

An exact projection-based algorithm for bilevel mixed-integer problems with nonlinearities

Maximilian Merkert¹, Galina Orlinskaya², Dieter Weninger²

¹OVGU Magdeburg

²FAU Erlangen-Nürnberg

December 11, 2020

Abstract

We aim at solving bilevel mixed-integer optimization problems with lower-level integer variables and including nonlinear terms such as, e.g., products of upper-level and lower-level variables. Problems of this type are extremely challenging as a single-level reformulation suitable for off-the-shelf solvers is not available in general. In order to solve these problems, we enhance an approximative projection-based algorithm for mixed-integer linear bilevel programming problems from the literature to become exact under one additional relatively mild assumption. This assumption still allows for discrete and continuous variables on both levels, but forbids continuous upper-level variables to appear in lower-level constraints and thus ensures that a bilevel optimum is attained. In addition, we extend the algorithm to make it applicable to a wider problem class and demonstrate which types of nonlinearities can be handled by our algorithm on each level. We also discuss computational experiments on modified library instances.

1 Introduction

Hierarchical decision making by different agents occurs in a number of real life problems including, e.g., energy policy and markets [GOS⁺20], pricing schemes [LV13], supply chain management [YGZY18], taxation [LMS98], and transportation network design [BABBL92]. In this paper we consider the setting with two agents, the *leader* on the upper level and the *follower* on the lower level, known in game theory as a Stackelberg game (see [Sta52]). Furthermore, discrete decisions are allowed on both levels.

The motivation for this paper is twofold. First, optimal graph aggregations for network design problems as in [BLM⁺15] can be determined using bilevel modelling. Second, these kind of bilevel problems arise if one introduces controllable energy generation facilities on the lower level of energy supply tariff design problems described in [GOS⁺20]. Thus, we want an exact solution method for the following general problem class:

$$\begin{aligned}
& \max_{x^u, y^u} && f(x^u, y^u, x^l, y^l) \\
& \text{s.t.} && F(x^u, y^u, x^l, y^l) \leq 0 \\
& && x^u \in \mathbb{R}_+^{m_R}, y^u \in \mathbb{Z}_+^{m_Z} \\
& \max_{x^l, y^l} && g(x^u, y^u, x^l, y^l) \\
& \text{s.t.} && G(y^u, x^l, y^l) \leq 0 \\
& && x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z}
\end{aligned} \tag{1}$$

Here we denote the leader's variables, objective function and constraints by $x^u \in \mathbb{R}_+^{m_R}$ and $y^u \in \mathbb{Z}_+^{m_Z}$, f , and $F(x^u, y^u, x^l, y^l) \leq 0$, respectively. The follower's optimization problem, parameterized by upper level variables (x^u, y^u) , constitutes maximizing objective function g over variables $x^l \in \mathbb{R}_+^{n_R}$ and $y^l \in \mathbb{Z}_+^{n_Z}$ under constraints $G(y^u, x^l, y^l) \leq 0$.

In this paper we take the algorithm proposed in [YGZY18] and

1. show that the realization there is liable to failure, such as for example not being able to find an optimal bilevel solution or even classifying a feasible bilevel problem as infeasible.
2. propose a way of making the algorithm exact under the additional assumption that continuous leader variables do not appear in follower constraints. Without this assumption a compact bilevel feasible set cannot be guaranteed from a theoretical point of view, i.e. a bilevel optimum may be unattainable. Even harsher assumptions on this point are present in [FLMS18], where continuous leader variables are not allowed to appear in the follower's problem at all. The same assumption is present in [TRD20] and the corresponding software MibS [Ral15].
3. extend the algorithm to a nonlinear setting and dispense with the assumption of feasibility of the bilevel problem required in [YGZY18].

Altogether, the algorithm proposed in this paper is able to find exact global optimal solutions for more complex bilevel problem classes than the methods suggested in the literature, such as, e.g., [FLMS18, TRD20]. This is not only due to the fact that we allow continuous variables of the leader to be present in the follower's objective function. Any

kind of a MINLP, which an off-the-shelf solver can handle, is allowed as the leader’s problem, while the follower’s problem considered in lower-level continuous variables only has to be convex, bounded, and satisfy Slater’s condition. In a recent survey on bilevel optimization [Dem18], no exact solution method for this problem class is mentioned, nor is any other existing method—apart from complete enumeration of the follower’s integer variables—known to the authors of this paper.

The algorithm proposed in this paper builds upon a MINLP solver of the user’s choice, which iteratively derives upper and lower bounds on the bilevel optimum by solving corresponding single-level optimization problems. Thus our method incorporates all merits of an established solver implementation (stability, efficiency, multi-threading support, etc.) and automatically profits from any further development of MINLP solvers.

1.1 Paper Structure

Here we briefly outline the structure of this paper. In the next Subsection 1.2 there is a short overview of selected methods and existing literature for bilevel optimization problems, with a focus on problems with integer follower variables.

Our setting including the necessary assumptions is presented in Section 2. In Subsection 2.1 we briefly state how the key features of Algorithm 1 described in Section 2 are realized in [YGZY18], and then analyze its approximative property in Subsection 2.2. In Sections 3.1-3.2 we propose an alternative exact realization which guarantees that a bilevel optimum of a feasible bilevel problem is always found by Algorithm 1. We initially restrict ourselves to the in case in which the follower problem is linear. In the end of Section 3 we give a proof that Algorithm 2 proposed in this paper either finds a global optimal solution of a bilevel problem satisfying Assumptions 1-4 or proves the infeasibility of this problem in finitely many iterations. Numerical results are presented in Section 4. In Section 5 we provide extensions of Algorithm 2 to allow nonlinear follower problems. Finally, we conclude with Section 6.

1.2 Existing Literature

A common way of solving bilevel problems is reformulating them into a single-level optimization problem, which is obtained by adding optimality conditions for the lower-level problem as constraints to the so-called *High Point Relaxation (HPR)* of the original bilevel problem, see, e.g., [Bar98, Dem02]. However, this approach requires that general optimality conditions for the lower-level problem can be stated in such a way that some general global optimization solver is able to handle the reformulated single-level optimization problem. Consequently, this method is suitable only for special classes of bilevel problems and in particular does not allow for lower-level integer variables.

A number of methods have been proposed for handling bilevel problems with lower-level integer variables, most of them concentrating on various linear problem classes. Branch and cut approaches for approximating the optimal-value function have been used in [DR09, TRD20, DK17], while [FLMS16b, FLMS17, FLMS18] extend and apply intersection cuts to bilevel problems. The approach published in [DR09] is publicly available as MibS solver [Ral15]. In addition, the algorithm presented in [FLMS16b] is accessible as software [FLMS16a].

For nonlinear mixed-integer cases, algorithms proposed in [Mit10, DGM18] or the Branch-and-Sandwich algorithm from [KA15, KA14] are proven to achieve ϵ -optimality. For a more comprehensive list of currently available solution methods for bilevel optimization problems the reader is referred to [Dem18].

In [ZA14] the authors adapt the above-mentioned classical approach of solving bilevel problems via formulating necessary and sufficient optimality conditions of the lower level to a setting where both levels are mixed-integer problems. They show that if a mixed-integer linear bilevel optimization problem has the so-called *relatively complete response property*, iteratively solving its HPR with successively added follower optimality conditions for some fixed values of lower-level integer variables produces an exact solution of the original bilevel problem. Problem (1) has the *relatively complete response property* if every combination of HPR-feasible (x^u, y^u) and HPR-feasible y^l can be extended to a HPR-feasible solution (x^u, y^u, x^l, y^l) by a suitable HPR-feasible x^l . Henceforth we shall call the HPR with optimality conditions of the lower level for some $y^{l,k}$ with iteration index k the *master problem* (MP). The relatively complete response property guarantees that the master problem constructed in this way is a relaxation of the original bilevel problem. Successively adding follower optimality conditions for different $y^{l,k}$ to (MP) produces a more precise approximation of the *optimal-value function* of the original bilevel problem, which becomes exact if all HPR-feasible discrete decisions for the follower have been enumerated. Note that complete enumeration of all possible follower integer configurations is not always necessary for solving the bilevel problem via the procedure described above.

A similar approach is also used for computing binary quasi-equilibria in [HS18], where the authors propose a transformation of a mixed-integer equilibrium problem into a mixed-integer bilevel problem, which they then solve by enumerating all possible integer configurations and formulate corresponding sets of follower optimality conditions. In this case the feasibility of the follower problem is not affected by the leader's decisions, and the relatively complete response property holds.

In [YGZY18] the authors propose an extension of an algorithm from [ZA14], which aims at handling lower-level integer variables in linear bilevel problems without the relatively complete response property. The key idea is to consider a set of upper-level variables which allow certain lower-level integer configurations, and add corresponding lower-level optimality conditions to the HPR only for this upper-level variables set.

2 General Setting and Algorithm Structure

In this section we first state our assumptions on the class of bilevel problems the approach proposed in this paper can solve. Afterwards we introduce some necessary bilevel terminology in order to describe the general algorithm idea as proposed in [YGZY18]. Then, in Subsection 2.1 we state how the key features of the general projection algorithm are realized in [YGZY18]. Finally, in Subsection 2.2 we give an example on how this realization makes the algorithm prone to serious failure.

The algorithm proposed in the current paper requires the following assumptions:

Assumption 1. *All upper-level variables have finite bounds in the HPR. All lower-level variables have finite bounds in the follower problem. The objective and constraint functions f, F, g and G are continuous in their respective closed boxes.*

Assumption 1 is a usual assumption which ensures that all terms occurring in the formulations are bounded. We will make use of it when reformulating various indicator constraints as well as for showing that the proposed algorithm terminates after finitely many iterations.

Assumption 2. *For any fixed upper level decisions \bar{x}^u and \bar{y}^u , and lower-level integer decisions \bar{y}^l , the follower problem is convex and in case of feasibility satisfies Slater's condition, i.e. $g(\bar{x}^u, \bar{y}^u, \cdot, \bar{y}^l)$ is concave, $G(\bar{y}^u, \cdot, \bar{y}^l)$ are convex, and the relative interior of the feasible region is nonempty (existence of a 'strictly feasible' point) for all \bar{x}^u, \bar{y}^u and \bar{y}^l if $G(\bar{y}^u, x^l, \bar{y}^l) \leq 0$ has a solution in x^l . Moreover, functions g and G are continuously differentiable within lower-level variable bounds.*

The main consequence of Assumption 2 is that, considering all upper-level variables as parameters, we can formulate necessary and sufficient optimality conditions for the follower problem with fixed lower-level integer variables. Note that although there are optimal-value function formulations [WH16, Wil17] as well as optimality conditions based on extended duality [BOW16] for IPs/MIPs, they do not allow a single level reformulation of (BLP) that can be directly handed over to an off-the-shelf solver.

Assumption 3. *The high point relaxation (HPR) of the original bilevel problem together with the necessary and sufficient optimality conditions for the lower level as assumed in Assumption 2 is of a problem class that can be handled by some off-the-shelf optimization solver.*

The algorithm presented in this paper has a framework that builds upon solvers for problems of the type given in Assumption 3. Hence we need this general assumption. In practice (and in our computational experiments), we mainly think of this problem class to be mixed-integer nonlinear problems (MINLPs) with polynomial nonlinearities. Note that the Assumptions 1-3 are also present in [YGZY18].

Assumption 4. *The lower level constraints do not contain any continuous upper level variables.*

Without Assumption 4 we might not be able to attain the bilevel optimum despite Assumption 1, see [VSJ96]. Indeed, this assumption justifies using ‘max’ instead of ‘sup’ in formulation (BLP). Assumptions with this effect are widely used in the literature on different levels of restrictiveness. For instance, the requirement of an all-integer leader and/or follower problem is frequently encountered. The main point of Assumption 4 is that continuous leader variables do not influence the follower’s feasible set. This is very similar, yet slightly less restrictive, than Assumption 2 from [FLMS18] and Assumption 1 from [TRD20], where continuous leader variables are also banned from the follower objective function. This assumption has already been mentioned in earlier work by the same authors (see e.g. [FLMS17]) as being very important for their bilevel solver based on intersection cuts. Indeed, for the linear case where no continuous leader variables are present in the follower’s problem at all, [VSJ96] shows that a bilevel optimum is attained. To the best of the authors’ knowledge, no such statement exists yet for nonlinear cases satisfying Assumption 4, but possibly with continuous leader variables appearing in the follower objective function. This paper offers a constructive proof that a bilevel optimum is attained in this setting too. See Corollary 3.17 in Section 3 for more details.

