b - Cx_I \}.
\]
The rational reaction set of the lower level reads
\[
M(x_I) = \arg\min\{q(y) : y \in P(x_I)\}.
\]
Since \( G_I \) is semidefinite, the lower level may not have a unique solution, i.e., \( M(x_I) \) may not be a singleton. In such a case, we assume the optimistic bilevel solution, i.e., \( y \in M(x_I) \) is chosen in favor of the upper level; see, e.g., Chapter 1 in [11].

In Problem (1), this is indicated by “\( \max_{x,y} \)” in the upper-level objective, i.e., the upper-level maximizes over \( x \) and \( y \). Further, note that our solution approach explicitly allows for \( M(x_I) = \emptyset \) for some \( x \in P_u \). Finally, the bilevel feasible set is given by
\[
\mathcal{F} = \{(x,y) : x \in P_u, y \in M(x_I)\}.
\]
If \( \mathcal{F} = \emptyset \), then Problem (1) is infeasible.

2.2. Strong-Duality-Based Nonconvex Single-Level Reformulation. We now use strong duality of the lower-level problem to transform Problem (1) into an equivalent nonconvex single-level problem. The parametric Lagrangian dual problem of the parametric lower level
\[
\min_y q(y) \quad \text{s.t.} \quad Dy \geq b - Cx_I
\]
is given by
\[
\max_{\lambda \geq 0} g(x_I; \lambda),
\]
with \( g(x_I; \lambda) = \inf_y \mathcal{L}(x_I; y, \lambda) \); see [8]. In our setup, the Lagrangian \( \mathcal{L} \) is given by
\[
\mathcal{L}(x_I; y, \lambda) = \frac{1}{2} y^\top G_I y + d_I^\top y - \lambda^\top (Cx_I + Dy - b).
\]
Since \( \mathcal{L}(x_I; y, \lambda) \) is convex and differentiable in \( y \), the infimum is given by
\[
\nabla_y \mathcal{L}(x_I; y, \lambda) = G_I y + d_I - D^\top \lambda = 0.
\]
In order to denote the Lagrangian dual problem in its general form (4), we could use Expression (6) to obtain \( y = G_I^{-1}(D^\top \lambda - d_I) \). This can then be used to substitute the primal variable \( y \) in the Lagrangian (5) to obtain
\[
g(x_I; \lambda) = -\frac{1}{2} (D^\top \lambda - d_I)^\top G_I^{-1} (D^\top \lambda - d_I) - (Cx_I - b)^\top \lambda.
\]
However, this only works if \( G_I \) is regular, e.g., if \( G_I \) is strictly definite. In the more general case of semidefinite matrices, we can explicitly keep the primal variable \( y \) and substitute \( D^\top \lambda = G_I y + d_I \) in the Lagrangian (5). This yields the dual problem
\[
\max_{y,\lambda} \bar{g}(x_I; y, \lambda) = -\frac{1}{2} y^\top G_I y - (Cx_I - b)^\top \lambda
\]
s.t. \( G_I y + d_I = D^\top \lambda, \lambda \geq 0 \).

Note that \( \bar{g}(x_I; \cdot) \) is a concave-quadratic function in \( y \) and \( \lambda \) because \( G_I \) is positive semidefinite. Thus, the dual problem (7) is a parametric concave-quadratic maximization problem over affine-linear constraints. Since Problem (7) does not involve the inverse \( G_I^{-1} \), we use formulation (7) also in the case when \( G_I \) is strictly positive definite.

In the following, we only consider \( x \in P_u \), i.e., upper-level variables for which the parametric lower-level problem is feasible. The parametric lower-level problem (3) is a convex-quadratic minimization problem over affine-linear constraints. Consequently, duality conditions apply without requiring additional constraint qualifications; see
With this we obtain works best in a bilevel context. We express the integer variables
This strong-duality inequality is convex in
variables $x$ are integer, this product can be reformulated using a binary expansion.
For the ease of presentation, we w.l.o.g. assume in the following that
Here, $s_j = \log_2(x_j^*) + 1$ many auxiliary binary variables $s_{jr}$:
With this we obtain
Now, we replace the binary-continuous products $s_{jr}$ by introducing auxiliary continuous variables $w_{jr}$, and enforce
by the additional constraints
In this formulation, we need bounds $\bar{X}_j \geq \sum_{i=1}^{m_1} c_{ij} \lambda_i$, and $\bar{X}_j \leq \sum_{i=1}^{m_1} c_{ij} \lambda_i$ which are, in practice, often derived by some suitable big-M. The need for such bounds is also a major drawback in the KKT-based reformulation (2). In [43], it is shown that wrong
big-Ms can lead to suboptimal solutions or points that are actually bilevel infeasible. Unfortunately, even verifying that the bounds are correctly chosen is, in general, at least as hard as solving the original bilevel problem; see [34]. Thus, if possible, big-Ms should be derived using problem-specific knowledge. However, we point out that our solution approaches introduced in Section 3 compute bilevel-feasible points independent of the big-Ms. Thus, in case of too small big-Ms, our algorithms may terminate with a suboptimal solution but never compute bilevel-infeasible points. We discuss this also later in Section 4.3.

Using (11) and (13), we rewrite Constraint (9) as
\[ \hat{c}(y, \lambda, w) := y^\top G_l y + d_l^\top y - b^\top \lambda + \sum_{j=1}^{|I|} \sum_{r=1}^{r_j} 2^{r-1} w_{jr} \leq 0, \] (14)
which is convex in \( y \) and linear in \( \lambda \) and \( w \). Note that \( 2^{r-1} \leq x^+_j \) holds, i.e., for reasonable bounds \( x^+_j \), the exponential coefficients \( 2^{r-1} \) can be considered as numerically stable.

Finally, the single-level problem (10) can be stated equivalently as
\[
\begin{align*}
\min_{x, y, \lambda, w, s} & \quad q_u(x, y) \\
\text{s.t.} & \quad (x, y) \in P, \\
& \quad G_l y + d_l = D^\top \lambda, \quad \lambda \geq 0, \\
& \quad \lambda^\top C x_l \quad \text{linearization: (11), (13),} \\
& \quad \text{Strong duality: (14).}
\end{align*}
\] (15)
This is a convex MIQCQP that has more binary variables and constraints compared to the nonconvex MIQCQP (10). We denote the feasible set of Problem (15) by \( \Omega \), feasible points by \((x, y, \lambda, w, s) \in \Omega \) and optimal points by \((x, y)^*\). By construction of Problem (15), we have the following equivalence result.

**Lemma 1.** The feasible set of Problem (15) projected on the \((x, y)\)-space equals the bilevel feasible set of the bilevel problem (1), i.e., \( \mathcal{F} = \text{Proj}_{(x, y)}(\Omega) \) holds. In addition, for every global optimal solution \((x^*, y^*, \lambda^*, w^*, s^*)\) of Problem (15), \((x^*, y^*)\) is a global optimal solution for Problem (1) and every global optimal solution \((x^*, y^*)\) of Problem (1) is part of an optimal solution \((x^*, y^*)\) of Problem (15).

3. Two Outer-Approximation Solution Approaches

Problem (15) is an MIQCQP in which the single quadratic constraint is convex. Such problem classes can be solved, e.g., directly by modern solvers like Gurobi or CPLEX. On the other hand, Problem (15) belongs to the broader class of convex MINLPs. For such problems, a variety of approaches exist, e.g., nonlinear branch-and-bound or multi- and single-tree methods based on outer approximation, generalized Benders decomposition, and extended cutting-planes; see [6] for a detailed survey of these methods. In this section, we introduce outer-approximation techniques that are tailored for mixed-integer quadratic bilevel problems. The general idea is to relax the convex-quadratic strong-duality inequality (14) of the lower level in Problem (15) to obtain an MIQP. Strong duality is then resolved by iteratively adding linear outer-approximation cuts. In its simplest form, this is a direct application of Kelley’s cutting-planes approach [33], which would add linear outer-approximation cuts until the strong-duality inequality (14) is satisfied up to a certain tolerance. Our preliminary numerical results indicate that this requires an enormous amount of iterations. We thus discuss a more sophisticated multitree approach and its single-tree variant in Sections 3.1 and 3.2.
3.1. A Multitree Outer-Approximation Approach. The well-known multitree outer approximation for convex MINLPs was first proposed in [15] and has been enhanced in [7, 22]. It alternatingly solves a mixed-integer linear master problem and a convex nonlinear problem (NLP) as a subproblem. The master problem is a mixed-integer linear relaxation of the original convex MINLP and is tightened subsequently by adding linear outer-approximation cuts for the convex nonlinearities. The convex nonlinear subproblem results from fixing all integer variables to the solution of the master problem in the original convex MINLP. Under suitable assumptions, every feasible integer solution of the master problem is visited at most once, and the algorithm terminates after a finite number of iterations with the correct solution. The algorithm proposed in this subsection is very much inspired by this scheme.

The master problem that we solve in every iteration \( p \geq 0 \) is given by

\[
\min_{x,y,\lambda,w,s} q_u(x, y) \\
\text{s.t.} \quad (x,y) \in P, \\
G_1 y + d_1 = D^\top \lambda, \quad \lambda \geq 0, \\
\lambda^\top C x_I \text{ linearization: (11), (13)}, \\
\hat{c}(\hat{y}^\ell; y, \lambda, w) \leq 0, \quad \ell = 0, \ldots, p - 1,
\]

where \( \hat{c}(\hat{y}; y, \lambda, w) \leq 0 \) is the linear outer approximation cut that is added to the master problem after every iteration. Thus, in iteration \( p \), the master problem \( (M_p) \) contains \( p \) outer-approximation cuts. We shed some more light on \( \hat{c} \) in a bit, but first emphasize that Problem \( (M_p) \) is a convex MIQP. This is in contrast to the standard outer-approximation literature, where the objective function is relaxed and iteratively approximated as well, resulting in a mixed-integer linear master problem. The rationale is that, in our implementation, the main working horse is a state-of-the-art solver like Gurobi or CPLEX. In recent years, these solvers made significant progress in solving convex MIQPs effectively. Thus, we want to exploit these highly evolved solvers as much as possible.

Now, we give some more details on the linear function \( \hat{c} \), which is derived by the first-order Taylor approximation of the convex strong-duality inequality (14): For a general convex function \( h(v) \), the first-order Taylor approximation at \( \bar{v} \) reads

\[
h(\bar{v}) + \nabla_v h(\bar{v})^\top (v - \bar{v}).
\]

Applied to (14), this gives

\[
\hat{c}(\hat{y}; y, \lambda, w) := 2\hat{y}^\top G_1 y + d_1^\top y - b^\top y + \sum_{j \in I} \sum_{r=1}^{r_j} 2^{r-1} w_{j,r} - \hat{y}^\top G_1 \hat{y}. \tag{16}
\]

Since \( \hat{c}(y, \lambda, w) \) is linear in \( \lambda \) and \( w \), the first-order Taylor approximation (16) is parameterized solely by \( \hat{y} \). The effectiveness of the proposed outer-approximation approach will depend on the actual selection of \( \hat{y} \), but \( \hat{c}(\hat{y}; y, \lambda, w) \leq 0 \) is a valid inequality no matter how \( \hat{y} \) is obtained. This is shown in the following lemma, in which \( M^p \) denotes the feasible set of \( (M_p) \).

**Lemma 2.** For every iteration \( p \geq 1 \), \( \Omega \subseteq M^p \subseteq M^{p-1} \) holds.

The consequence of Lemma 2 is that every master problem \( (M_p) \) is a relaxation of the single-level reformulation (15).

**Proof.** Let \( z = (x, y, \lambda, w, s) \in \Omega \) be a feasible point of Problem (15). In particular, \( z \) fulfills strong duality, i.e., \( \hat{c}(y, \lambda, w) = 0 \). Obviously, \( z \in M^p \) holds because \( M^0 \) corresponds exactly to \( \Omega \) without the strong-duality inequality (14). In addition,
since \( \hat{c} \) is convex, its first-order Taylor approximation is a global underestimator at any point \( \bar{y} \), i.e.,

\[
\hat{c}(\bar{y}; y, \lambda, w) \leq \hat{c}(y, \lambda, w) = 0.
\]

This holds in particular for the choice \( \bar{y} = \bar{y}^\ell \) for any \( \ell = 1, \ldots, p \) and \( p \geq 1 \). Hence, \( z \in M^p \). Further, \( M^p \subseteq M^{p-1} \) follows by construction. \( \square \)

In the following, we give details on how to select the linearization points \( \bar{y} \). We therefore assume that the master problem (M\( ^p \)) is feasible and denote its solution by \( z^p = (x^p, y^p, \lambda^p, w^p, s^p) \). According to Kelley’s cutting plane approach [33], one would add an outer-approximation cut (16) at the solution of the master problem, i.e., one would add the inequality \( \hat{c}(y^p; y, \lambda, w) \leq 0 \). Kelley has shown that this yields a convergent approach. However, in many cases this will turn out to be inefficient, because these cuts are rather weak. The outer approximation methods in the spirit of [7, 15, 22] additionally solve a convex nonlinear subproblem that results from fixing the integer variables in the original convex MINLP (or an auxiliary feasibility problem if the subproblem is infeasible) to obtain suitable linearization points \( \bar{y} \). In our context, the subproblem is given by fixing \( x_I = x_I^\ell \) and \( s = s^\ell \) in the convex MINLP (15), which yields the convex QCQP:

\[
\min_{x^p_I, y^p, \lambda^p, w^p} q_u(x^p_I, x^p_I, y^p) \quad \text{s.t.} \quad (x^p_I, x^p_I, y^p) \in P, \quad G_I y + d_I = D^\top \lambda, \quad \lambda \geq 0, \quad w_{jr} = s^p_{jr} \sum_{i=1}^{m_j} c_{ij} \lambda_i, \quad j \in I, \ r \in [r_j], \quad \hat{c}(y, \lambda, w) \leq 0.
\]

We denote the feasible set of (S\( ^p \)) by \( S^p \) and first assume \( S^p \neq \emptyset \). The infeasible case is discussed afterward. Let \( (\bar{x}^p_R, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p) \) be the solution of the subproblem (S\( ^p \)). For the correctness of our proposed algorithms, we need a technical assumption that is also used in [7, 15, 22].

**Assumption 1.** For every feasible subproblem (S\( ^p \)), the Abadie constraint qualification holds at the solution \( (\bar{x}^p_R, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p) \).

The following lemma now shows that it is indeed a good idea to linearize the strong-duality inequality (14) at the solution of the subproblem \( \bar{y}^p \) instead of at the solutions of the master problem \( y^p \).

**Lemma 3.** Let \( z^p = (x^p_I, x^p_R, y^p, \lambda^p, w^p, s^p) \) be an optimal solution of the master problem (M\( ^p \)) and assume that the subproblem (S\( ^p \)) is feasible and has the optimal solution \( (\bar{x}^p_R, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p) \). Suppose further that Assumption 1 holds and consider the new master problem that is obtained by adding the outer-approximation cut \( \hat{c}(\bar{y}^p; y, \lambda, w) \leq 0 \) to (M\( ^p \)). Then, for any feasible point of the form \( z = (x^p_I, x^p_R, y, \lambda, w, s^p) \) of this problem the following holds:

\[
q_u(x^p_I, x^p_R, y) \geq q_u(x^p_I, x^p_R, \bar{y}^p).
\]

For the proof of this lemma we need some theory about standard cones of nonlinear optimization. The required basics can be found, e.g., in [42].

**Proof.** We consider \( x^p_I \) and \( w^p \) fixed and assume that (S\( ^p \)) is feasible with optimal solution \( (\bar{x}^p_R, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p) \). Thus, \( (\bar{y}^p, \bar{\lambda}^p, \bar{w}^p) \) fulfills the convex strong-duality inequality (14). Since weak duality holds anyway, we obtain \( \hat{c}(\bar{y}^p, \bar{\lambda}^p, \bar{w}^p) = 0 \). Now, let \( z = (x^p_I, x^p_R, y, \lambda, w, s^p) \) be feasible for (M\( ^p \)) and let \( \hat{c}(\bar{y}^p; y, \lambda, w) \leq 0 \), i.e., \( z \) is
feasible for a suitably chosen master problem in iteration $p + 1$. In the following we abbreviate the vector $v = (y, \lambda, w)$ for the ease of presentation. Then, we have
\[ 0 \geq \hat{c}(\bar{y}^p; v) = \hat{c}(\bar{y}^p) + \nabla_v \hat{c}(\bar{y}^p)(v - \bar{v}^p), \]
i.e., $\nabla_v \hat{c}(\bar{y}^p)^\top(v - \bar{v}^p) \leq 0$, which means that $(v - \bar{v}^p)$ is in the linearized tangent cone $T^{lin}_{S^p}(\bar{y}^p)$. Due to Assumption 1, $T^{lin}_{S^p}(\bar{y}^p)$ equals the tangent cone $T_{S^p}(\bar{v}^p)$, which gives $(v - \bar{v}^p) \in T_{S^p}(\bar{v}^p)$. For all directions $d \in T_{S^p}(\bar{v}^p)$, we know that the property
\[ \nabla_v q_a(x^p_I, x^p_{R}, \bar{y}^p)^\top d \geq 0 \] holds. Thus, we have
\[ q_a(x^p_I, x^p_{R}, \bar{y}^p) \geq q_a(x^p_I, x^p_{R}, \bar{y}^p) + \nabla_v q_a(x^p_I, x^p_{R}, \bar{y}^p)(v - \bar{v}^p) \geq q_a(x^p_I, x^p_{R}, \bar{y}^p). \]

The first inequality follows because $q_a$ is convex, i.e., its first-order Taylor approximation is a global underestimator. The second inequality follows from Inequality (17).

In contrast to the slightly different setting in [22], using the solution of the subproblem (S$^p$) as the linearization point of the outer-approximation inequality (14) does not explicitly cut off the related integer solution $x^p_I$. The reason is our modified master problem, that does not linearize and approximate the convex objective function. Nevertheless, Lemma 3 lets us conclude that every integer assignment that satisfies the strong duality. Since the subproblem (S$^p$) needs to be visited only once, because the objective cannot be improved by visiting such a solution for a second time. This will be one of the key properties to prove convergence of our algorithm.

We now consider the case of an infeasible subproblem (S$^p$). In [15], it is argued that in order to eliminate $x^p_I$ from further consideration, an integer no-good-cut must be introduced. In our application, this is a straightforward task. The subproblem (S$^p$) is fully parameterized by fixed upper-level variables $x^p_I$. For these variables we have a binary expansion available anyway (the variables $s$) so that a simple binary no-good-cut on $s$ can be used. However, such no-good-cuts are known to cause numerical instabilities. As a remedy, in [22], it is proposed to derive cutting planes from an auxiliary feasibility problem that indeed cut off the integer solution $x^p_I$. The feasibility problem minimizes the constraint violations of the infeasible subproblem in some suitable sense, e.g., via the $\ell^1$- or the $\ell^\infty$-norm. Recap that $z^p$ is a solution of the master problem (M$^p$). In particular, $(y^p, \lambda^p)$ is primal-dual feasible for the lower-level problem (3) with fixed $x^p_I$. Thus, the latter problem also has an optimal solution that fulfills strong duality. Since the subproblem (S$^p$) is infeasible, this optimal solution must be infeasible for the upper-level constraints. On the other hand, $z^p$ must be feasible for the subproblem (S$^p$) without the strong-duality inequality, because $z^p$ is feasible for Problem (M$^p$). Thus, a simple feasibility problem in the sense of [22] is given by
\[
\min_{x^p_I, y, \lambda, w} \hat{c}(y, \lambda, w)
\text{ s.t. } (x^p_I, x^p_{R}, y) \in P, \\
G_I y + d_I = D^\top \lambda, \quad \lambda \geq 0,
\]
\[ w_{jr} = s^p_{jr} \sum_{i=1}^{m_j} c_{ij} \lambda_i, \quad j \in I, \quad r \in [\bar{r}_j], \tag{FP} \]
whose objective value is strictly greater than zero, since otherwise the subproblem (S$^p$) would be feasible. For a solution of (FP), we obtain the following proposition by adapting Lemma 1 of [22].
Lemma 4. Let $z^p$ be a solution of the master problem (MP), let the subproblem (SP) be infeasible, and let $(\bar{x}^p_I, \bar{y}^p, \lambda, \bar{w})$ be a solution of the feasibility problem (FP). Then, $\bar{c}(\bar{y}^p, \lambda, \bar{w}) > 0$ and every $z = (x^p_I, x^p_R, y, \lambda, w, s^p) \in \mathcal{M}^p$ is infeasible for the constraint

$$\bar{c}(\bar{y}^p, y, \lambda, w) \leq 0. \tag{18}$$

Proof. Consider a fixed $x^p_I$ and assume (SP) to be infeasible, which means that (FP) has an optimal solution $(\bar{x}^p_I, \bar{y}^p, \lambda^p, \bar{w}^p)$ with $\bar{c}(\bar{y}^p, \lambda^p, \bar{w}^p) > 0$. For the ease of presentation, we again use the abbreviation $v = (y, \lambda, w)$ and we rewrite the linear constraint set of (FP) to obtain

$$\min_{x^p_R, v} \bar{c}(v) \quad \text{s.t.} \quad \bar{A}x^p_R + \bar{B}v \leq \bar{b}, \quad \bar{D}v = \bar{d}. \tag{19}$$