Note that whenever the follower’s optimal solution is not unique, the bilevel optimal solution as well as its objective value may be so too, see, e.g., [Bar98]. As this issue is hard to deal with in practice, a common way is to consider the following reformulation of the original bilevel problem (1):

$$\begin{aligned}
 \max_{x^u, y^u, x^l, y^l} \quad & f(x^u, y^u, x^l, y^l) && \text{(BLP)} \\
 \text{s.t.} \quad & F(x^u, y^u, x^l, y^l) \leq 0 \\
 & G(y^u, x^l, y^l) \leq 0 \\
 & g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u) && \text{(optimal-value function constraint)} \\
 & x^u \in \mathbb{R}_+^{m_R}, y^u \in \mathbb{Z}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z},
 \end{aligned}$$

where

$$\theta(x^u, y^u) = \max_{x^l, y^l} \{g(x^u, y^u, x^l, y^l) : G(y^u, x^l, y^l) \leq 0, x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z}\} \quad (3)$$

is an *optimal-value function*, cf. [Bar98]. The reformulation (BLP) implies a so-called *optimistic assumption*, i.e., whenever multiple optimal lower-level solutions exist, the one allowing best results for the leader will be chosen.

The *High Point Relaxation (HPR)* mentioned earlier is obtained from (BLP) by dropping the *(optimal-value function constraint)*.

Next, we provide some definitions from [YGZY18] that are needed to state their general algorithm, including a master problem and two subproblems. By k we denote the iteration index in the following.

Definition 2.1. For any HPR-feasible $y^{l,k}$, let $P(y^{l,k})$ be defined as

$$P(y^{l,k}) = \{(y^u, x^l) : G(y^u, x^l, y^{l,k}) \leq 0, y^u \in \mathbb{Z}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R}\}. \quad (4)$$

Definition 2.2. For any HPR-feasible $y^{l,k}$, by $\text{Proj}_{(y^u)}P(y^{l,k})$ we denote

$$\text{Proj}_{(y^u)}P(y^{l,k}) = \{y^u \in \mathbb{Z}_+^{m_Z} : \exists x^l \in \mathbb{R}_+^{n_R} \text{ with } (y^u, x^l) \in P(y^{l,k})\}. \quad (5)$$

The HPR of the original bilevel problem together with optimality conditions of the lower level for some fixed lower-level feasible $y^{l,k}$ is denoted by *master problem* (MP):

$$\begin{aligned} \max_{x^u, y^u, x^l, y^l} \quad & f(x^u, y^u, x^l, y^l) & \text{(MP)} \\ \text{s.t.} \quad & F(x^u, y^u, x^l, y^l) \leq 0 \\ & G(y^u, x^l, y^l) \leq 0 \\ & x^u \in \mathbb{R}_+^{m_R}, y^u \in \mathbb{Z}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z} \\ & \forall y^{l,k} \in Y^L : \\ & \left[y^u \in \text{Proj}_{(y^u)}P(y^{l,k}) \right] & \text{PROJECTION TEST FOR } y^{l,k} \\ & \implies & \text{IMPLICATION OF OPT. PACKAGE} \\ & \left[g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k}) \right], & \text{OPTIMALITY PACKAGE FOR } y^{l,k} \end{aligned}$$

where

$$\theta(\bar{x}^u, \bar{y}^u, y^{l,k}) = \max_{x^l} \{g(\bar{x}^u, \bar{y}^u, x^l, y^{l,k}) : G(\bar{y}^u, x^l, y^{l,k}) \leq 0, x^l \in \mathbb{R}_+^{n_R}\} \quad (7)$$

is the optimal objective function value of the lower-level problem where all but continuous lower-level variables x^l are fixed, and Y^L a set of some lower-level feasible integer configurations, which formation will be shown in Algorithm 1. We deviate from notation purity for the sake of more intuitive understanding and use the same symbol θ as for the optimal-value function defined in (3). Particular realizations of PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE will be discussed later in this paper.

The following subproblem provides the optimal objective function value of the lower level for some fixed upper-level decision (\bar{x}^u, \bar{y}^u) , i.e. the value of θ at the point (\bar{x}^u, \bar{y}^u) . We call it *follower optimality* (FO) problem:

$$\begin{aligned} \theta(\bar{x}^u, \bar{y}^u) = \max_{x^l, y^l} \quad & g(\bar{x}^u, \bar{y}^u, x^l, y^l) \\ \text{s.t.} \quad & G(\bar{y}^u, x^l, y^l) \leq 0 \\ & x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z} \end{aligned} \quad \text{(FO)}$$

The last required subproblem checks whether for certain fixed (\bar{x}^u, \bar{y}^u) there exists an optimal solution of the lower level problem that together with (\bar{x}^u, \bar{y}^u) satisfies the upper

level constraints, i.e. whether (\bar{x}^u, \bar{y}^u) can be extended to a bilevel feasible solution. In addition to that, if such a lower level solution exists, the one that produces best results for the upper level will be chosen, thus realizing the optimistic assumption. We call this subproblem *bilevel feasibility* (BF) problem:

$$\begin{aligned}
& \max_{x^l, y^l} f(\bar{x}^u, \bar{y}^u, x^l, y^l) & (\text{BF}) \\
& \text{s.t. } F(\bar{x}^u, \bar{y}^u, x^l, y^l) \leq 0 \\
& \quad G(\bar{y}^u, x^l, y^l) \leq 0 \\
& \quad g(\bar{x}^u, \bar{y}^u, x^l, y^l) \geq \theta(\bar{x}^u, \bar{y}^u) \\
& \quad x^u \in \mathbb{R}_+^{m_R}, y^u \in \mathbb{Z}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R}, y^l \in \mathbb{Z}_+^{n_Z}
\end{aligned}$$

Now we are ready to state a general form of the algorithm as proposed in [YGZY18] for feasible bilevel problems:

Algorithm 1 General algorithm for feasible bilevel problems with integer variables

Set $LB \leftarrow -\infty, UB \leftarrow +\infty, \xi \leftarrow 0, k \leftarrow 0, Y_0^L \leftarrow \emptyset$
while $UB - LB > \xi$ **do**
 Solve (MP) for Y_k^L to obtain $(x_k^{u,*}, y_k^{u,*}, x_k^{l,*}, y_k^{l,*})$
 Set UB to the optimal objective function value of (MP)
 Solve (FO) for $(x_k^{u,*}, y_k^{u,*})$ to get $(\hat{x}_k^1, \hat{y}_k^1)$ and objective function value $\theta_k(x_k^{u,*}, y_k^{u,*})$
 Solve (BF) for $(x_k^{u,*}, y_k^{u,*})$ and $\theta_k(x_k^{u,*}, y_k^{u,*})$
 if (BF) is feasible **then**
 Denote optimal solution of (BF) as $(\tilde{x}_k^1, \tilde{y}_k^1)$
 $LB \leftarrow \max\{LB, \text{optimal objective function value of (BF)}\}$
 $y^{l,k} \leftarrow \tilde{y}_k^1$
 else
 $y^{l,k} \leftarrow \hat{y}_k^1$
 Add PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for $y^{l,k}$ to (MP)
 $Y_{k+1}^L \leftarrow Y_k^L \cup \{y^{l,k}\}$
 $k \leftarrow k + 1$
Terminate and return the optimal solution

All of our assumptions apart from Assumption 4 stem directly from this general algorithm form and are also assumed in [YGZY18]. They guarantee that we are able to solve the master problem (MP) or determine that it is infeasible (Assumption 3), formulate OPTIMALITY PACKAGES for fixed $y^{l,k}$ (Assumption 2) and, if the bilevel problem is feasible, arrive at a bilevel optimal solution after finitely many iterations (Assumption 1). As-

sumption 4, that has already been motivated from a theoretical viewpoint, will be shown to be essential in practice too in Section 3.1.

Remark 2.3. For all HPR-feasible (\bar{x}^u, \bar{y}^u) we have

$$\theta(\bar{x}^u, \bar{y}^u) = \max_{y^{l,k}} \{ \theta(\bar{x}^u, \bar{y}^u, y^{l,k}) : (\text{FO}) \text{ is feasible for } (\bar{x}^u, \bar{y}^u, y^{l,k}) \}.$$

The following statements—Lemma 2.4, Lemma 2.6, and Theorem 2.7—were already shown in [YGZY18]. We list them here in our notation for completeness. Corollary 2.5 is not explicitly stated in [YGZY18] (due to their assumption that a feasible solution exists), but is a direct consequence of Lemma 2.4.

Lemma 2.4. For any set Y^L of lower-level feasible integer variable configurations the master problem (MP) is a relaxation of the original bilevel problem (BLP).

If Y^L comprises a complete set of lower-level feasible integer variable configurations, the master problem (MP) is equivalent to the original bilevel problem (BLP).

Corollary 2.5. If the master problem (MP) is infeasible, then the original bilevel problem (BLP) is infeasible. If the original bilevel problem (BLP) is infeasible, the master problem (MP) for a complete set of HPR-feasible lower-level integer variable configurations Y^L is infeasible.

Lemma 2.6. Algorithm 1 generates some $y^{l,k}$ to be added to Y_k^L at the end of each iteration k . If thus generated $y^{l,k}$ is already contained in Y_k^L , i.e. has been generated in some previous iteration, Algorithm 1 terminates with a bilevel optimal solution at the latest in iteration $k + 1$.

Theorem 2.7. Given a feasible bilevel problem (BLP) satisfying Assumptions 1-4, Algorithm 1 finds a bilevel optimal solution in finitely many iterations.

The proofs of statements analogous to Lemma 2.6 and Theorem 2.7 that are needed for the exact PROJECTION TEST as described in Section 3.1, can be found in Section 3.3.

Note that in the worst case, Algorithm 1 can result in a complete enumeration of all lower-level feasible y^l . See Example A.1 in the appendix for an unfavorable instance in this regard.

For practical purposes, in the remainder of this paper we focus on the case in which the follower problem is linear, i.e. g and G are linear functions for fixed upper-level decisions \bar{y}^u and \bar{x}^u . In this case we can write

$$g(x^u, y^u, x^l, y^l) = g_R(\bar{x}^u, \bar{y}^u)x^l + g_Z(\bar{x}^u, \bar{y}^u)y^l$$

(a lower-level objective function term that is constant for fixed y^u and x^u would be meaningless) and

$$G(y^u, x^l, y^l) = G_R(\bar{y}^u)x^l + G_Z(\bar{y}^u)y^l + G_c(\bar{y}^u)$$

for vector-valued functions g_R, g_Z and matrix-valued functions G_R, G_Z, G_c of the upper-level decisions. Extensions to the general case characterized by Assumptions 1-4 will be discussed in Section 5.

2.1 Projection Test, Implications and Optimality Packages as Realized in the Literature

In this subsection we describe the setting and the way PROJECTION TEST, IMPLICATION OF OPTIMALITY PACKAGE and OPTIMALITY PACKAGE for $y^{l,k}$ necessary for Algorithm 1 are realized in [YGZY18].

Their set of assumptions include Assumptions 1-3 and an additional assumption that the inducible region is nonempty, i.e. the bilevel problem is feasible. However, Assumption 4 is not present and the setting treats only bilevel problems where the corresponding HPR is linear.

PROJECTION TEST (PT) determining whether $y^u \in \text{Proj}_{(y^u)} P(y^{l,k})$ is realized with the following subproblem formed at the end of each iteration for the resulting $y^{l,k}$:

$$\begin{aligned} \min_{x^{l,k}, t^k} \quad & \mathbb{1}_s^\top t^k \\ \text{s.t.} \quad & G(y^u, x^l, y^{l,k}) - t^k \leq 0 \\ & x^{l,k} \in \mathbb{R}_+^{n_R}, t^k \in \mathbb{R}_+^s, \end{aligned} \tag{PT_0}$$

where $\mathbb{1}_s$ is an s -dimensional all-ones vector $(1, \dots, 1)^\top$, s is the number of constraints in the follower's problem and $x^{l,k}$ are copies of the lower-level continuous variables x^l , introduced specifically for PROJECTION TEST in iteration k .