Problem (FP), and hence Problem (19), minimizes a convex function over affine-linear constraints. Thus, $(\bar{x}^p_I, \bar{v}^p)$ fulfills the KKT conditions of Problem (19), i.e., primal feasibility, stationarity, non-negativity of multipliers of inequality constraints, and complementarity:

$$\bar{A}x^p_R + \bar{B}v^p \leq \bar{b}, \quad \bar{D}v^p = \bar{d}, \tag{20a}$$

$$\bar{A}^T \alpha = 0, \tag{20b}$$

$$-\bar{B}^T \alpha - \bar{D}^T \delta = \nabla_v \bar{c}(v^p), \tag{20c}$$

$$\alpha \geq 0, \tag{20d}$$

$$\alpha^T \bar{A}x^p_R + \alpha^T \bar{B}v^p = \alpha^T \bar{b}. \tag{20e}$$

We recap that $\bar{c}$ is derived from the first-order Taylor approximation, i.e., it holds

$$\bar{c}(\bar{y}^p; v) = \bar{c}(v^p) + \nabla_v \bar{c}(v^p)^\top (v - v^p).$$

This can be expanded to

$$\bar{c}(\bar{y}^p; v) = \bar{c}(v^p) - (\bar{B}^T \alpha + \bar{D}^T \delta)^\top (v - v^p) \tag{21a}$$

$$= \bar{c}(v^p) - \alpha^T \bar{B}v + \alpha^T \bar{B}v^p - \delta^\top (\bar{D}v - \bar{D}v^p) \tag{21b}$$

$$= \bar{c}(v^p) - \alpha^T (\bar{B}v - \bar{b}) - \delta^\top (\bar{D}v - \bar{d}), \tag{21c}$$

by using KKT stationarity (20c) (for (21a)) and re-ordering the terms (for (21b)). Further, we replaced $\alpha^T \bar{B}v^p$ by $\alpha^T \bar{b}$ according to (20e) and (20b) and $\bar{D}v^p$ by $\bar{d}$ according to (20a) to obtain (21c). Now, let $z = (x^p_I, x^p_R, v, s^p) \in \mathcal{M}^p$. This implies that $v$ must be feasible for (FP), respectively (19). In particular, we know that $\bar{D}v - \bar{d} = 0$. Applying this to (21) and using (20b) yields

$$\bar{c}(\bar{y}^p; v) \geq \bar{c}(v^p) + \alpha^T \bar{A}x^p_R = \bar{c}(v^p) > 0.$$

Thus, $v$ violates (18). \hfill \Box

We recap that, with the last lemma, we can derive cutting planes that cut off $x^p_I$ when the subproblem (SP) is infeasible. This is another key property for the convergence of our approach. With this result, we are ready to present the multitree outer approximation in Algorithm 1. In every iteration $p$, Algorithm 1 first solves the master problem (MP) to obtain a solution $z^p$. According to Lemma 2, $\mathcal{M}^p \subseteq \mathcal{M}^{p-1}$ and we can update the lower bound $\phi$ by $\phi_a(x^p_I, x^p_R, y^p)$. Next, either the subproblem (SP) or, in case of infeasibility, the feasibility problem (FP) is solved. If the subproblem is feasible with solution $(\bar{x}^p_I, \bar{y}^p, \lambda^p, \bar{w}^p)$, then the point $(x^p_I, x^p_R, y^p, \lambda^p, \bar{w}^p, s^p)$ is feasible for the convex single-level reformulation (15) and the upper bound is updated if $\phi_a(x^p_I, x^p_R, y^p) < \Phi$. In this case, also $z^*$ is updated. We terminate when $\phi \geq \Phi$ is achieved and return the best solution $z^*$. We now show the correctness of this approach.
Algorithm 1 Multitree Outer Approximation for MIQP-QP Bilevel Problems.

1: Initialize $\phi = -\infty$, $\Phi = \infty$, and $p = 0$.
2: while $\phi < \Phi$ do
3:    Solve the master problem $(M^p)$.
4:    if $(M^p)$ is infeasible then
5:        Return “The bilevel problem is infeasible”.
6:    else
7:        Let $z^p$ be the optimal solution of $(M^p)$ and set $\phi = q_a(x^p, y^p)$.
8:    end if
9:    Solve the subproblem $(S^p)$, or the feasibility problem $(F^p)$ if $(S^p)$ is infeasible,
10:   and obtain $(\bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p)$.
11:   if $(S^p)$ is feasible and $q_a(x^*_f, \bar{x}_R^p, \bar{y}^p) < \Phi$ then
12:       Set $z^* = (x^*_f, \bar{x}_R^p, \bar{y}^p, \bar{\lambda}^p, \bar{w}^p, s^p)$ and $\Phi = q_a(x^*_f, \bar{x}_R^p, \bar{y}^p)$.
13:   end if
14:   Add the outer approximation cut $\tilde{c}(\bar{y}^p; y, \lambda, w) \leq 0$ to $(M^p)$.
15:   Set $p ← p + 1$.
16: end while
17: Return $z^*$.

Theorem 1. Algorithm 1 terminates after a finite number of iterations at an optimal solution of Problem (1) or with an indication that the problem is infeasible.

The following proof is adapted from [22].

Proof. First of all, note that all master problems are bounded since $P$ is assumed to be bounded. We next show finiteness of Algorithm 1. If Problem (1) is feasible, then we can follow from Lemma 3 that at most one integer solution is visited twice. Whenever an integer solution is visited for a second time, $\phi \geq \Phi$ holds and the algorithm terminates. On the other hand, if Problem (1) is infeasible, then every subproblem $(S^p)$ is infeasible. According to Lemma 4, the integer solution $x^*_f$ is infeasible for the master problem in iteration $p + 1$, which results in an infeasible master problem after a finite number of iterations. Thus, finiteness follows from the finite number of integer solutions for Problem (1).

Second, we show that Algorithm 1 always terminates at a solution of Problem (1), if it is feasible. We denote a (possibly non-unique) optimal solution of Problem (1) by $(x^*_f, x^*_R, y^*)$ with objective function value $q_a^* = q_a(x^*_f, x^*_R, y^*)$. Now assume that Algorithm 1 terminates with a solution $(x^*_f, x^*_R, y^*, \lambda^*, w^*, s^*)$ and objective function value $\Phi = q_a(x^*_f, x^*_R, y^*)$. It is obvious that $(x^*_f, x^*_R, y^*, \lambda^*)$ is feasible for the nonconvex single-level reformulation (10). According to Lemma 1, $(x^*_f, x^*_R, y^*)$ is then feasible for the original bilevel problem, which gives $q_a^* \leq \Phi = q_a(x^*_f, x^*_R, y^*)$. On the other hand, Lemma 2 together with Lemma 1 state that every master problem $(M^p)$ is a relaxation of the original bilevel problem (1), i.e., $\phi \leq q_a^*$. Since Algorithm 1 only terminates when $\phi \geq \Phi$, we obtain $q_a(x^*_f, x^*_R, y^*) = q_a(x^*_f, x^*_R, y^*)$. □

In the remainder of this subsection we discuss some enhancements of Algorithm 1.

Additional outer-approximation cuts: Since the outer-approximation cuts of the form of (16) are globally valid, we can add cuts also for points other than the solution of the subproblem. One point that comes for free is the solution $z^p$ of the master problem, i.e., we can add a cut (16) for the point $y^p$. This is an outer-approximation cut in the sense of Kelley [33]. Further, we can add outer-approximation cuts for all feasible solutions that are encountered in the process of solving the master problem.
Early termination of the master problem: It is sufficient for the correctness of the entire algorithm that the master problem provides a new integer-feasible point. Since the outer-approximation cuts are constructed in a way that already visited integer solutions of the previous iterations have on objective value worse or equal to the incumbent $\Phi$, we can stop the master problem with the first improving integer-feasible solution, i.e., a solution that has an objective value better than the incumbent $\Phi$; see also [22]. This strategy mimics the single-tree approach that is stated in Section 3.2.

Warmstarting the master problem: Warmstarting mixed-integer problems can be a very effective strategy because it may produce a tight initial upper bound and help to keep the branch-and-bound tree small. Since the incumbent solution $z^*$ is feasible for every master problem, it is reasonable to warmstart the master problems with this solution.

The performance of the plain Algorithm 1, as well as the effectiveness of the mentioned enhancements are evaluated in Section 5.

3.2. A Single-Tree Outer-Approximation Approach. The multitree outer-approximation approach from Section 3.1 can also be cast into a single branch-and-bound tree. In the context of general convex MINLPs, this approach is known as LP/NLP-based branch-and-bound (LP/NLP-BB) and was first introduced in [44]. LP/NLP-BB avoids the time-consuming solution of subsequently updated mixed-integer master problems by branching on the integer variables of an initial master problem and solving continuous relaxations of the subsequently updated master problem at every branch-and-bound node. Whenever such a relaxation results in a new integer solution with a better objective value than the incumbent, the solution process is interrupted. In this event the convex nonlinear subproblem with fixed integer variables is solved and the master problem is updated by an outer-approximation cut derived from the solution of the subproblem. In this view, LP/NLP-BB can be interpreted as a branch-and-cut algorithm that requires the solution of NLPs to separate cuts. For additional implementation details we refer to [1, 7].

If we apply such a single-tree approach to our setup, the initial master problem is given by Problem (M$_p$) for $p = 0$, i.e., this corresponds exactly to the initial multitree master problem. In this view, the problem that is solved at every branch-and-bound node is the continuous relaxation of Problem (M$_p$) with bounds $\ell, u \in \mathbb{R}^{|I|}$ on the integer variables, i.e., the QP:

$$\min_{x, y, \lambda, w, s} q_0(x, y)$$

s.t. $Ax + By \geq a, \quad Cx_I + Dy \geq b,$

$l \leq x_I \leq u,$

$G_0 y + d = D^T \lambda, \quad \lambda \geq 0,$

$\lambda^T C x_I$ linearization: (11), (13),

$\bar{c}(\bar{y}; y, \lambda, w) \leq 0, \quad \ell = 0, \ldots, p - 1.$

The index $p$ in (N$_p^p(l, u)$) corresponds to the number of added strong-duality cuts. The specific values for $l$ and $u$ follow from branching. As opposed to the textbook LP/NLP-BB that solves an LP at every branch-and-bound node, we solve a QP. Thus, we rather perform a QP/NLP-BB.

We now state a tailored single-tree approach for bilevel problems of the form (1) in Algorithm 2. The rationale is the following. The algorithm subsequently solves QPs of the form (N$_p^p(l, u)$), starting with the root node problem for $p = 0$, $l = x^-$. 
Algorithm 2: Single-Tree Outer Approximation for MIQP-QP Bilevel Problems.

1: Initialize \( \Phi = \infty \), \( p = 0 \), \( l = x^- \), \( u = x^+ \), and \( z^* = \text{none} \).
2: Initialize the set of open node problems \( O := \{(N^p(l, u))\} \).
3: while \( O \neq \emptyset \) do
4: Remove a QP \( (N^p(l, u)) \) from \( O \) and solve it to obtain a solution \( z^{l,u} \).
5: if \( (N^p(l, u)) \) is infeasible or \( q_u(x^{l,u}, y^{l,u}) \geq \Phi \) then
6: Subtree can be pruned. Continue.
7: else if \( z^{l,u} \) is integer feasible and \( q_u(x^{l,u}, y^{l,u}) < \Phi \) then
8: Set \( x^I_f = x^{l,u} \) and \( s^p = s^{l,u} \) and solve the subproblem \( (S^P) \) or the feasibility problem \( (P^P) \) to obtain \( (\bar{x}_l^R, \bar{y}_R, \bar{\lambda}_p, \bar{w}_p) \).
9: if \( (S^P) \) is feasible and \( q_u(x^I_f, x^P_R, \bar{y}_R) < \Phi \) then
10: Set \( z^* = (x^I_f, x^P_R, \bar{y}_R, \bar{\lambda}_p, \bar{w}_p) \) and \( \Phi = q_u(x^I_f, x^P_R, \bar{y}_R) \).
11: end if
12: Re-add the problem: \( O \leftarrow O \cup \{(N^p(l, u))\} \).
13: Add the outer-approximation cut \( \tilde{c}(\bar{y}^P, y, \lambda, w) \leq 0 \) to all problems in \( O \).
14: Set \( p \leftarrow p + 1 \).
15: else
16: Branch on a fractional \( x^P_i \), \( i \in I \), to obtain new bounds \( l^I, u^I \) and \( l^2, u^2 \).
17: Update \( O \leftarrow O \cup \{M^P(l^I, u^I), M^P(l^2, u^2)\} \).
18: end if
19: end while
20: Return \( z^* \) or, if \( z^* \) is \text{none}, return “The bilevel problem is infeasible”.

and \( u = x^+ \). Whenever such a QP is infeasible or its objective function value cannot improve the incumbent, then this problem can be removed from the set of open problems \( O \) once and for all. In case the solution of Problem \( (N^p(l, u)) \) is integer feasible, then the corresponding subproblem \( (S^P) \)—or the feasibility problem \( (P^P) \) if the subproblem is infeasible—is solved. One key difference of LP/NLP-BB compared to a standard branch-and-bound is that this solved subproblem must not be pruned but needs to be updated with an appropriate outer-approximation cut. Moreover, all other open problems in \( O \) need to be updated as well. Finally, if Problem \( (N^p(l, u)) \) is feasible but the solution is not integer feasible, one branches on a fractional integer variable to obtain two new open problems. Similar to the multitree approach, only one integer assignment—the optimal one—is computed twice during the solution process. Finiteness follows again from the finite number of possible integer assignments. At some point all problems in the set \( O \) are infeasible, \( O \) is emptied, and the algorithm terminates. If all subproblems turned out to be infeasible, then \( z^* \) is never updated and the infeasibility of the bilevel problem is correctly detected. All together, we obtain the following correctness theorem.

**Theorem 2.** Algorithm 2 terminates after a finite number of iterations with an optimal solution of Problem (1) or with an indication that the problem is infeasible.

Since all arguments mainly follow from the proof of Theorem 1, we refrain from a formal proof of Theorem 2. We close this subsection by discussing some possible enhancements of Algorithm 2.

**Additional outer-approximation cuts:** Similar to the multitree approach, we can enhance Algorithm 2 by adding outer-approximation cuts for all integer-feasible solutions \( z^{l,u} \) with \( q_u(x^{l,u}, y^{l,u}) \geq \Phi \), i.e., integer-feasible solutions that do not fulfill the if-condition in Line 7 in Algorithm 2. This can be done, e.g., by storing these non-improving integer-feasible solutions...
and adding outer-approximation cuts for these solutions together with the cut that is added in Line 13.

**Advanced initialization:** Initializing the single-tree approach with a bilevel-feasible solution may be beneficial for various reasons. First, in the initial master problem \((M^p)\) for \(p = 0\), the constraints for the binary expansion (11) and for the linearization (13) are redundant, because they are not yet coupled by any outer-approximation cut. When the master problem is equipped with an initial outer-approximation cut, however, then all parts of the model are coupled and the solver can effectively presolve the entire model before solving the root node problem, i.e., \((N^p(\ell, u))\) for \(p = 0\), \(\ell = x^-\), and \(u = x^+\). In addition, this initial outer-approximation cut results in a tighter root node problem.

Second, an initial bilevel-feasible solution can be used to pass an incumbent solution \(z^*\) to Algorithm 2 and to compute an initial upper bound \(\Phi\). This may allow to prune parts of the search tree right in the beginning. An initial bilevel-feasible point can be obtained, e.g., by finding a feasible or optimal point for Problem \((M^p)\) for \(p = 0\) and solving the corresponding subproblem. This mimics the first iteration of the multitree approach.

We evaluate the plain Algorithm 2 and these enhancements in Section 5.

### 3.3. Exploiting the Bilevel Structure

The two outer approximation algorithms stated in the previous sections are an application of the approaches in [7, 22] and [44], respectively. The effectiveness of both algorithms will depend, among other aspects, on the following properties:

(i) The ability to solve the master problem(s) effectively.

(ii) The number of integer-feasible solutions of the master problem that need to be evaluated.

(iii) The ability to solve the subproblems effectively.

Aspect (i) is addressed by the various enhancements stated in Section 3.1 and Section 3.2, respectively. For the latter two aspects, we can exploit the specific bilevel structure of Problem (1). We explain this on the example of the multitree algorithm 1, but the same explanations hold for the single-tree approach Algorithm 2 as well. We first discuss aspect (ii). In general, the number of integer-feasible solutions of the master problem coincides with the number of subproblems that need to be solved. In the worst case, the algorithm needs to consider every integer-feasible solution of the initial master problem. However, for bilevel problems of the form (1), there is hope that one needs to evaluate only a few subproblems. The hypothesis is the following. Both the upper- and the lower-level objective functions are convex-quadratic in \(y\) and are to be minimized. This means that some parts in the upper- and the lower-level objective (namely the quadratic parts in \(y\)) have “more or less” the same optimization directions. Thus, although the linear parts of the two objective functions may have opposite optimization directions, explicit min-max problems for which the upper level minimizes a function that the follower maximizes, cannot arise for quadratic bilevel problems of the form (1), unless all matrices \(H_u, G_u,\) and \(G_l\) are 0. Thus, the solution of the early master problems (that mainly abstract from lower-level optimality) might already be a good estimate of the optimal lower-level solution, depending on how competitive the two objective functions are. As a consequence, it might be quite likely that the first few solutions of the master problem already contain a close-to-optimal or even optimal integer upper-level decision \(x_I\). Hence, the solution of the respective subproblem already provides a very tight upper bound \(\Phi\). This is, of course, instance specific, and we discuss this in more detail in Section 5.
Then, the following properties hold:

**Proposition 1.** Let $z^p$ be a solution of the master problem $(M^p)$. Further, let $q^p_0(x^p_I)$ be the optimal objective value of the parametric lower-level problem (3) for $x_I = x^p_I$ and $(\tilde{x}_R^p, \tilde{y}^p)$ be the solution of the QCQP

\[
\min_{x_R, y} q_a(x^p_I, x_R, y)
\text{ s.t. } (x^p_I, x_R, y) \in P,
\]

\[
q_i(y) \leq q^p_0(x^p_I).
\]

Then, the following properties hold:

(i) $(x^p_I, \tilde{x}_R^p, \tilde{y}^p) \in F$, i.e., it is a bilevel feasible point of Problem (1) in the optimistic sense.