Remark 2.8. *Given $y^{l,k}$ generated by Algorithm 1 and a HPR-feasible y^u . If the optimal value of (PT₀) is 0, then $y^u \in \text{Proj}_{(y^u)} P(y^{l,k})$ applies, and otherwise $y^u \notin \text{Proj}_{(y^u)} P(y^{l,k})$.*

Then (PT₀), its dual feasibility conditions as well as both types of complementarity constraints are added to (MP) in order to replace

$$\left[y^u \in \text{Proj}_{(y^u)} P(y^{l,k}) \right] \implies \left[g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k}) \right]$$

by

$$\left[\mathbb{1}_s^\top t^k = 0 \right] \implies \left[g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k}) \right] \tag{8a}$$

$$\text{(PT}_0\text{)} \tag{8b}$$

$$\langle \text{dual feasibility of (PT}_0\text{)} \rangle \tag{8c}$$

$$\langle \text{complementarity constraint for (PT}_0\text{) and its dual} \rangle . \tag{8d}$$

The authors of [YGZY18] state that constraint (8a) cannot be handled by off-the-shelf solvers directly and propose an approximative reformulation dependent on some $\epsilon \geq 0$:

$$\left[\psi^k = 1 \right] \implies \left[g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k}) \right] \tag{9a}$$

$$\epsilon(1 - \psi^k) \leq \mathbb{1}_s^\top t^k, \quad \psi^k \in \{0, 1\} \tag{9b}$$

(8b)-(8d).

In Section 2.2 we elaborate on the theoretical meaning of this approximation and illustrate how it can lead to algorithm failure in practice.

The algorithm from [YGZY18] then utilizes a feature for specifying indicator constraints provided by some mixed-integer solvers such as, e.g., CPLEX, in order to realize implication (9a).

In total the approximative PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE add one new binary as well as $(2n_R + 2s)$ new continuous variables, and $(6n_R + 4s + 2)$ new constraints to (MP) in each iteration.

All $(3n_R + 2s)$ complementarity constraints from both PROJECTION TEST and OPTIMALITY PACKAGE are then linearized using a big-M technique, as the master problem has to remain linear in [YGZY18] in order to be able to employ the indicator constraints as mentioned above. Note that, unless upper bounds for dual variables of the lower level are known, the big-M linearization technique may lead to suboptimal solutions in the bilevel context, see, e.g., [PM19] for more information.

2.2 Consequences of the Approximative Projection Test

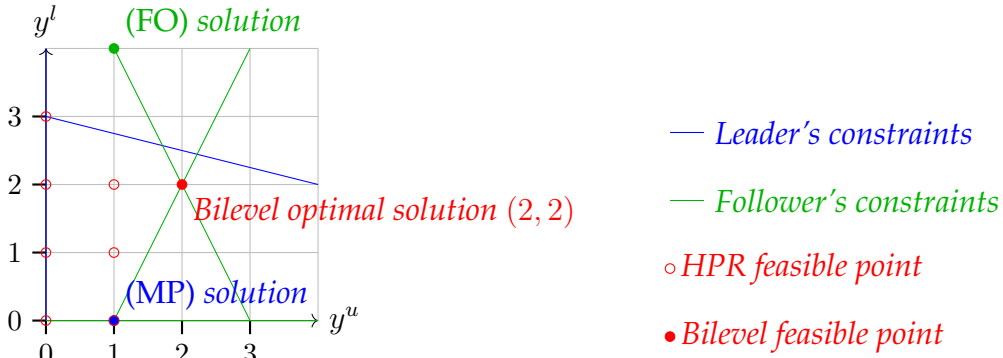
In this section we first explain some theoretical background regarding constraint (8a). Then we show how an inappropriate choice of ϵ used for the approximation (9a)-(9b) can lead to algorithm failure even on a very simple bilevel problem satisfying Assumption 4. In particular, we show that a value for ϵ ensuring correctness of Algorithm 1 with PROJECTION TEST as described in [YGZY18] depends also on the constraint representation of the bilevel problem.

First of all, note that constraint (8a) to be approximated in PROJECTION TEST implies optimizing over non-closed sets, as will be explained later in more detail in Section 3, see (11) and (12) therein. Hence, its realization poses problems not just from a practical viewpoint of, e.g., unavailable solver features. Recall that, as stated in Section 2, sometimes no bilevel optimum can be attained precisely because the bilevel feasible region is not compact, cf. Example 6.2.1(ii) from [Bar98]. Assumption 4 eliminates this theoretical problem, i.e. when upper-level variables do not influence the follower's feasible region, the bilevel optimum under the optimistic assumption can always be attained.

However, the approximation (9a)-(9b) can still cause problems even for bilevel problems satisfying Assumptions 1-4. An ϵ that is on the one hand small enough to ensure correctness of Algorithm 1 with PROJECTION TEST as described in [YGZY18], and on the other hand big enough so as not to lead to numerical untractability, may be hard to find in practice. The following example shows how all bilevel feasible points can be cut off and a problem instance gets erroneously classified as infeasible just because of constraint scaling.

Example 2.9. Let $\nu \geq 0$ be some parameter. We solve the following bilevel problem employing the algorithm from [YGZY18]:

$$\begin{aligned} \max_{y^u} \quad & y^u - y^l \\ \text{s.t.} \quad & y^u + 4y^l \leq 12, \quad y^u \in \mathbb{Z}_+ \\ \max_{y^l} \quad & y^l \\ \text{s.t.} \quad & \nu y^u + \frac{\nu}{2}y^l \leq 3\nu, \quad 2y^u - y^l \leq 2, \quad y^l \in \mathbb{Z}_+ \end{aligned}$$



We apply Algorithm 1 to this example problem and start with $k = 0$, $UB = \infty$, $LB = -\infty$:
Problem (MP)

$$\begin{aligned} \max_{y^u, y^l} \quad & y^u - y^l \\ \text{s.t.} \quad & y^u + 4y^l \leq 12, \quad \nu y^u + \frac{\nu}{2}y^l \leq 3\nu, \quad 2y^u - y^l \leq 2, \quad y^u, y^l \in \mathbb{Z}_+ \end{aligned}$$

yields $(y_0^{u,*}, y_0^{l,*}) = (1, 0)$ together with $UB = 1$. Problem (FO) for $y_0^{u,*} = 1$

$$\begin{aligned} \max_{y^l} \quad & y^l \\ \text{s.t.} \quad & \nu y_0^{u,*} + \frac{\nu}{2}y^l \leq 3\nu, \quad 2y_0^{u,*} - y^l \leq 2, \quad y^l \in \mathbb{Z}_+ \end{aligned}$$

yields $\hat{y}_0^l = 4$ and $\theta_0(y_0^{u,*} = 1) = 4$. Finally, problem (BF)

$$\begin{aligned} \max_{y^l} \quad & -y^l \\ \text{s.t.} \quad & \nu y_0^{u,*} + \frac{\nu}{2}y^l \leq 3\nu, \quad 2y_0^{u,*} - y^l \leq 2, \quad y_0^{u,*} + 4y^l \leq 12, \quad y^l \geq \theta_0(y_0^{u,*} = 1) = 4, \quad y^l \in \mathbb{Z}_+ \end{aligned}$$

is infeasible as the two last constraints imply $4 \leq y^l \leq \frac{11}{4}$. So we add $y^{l,0} = \hat{y}_0^l = 4$ to Y_1^L . Thus, approximative PROJECTION TEST (PT₀) reads:

$$\begin{aligned} \min_{t_1^0, t_2^0} \quad & t_1^0 + t_2^0 \\ \text{s.t.} \quad & -t_1^0 + \nu y^u + \frac{\nu}{2} y^{l,0} \leq 3\nu, \quad -t_2^0 + 2y^u - y^{l,0} \leq 2, \quad t_1^0, t_2^0 \in \mathbb{R}_+ \end{aligned}$$

For any optimal solution $(t_1^{0,*}, t_2^{0,*})$ of (PT₀) we have

$$t_1^{0,*} = \max \{ \nu (y^u - 1), 0 \}, \quad t_2^{0,*} = \max \{ 2 (y^u - 3), 0 \}.$$

For $0 \leq t_1^0 + t_2^0 < \epsilon$, constraint (9b)

$$\epsilon(1 - \psi^0) \leq t_1^0 + t_2^0, \quad \psi^0 \in \{0, 1\}$$

implies $\psi^0 = 1$.

Assuming $t_1^0 = t_1^{0,*}$ and $t_2^0 = t_2^{0,*}$ are optimal solutions of (PT₀), constraint (9a) for $y^{l,0} = 4$, given by

$$[\max \{ \nu (y^u - 1), 0 \} + \max \{ 2 (y^u - 3), 0 \} < \epsilon] \rightarrow [y^l \geq 4],$$

is added to problem (MP). Observe that there are only three HPR-feasible y^u in this example problem, namely $\{0, 1, 2\}$. For $y^u = 0$ and $y^u = 1$ the optimal objective function value of (PT₀) is 0 irrespective of the value of ν . In this case the above constraint correctly activates the corresponding OPTIMALITY PACKAGE, i.e. $[y^l \geq 4]$ for $y^u \in \{0, 1\}$. In particular, it means that there are no bilevel feasible solutions for $y^u \in \{0, 1\}$. The correct rational response of the follower to the only remaining HPR-feasible point $y^u = 2$ is $y^l = 2$. However, for $\nu < \epsilon$ the above constraint will impose the implication $[y^u = 2] \rightarrow [y^l \geq 4]$, which cuts off the only bilevel feasible point $(2, 2)$ and leads to the instance being classified as infeasible.

So, for any $\epsilon > 0$ to be chosen for the approximative PROJECTION TEST there exists a half-space representation of the lower-level problem from Example 2.9 such that Algorithm 1 will erroneously classify the corresponding bilevel problem as infeasible.

For numerical reasons, optimization solvers usually employ some kind of scaling procedure on the problem before the actual solving process starts, see, e.g., [Chv83]. This obviously changes the original representation of the problem and thus has, as shown on Example 2.9, an influence on the suitability of the choice of ϵ . Unfortunately, in most cases the exact scaling process of the solver used is neither known nor accessible to the user.

3 Exact Algorithm Realization Suitable for Nonlinearities

In this section we describe our realization of PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for each $y^{l,k} \in Y_k^L$ in the master problem (MP), i.e.

$$\left[y^u \in \text{Proj}_{(y^u)} P(y^{l,k}) \right] \implies [g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k})],$$

which makes Algorithm 1 exact and allows its extension to a nonlinear setting. Analogously to the implementation described in Section 2.1, we employ a binary variable ψ^k for every $y^{l,k} \in Y_k^L$ to separate the above routines of the algorithm:

$$\left[y^u \in \text{Proj}_{(y^u)} P(y^{l,k}) \right] \implies [\psi^k = 1] \quad (10a)$$

$$[\psi^k = 1] \implies [g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k})]. \quad (10b)$$

Subsection 3.1 deals with (10a), and Subsection 3.2 handles (10b). At the end of this section we present our algorithm version with the proof of its correctness and finite termination.

3.1 Exact Projection Test

As has been seen in the Example 2.9, approximative PROJECTION TEST applies OPTIMALITY PACKAGE for $y^{l,k}$ also to some $y^u \notin \text{Proj}_{(y^u)} P(y^{l,k})$. Here we aim at precise handling by adding the implication for $y^u \in \text{Proj}_{(y^u)} P(y^{l,k})$, but abstaining from adding it for any $y^u \notin \text{Proj}_{(y^u)} P(y^{l,k})$. Note that $\text{Proj}_{(y^u)} P(y^{l,k})$ is a discrete set, so the required implication (10a) could be imposed by some kind of enumeration procedure. However, this would mean that Algorithm 1 would enumerate not only y^l , but combinations of y^l and y^u . Therefore we want to avoid enumeration of y^u whenever possible and use dual formulations for PROJECTION TEST, similar to the technique described in Section 2.1.

For this we utilize the linear relaxation of $P(y^{l,k})$, denoted by $P_{\text{lin}}(y^{l,k})$, i.e.

$$P_{\text{lin}}(y^{l,k}) = \{ (y^u, x^l) : G(y^u, x^l, y^{l,k}) \leq 0, y^u \in \mathbb{R}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R} \}$$

(cf. Definition 2.1). Correspondingly, we denote the projection of $P_{\text{lin}}(y^{l,k})$ to the space of upper-level discrete variables by

$$\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k}) = \{ y^u \in \mathbb{R}_+^{m_Z} : \exists x^l \in \mathbb{R}_+^{n_R} \text{ with } (y^u, x^l) \in P_{\text{lin}}(y^{l,k}) \}.$$

Note that while $\text{Proj}_{(y^u)} P(y^{l,k})$ and $\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$ contain exactly the same integer points, $\text{Proj}_{(y^u)} P(y^{l,k})$ is a discrete set whereas $\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$ is a continuous closed set.