(ii) The solution $(\tilde{x}_R^p, \tilde{y}^p)$ of the subproblem $(S^p)$ and the solution of Problem (22) correspond in the sense that $(\tilde{x}_R^p, \tilde{y}^p) = (\tilde{x}_R^p, \tilde{y}^p)$.

Since the outer-approximation cut is only parameterized by $y$, the solution of Problem (22) yields the same cut than the solution of Problem $(S^p)$.

**Remark 1.** We can replace Step 9 of Algorithm 1 (or Step 8 of Algorithm 2) by subsequently solving the parametric lower-level problem (3) with fixed integer linking variables $x_I$ and Problem (22). In other words, instead of solving subproblem $(S^p)$, we can solve a convex QP and a convex QCQP that is considerably smaller than $(S^p)$.

**Remark 2.** If the matrix $G_1$ is positive definite, then the lower-level problem has a unique solution and $M(x_I)$ is a singleton. In this case we can replace Step 9 of Algorithm 1 (or Step 8 of Algorithm 2) by subsequently solving the parametric lower-level QP (3) with fixed integer upper-level variables $x^p_I$ and solving the upper-level problem

\[
\min_{x_R} q_a(x^p_I, x_R, \tilde{y}^p) \text{ s.t. } A(x^p_I, x_R) + By = a,
\]

in which all integer upper-level variables are fixed to $x^p_I$ and all lower-level variables are fixed to the unique solution $\tilde{y}^p$. Problem (23) is a QP as well.

Especially Remark 2 can be expected to decrease the running time needed for the subproblem substantially, because it replaces a large QCQP by two considerably easier QPs. We discuss the effectiveness of both remarks in Section 5.

4. Implementation

In this section, we first discuss the software setup of the two outer-approximation approaches stated in Section 3 as well as of the two benchmark approaches, namely solving the KKT-based reformulation (2) and solving the strong-duality-based reformulation (15) directly. Then, in Section 4.2, we discuss the handling of Assumption 1 before we discuss some details regarding the subproblem in Section 4.3. Finally, we briefly sketch the implementation of the single-tree approach in Section 4.4.

4.1. Software Setup. We implemented all solution approaches using C++11 with GCC 7.3.0 as the compiler. All optimization problems, i.e., all convex (MI)(QC)QP problems, are solved by Gurobi 8.1.1 using its C interface. Throughout all computations we set the `NumericFocus` parameter to a value of 3, which results in increased numerical accuracy. Further, we tightened Gurobi’s integer feasibility tolerance from its default value $10^{-5}$ to $10^{-9}$. The rationale is to prevent numerical inaccuracies caused by products of binary variables and big-M values, e.g., in the Constraints (13). Similar products also appear when using the KKT-based reformulation (2).
4.2. The Role of Assumption 1. In theory, Assumption 1 is crucial for the termination of the methods described in Section 3. An indication that this assumption is not fulfilled in practice is cycling, i.e., a certain integer solution is computed more than once. In preliminary numerical tests, cycling hardly ever occurred for any of the two outer-approximation approaches, such that we disabled an expensive cycling management (e.g., storing all integer solutions and adding a no-good-cut if an integer solution is computed for the second time) in our final computations.

4.3. Solving the Subproblem. Both, Algorithm 1 and Algorithm 2 require the solution of the subproblem $(S^p)$. It is easy to see that the variables $w$ and $s$ can be projected out of this QCQP according to Equation (12). The resulting problem is then equivalent to fixing $x_I = x_I^p$ directly in the original nonconvex MINLP (10). Note that for fixed integer variables, (10) is a convex QCQP as well. While the full subproblem $(S^p)$ is more in line with the standard literature on outer approximation for convex MINLPs and makes it easier to prove correctness, it is certainly better to use Problem (10) with fixed integer variables in the actual implementation for the following reasons. First, Problem (10) is smaller than $(S^p)$, and second and more importantly, Problem (10) is big-$M$ free. As already discussed in Section 2, a wrong big-$M$ can result in terminating with points that are actually bilevel-infeasible or suboptimal. Using Problem (10) as the subproblem, the former case can never appear. This is a huge advantage compared to solving the two single-level reformulations (2) and (15) directly, which may indeed terminate with points that are actually bilevel-infeasible.

In Remarks 1 and 2, we discuss that one may solve the subproblem in a bilevel-specific manner. Both approaches are big-$M$ free as well. However, our preliminary computational results revealed that applying Remark 1 is not beneficial at all in practice. The main problem is that Gurobi very often fails to solve the QCQP (22) and gets stuck with a suboptimal solution. This happens in particular also for instances that can be solved using Problem (10) with fixed integer variables (or with the standard subproblem $(S^p)$). This is surprising, because Problem (10) contains also dual lower-level constraints and is a significantly larger QCQP compared to Problem (22). On the other hand, the quadratic constraint in (22) has much less degrees of freedom than the quadratic constraint in (10), because it does not contain the dual terms but the optimal objective value of the parametric lower level. This might explain the observed behavior. We thus leave the application of Remark 1 out of our numerical analysis in Section 5 and recommend to use Problem (10) (or $(S^p)$) in case $G_I$ is not strictly positive definite, i.e., if the lower-level problem does not have a unique solution.

4.4. Implementation of the Single-Tree Approach. While an implementation of the multitree approach of Section 3.1 is fairly straightforward, the implementation of the single-tree approach of Section 3.2 deserves a bit more explanation. The overall goal is to exploit the powerful branch-and-bound implementation of MIP solvers as much as possible. This is in line with the authors of [1], who argue that the performance of LP/NLP-BB methods can be greatly improved if they are implemented within modern MIP solvers. In particular, we want to exploit Gurobi’s branch-and-bound framework. We recap that Algorithm 2 subsequently solves continuous relaxations $(N^p(l,u))$ of Problem $(M^p)$ at every branch-and-bound node. Thus, the implementation of the single-tree approach is realized by solving the initial master problem, i.e., Problem $(M^p)$ for $p = 0$, with Gurobi. The MIP solver then takes care of the full branch-and-bound framework and the solution of the node QPs $(N^p(l,u))$. In the course of solving this initial master problem, it is subsequently updated. Whenever Gurobi finds a new integer-feasible solution,
we check via a callback whether the corresponding objective value is better than the bilevel incumbent. This requires some additional bookkeeping of the bilevel incumbent and the best bilevel solution. If necessary (see Algorithm 2), the callback calls a subroutine that solves the subproblem (or the feasibility problem) and generates outer-approximation cuts. These cuts are added to the master problem as so-called lazy constraint cuts, i.e., cuts that are allowed to safely cut off integer-feasible points, before the solution process is resumed. This implicitly updates all open node problems \((N^p(l, u))\). Note however that using Gurobi’s lazy constraint callbacks requires to set the parameter LazyConstraints to 1, which avoids certain reductions and transformations during the presolve that are incompatible with lazy constraints.

5. Computational Study

We now provide detailed numerical results for the methods proposed in the previous sections. All computational experiments have been executed on a compute cluster using compute nodes with Xeon E3-1240 v6 CPUs with 4 cores, 3.7 GHz, and 32 GB RAM; see [46] for more details.

The evaluations and comparisons mostly rely on performance profiles according to [14]. For every test instance \(i\) we compute ratios \(r_{i,s} = t_{i,s} / \min\{t_{i,s} : s \in S\}\), where \(S\) is the set of the solution approaches and \(t_{i,s}\) is the running time of a solver \(s\) for instance \(i\), given in wall-clock seconds. Each performance profile plot in this section shows the percentage of instances (y-axis) for which the performance ratio \(r_{i,s}\) of approach \(s\) is within a factor \(\tau \geq 1\) (log-scaled x-axis) of the best possible ratio. Note that for each solution attempt, we set a time limit of 3600 s. In addition, we set a minimum running time of 0.5 s, i.e., all running times faster than 0.5 s are fixed to 0.5 s. This is done to minimize the impact of external effects (like, e.g., CPU temperature) on very short running times.

The test set \(I\) that we use for our numerical study is specified in Section 5.1. In Section 5.2 and 5.3, we evaluate the results for different variants of the multi- and the single-tree approach, respectively. Finally, in Section 5.4, we compare both methods and also test their performance against solving the KKT-based reformulation (2) and the strong-duality-based reformulation (15).

5.1. Test Set. Our test set is taken from the MIQP-QP test set that is also used in [35]. Most of these instances are generated based on MILP-MILP instances from the literature by adding quadratic terms to both objective functions and by relaxing integrality conditions in the lower-level problem; see Section 6.1 in [35] for more details on the construction of the instances. Note that we enforced continuous linking variables to be integer, if necessary. Some of the instance classes in [35] are too difficult to be solved to global optimality, i.e., for most instances of the respective instance class, every tested solver exceeds the time limit. Thus, we excluded these instance classes from our test set. On the other hand, we used the instance classes DENE GRE, INTER-ASSIG, and INTER-KP. These instances are generated by the same authors in the same way as the other instances but turned out to be too easy for considerations of local optimality and thus are left out in the final computations in [35].

The various test set classes yield a total of 757 instances. From these instances, we removed 72 instances that are too easy, i.e., each solver (this includes also the benchmark solvers) can solve the instance within 0.5 s. Further, we exclude 45 instances that cannot be solved by any solver, because they are either too hard in terms of running time or memory requirements, or infeasible. Finally, we also exclude 18 instances for which the optimal objective values provided by different solvers do not match. The reason for this behavior may be due to numerical problems
and/or wrong big-M values. Our final test set $\mathcal{I}$ contains $|\mathcal{I}| = 622$ instances, which is—to the best of our knowledge—the largest test set of MIQP-QP bilevel problems considered for global optimality. We explicitly point out that every instance in $\mathcal{I}$ can be solved by at least one method discussed in this work within 3600 s. The largest instances in $\mathcal{I}$ have several thousand variables and constraints. We display the sizes and densities of the instances contained in $\mathcal{I}$ in Figure 1. More details on the instances can be found in Table 1, in which we give an overview of the different instance classes that we used, along with a reference (“Ref”) to the corresponding original MILP-MILP test set and the resulting size of the filtered instance class (“Size”). We also specify the minimum and maximum number of upper-level and lower-level variables ($n_x, \text{dim}_Y$) and constraints ($m_u, m_l$), as well as the minimum and maximum number of linking variables ($|\mathcal{I}|$) and of the maximum upper bound of the linking variables ($\max_{i \in \mathcal{I}} x^*_i$). Except for the instance classes INTER-CLIQUE and INTER-FIRE that use a big-M value of $10^5$, we set the big-M to $10^6$.