The implication

$$\left[y^u \in \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k}) \right] \implies [\psi^k = 1], \quad (11)$$

which is equivalent to the disjunction

$$\left[y^u \notin \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k}) \right] \vee [\psi^k = 1], \quad (12)$$

however, cannot be modeled directly, as the complement of $\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$ is an open set. Consequently, for linearly relaxed $y^u \in \mathbb{R}_+^{m_z}$ the set on which (11) is true is not a closed set.

One option to handle this obstacle is to use an approximation as it is done in [YGZY18], described in Section 2.1. However, as discussed in Section 2.2, we want to avoid this and obtain an exact algorithm. So our idea is the following:

- (a) Find an open subset $U \subseteq \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$ for which the implication $[y^u \in U] \implies [\psi^k = 1]$ can be modeled using standard duality theory.
- (b) Model the remaining requirement $[y^u \in (\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U)] \implies [\psi^k = 1]$ via disjunction over $y^u \in (\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U)$.

Since (b) will involve some kind of enumeration, U should be chosen as large as possible such that (a) can still be modeled conveniently. We choose $U = \text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})^\circ)$, i.e. the interior of $P_{\text{lin}}(y^{l,k})$ projected to the space of discrete upper-level variables. Note that the projection is an open map, so U is indeed an open set in the target space.

Remark 3.1. *Since we would like U to be as large as possible, $U = \text{Proj}_{(y^u)}^\circ(P_{\text{lin}}(y^{l,k}))$ seems to be the canonical candidate as it is by definition the largest open subset of $\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$. However, this set is more difficult to describe. In particular, note that $\text{Proj}_{(y^u)}^\circ(P_{\text{lin}}(y^{l,k})) \neq \text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})^\circ)$ in general as can be seen from the following example:*

Suppose $G: \mathbb{R}^3 \mapsto \mathbb{R}$ is a scalar function with $G(y^u, x^l, y^{l,k}) = x^l - (y^u - 1)^2$. In this case, for any $y^{l,k}$ we have

$$\text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k}) = \{y^u \in \mathbb{R}_+ : \exists x^l \in \mathbb{R}_+ \text{ with } x^l \leq (y^u - 1)^2\} = \mathbb{R}_+,$$

and hence its interior is equal to $\mathbb{R}_{>0}$. However, for $y^u = 1$ there is no $x^l \in \mathbb{R}_+$ such that $x^l < (y^u - 1)^2$ and therefore

$$1 \notin \{y^u \in \mathbb{R}_+ : \exists x^l \in \mathbb{R}_+ \text{ with } x^l < (y^u - 1)^2\} = \text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})^\circ).$$

One may object that $U = \emptyset$ if $P_{\text{lin}}(y^{l,k})$ is not full-dimensional, which happens regularly in practical instances due to equations in the follower problem. Nonetheless, an adjustment for U in this case is possible and will be presented in Section 3.1.1.

W.l.o.g. we assume in Section 3.1 and Section 3.2 that there are no lower-level constraints which depend on y^l only, i.e. where neither x^l nor y^u are present. Indeed, as such constraints are always satisfied for a $y^{l,k}$ that resulted from solving (FO) or (BF) and have no influence on other variables, they can be safely disregarded once discrete lower-level variables are fixed to $y^{l,k}$.

We introduce the following auxiliary optimization problem for PROJECTION TEST:

$$\begin{aligned} \max_{x^{l,k}, t^k} \quad & t^k & (\text{PT}) \\ \text{s.t.} \quad & G_R(y^u)x^{l,k} + (t^k)_s \leq -G_c(y^u) - G_Z(y^u)y^{l,k} \\ & x^{l,k} \in \mathbb{R}_+^{n_R}, t^k \in \mathbb{R}, \end{aligned}$$

where $(t^k)_s = (t^k, \dots, t^k)^\top \in \mathbb{R}^s$ is the vector having t^k in every component.

It is easy to see that $y^u \in \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$ if and only if (PT) has a feasible solution with $t \geq 0$. Furthermore, we have the following:

Lemma 3.2. *For any fixed $y^{l,k}$ and \bar{y}^u the problem (PT) has a nonempty feasible region. Moreover, the optimal solution part $t^{k,*}$ is unique, and the following equivalences hold:*

$$\text{Case 1: } t^{k,*} > 0 \iff \bar{y}^u \in \text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})^\circ) = U$$

$$\text{Case 2: } t^{k,*} = 0 \iff \bar{y}^u \in \left(\text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})) \setminus U \right)$$

$$\text{Case 3: } t^{k,*} < 0 \iff \bar{y}^u \notin \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$$

Proof. Recall that due to Assumption 1 all variables of both levels have finite bounds. Since t^k is a free slack variable present in every constraint of (PT), with $x^{l,k} = 0$ it can be chosen such that all constraints are satisfied. Since the objective function of (PT) depends only on one decision variable t^k , the optimal solution part $t^{k,*}$ is unique. For the optimal solution $t^{k,*}$ at least one of the constraints is satisfied with equality. Thus $t^{k,*}$ is per construction the maximal slack applicable to all lower-level constraints with lower-level variables fixed to $y^{l,k}$. Since $y^{l,k}$ is a constant, Case 1 is equivalent to the existence of some $\bar{x}^{l,k}$ such that

$$(\bar{y}^u, \bar{x}^{l,k}) \in \{ (y^u, x^l) : G(y^u, x^l, y^{l,k}) < 0, y^u \in \mathbb{R}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R} \} = P_{\text{lin}}(y^{l,k})^\circ$$

if G does not contain any tautological equations for the particular $y^{l,k}$. Recall that w.l.o.g. we have excluded the possibility of lower-level equations containing no other variables apart from y^l , so this translates to $\bar{y}^u \in \text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})^\circ)$. Consequently, Case 2 corresponds to

$$\bar{y}^u \in \left(\text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})) \setminus U \right).$$

This completes the proof. \square

Let us now consider task (a), corresponding to the first case from Lemma 3.2. For some sufficiently large number M_μ (details of which we discuss in Section 3.1.1), we introduce the following inequality:

$$t^{k,*} \leq \psi^k M_\mu \tag{13}$$

It ensures that $\left[\bar{y}^u \in \text{Proj}_{(y^u)} \left(P_{\text{lin}} \left(y^{l,k} \right)^\circ \right) \right] \implies [\psi^k = 1]$ due to Case 1 of Lemma 3.2.

We can formulate the dual problem to (PT):

$$\begin{aligned} \min_{\mu} \quad & - \left(G_c(y^u) + G_Z(y^u)y^{l,k} \right)^\top \mu & \text{(PT dual)} \\ \text{s.t.} \quad & G_R(y^u)^\top \mu \geq 0^{n_R} \\ & \mu^\top \mathbb{1}_s = 1 \\ & \mu \in \mathbb{R}_+^s \end{aligned}$$

with an s -dimensional all-ones vector $\mathbb{1}_s = (1, \dots, 1)^\top$.

We will show that we need only dual feasibility constraints to be able to reformulate (13) as

$$- \left(G_c(y^u) + G_Z(y^u)y^{l,k} \right)^\top \mu \leq \psi^k M_\mu. \quad (13a)$$

Lemma 3.3. *Let $\psi^k \in \{0, 1\}$ and M_μ be some sufficiently large number. Let strong duality hold for (PT) and (PT dual). Adding (PT dual) and (13a) to the master problem (MP) implies*

$$\begin{aligned} \left[y^u \in \text{Proj}_{(y^u)} \left(P_{\text{lin}} \left(y^{l,k} \right)^\circ \right) \right] & \implies (\psi^k = 1) \\ \left[y^u \notin \text{Proj}_{(y^u)} \left(P_{\text{lin}} \left(y^{l,k} \right)^\circ \right) \right] & \implies \text{(PT dual) and (13a) impose no restrictions on (MP)} \end{aligned}$$

Proof. The weak duality theorem implies

$$- \left(G_c(y^u) + G_Z(y^u)y^{l,k} \right)^\top \mu \geq t^k$$

for every feasible solution of (PT) and (PT dual). Lemma 3.2 states that $t^{k,*} > 0$ if and only if $y^u \in \text{Proj}_{(y^u)} \left(P_{\text{lin}} \left(y^{l,k} \right)^\circ \right)$ and, therefore,

$$- \left(G_c(y^u) + G_Z(y^u)y^{l,k} \right)^\top \mu \geq t^{k,*} > 0$$

for every μ satisfying (PT dual). Then ψ^k is set to 1 due to (13a).

Note that in the master problem (MP) the variables μ appear only in constraints from (PT dual) and in (13a), which do not involve any other variables apart from ψ^k . Thus ψ^k is the only variable that links these constraints to the rest of (MP). Consequently, if μ can be chosen such that no restriction is imposed on the value of ψ^k , i.e. such that the left-hand side of (13a) is nonpositive, these constraints exert no influence on (MP). This is indeed possible due to strong duality and Cases 2 and 3 of Lemma 3.2. □

Remark 3.4. *Note that unless strong duality holds between (PT) and (PT dual), the above approach may erroneously classify some upper-level variable values as belonging to a certain projection when in fact they do not. However, the required strong duality is satisfied due to Assumption 2.*

Thus, adding (PT dual) and (13a) to (MP) ensures correct handling for the Cases 1 and 3 from Lemma 3.2 and completes task (a) from p. 15.

Next we consider task (b) handling Case 2. We are unable to add the implication

$$\left[y^u \in \left(\text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})) \setminus U \right) \right] \implies [\psi^k = 1] \quad (14)$$

directly to the formulation for the same reason we could not do so for implication (11).

Instead, we may add a weaker implication to (MP) that considers only the optimal solution values $y_k^{u,*}$ of the upper-level discrete variables in the current algorithm iteration:

$$[y^u = y_k^{u,*}] \implies [\psi^k = 1] \quad (15)$$

An obvious way to enforce implication (15) is to employ a so-called *no-good cut* [HO03, DFLL10], which cuts off a specific solution. A similar approach was pursued in [Wen16] for a Benders' decomposition algorithm with a mixed-integer linear subproblem. In it, each assignment of discrete variables corresponds to a non-convex set, which is the union of the complementary set and the boundary of a polyhedron. The no-good cut is used to exclude cases in further iterations that lead to points on faces of the polyhedron.

A way to realize a no-good cut for implication (15) is to use a quadratic constraint:

$$1 - \sum_{i=1}^{m_Z} (y_i^u - y_{k,i}^{u,*})^2 \leq \psi^k. \quad (16)$$

In case all upper-level integer variables are binary, instead of (16) one may use the linear constraint

$$1 + \sum_{y_{k,i}^{u,*}=1} (y_i^u - 1) - \sum_{y_{k,i}^{u,*}=0} y_i^u \leq \psi^k. \quad (17)$$

Remark 3.5. Note that (14) can be imposed on the master problem gradually by adding suitable no-good constraints realizing the implication (15) for some $y^{u,k} \in \left(\text{Proj}_{(y^u)} (P_{\text{lin}}(y^{l,k})) \setminus U \right) \cap \mathbb{Z}_+^{m_Z}$, where $y^{u,k}$ is the upper-level discrete part of the optimal solution of (MP) in iteration k .

The required implication (10a) can be imposed on the master problem by adding (PT dual), (13a) and a finite number of no-good constraints realizing (14), which at the same time imposes no implications on any $y^u \notin \text{Proj}_{(y^u)} P(y^{l,k})$. The number of required no-good constraints is finite because the number of HPR-feasible y^u is finite. Note that Assumption 4 is essential for this to work as it ensures that only discrete upper-level variables influence the set of feasible follower responses.

3.1.1 Projection Test with Equalities

If the description of $P_{\text{lin}}(y^{l,k})$ contains equations, $P_{\text{lin}}(y^{l,k})^\circ$ is empty, i.e. Case 1 from Lemma 3.2 never occurs. Thus, the primarily desired implication (11) would be imposed

by (15) only, which would lead to adding a lot of no-good cuts and, consequently, a large number of algorithm iterations. In order to avoid this, we propose a special handling for the case of equality constraints being present in the lower-level problem formulation.

First we divide the lower-level matrix-valued constraint functions G_R, G_c and G_Z into two functions each according to their appearance in equality and inequality constraints, i.e., G_R^E, G_c^E, G_Z^E and G_R^I, G_c^I, G_Z^I , respectively. We denote the number of inequality constraints of the lower level by $s^I \leq s$.

Our aim is to substitute

$$U = \text{Proj}_{(y^u)} \left(P_{\text{lin}}(y^{l,k})^\circ \right) \text{ by } U^I = \text{Proj}_{(y^u)} \left(P_{\text{lin}}^I(y^{l,k})^\circ \cap P_{\text{lin}}(y^{l,k}) \right),$$

where $P_{\text{lin}}^I(y^{l,k}) = \{(y^u, x^l) : G^I(y^u, x^l, y^{l,k}) \leq 0, y^u \in \mathbb{R}_+^{m_Z}, x^l \in \mathbb{R}_+^{n_R}\}$ and $G^I(y^u, x^l, y^l)$ are the inequality constraints of the lower level. Hence,

$$U^I = \{y^u \in \mathbb{R}_+^{m_Z} \mid \exists x^l \in \mathbb{R}_+^{n_R} : G^I(y^u, x^l, y^{l,k}) < 0, G^E(y^u, x^l, y^{l,k}) = 0\}.$$

To achieve this we modify the primal projection test problem (PT) by abstaining from adding slack variables to equality constraints of the lower level:

$$\begin{aligned} \max_{x^{l,k}, t^k} \quad & t^k & (\text{PTEq}) \\ \text{s.t.} \quad & G_R^I(y^u)x^{l,k} + (t^k)_{s^I} \leq -G_c^I(y^u) - G_Z^I(y^u)y^{l,k} \\ & G_R^E(y^u)x^{l,k} = -G_c^E(y^u) - G_Z^E(y^u)y^{l,k} \\ & x^{l,k} \in \mathbb{R}_+^{n_R}, t^k \in \mathbb{R} \end{aligned}$$

with $(t^k)_{s^I} = (t^k, \dots, t^k)^\top \in \mathbb{R}^{s^I}$ a vector having t^k in every component.

Note that in contrast to (PT), (PTEq) is not necessarily feasible. However, we will show that its dual is always feasible.

With the dual variables μ^E and μ^I corresponding to equality and inequality constraints of (PTEq), respectively, we formulate the dual problem to (PTEq):

$$\begin{aligned} \min_{\mu^I, \mu^E} \quad & - (G_c^I(y^u) + G_Z^I(y^u)y^{l,k})^\top \mu^I - (G_c^E(y^u) + G_Z^E(y^u)y^{l,k})^\top \mu^E & (\text{PTEq dual}) \\ \text{s.t.} \quad & G_R^I(y^u)^\top \mu^I + G_R^E(y^u)^\top \mu^E \geq 0^{n_R} \\ & \mu^{I^\top} \mathbb{1}_{s^I} = 1 \\ & \mu^I \in \mathbb{R}_+^{s^I} \end{aligned}$$

with an s^I -dimensional all-ones vector $\mathbb{1}_{s^I} = (1, \dots, 1)^\top$.

Lemma 3.6. *The optimization problem (PTEq dual) always has a nonempty feasible region.*

Proof. Recall that according to Lemma 3.2, (PT) always has a nonempty feasible region. Thus, although (PTEq) is not necessarily feasible, its infeasibility can arise only from violated equality constraints. Therefore (PTEq) without equality constraints is always feasible with finite optimal value, as consequently is the case for its dual

$$\begin{aligned}
\min_{\mu^I} \quad & - (G_c^I(y^u) + G_Z^I(y^u)y^{l,k})^\top \mu^I && \text{(PTwithoutEq dual)} \\
\text{s.t.} \quad & G_R^I(y^u)^\top \mu^I \geq 0^{n_R} \\
& \mu^I \mathbb{1}_{s^I} = 1 \\
& \mu^I \in \mathbb{R}_+^{s^I}.
\end{aligned}$$

For any feasible solution $\bar{\mu}^I$ of (PTwithoutEq dual), $(\bar{\mu}^I, \mu^E = 0)$ constitutes a feasible solution of (PTEq dual). \square

Corollary 3.7. *If (PTEq) is infeasible, (PTEq dual) is unbounded.*

Proof. The dual of an infeasible linear optimization problem can be either infeasible or unbounded according to the LP theory, see Corollary 2.5 from [WN99]. Lemma 3.6 shows that the first case can not occur with the considered primal dual pair. Therefore the second case is true and an infeasibility of (PTEq) implies unboundedness of (PTEq dual). \square

Analogously to (13a), we introduce the following constraint to complete the projection test (PTEq dual) with equalities:

$$- (G_c^I(y^u) + G_Z^I(y^u)y^{l,k})^\top \mu^I - (G_c^E(y^u) + G_Z^E(y^u)y^{l,k})^\top \mu^E \leq \psi^k M_\mu. \quad (13b)$$

Now we are ready to provide the analogon of Lemma 3.3 for the case of equality constraints being present on the lower level:

Lemma 3.8. *Let $\psi^k \in \{0, 1\}$ and M_μ be some sufficiently large number. If strong duality holds for (PTEq) and (PTEq dual) whenever (PTEq) is feasible, adding (PTEq dual) and (13b) to the master problem (MP) implies:*

$$\begin{aligned}
\left[\bar{y}^u \in \text{Proj}_{(y^u)} \left(P_{\text{lin}}^I(y^{l,k})^\circ \cap P_{\text{lin}}(y^{l,k}) \right) \right] &\implies \psi^k = 1 \\
\left[\bar{y}^u \notin \text{Proj}_{(y^u)} \left(P_{\text{lin}}^I(y^{l,k})^\circ \cap P_{\text{lin}}(y^{l,k}) \right) \right] &\implies \text{no additional restrictions on the master problem}
\end{aligned}$$

Proof. If $\bar{y}^u \in U^I = \text{Proj}_{(y^u)} (P_{\text{lin}}^I(y^{l,k})^\circ \cap P_{\text{lin}}(y^{l,k}))$, all constraints of (PTEq) can be satisfied and the maximal slack $t^{k,*}$ for inequality constraints is strictly positive, analogously to the first case of Lemma 3.2. Thus the dual (PTEq dual) is also feasible and, due to weak duality, (13b) enforces $\psi^k = 1$:

$$0 < t^{k,*} \leq - (G_c^I(y^u) + G_Z^I(y^u)y^{l,k})^\top \mu^I - (G_c^E(y^u) + G_Z^E(y^u)y^{l,k})^\top \mu^E \leq \psi^k M_\mu.$$

Thus we obtain the first implication of the lemma and proceed with the second one.

If $\bar{y}^u \notin U^I$, then either (PTEq) is infeasible, or has $t^{k,*} \leq 0$. If (PTEq) is infeasible, then the minimization problem (PTEq dual) is unbounded according to Corollary 3.7. For $t^{k,*} \leq 0$, strong duality holds between (PTEq) and (PTEq dual):

$$- (G_c^I(y^u) + G_Z^I(y^u)y^{l,k})^\top \mu^{I*} - (G_c^E(y^u) + G_Z^E(y^u)y^{l,k})^\top \mu^{E*} = t^{k,*} \leq 0.$$

Thus, for $\bar{y}^u \notin U^I$, a feasible solution (μ^I, μ^E) of (PTEq dual) can be chosen such that its objective function, which is also the left-hand side of (13b), is strictly less than or equal to 0 regardless of the feasibility of (PTEq). Therefore (13b) imposes no restrictions on the value of ψ^k if $\bar{y}^u \notin U^I$, and this completes the proof analogously to the proof of Lemma 3.3. \square

Now we address the issue of computation of M_μ . Note that it is sufficient to choose M_μ big enough such that for every HPR feasible y^u there is a feasible solution of (PTEq dual) that satisfies (13b). Then an optimal solution of (PTEq dual) also satisfies (13b) for every HPR feasible y^u , and, consequently, adding (13b) to the master problem (MP) produces the effect desired by Lemma 3.8. In the proof of Lemma 3.6 we established that for any y^u there always is a feasible solution for (PTEq dual) with $\mu^E = 0$. Therefore, a suitable M_μ is, e.g., the optimal objective function value of (PTwithoutEq dual) combined with the HPR constraint set and variables of (BLP). The resulting auxiliary optimization problem is always feasible, as (PTwithoutEq dual) is feasible for any HPR feasible y^u , and has a finite objective function value as all variables in there are bounded.

Remark 3.9. *The above proposed PROJECTION TEST including no-good cuts allows an exact realization of Algorithm 1 and also results in some more changes compared to the approximative PROJECTION TEST described in Section 2.1:*

- *We need only dual feasibility constraints together with (13a) or (13b), respectively, instead of all necessary optimality conditions for (PT) or (PTEq), respectively.*
- *Consequently, neither $x^{l,k}$ as ‘copies’ of continuous lower-level variables, nor t^k for each $y^{l,k}$ are needed.*
- *We introduce one implication, i.e. one no-good constraint, more per iteration k of the algorithm. Altogether, we introduce one new binary and s continuous variables as well as $n_R + 3$ constraints for the exact PROJECTION TEST in each iteration.*
- *We may have to enumerate a lot of or, in the worst case, all integer points of some sets $\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U^I$.*

Now we redo example 2.9 using the above proposed improvement of the algorithm. For $k = 0$, the steps solving (MP), (FO) and (BF) are identical, so we start from (PT). As the lower level does not contain equations, we employ PROJECTION TEST with U :

Example 3.10. (PT) ‘Checking what y^u allow $y^{l,0} = 4$ ’

$$\begin{aligned} \max_{t^0} \quad & t^0 \\ \text{s.t.} \quad & t^0 \leq \nu(1 - y^u) \\ & t^0 \leq 2(3 - y^u) \\ & t^0 \in \mathbb{R} \end{aligned}$$

Let us consider all three possible cases regarding $\text{Proj}_{(y^u)} P_{lin}(y^{l,0})$:

Case 1: $t^{0,*} > 0$ for $y^u = 0 \iff (y^u = 0) \in \text{Proj}_{(y^u)}(P_{lin}(y^{l,0})^\circ)$

Case 2: $t^{0,*} = 0$ for $y^u = 1 \iff (y^u = 1) \in (\text{Proj}_{(y^u)}(P_{lin}(y^{l,0})) \setminus U)$

Case 3: $t^{0,*} < 0$ for $y^u \geq 2 \iff (y^u \geq 2) \notin \text{Proj}_{(y^u)} P_{lin}(y^{l,0})$

The dual problem (PT dual) reads:

$$\begin{aligned} \min_{\mu_1^0, \mu_2^0} \quad & \nu(1 - y^u)\mu_1^0 + 2(3 - y^u)\mu_2^0 \\ \text{s.t.} \quad & \mu_1^0 + \mu_2^0 = 1 \\ & \mu_1^0, \mu_2^0 \in \mathbb{R}_+ \end{aligned}$$

Taking dual feasibility constraints together with the proposed inequality (13a)

$$\nu(1 - y^u)\mu_1^0 + 2(3 - y^u)\mu_2^0 \leq \psi^0 M_\mu.$$

we obtain that $\psi^0 = 1$ is implied exactly for $y^u = 0$ corresponding to case 1, and $\psi^0 \in \{0, 1\}$ for larger values of y^u . Indeed, consider all three possible cases regarding $\text{Proj}_{(y^u)} P_{lin}(y^{l,0})$:

Case 1: for $y^u = 0$ we have $\nu\mu_1^0 + 6\mu_2^0 \leq \psi^0 M_\mu$ which implies $\psi^0 = 1$ since $\mu_1^0 + \mu_2^0 = 1$ and $\mu_1^0, \mu_2^0 \geq 0$.

Case 2: for $y^u = 1$ we have $4\mu_2^0 \leq \psi^0 M_\mu$. Hence, $\psi^0 = 0$ is possible as $(\mu_1^0, \mu_2^0) = (1, 0)$ is a feasible solution for (PT dual).

Case 3: for $y^u \geq 2$ the objective coefficient of μ_1^0 is negative such that the feasible solution $(\mu_1^0, \mu_2^0) = (1, 0)$ even gives a negative objective value for (PT dual), again allowing $\psi^0 = 0$.

By adding the no-good cut corresponding to $y_0^{u,*}$, i.e. the implication $[y^u = y_0^{u,*} = 1] \implies [\psi^0 = 1]$, Case 2 is covered as well. Altogether, PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for $y^{l,0}$ to be added to the master problem (MP) comprise

$$[0 \leq y^u \leq 1] \rightarrow [y^l \geq 4].$$

Taken together with the leader's constraint $y^u + 4y^l \leq 12$, this implies $y^u \geq 2$, which leaves just one integer feasible point in (MP) for $k = 1$, namely $(y_1^{u,*}, y_1^{l,*}) = (2, 2)$, $UB = 0$. As the point $(2, 2)$ is bilevel-feasible, LB resulting from (FO) and (BF) is also 0, and the algorithm terminates with an optimal solution.