Finally, we emphasize that for all instances in $\mathcal{I}$, the matrix $G_l$ is positive definite, which renders the solutions of the lower-level problems unique. This means, that we can evaluate the application of Remark 2 for the full test set $\mathcal{I}$. We do not consider semidefinite matrices $G_l$, because applying Remark 1 turned out to be inefficient in our preliminary computational results; see the discussion in Section 4.3. Instead, we suggest to use the standard subproblem ($S^p$) for instances with multiple lower-level solutions, and test this approach also on the test set $\mathcal{I}$.

5.2. Evaluation of the Multitree Approach. We now evaluate the different parameterizations of the multitree approach as described in Section 3.1. Since in
our test set $G_l$ is positive definite, we use Remark 2 as the default method for solving the subproblem and compare the following variants, which are in line with the enhancements for the master problem proposed at the end of Section 3.1:

- **MT**: A basic variant without any enhancements, i.e., the plain Algorithm 1.
- **MT-K**: Every master problem is solved to optimality but additional Kelley-type cutting planes (“K”) are used.
- **MT-K-F**: The master problem terminates as soon as a first (“F”) improving integer-feasible solution is found and after every iteration additional Kelley-type cutting planes are added for every non-improving integer-feasible solution found by the master problem.
- **MT-K-F-W**: Like MT-K-F but every master problem is warmstarted (“W”) using the best available bilevel-feasible solution.

Later on, we compare the best setting among those four variants with an equivalent setting that uses the subproblem $S_p$ (in fact, Problem (10) with fixed integer variables; see Section 4) instead of the strategy of Remark 2.

Figure 2 (left) shows the performance profile of the four variants. It turns out that MT-K outperforms MT, which means that adding Kelley-type cutting planes improves the performance. In addition, MT-K-F dominates MT-K. Thus, stopping the master problem with the first integer-feasible solution that has a better objective function value than the bilevel incumbent is beneficial, too, although this strategy might result in more iterations and less Kelley-type cutting planes added. MT-K-F in turn is dominated by MT-K-F-W, which obviously outperforms all other tested variants. In fact, MT-K-F-W is the fastest method for around 75% of the instances. Further, it is the most reliable approach and solves around 99% of the instances in $I$.

Overall, according to Figure 2, MT-K-F-W is the clear winner among the four tested variants, although differences across the four approaches are rather mild, as the maximum factor $\tau$ is around 10.

In Figure 2 (right), we compare the “winner setting” MT-K-F-W with MT-STD, a variant with the same settings than the former approach but that uses the “standard” subproblem $S_p$ instead of the bilevel-specific strategy proposed in Remark 2. In contrast to the other approaches, MT-STD is also applicable for bilevel problems with multiple lower-level solutions. This comes at the cost of longer running times. The performance profile shows that MT-STD is clearly dominated by MT-K-F-W, which is the faster method for around 95% of the instances.

The conclusions drawn from the performance profiles are underlined by the running times displayed in Table 2. Note that, in order to have a fair comparison,
Table 2. Running times (in seconds), number of solved subproblems and time spent in the subproblem for all tested multitree variants.

|         | running time | solved subproblems | time in subproblems |
|---------|--------------|--------------------|---------------------|
|         | mean | median | mean | median | mean | median |
| MT      | 122.22 | 1.47   | 3.06 | 1.00   | 0.43 | 0.03   |
| MT-K    | 124.44 | 1.24   | 2.03 | 1.00   | 0.42 | 0.02   |
| MT-K-F  | 116.39 | 1.39   | 3.47 | 1.00   | 0.43 | 0.03   |
| MT-K-F-W| 77.81  | 1.09   | 3.61 | 1.00   | 0.43 | 0.03   |
| MT-STD  | 124.75 | 1.25   | 3.52 | 1.00   | 39.46| 0.20   |

for the numbers in Table 2, we used only instances that every multitree solver can solve. The first observation is that for all tested approaches, the median running times are much shorter than the mean running times. This indicates that many instances are easy for all multitree approaches. Further, it can be seen, that MT-K-F-W has significantly shorter mean and median running times than the other variants. Table 2 also shows the mean and median number of solved subproblems (or feasibility problems), which corresponds to the number of evaluated integer-feasible solutions (or to the number of iterations minus 1, since per construction the last iteration computes a specific integer solution for the second time). The table reveals that adding Kelley-type outer-approximation cuts reduces the number of solved subproblems by more than 30% in the mean (MT-K vs. MT). On the other hand, terminating the master problem early increases the number of solved subproblems by around 75% in the mean (MT-K-F vs. MT-K). This is expected and these additional iterations are obviously overcompensated by a reduction in the running time per iteration. The median numbers of solved subproblems are equal to 1 for all multitree solvers, which means that for at least 50% of the instances (that all of the multitree approaches can solve), only one integer-feasible solution needs to be computed. This seems to be surprising at a first glance, but is in line with our discussion in Section 3.3: Since both the upper- and lower-level objective function are convex-quadratic in \( y \) and to be minimized, it can be expected that the quadratic parts in the upper- and the lower-level objective have “more or less” the same optimization directions for many instances. For these instances, the outer-approximation cuts are very effective. We finally point out that the time spent for solving the subproblem dramatically increases, when \((S^p)\) is used. This explains the significantly longer running times of MT-STD compared to MT-K-F-W and justifies the exploitation of the bilevel structure as proposed in Section 3.3.

5.3. Evaluation of the Single-Tree Approach. We now analyze the single-tree approach described in Section 3.2 in the following variants:

**ST:** A basic variant without any enhancements as stated in Algorithm 2.

**ST-K:** Additional Kelley-type cutting planes (“K”) are added for every non-improving integer-feasible solution found.

**ST-K-C:** Like ST-K but an initial bilevel-feasible solution is computed (if available) in order to add an initial outer-approximation cut (“C”).

**ST-K-C-S:** Like ST-K-C but the initial bilevel-feasible solution is used to set start values (“S”) for \( z^* \) and \( \Phi \).

Again, all these variants apply Remark 2 to solve the subproblems. Later, we compare the winner setting to an equivalent setting that uses subproblem \((S^p)\) instead.
We compare the running times of the four variants in the performance profile in Figure 3 (left). The first observation is that the plots for ST and ST-K match almost exactly. This means that adding additional Kelley-type cutting planes for all non-improving integer-feasible solutions does not make any difference, which is in contrast to the results for the multitree approach. One explanation might be that the performance boost that is observed in the multitree method is mainly due to more effective presolving of the individual master problems when additional cuts are added. This is, of course, not possible in the single-tree approach. On the contrary, adding an initial outer-approximation cut is very beneficial. The methods without this initial cut, ST and ST-K, are clearly dominated by ST-K-C. The latter is in turn dominated by ST-K-C-S, the variant that also sets start values according to the initial bilevel-feasible solution. ST-K-C-S is the fastest method for more than 75% of the instances and it can solve more instances than any other single-tree variant. Overall, the differences between the four variants are rather small, as the maximum factor \( \tau \) is again around 10. Still, ST-K-C-S is the clear “winner setting” for the single-tree approach.

In Figure 3 (right), we compare this winner setting to ST-STD; a variant with the same settings but that uses the “standard” subproblem (\( S^p \)). We recap, that ST-STD is also applicable for bilevel problems with multiple lower-level solutions. The performance profile shows that this comes, like for the multitree approach, at the cost of longer running times. In fact, ST-K-C-S is the faster method for around 95% of the instances and it also solves more instances than ST-STD.

The conclusions drawn from the performance profiles in Figure 3 are also visible in Table 3, which displays statistics on running times and on the number of solved subproblems. We recap that the instances underlying the analysis in Table 3 are a subset of \( I \), namely the instances that all single-tree variants (including ST-STD) can solve. This makes a comparison of Table 2 and Table 3 impossible, because the underlying instances are different. Comparing median with mean running times in Table 3, we see that also for all single-tree approaches, many instances are rather easy to solve. Further, ST-K-C-S has the lowest mean running time, which is around 50% faster compared to ST. Additionally, the median running times are dominated by ST-K-C-S. Looking at the solved subproblems, it is interesting to see that adding additional Kelley-type cutting planes decreases the number of solved subproblems, although in terms of running times no effect could be observed (ST-K vs. ST). Adding an initial outer-approximation cut further reduces the number of solved subproblems, whereas setting start values has no effect on the iterations.
Table 3. Running times (in seconds), number of solved subproblems and time spent in the subproblem for all tested single-tree variants.

|                | running time | solved subproblems | time in subproblems |
|----------------|--------------|--------------------|---------------------|
|                | mean median  | mean median        | mean median         |
| ST             | 214.10 0.96  | 7.72 2.00          | 0.99 0.09           |
| ST-K           | 219.56 0.95  | 6.78 2.00          | 0.98 0.08           |
| ST-K-C         | 182.86 0.87  | 6.04 1.00          | 0.45 0.05           |
| ST-K-C-S       | 114.38 0.77  | 6.09 1.00          | 0.45 0.05           |
| ST-STD         | 141.99 0.96  | 5.87 1.00          | 28.54 0.22          |

Interestingly, ST-STD needs to solve considerably less subproblems than ST-K-C-S. Since these two methods only differ in the solution method of the subproblem, we have no proper explanation for this behavior. ST and ST-K spent double the time for solving the subproblems compared to ST-K-C and ST-K-C-S, because they need to solve two subproblems in the mean (compared to one for ST-K-C and ST-K-C-S). Again, it is obvious that the time spent for the subproblem significantly increases when \( S^p \) is used.