3.2 Implications and Optimality Packages

Recall that OPTIMALITY PACKAGE for $y^{l,k}$ in the master problem (MP) consists of the following constraint:

$$g(x^u, y^u, x^l, y^l) = g_R(x^u, y^u)x^l + g_Z(x^u, y^u)y^l \geq \theta(x^u, y^u, y^{l,k}),$$

where $\theta(x^u, y^u, y^{l,k})$ is the optimal objective function value of the follower's problem with all but continuous lower-level variables x^l fixed. Assumption 2 ensures that $\theta(x^u, y^u, y^{l,k})$ can be formulated by using necessary and sufficient optimality conditions of the lower level with fixed y^l , such as, e.g., KKT conditions. Let us denote the lower-level dual variables in iteration k by π^k . For a linear follower problem we can choose OPTIMALITY PACKAGE comprised of primal and dual feasibility constraints for the lower level as well as the strong duality equality with fixed integer variables $y^{l,k}$:

$$\begin{aligned} g_R(x^u, y^u)x^l + g_Z(x^u, y^u)y^l &\geq g_R(x^u, y^u)x^{l,k} + g_Z(x^u, y^u)y^{l,k} \\ G_R(y^u)x^{l,k} &\leq -G_c(y^u) - G_Z(y^u)y^{l,k} \\ G_R(y^u)^\top \pi^k &\geq g_R(x^u, y^u) \\ g_R(x^u, y^u)x^{l,k} &= (-G_c(y^u) - G_Z(y^u)y^{l,k}) \pi^k \\ x^{l,k} &\in \mathbb{R}_+^{n_R}, \pi^k \in \mathbb{R}_+^s. \end{aligned} \tag{18}$$

For our approach we need to implement logical implications of the form

$$\begin{aligned} [\psi^k = 1] &\implies \text{OPTIMALITY PACKAGE for } y^{l,k} \text{ is active, i.e. (18) is added to (MP)} \\ [\psi^k = 0] &\implies \text{OPTIMALITY PACKAGE for } y^{l,k} \text{ is inactive,} \\ &\text{i.e. no additional restrictions are imposed on the master problem.} \end{aligned} \tag{19}$$

Such implications are often realized by so-called *indicator constraints* [BLTW15, BBF⁺16]. An indicator constraint is a way to express logical relationships among variables by designating a binary variable to control whether a specified constraint is active or not. Some solvers provide facilities for using indicator constraints, e.g., CPLEX, which are utilized in [YGZY18]. However, the problem class considered in our work cannot be handled by such a solver due to nonlinearities.

If we use big-M formulations for (19), we have to deduce a suitable big-M coefficient for arbitrary lower-level problems. A similar problem arises in a popular solution method

for linear bilevel optimization problems where the lower level is reformulated using KKT optimality conditions, which then are linearized using big-M formulations [YGZY18]. Finding a suitable big-M coefficient in this case is already challenging as is indicated in [PM19, KLPS19]. One possibility is to derive the correct big-M coefficient based on bound propagation. The dual variables of the lower level, however, do not have finite bounds in general. Nevertheless, in the following we show that we do not require bounds for dual variables π^k of the lower level in iteration k in order to realize the implication (19).

Consider the following reformulation of the implication (19):

$$\begin{aligned}
g_R(x^u, y^u)x^l + g_Z(x^u, y^u)y^l &\geq g_R(x^u, y^u)x^{l,k} + g_Z(x^u, y^u)y^{l,k} - (1 - \psi^k) M_\pi \\
G_R(y^u)x^{l,k} &\leq -G_c(y^u) - G_Z(y^u)y^{l,k} + (1 - \psi^k) M_\pi \mathbb{1}_s \\
G_R(y^u)^\top \pi^k &\geq g_R(x^u, y^u) - (1 - \psi^k) M_\pi \mathbb{1}_{n_R} \\
g_R(x^u, y^u)x^{l,k} &= (-G_c(y^u) - G_Z(y^u)y^{l,k}) \pi^k \\
x^{l,k} &\in \mathbb{R}_+^{n_R}, \pi^k \in \mathbb{R}_+^s
\end{aligned} \tag{20}$$

with

$$\begin{aligned}
M_\pi = \max\{ &\max_{x^u, y^u, x^l, y^l, y^{l,k}} \{g_Z(x^u, y^u)y^{l,k} - g_R(x^u, y^u)x^l - g_Z(x^u, y^u)y^l\}, \\
&\max_{i, y^u, y^{l,k}} (G_c(y^u) + G_Z(y^u)y^{l,k})^\top e_i, \\
&\max_{i, x^u, y^u} g_R(x^u, y^u)e_i, \\
&0\},
\end{aligned} \tag{21}$$

where e_i are the corresponding standard basis vectors. Similarly to the calculation of M_μ described in Section 3.1.1, the calculation of M_π can be done by solving three auxiliary optimization problems with the objectives as given in (21) and constraints and variables from the HPR of the original bilevel problem. Note that in these auxiliary problems $y^{l,k}$ are variables from a set of all lower-level feasible integer configurations.

Terms appearing in the M_π formula (21) that do not include any variables can be multiplied by ψ^k in (20). Then these terms can be disregarded in the M_π calculation.

Lemma 3.11. *For $\psi^k \in \{0, 1\}$ the inequality system (20) is equivalent to (19).*

Proof. The first part of the implication (19), i.e. the one for $\psi^k = 1$, is fulfilled trivially.

In case of $\psi^k = 0$, OPTIMALITY PACKAGE corresponding to $y^{l,k}$ is inactive, i.e. constraints (20) have to be satisfied for any HPR-feasible (x^u, y^u, x^l, y^l) . This is true with, e.g., $x^{l,k} = 0$ and $\pi^k = 0$. As both $x^{l,k}$ and π^k appear only in OPTIMALITY PACKAGE for $y^{l,k}$ and nowhere else in the master problem, in an inactive OPTIMALITY PACKAGE their value can be chosen freely if the first constraint of (20) is then valid for all HPR-feasible (x^u, y^u, x^l, y^l) . \square

OPTIMALITY PACKAGE formulation (20) can be simplified even further:

$$\begin{aligned}
g_R(x^u, y^u)x^l + g_Z(x^u, y^u)y^l &\geq (-G_c(y^u) - G_Z(y^u)y^{l,k}) \pi^k + g_Z(x^u, y^u)y^{l,k} - (1 - \psi^k) \mathbf{M}_\pi \\
G_R(y^u)^\top \pi^k &\geq g_R(x^u, y^u) - (1 - \psi^k) \mathbf{M}_\pi \mathbb{1}_{n_R} \\
\pi^k &\in \mathbb{R}_+^s
\end{aligned} \tag{22}$$

with

$$\mathbf{M}_\pi = \max\left\{ \max_{x^u, y^u, x^l, y^l, y^{l,k}} \{g_Z(x^u, y^u)y^{l,k} - g_R(x^u, y^u)x^l - g_Z(x^u, y^u)y^l\}, \max_{i, x^u, y^u} g_R(x^u, y^u)e_i, 0\right\}$$

Lemma 3.12. *Assume that strong duality holds for the follower problem in continuous lower-level variables. Then, for $\psi^k \in \{0, 1\}$ the inequality system (22) is equivalent to (19).*

Proof. The case of $\psi^k = 0$ can be treated analogously to Lemma 3.11. All inequalities of (22) are satisfied for $\pi^k = 0$. As π^k appears only in the OPTIMALITY PACKAGE corresponding to $y^{l,k}$, which is inactive for $\psi^k = 0$, the values of π^k can be chosen freely without influencing the result of the overall optimization problem.

Then consider the case $\psi^k = 1$, where the idea of the proof is similar to Lemma 3.2. Due to weak duality the relation

$$g_R(x^u, y^u)x^{l,k} \leq - (G_c(y^u) + G_Z(y^u)y^{l,k}) \pi^k \tag{23}$$

is always true and is even satisfied with equality for each optimal solution pair of the primal and dual follower problem. For $\psi^k = 1$, the first inequality of (20) as well as of (22) already ensures lower-level optimality for fixed $y^{l,k}$ by imposing a lower bound on the lower-level objective function value. Consequently, in the optimal solution of the master problem, π^k must constitute an optimal solution of the dual lower-level problem with fixed $y^{l,k}$. Indeed, for $\psi^k = 1$, the master problem is feasible only if (23) is satisfied with equality, which is possible due to strong duality. Thus only the dual feasibility conditions are required to correctly impose the lower bound on the lower-level objective function for fixed $y^{l,k}$. \square

Remark 3.13. *The OPTIMALITY PACKAGE formulation (22) dispenses with the copies of lower-level continuous variables $x^{l,k}$ and corresponding primal feasibility constraints together with the explicit strong duality constraint. Thus, only s continuous variables and $n_R + 1$ constraints are added to the master problem in each iteration to realize OPTIMALITY PACKAGE.*

Note that complete elimination of $x^{l,k}$ from the problem formulations is possible only in combination with the specific form of the PROJECTION TEST described in Section 3.1. Absence of the primal lower-level feasibility constraints makes calculation of \mathbf{M}_π easier and the number itself potentially smaller.

3.3 Algorithm Form, Correctness and Finite Termination

In the following we present Algorithm 2, a modification of Algorithm 1 with our exact PROJECTION TEST, that also can decide feasibility of (BLP). In order to handle Case 2 from Lemma 3.2, we need to keep track of all encountered upper-level integer solutions $y^{u,k}$ corresponding to every generated lower-level integer configuration $y^{l,k}$, denoted by $Y^U(y^{l,k})$. For each of these $y^{u,k}$ a no-good constraint of the form (16) or, if all leader integer variables are binary, (17), is added to the master problem (MP) as part of exact PROJECTION TEST. The rest of PROJECTION TEST is composed of (PTEq dual) and (13b), or (PT dual) and (13a), depending on whether a special equality constraint treatment is needed or not. OPTIMALITY PACKAGE comprises (22).

Analogously to Lemma 2.4, we formulate the following statements:

Lemma 3.14. *For any set Y^L of lower-level feasible integer variable configurations and corresponding sets $Y^U(y^{l,k})$, the master problem (MP) is a relaxation of the original bilevel problem (BLP).*

If Y^L comprises a complete set of lower-level feasible integer variable configurations while for the corresponding sets $Y^U(y^{l,k})$ the inclusion $(\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U) \cap \mathbb{Z}_+^{mz} \subseteq Y^U(y^{l,k})$ holds, the master problem (MP) is equivalent to the original bilevel problem (BLP).

Proof. Lemma 2.4 shows the first statement for the master problem which incorporates

$$\left[y^u \in \text{Proj}_{(y^u)} P(y^{l,k}) \right] \implies [g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k})] \text{ for all } y^{l,k} \in Y^L.$$

As Lemma 3.8 and the first part of Remark 3.5 indicate, exact PROJECTION TEST imposes $[g(x^u, y^u, x^l, y^l) \geq \theta(x^u, y^u, y^{l,k})]$ for a subset of $\text{Proj}_{(y^u)} P(y^{l,k})$. Therefore, for any set Y^L of lower-level feasible integer variable configurations and corresponding sets $Y^U(y^{l,k})$, master problem (MP) with exact PROJECTION TEST is a relaxation of the master problem from Lemma 2.4, and as such a relaxation of (BLP).

The second statement of this lemma is inferred from the second part of Lemma 2.4 and the second part of Remark 3.5. Note that not necessarily all $\bar{y}^u \in (\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U) \cap \mathbb{Z}_+^{mz}$ for a complete set of lower-level feasible integer variable configurations $y^{l,k}$ have to be enumerated in order to construct a single-level reformulation of the (BLP). \square

Even less no-good constraints may be needed to find a bilevel optimal solution of (BLP), as Algorithm 2 needs to add no-good constraints only for \bar{y}^u which form part of an optimal solution of (MP) in some iteration.

To show correctness and finite termination of Algorithm 2 we need the following pendant to Lemma 2.6:

Algorithm 2 Algorithm for bilevel problems with exact projection test

- 1: Set $LB \leftarrow -\infty, UB \leftarrow +\infty, \xi \leftarrow 0, k \leftarrow 0, Y_0^L \leftarrow \emptyset$
 - 2: **while** $UB - LB > \xi$ **do**
 - 3: Solve master problem (MP) for Y_k^L and $Y^U(y^{l,k}), y^{l,k} \in Y_k^L$
 - 4: **if** (MP) is feasible **then**
 - 5: Denote its optimal solution by $(x_k^{u,*}, y_k^{u,*}, x_k^{l,*}, y_k^{l,*})$
 - 6: Set UB to the optimal objective function value of (MP)
 - 7: Solve (FO) for $(x_k^{u,*}, y_k^{u,*})$, to get $(\hat{x}_k^l, \hat{y}_k^l)$ and $\theta_k(x_k^{u,*}, y_k^{u,*})$
 - 8: Solve bilevel feasibility problem (BF) for $(x_k^{u,*}, y_k^{u,*})$ and $\theta_k(x_k^{u,*}, y_k^{u,*})$
 - 9: **if** (BF) is feasible **then**
 - 10: Denote optimal solution of (BF) as $(\tilde{x}_k^l, \tilde{y}_k^l)$
 - 11: $LB \leftarrow \max\{LB, \text{optimal objective function value of (BF)}\}$
 - 12: $y^{l,k} \leftarrow \tilde{y}_k^l$
 - 13: **else**
 - 14: $y^{l,k} \leftarrow \hat{y}_k^l$
 - 15: $y^{u,k} \leftarrow y_k^{u,*}$
 - 16: **if** $y^{l,k} \notin Y_k^L$ **then**
 - 17: Create $Y^U(y^{l,k}) = \{y^{u,k}\}$
 - 18: Create PROJECTION TEST and OPTIMALITY PACKAGE for $y^{l,k}$
and add them to (MP)
 - 19: $Y_{k+1}^L \leftarrow Y_k^L \cup \{y^{l,k}\}$
 - 20: **else**
 - 21: $Y^U(y^{l,k}) \leftarrow Y^U(y^{l,k}) \cup \{y^{u,k}\}$
 - 22: For $y^{u,k}$ create IMPLICATION of OPTIMALITY PACKAGE for $y^{l,k}$,
i.e. add a no-good constraint (16) to (MP)
 - 23: $k \leftarrow k + 1$
 - 24: **else**
 - 25: The original bilevel problem (BLP) is infeasible, terminate
 - 26: Terminate and return the optimal solution
-

Lemma 3.15. *As long as the master problem (MP) remains feasible, Algorithm 2 generates some $y^{l,k}$ and $y^{u,k}$ to be added to Y_k^L and $Y^U(y^{l,k})$, respectively, at the end of each iteration k of the while-loop. If thus generated $y^{l,k}$ is already contained in Y_k^L and $y^{u,k}$ is already contained in $Y^U(y^{l,k})$, i.e. the pair $(y^{u,k}, y^{l,k})$ has been generated in some previous iteration, Algorithm 2 terminates with a bilevel optimal solution not later than in iteration $k + 1$.*

Proof. Let $(x_k^{u,*}, y_k^{u,*}, x_k^{l,*}, y_k^{l,*})$ be the solution of (MP) in iteration k . As the follower optimality subproblem (FO) is always feasible given HPR-feasible $(x_k^{u,*}, y_k^{u,*})$, a $y^{u,k} = y_k^{u,*}$ and $y^{l,k}$ are always generated in each iteration as long as the master problem (MP) remains feasible.

Now we prove the second part of the Lemma. If the bilevel feasibility subproblem (BF) is infeasible in iteration k , then $y^{l,k} = \hat{y}_k^1$ is part of the optimal solution of (FO) computed before, and consequently $\theta(x_k^{u,*}, y_k^{u,*}) = \theta(x_k^{u,*}, y_k^{u,*}, y^{l,k})$ holds. If (BF) is feasible in iteration k , then $y^{l,k} = \tilde{y}_k^1$ is part of its optimal solution and as such also satisfies $\theta(x_k^{u,*}, y_k^{u,*}) = \theta(x_k^{u,*}, y_k^{u,*}, y^{l,k})$.

If thus generated $y^{l,k}$ is already contained in Y_k^L and $y^{u,k}$ is already contained in $Y^U(y^{l,k})$, then the corresponding PROJECTION TEST together with IMPLICATION and OPTIMALITY PACKAGE for $y^{l,k}$ are also already present in the master problem (MP) in iteration k . In particular, the no-good constraint for $(y^{u,k}, y^{l,k})$ as a part of PROJECTION TEST implies OPTIMALITY PACKAGE for $y^{l,k}$, i.e. the constraint $g(x_k^{u,*}, y_k^{u,*}, x_k^{l,*}, y_k^{l,*}) \geq \theta(x_k^{u,*}, y_k^{u,*}, y^{l,k})$ is active in (MP) in iteration k . As $\theta(x_k^{u,*}, y_k^{u,*}) = \theta(x_k^{u,*}, y_k^{u,*}, y^{l,k})$, for the upper-level decision variables $(x_k^{u,*}, y_k^{u,*})$ this OPTIMALITY PACKAGE corresponding to $y^{l,k}$ is exactly the (optimal-value function constraint). Therefore $(x_k^{u,*}, y_k^{u,*}, x_k^{l,*}, y_k^{l,*})$ is a bilevel feasible solution. Since according to the first part of Lemma 3.14 it is also an optimal solution of a relaxation of the original bilevel problem (BLP), it is a bilevel optimal solution of (BLP). \square

Theorem 3.16. *Algorithm 2 with PROJECTION TEST as described in Subsection 3.1 and IMPLICATIONS and OPTIMALITY PACKAGES as described in Subsection 3.2 either finds a bilevel optimal solution or shows infeasibility of the original problem (BLP) in finitely many iterations.*

Proof. From Lemma 3.15 we can see that Algorithm 2 has three possible outcomes in each iteration of the while-loop:

- Master problem (MP) is infeasible, which by Corollary 2.5 means the infeasibility of the original bilevel problem (BLP).
- Some $y^{l,k}$ and $y^{u,k}$ generated in iteration k are already in Y_k^L and $Y^U(y^{l,k})$, respectively, which by the last statement of Lemma 3.15 leads to termination of the algorithm with a bilevel optimal solution.
- An $y^{l,k}$ generated in iteration k is not yet in Y_k^L , or $y^{u,k} \notin Y^U(y^{l,k})$.

This means that after each iteration the algorithm either terminates or generates a not previously encountered pair $(y^{u,k}, y^{l,k})$. Unless the algorithm terminates earlier according to the first two above listed cases, it enumerates all lower-level feasible integer configurations $y^{l,k}$ and, according to Remark 3.5 some of the upper-level feasible integer configurations $y^{u,k}$, thus constructing a master problem (MP) which is equivalent to the original bilevel problem (BLP) due to Part 2 of Lemma 3.14. As the number of lower-level feasible integer configurations $y^{l,k}$ as well as the number of HPR-feasible integer configurations $y^{u,k}$ is finite by Assumption 1, Algorithm 2 terminates after finitely many iterations. \square

Corollary 3.17. *An optimum is attained for a feasible bilevel problem satisfying Assumptions 1-4.*

Suitable realizations of PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE that are necessary to confirm this result for nonlinear follower problems will be given in Section 5.

Altogether, PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE as described in this section add to the master problem 1 binary and $2s$ continuous variables as well as $2n_R + 4$ constraints per iteration. Note that via elimination of $x^{l,k}$ as suggested in this paper, the number of additionally introduced nonlinearities is reduced compared to the realization from [YGZY18]. Also, in the current paper $2n_R$ continuous variables and $4n_R + 4s - 2$ less constraints are added to the master problem per iteration of the algorithm.

However, the number of iterations can be higher due to possibly enumerating a lot of or, in the worst case, all integer points of some sets $\text{Proj}_{(y^u)}(P_{\text{lin}}(y^{l,k})) \setminus U^1$. If such enumeration occurs, only one constraint is added per additional iteration of the algorithm, namely a no-good constraint for the encountered upper-level integer configuration. Thus, unless their number is very high, the impact of these additional iterations on the size and complexity of the master problem is moderate compared to PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE routines for each newly encountered lower-level integer configuration.

4 Computational Results

Given a suitable MINLP solver to be used for the master problem, our implementation is able to handle a MINLP on the upper level and a follower problem which is linear in continuous lower-level variables. To the best knowledge of the authors, no library of bilevel instances exists where on both levels discrete variables and nonlinearities such as products of upper- and lower-level are present. Therefore, instances of the desired class were created based on the first 10 MIPS instances from [RA16].

The original instances `miblp_20_15_50_0110_10_1` to `miblp_20_15_50_0110_10_10` are all-integer with 5 upper-level variables and 10 lower-level variables each. There are 20

linear lower-level and no upper-level constraints in each instance, both objective functions are linear as well. Notice that as the main difficulty in solving bilevel problems comes from the structure of the lower level, i.e. partial construction of its optimal-value function, absence of upper-level constraints does not impede representativity of the computations. However, as the ability of the proposed algorithm to detect infeasible instances has to be tested as well, some instances with upper-level constraints were constructed too.

We also included a nonlinear toy example (Example A.2) in the appendix with constraints and integer variables on both levels, where the behavior of our implementation can be clearly retraced and the solution found can easily be verified.

Computational results given in this section comprise altogether 120 instances derived from [RA16]. In order to obtain mixed-integer nonlinear bilevel problems, each of the original 10 instances was modified as follows:

- adding $m_R \in \{5, 10, 20\}$ continuous variables to the upper level and redeclaring every second lower-level integer variable as a continuous variable,
- adding $m_R \in \{5, 10, 20\}$ continuous variables to the upper level and redeclaring every fourth lower-level integer variable as a continuous variable,
- redeclaring every second or, respectively, fourth upper-level integer variable as a continuous variable and redeclaring every second lower-level integer variable as a continuous variable,

as well as adding a bilinear term to both lower- and upper-level objective functions. Thus, 80 new instances were created, where every modification produced a distinct combination of numbers of upper- and lower-level integer and continuous variables. In order to keep as close to the original bilevel library instances from [RA16] as possible, no nonlinearities apart from the above-mentioned terms in the objective functions of both upper and lower level were added. If an existing integer variable was redeclared as continuous, it retains all its bounds as well as coefficients in all constraints and objective functions. An exception is made if an upper-level integer variable is redeclared as a continuous one, in which case its coefficients in lower-level constraints are set to 0 in order to comply with Assumption 4.

The lower bound for added continuous upper-level variables is set to 0, and their upper bound is set to the maximum of the upper bounds of upper-level integer variables. No constraint coefficients need to be produced while adding continuous upper-level variables, since there are no upper-level constraints in the original instances, and lower-level constraints should not contain upper-level continuous variables due to Assumption 4.

Coefficients for the added continuous upper-level variables in the linear part of the objective functions on both levels are generated by rearranging the corresponding objective function coefficients of the existing discrete upper-level variables. We describe only

construction of the continuous part of the upper-level objective function, as the procedure for the continuous part of the lower-level objective function is exactly the same:

$$\begin{aligned}
 m_R = m_Z = 5 : \quad & \underbrace{(1, 2, 3, 4, 5)}_{\text{upper-level obj coeffs of } y^u} \rightarrow \underbrace{(3, 4, 5, 1, 2)}_{\text{upper-level obj coeffs of } x^u} \\
 m_R = 10 > m_Z = 5 : \quad & \underbrace{(1, 2, 3, 4, 5)}_{\text{upper-level obj coeffs of } y^u} \rightarrow \underbrace{(3, 4, 5, 1, 2, 3, 4, 5, 1, 2)}_{\text{upper-level obj coeffs of } x^u}
 \end{aligned}$$

Note that the number of added upper-level continuous variables is always a positive integer multiple of $m_Z = 5$, the number of upper-level integer variables.

Regardless of the way original instances have been made mixed-integer, the following bilinear term is added to the upper-level objective function of each instance:

$$x^u \frac{1}{ub(x^u)} B x^l,$$

with B a matrix comprised of an identity matrix $I_{\min\{m_R, n_R\}}$ extended with 0-entries to fit the required dimensions, and $ub(x^u)$ the largest upper bound of all x^u variables. The lower-level objective function of each instance receives the above bilinear term with a minus sign.