5.4. Comparison of the Approaches. We now compare our proposed algorithms with each other and with the two benchmark approaches introduced in Section 2. We briefly recap and discuss these benchmark approaches in the following. The first benchmark approach (KKT-MIQP) solves a reformulated and linearized single-level problem, where the lower level is replaced by its KKT conditions. The KKT complementarity conditions are linearized with a big-M formulation. This yields the convex MIQP(2), which can be solved directly with solvers like Gurobi or CPLEX. This approach is very popular and widely used in bilevel programming. Similarly, using strong duality of the lower level and convexifying the strong-duality inequality yields the convex MIQCQP(15). We call this approach SD-MIQCQP in the following. Instead of applying outer-approximation algorithms to this problem as proposed in Section 3, one can also solve this problem directly. Solvers like Gurobi then either apply a linear outer approximation or solve a continuous QCP relaxation at every branch-and-bound node. The exact method can be set via the MIQCPMethod parameter that we left at its default value \(-1\). This setting chooses automatically the best strategy. We emphasize again, that KKT-MIQP and SD-MIQCQP make use of a big-M value. This may result in bilevel-infeasible “solutions”, which is not the case for the proposed outer-approximation algorithms. Thus, we implemented an ex-post sanity check that computes the relative strong-duality error of the lower level for a given solution \((x, y, \lambda)\):

\[
\chi(x, y, \lambda) := \frac{c(x_I, y, \lambda)}{|q_l(y)|}.
\]
Whenever the error \(\chi(x, y, \lambda)\) exceeds the tolerance of \(10^{-4}\), we consider the instance as unsolved for the respective solver.

In Figure 4 we compare the best parameterizations of the multitree and single-tree approach (MT-K-F-W and ST-K-C-S) with the equivalents that use the standard subproblem (MT-STD and ST-STD) and with the two benchmark approaches (KKT-MIQP and SD-MIQCQP). The first observation is that SD-MIQCQP clearly outperforms KKT-MIQP. In particular, SD-MIQCQP can solve around 80% of the instances in \(I\), whereas KKT-MIQP can only solve around 30%. These results are in line with the study in [52], that reveals, that a strong-duality-based reformulation outperforms a KKT-based reformulation for mixed-integer linear bilevel problems.
with considerably large lower-level problems. It is still an interesting result since the KKT reformulation is the most used approach in practical bilevel programming.

We further observe, that all of the proposed outer-approximation approaches outperform the two benchmark approaches significantly. They can solve more instances faster and can also solve almost every instance in the test set $I$. In particular, the outer-approximation algorithms are capable of solving the largest instances in the test set with several thousand variables and constraints, for which the benchmark approaches KKT-MIQP and SD-MIQCQP fail.

This underlines the effectiveness of our methods in that extent that the charm of solving the MIQCQP (15) directly, i.e., applying SD-MIQCQP, lies, among other things, in the exploitation of the numerical stability of modern solvers. This contains, e.g., an elaborate numerical polishing of the outer-approximation cuts and managing these cuts in cut pools—numerical details that we mostly abstracted from in our implementation. Incorporating such aspects in our implementation would certainly not harm our results, but it can be expected that a more elaborated implementation would lead to an even greater domination of our approaches compared to the benchmark approaches. It is also interesting to see that the outer-approximation approaches that use the standard subproblem ($S^p$) outperform the benchmarks. This means that also for instances with multiple lower-level solutions, i.e., instances with a positive semidefinite matrix $G_l$, our proposed outer-approximation algorithms MT-STD and ST-STD are the better choice as compared to the benchmarks KKT-MIQP and SD-MIQCQP.

In order to reliably compare the multi- and single-tree methods, we look at a separate plot that compares MT-K-F-W and ST-K-C-S; see Figure 4 (right). This performance profile shows that the single-tree approach can solve slightly more instances faster than the multitree approach. However, from a factor of $\tau = 2$ onward, the plots of the two methods match almost exactly. Further, both approaches are equally reliable, such that on the full test set $I$, there seems to be hardly any difference in the performance of the two approaches. This is supported by Table 4 that provides statistics of running times and the number of solved subproblems for the subset of instances that both approaches, MT-K-F-W and ST-K-C-S, can solve to optimality. It can be seen that MT-K-F-W has shorter running times in the mean and needs to solve slightly less subproblems in the mean. In contrast, ST-K-C-S has significantly shorter running times in the median. Thus, also from Table 4, we cannot identify a dominating approach.
Table 4. Running times (in seconds), number of solved subproblems and time spent in the subproblem for MT-K-F-W and ST-K-C-S.

|                  | running time | solved subproblems | time in subproblems |
|------------------|--------------|--------------------|---------------------|
|                  | mean median  | mean median        | mean median         |
| MT-K-F-W         | 106.56 1.40  | 5.30 1.00          | 0.46 0.03           |
| ST-K-C-S         | 141.28 0.93  | 5.98 1.00          | 0.54 0.06           |

Figure 5. Log-scaled performance profiles for instance classes that are dominated by MT-K-F-W (left) and for all other instance classes (right).

However, if we look at the results in more detail, we observe significant performance differences on certain subsets of $\mathcal{I}$. We therefore look at the two performance profiles in Figure 5. The left plot is a performance profile for the subset $\mathcal{I}_{MT} \subset \mathcal{I}$ that contains the 186 instances of the classes CLIQUE, INTER-FIRE, and XULARGE. Although both approaches perform equally reliable, MT-K-F-W is the clear winner on these classes of instances. Looking at the characteristics in Table 1, one can see that the classes in $\mathcal{I}_{MT}$ are among the classes with the most lower-level constraints. This may indicate that the multitree approach is more suitable for instances with large lower levels than the single-tree approach. On the other hand, the reason for this behavior may be the lack of model reductions during presolve in the single-tree implementation due to the usage of Gurobi callbacks; see also the discussion in Section 4.4. The opposite pattern can be seen in Figure 5 (right) that shows a performance profile for the subset $\mathcal{I} \setminus \mathcal{I}_{MT}$, that contains the remaining 436 instances that are not dominated by the multitree approach. The single-tree approach is clearly the better approach on this subset. The instances in this subset have smaller lower-level problems than the instances in $\mathcal{I}_{MT}$ but do not share other characteristics in terms of sizes and/or structure; see Table 1. Thus, the reason for the observed dominance may simply lie in the nature of the approach: the single-tree approach needs to build and search only one branch-and-bound tree.

6. Conclusion

In this paper, we considered bilevel problems with a convex-quadratic mixed-integer upper level and a convex-quadratic lower level. Further, all linking variables are assumed to be bounded integers. For such problems, we proposed an equivalent transformation to a single-level convex MINLP and developed a multi- and a single-tree outer-approximation algorithm that we derived from algorithms for general
convex MINLPs. We further proposed enhancements of these algorithms that exploit the bilevel-specific structure of the problem. Finally, we proved the correctness of the methods and carried out an extensive numerical study.

This study revealed that the bilevel-specific enhancements clearly speed up the two proposed outer-approximation algorithms. In most cases, the algorithms only need very few iterations and the involved subproblems can be solved very fast. Moreover, all parameterizations of the outer-approximation algorithms clearly outperform known benchmark approaches. In particular, the benchmark approaches fail for larger instances with several thousand variables and constraints, which the proposed algorithms can solve. Even when certain enhancements that cannot be applied to instances with lower-level problems with multiple solutions are disabled, the novel algorithms perform significantly better than the benchmark approaches.

We also compared the multi- and single-tree approach. The performance profile in Figure 4 (right) indicates that the single-tree approach is, in general, the faster method. However, Figure 5 reveals that the multistorey approach is the faster method on the subset of larger instances. Since the multistorey approach is not substantially slower on the remaining instance set, we consider the multistorey approach to be the method of our choice.

For both methods several questions remain open. For instance: Can we extend the algorithmic framework in order to drop the integrality condition on the linking variables? This would require, e.g., a spatial branching on linking variables but may also introduce some pitfalls like unattainable bilevel solutions; see [40]. Another question is whether one can introduce integer variables to the lower level, i.e., considering MIQP-MIQP bilevel problems. Certainly, the strong-duality-based reformulation for convex lower-level problems would not be applicable anymore, but one could use a more general single-level reformulation like the value-function reformulation, as it is done, e.g., for MIP-MIP bilevel problems in [18].

ACKNOWLEDGMENTS

This research has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. We also thank the Deutsche Forschungsgemeinschaft for their support within project A05 and B08 in the "Sonderforschungsbereich/Transregio 154 Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks".

REFERENCES

[1] K. Abhishek, S. Leyffer, and J. Linderoth. “FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs.” In: INFORMS Journal on Computing 22.4 (2010), pp. 555–567. doi: 10.1287/ijoc.1090.0373.

[2] J. M. Arroyo. “Bilevel programming applied to power system vulnerability analysis under multiple contingencies.” In: IET Generation, Transmission & Distribution 4.2 (2010), pp. 178–190. doi: 10.1049/iet-gtd.2009.0098.

[3] A. Baggio, M. Carvalho, A. Lodi, and A. Tramontani. Multilevel approaches for the critical node problem. Tech. rep. Ecole Polytechnique de Montral, 2016.

[4] J. F. Bard. “Convex two-level optimization.” In: Mathematical Programming 40.1 (1988), pp. 15–27. doi: 10.1007/BF01580720.

[5] J. F. Bard and J. T. Moore. “A branch and bound algorithm for the bilevel programming problem.” In: SIAM Journal on Scientific and Statistical Computing 11.2 (1990), pp. 281–292. doi: 10.1137/0911017.
26 REFERENCES

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

[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, et al. “An algorithmic framework for convex mixed integer nonlinear programs.” In: Discrete Optimization 5.2 (2008), pp. 186–204. DOI: 10.1016/j.disopt.2006.10.011.

[8] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004. DOI: 10.1017/cbo9780511804441.

[9] A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. “Bilevel Knapsack with Interdiction Constraints.” In: INFORMS Journal on Computing 28.2 (2016), pp. 319–333. DOI: 10.1287/ijoc.2015.0676.

[10] O. Daxhelet and Y. Smeers. “The EU regulation on cross-border trade of electricity: A two-stage equilibrium model.” In: European Journal of Operational Research 181.3 (2007), pp. 1396–1412. DOI: 10.1016/j.ejor.2005.12.040.

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

[12] S. Dempe and A. B. Zenkoho. “Bilevel road pricing: theoretical analysis and optimality conditions.” In: Annals of Operations Research 196.1 (2012), pp. 223–240. DOI: 10.1007/s10479-011-1023-z.

[13] S. DeNegre. “Interdiction and discrete bilevel linear programming.” PhD thesis. Lehigh University, 2011.

[14] E. D. Dolan and J. J. Moré. “Benchmarking Optimization Software with Performance Profiles.” In: Mathematical Programming 91 (2 2002), pp. 201–213. DOI: 10.1007/s101070100263.

[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] T. A. Edmunds and J. F. Bard. “Algorithms for nonlinear bilevel mathematical programs.” In: IEEE Transactions on Systems, Man, and Cybernetics 21.1 (1991), pp. 83–89. DOI: 10.1109/21.101139.

[17] J. Egerer, V. Grimm, T. Kleinert, M. Schmidt, and G. Zöttl. The Impact of Neighboring Markets on Renewable Locations, Transmission Expansion, and Generation Investment. Tech. rep. Dec. 2019. URL: http://www.optimization-online.org/DB_FILE/2019/12/7508.pdf.

[18] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. “A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs.” In: Operations Research 65.6 (2017), pp. 1615–1637. DOI: 10.1287/opre.2017.1650.

[19] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. “Interdiction games and monotonicity, with application to knapsack problems.” In: INFORMS Journal on Computing 31.2 (2019), pp. 390–410. DOI: 10.1287/ijoc.2018.0831.

[20] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. “On the use of intersection cuts for bilevel optimization.” In: Mathematical Programming 172.1-2 (2018), pp. 77–103. DOI: 10.1007/s10107-017-1189-5.

[21] M. Fischetti, M. Monaci, and M. Sinnl. “A dynamic reformulation heuristic for Generalized Interdiction Problems.” In: European Journal of Operational Research 267.1 (May 2018), pp. 40–51. DOI: 10.1016/j.ejor.2017.11.043.

[22] 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.
REFERENCES 27

[23] J. Fortuny-Amat and B. McCarl. “A Representation and Economic Interpretation of a Two-Level Programming Problem.” In: The Journal of the Operational Research Society 32.9 (1981), pp. 783–792. DOI: 10.1057/jors.1981.156.

[24] L. P. García, A. J. Conejo, R. García-Bertrand, and R. Romero. “A bilevel approach to transmission expansion planning within a market environment.” In: IEEE Transactions on Power Systems 24.3 (2009), pp. 1513–1522. DOI: 10.1109/TPWRS.2009.2021280.

[25] P. García-Herreros, L. Zhang, P. Misra, E. Arslan, S. Mehta, and I. E. Grossmann. “Mixed-integer bilevel optimization for capacity planning with rational markets.” In: Computers & Chemical Engineering 86 (2016), pp. 33–47. DOI: 10.1016/j.compchemeng.2015.12.007.

[26] V. Grimm, J. Grübel, L. Schewe, M. Schmidt, and G. Zöttl. “Nonconvex equilibrium models for gas market analysis: Failure of standard techniques and alternative modeling approaches.” In: European Journal of Operational Research 273.3 (2019), pp. 1097–1108. DOI: 10.1016/j.ejor.2018.09.016.

[27] V. Grimm, G. Orlinskaya, L. Schewe, M. Schmidt, and G. Zöttl. Optimal Design of Retailer-Prosumer Electricity Tariffs Using Bilevel Optimization. Tech. rep. 2019. URL: http://www.optimization-online.org/DB_HTML/2019/07/7297.html.

[28] V. Grimm, L. Schewe, M. Schmidt, and G. Zöttl. “A Multilevel Model of the European Entry-Exit Gas Market.” In: Mathematical Methods of Operations Research (2018). DOI: 10.1007/s00186-018-0647-z.

[29] P. Hansen, B. Jaumard, and G. Savard. “New branch-and-bound rules for linear bilevel programming.” In: SIAM Journal on Scientific and Statistical Computing 13.5 (1992), pp. 1194–1217. DOI: 10.1137/0913069.

[30] X. He, C. Li, T. Huang, and C. Li. “Neural network for solving convex quadratic bilevel programming problems.” In: Neural Networks 51 (2014), pp. 17–25. DOI: 10.1016/j.neunet.2013.11.015.

[31] X. Hu and D. Ralph. “Using EPECs to Model Bilevel Games in Restructured Electricity Markets with Locational Prices.” In: Operations Research 55.5 (2007), pp. 809–827. DOI: 10.1287/opre.1070.0431.

[32] R. G. Jeroslow. “The polynomial hierarchy and a simple model for competitive analysis.” In: Mathematical Programming 32.2 (1985), pp. 146–164. DOI: 10.1007/BF01586088.

[33] J. E. Kelley Jr. “The Cutting-Plane Method for Solving Convex Programs.” In: Journal of the Society for Industrial and Applied Mathematics 8.4 (1960), pp. 703–712. DOI: 10.1137/0108053.

[34] T. Kleinert, M. Labbé, F. Plein, and M. Schmidt. “There’s No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization.” In: Operations Research (2019). URL: http://www.optimization-online.org/DB_HTML/2019/04/7172.html. Forthcoming.

[35] T. Kleinert and M. Schmidt. “Computing Stationary Points of Bilevel Problems with a Penalty Alternating Direction Method.” In: INFORMS Journal on Computing (2019). URL: http://www.optimization-online.org/DB_HTML/2019/03/7117.html. Forthcoming.

[36] T. Kleinert and M. Schmidt. “Global Optimization of Multilevel Electricity Market Models Including Network Design and Graph Partitioning.” In: Discrete Optimization 33 (2019), pp. 43–69. DOI: 10.1016/j.disopt.2019.02.002.

[37] M. Labbé, P. Marcotte, and G. Savard. “A bilevel model of taxation and its application to optimal highway pricing.” In: Management Science 44.12-part-1 (1998), pp. 1608–1622. DOI: 10.1287/mnsc.44.12.1608.
[38] L. Lozano and J. C. Smith. “A value-function-based exact approach for the bilevel mixed-integer programming problem.” In: *Operations Research* 65.3 (2017), pp. 768–786. DOI: 10.1287/opre.2017.1589.

[39] Y. Lv, Z. Chen, and Z. Wan. “A neural network for solving a convex quadratic bilevel programming problem.” In: *Journal of Computational and Applied Mathematics* 234.2 (2010), pp. 505–511. DOI: 10.1016/j.cam.2009.12.041.

[40] J. T. Moore and J. F. Bard. “The mixed integer linear bilevel programming problem.” In: *Operations Research* 38.5 (1990), pp. 911–921. DOI: 10.1287/opre.38.5.911.

[41] J. M. Morales, P. Pinson, and H. Madsen. “A transmission-cost-based model to estimate the amount of market-integrable wind resources.” In: *IEEE Transactions on Power Systems* 27.2 (2012), pp. 1060–1069. DOI: 10.1109/TPWRS.2011.2177281.

[42] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd. Berlin: Springer, 2006. DOI: 10.1007/978-0-387-40065-5.

[43] S. Pineda and J. M. Morales. “Solving Linear Bilevel Problems Using Big-Ms: Not All That Glitters Is Gold.” In: *IEEE Transactions on Power Systems* (2019). DOI: 10.1109/TPWRS.2019.2892607.

[44] I. Quesada and I. E. Grossmann. “An LP/NLP based branch and bound algorithm for convex MINLP optimization problems.” In: *Computers & Chemical Engineering* 16.10-11 (1992), pp. 937–947. DOI: 10.1016/0098-1354(92)80028-8.

[45] T. Ralphs. Cor@l: Bilevel Optimization Problem Library. Accessed: 2019-12-12. URL: http://coral.ise.lehigh.edu/data-sets/bilevel-instances/.

[46] Regionales Rechenzentrum Erlangen. Woodcrest Cluster. URL: https://www.anleitungen.rrze.fau.de/hpc/woody-cluster/ (visited on 12/12/2019).

[47] S. Tahernejad, T. K. Ralphs, and S. T. DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. Tech. rep. Lehigh University, 2016.

[48] Y. Tang, J.-P. P. Richard, and J. C. Smith. “A class of algorithms for mixed-integer bilevel min–max optimization.” In: *Journal of Global Optimization* 66.2 (2015), pp. 225–262. DOI: 10.1007/s10898-015-0274-7.

[49] R. J. Vanderbei. *Linear Programming*. Springer, 2014. DOI: 10.1007/978-1-4614-7630-6.

[50] L. Vicente, G. Savard, and J. Júdice. “Descent approaches for quadratic bilevel programming.” In: *Journal of Optimization Theory and Applications* 81.2 (1994), pp. 379–399. DOI: 10.1007/BF02191670.

[51] P. Xu and L. Wang. “An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions.” In: *Computers & Operations Research* 41 (2014), pp. 309–318. DOI: 10.1016/j.cor.2013.07.016.

[52] M. H. Zare, J. S. Borrero, B. Zeng, and O. A. Prokopyev. “A note on linearized reformulations for a class of bilevel linear integer problems.” In: *Annals of Operations Research* 272.1-2 (2019), pp. 99–117. DOI: 10.1007/s10479-017-2694-x.

(T. Kleinert) (A) Friedrich-Alexander-Universität Erlangen-Nürnberg, Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany; (B) Energie Campus Nürnberg, Förther Str. 250, 90429 Nürnberg, Germany

Email address: thomas.kleinert@fau.de
REFERENCES

(V. Grimm) (A) Friedrich-Alexander-Universität Erlangen-Nürnberg, Economic Theory, Lange Gasse 20, 90403 Nürnberg, Germany; (B) Energie Campus Nürnberg, Fürther Str. 250, 90429 Nürnberg, Germany
Email address: veronika.grimm@fau.de

(M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany
Email address: martin.schmidt@uni-trier.de