As none of the instances described so far proved to be infeasible, bilevel optimization problems with upper-level constraints were constructed, again based on instances `miblp_20_15_50_0110_10_1` to `miblp_20_15_50_0110_10_10` from [RA16]. For each of the original 10 instances three instances were produced by shifting all but every second, fifth or tenth lower-level constraint to the upper level. No other modifications were done to obtain these 30 new instances, which therefore remain all-integer and linear on both levels.

We used Gurobi 9.0 [GO20] as MINLP solver, Pyomo 5.6.9 for modeling and CPython 3.7.6 for the implementation of Algorithm 2. Notice that Gurobi allows only products of variables as nonlinearities. To extend Algorithm 2 to more general MINLPs, other global nonlinear solvers such as, e.g., SCIP [GAB⁺20] or BARON [Sah20] can be used.

Computations were performed on Xeon E3-1240 v6 CPUs (4 cores, HT disabled, 3.7 GHz base frequency) with 32 GB RAM. Runtimes are stated excluding instance loading from MPS and AUX files as well as their modification, but including big-M calculations. The time limit for each instance is 2 h.

The duality gap measures the maximum relative deviation from optimality of the best feasible solution found. As proposed in [KAA⁺11], we have calculated the gap for a given lower bound LB and upper bound UB by the following formula:

$$\text{gap}(LB, UB) = \begin{cases} 0, & \text{if } LB = UB, \\ \infty, & \text{if } LB \cdot UB \leq 0, \\ \frac{|LB-UB|}{\min\{|LB|, |UB|\}}, & \text{else.} \end{cases}$$

Table 1: Computational results for original instances miblp_20_15_50_0110_10_1 to miblp_20_15_50_0110_10_10 from [RA16]

instance number	m_Z	m_R	n_Z	n_R	runtime in s	completed iterations	optimality packages
1	5	0	10	0	2.6	4	4
2	5	0	10	0	581.9	19	11
3	5	0	10	0	39.3	15	11
4	5	0	10	0	2.0	5	4
5	5	0	10	0	2.1	5	4
6	5	0	10	0	7.0	9	8
7	5	0	10	0	1.6	5	1
8	5	0	10	0	2.8	6	6
9	5	0	10	0	131.5	22	20
10	5	0	10	0	2.6	6	5

The original 10 instances from [RA16] were solved to optimality, and key characteristics of the runs are listed in Table 1 separately for each instance. From the 80 MINLP instances all but 5 were solved to optimality, while the remaining 5 instances had a relative optimality gap under 0.35%. Consolidated statistics of the runs on all 80 MINLP instances are given in Table 2 with the corresponding minimum, maximum, arithmetic mean and standard deviation for the runtime. Detailed data for each instance separately is listed in Table 6 in Appendix A.3.

From the 30 instances with upper-level constraints 14 were solved to optimality and 12 were proven infeasible. For 3 of the 4 remaining instances no decision on the feasibility could be made, while the last instance was proven feasible but not solved to optimality

Table 2: Computational results for 80 MINLP instances

	gap in %	runtime in s	completed iterations	optimality packages
mean	0.01	474.5	291.0	5.5
std	0.04	1749.3	1152.4	3.0
min	0.00	0.7	2	1
1st quartile	0.00	1.9	5	3
2d quartile	0.00	4.1	7	5
3d quartile	0.00	9.4	12	7
max	0.35	7201.0	6278	12

Table 3: Computational results for 30 IP instances with upper-level constraints

	gap in %	runtime in s	completed iterations	optimality packages
mean	inf	1189.1	96.5	15.0
std	inf	2564.3	439.5	11.1
min	0	0.9	2	2
1st quartile	0	2.9	8	7
2d quartile	0	7.8	14	11.5
3d quartile	0	208.1	24.8	20
max	inf	7200.6	2423	36

within the time limit. Consolidated statistics of the runs on these 30 instances are given in Table 3, and the full data for each instance separately can be found in Table 7 in Appendix A.3.

From altogether 120 instances, thereof 10 original library IP, 80 MINLP and 30 IP with upper-level constraints, only 9 were neither solved to optimality nor proven infeasible within the time limit. In the case of these 9 instances we can observe two possible causes of the algorithm not terminating within the time limit.

First, the number of no-good cuts can be too large, which raises the iteration count to several thousands. This behavior in particular is present in all 5 MINLP instances which were not solved to optimality and the penultimate IP instance with upper-level constraints. The exact number of no-good cuts for each instance can be inferred by subtracting the number of optimality packages from the total number of iterations listed in Appendix A.3. However, for all 5 instances the algorithm found a solution with a gap less than or equal to 0.35%, which can be considered an acceptable result for such a challenging problem type.

Second, the number of optimality packages can cause the master problem (MP) to become too hard and thus demand too much time to solve. This is the case with the remaining 3 instances, which are all linear on both levels and with upper-level constraints. Two of these instances acquired more than 30 optimality packages, and in one case the last completed master problem solve took over an hour. See Appendix A.3 for details on each instance.

Mean run time of the algorithm for all 120 instances is 10 min 20 s, median run time is less than 10 s. So more than 90% of the test instances were either solved to optimality or proven infeasible by Algorithm 2 within 2 h. Over 95% of the instances were either solved up to a relative optimality gap under 1% or proven infeasible before hitting the time limit of 2 h.

5 Algorithm Extension for Nonlinear Follower

For the presentation of the results so far we have assumed the follower problem to be linear for all x^u, y^u . In this section, we describe how to extend Algorithm 2 to the more general setting described by Assumptions 1-4. The pseudo-code description of Algorithm 2 on p. 27 stays exactly the same, but we have to generalize PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for the nonlinear setting. In order to do this, we will use the Wolfe dual for replacing linear programming duality, which also provides us with the required strong duality statements.

5.1 Exact Projection Test for Nonlinear Follower

The general concept behind our exact PROJECTION TEST as described in Section 3.1 remains the same, and the adaptation of (PT) to the nonlinear setting is straightforward:

$$\begin{aligned} \max_{x^{l,k}, t^k} \quad & t^k & \text{(PT)} \\ \text{s.t.} \quad & G(y^u, x^{l,k}, y^{l,k}) + (t^k)_s \leq 0 \\ & x^{l,k} \in \mathbb{R}_+^{n_R}, t^k \in \mathbb{R}. \end{aligned}$$

Note, however, that this is now a convex continuous problem—by Assumption 2—instead of a linear program. Its *Wolfe dual* is given as follows:

$$\begin{aligned} \min_{x^{l,k}, \mu} \quad & -\mu^\top G(y^u, x^{l,k}, y^{l,k}) + \mu^\top \nabla_{x^l} G(y^u, x^{l,k}, y^{l,k}) x^{l,k} & \text{(PT W-dual)} \\ \text{s.t.} \quad & \nabla_{x^l} G(y^u, x^{l,k}, y^{l,k})^\top \mu \geq 0^{n_R} \\ & \mu^\top \mathbb{1}_s = 1 \\ & \mu \in \mathbb{R}_+^s, \end{aligned}$$

where the partial derivative $\nabla_{x^l} G$ exists by Assumption 2. Note that the Lagrangian multipliers for the non-negativity conditions for $x^{l,k}$ have been eliminated from the formulation already, as well as the slack variable t^k .

Eliminating all primal variables from the Wolfe dual is unfortunately not possible in general, but only for special cases, including e.g. problems with linear constraints and strictly convex quadratic objective [NW06, Example 12.12]. It requires regularity of the partial derivative of the Lagrangian w.r.t. the primal variables, thus enabling usage of the implicit function theorem in order to consider $x^{l,k}$ as a function of μ . Thus, in contrast to (PT dual) for the case of a linear follower problem discussed in Section 3.1, primal variable copies are in general needed for (PT W-dual).

The statement of Lemma 3.2 holds also for the nonlinear version of projection test (PT) since the arguments in its proof do not depend on the linearity of the lower-level con-

straints. In particular, (PT) is feasible for any fixed $y^{l,k}$, y^u , and $x^{l,k}$. Additionally, Assumption 1 guarantees that its optimum is finite and attained by some $(x_0^{l,k}, t_0^k)$. Furthermore, (PT) satisfies Slater's condition in case of feasibility due to Assumption 2. Therefore there exists $\mu_0 \in \mathbb{R}_+^s$ such that $(x_0^{l,k}, t_0^k, \mu_0)$ is optimal for (PT W-dual), i.e. we have strong duality; cf. [Wol61]. Thus we obtain Lemma 3.3 for an accordingly adapted version of (13a). Hence, as no-good cuts handling Case 2 from Lemma 3.2 are independent of the lower-level problem class, Remark 3.5 holds for the nonlinear follower problem when (PT W-dual) and the corresponding analagon of (13a) are employed.

Note that using the Wolfe dual for solving bilevel problems with nonlinear lower level is not limited to our particular algorithm framework. The Wolfe dual can be employed to obtain single-level reformulations of bilevel optimization problems if, e.g., all variables have finite bounds on their respective levels, and the lower level is convex and satisfies Slater's condition. So far the majority of the single level reformulations of bilevel problems in the literature rely on optimality conditions of the lower level expressed either with strong duality in the linear case or the full KKT system in the nonlinear case. In contrast, it seems that using the Wolfe dual in bilevel optimization has been explored very little.

5.2 Projection Test with Equalities for Nonlinear Follower

We can implement special handling of equations as discussed in Subsection 3.1.1 also for the more general case. However, the situation is slightly more complicated.

Let G^I and G^E denote the functions defining the lower-level inequalities and equations, respectively. Note that by convexity due to Assumption 2, any equations must be linear in the continuous follower variables, so we can write $G^E(y^u, x^l, y^l)$ as $G^E(y^u, x^l, y^l) = G_R^E(y^u, y^l)x^l + G_c^E(y^u, y^l)$ for coefficient functions G_R^E and G_c^E .

We again modify the primal problem (PT) by applying slack variables only to the inequality constraints:

$$\begin{aligned}
 & \max_{x^{l,k}, t^k} && t^k && && \text{(PTEq)} \\
 & \text{s.t.} && G^I(y^u, x^{l,k}, y^{l,k}) + (t^k)_s \leq 0 \\
 & && G^E(y^u, x^{l,k}, y^{l,k}) = 0 \\
 & && x^{l,k} \in \mathbb{R}_+^{n_R}, t^k \in \mathbb{R}.
 \end{aligned}$$

The Wolfe dual of this problem is given by

$$\begin{aligned}
\min_{x^{l,k}, \mu^I, \mu^E} \quad & -\mu^{I\top} G^I(y^u, x^{l,k}, y^{l,k}) + \mu^{I\top} \nabla_{x^l} G^I(y^u, x^{l,k}, y^{l,k}) x^{l,k} - \mu^{E\top} G_c^E(y^u, y^{l,k}) \\
\text{s.t.} \quad & \nabla_{x^l} G^I(y^u, x^{l,k}, y^{l,k})^\top \mu^I + G_R^E(y^u, y^{l,k})^\top \mu^E \geq 0^{n_R} \quad (\text{PTEq W-dual}) \\
& \mu^{I\top} \mathbb{1}_{s^I} = 1 \\
& \mu^I \in \mathbb{R}_+^{s^I}.
\end{aligned}$$

We again face the problem that (PTEq) could—in contrast to (PT)—be infeasible. We have to show that also in this case, (PTEq W-dual) is feasible and admits a solution with objective value ≤ 0 . This would ensure that the strong duality constraint of the form of (13b) does not impose any restriction to the master problem if (PTEq) is infeasible. In fact, we will be able to show that (PTEq W-dual) is unbounded in that case, matching Corollary 3.7.

Lemma 5.1. *If (PTEq) is infeasible, (PTEq W-dual) is unbounded.*

Proof. We can show that (PTEq W-dual) is feasible in a way that is completely analogous to the proof of Lemma 3.6, obtaining a feasible solution of the form $(\bar{x}^{l,k}, \bar{\mu}^I, \mu^E = 0)$. However, this does not directly imply the statement since the Wolfe dual can in general have a finite optimum even if the primal is infeasible [Wol61].

To complete the proof, we will find an unbounded ray of (PTEq W-dual). Consider a version of the primal problem without the inequalities $G^I(y^u, x^{l,k}, y^{l,k}) + (t^k)_s \leq 0$. It is a linear problem and still infeasible if (PTEq) is. Due to the results in Subsection 3.1.1, its dual

$$\begin{aligned}
\min_{x^{l,k}, \mu^I, \mu^E} \quad & -\mu^{E\top} G_c^E(y^u, y^l) \\
\text{s.t.} \quad & G_R^E(y^u, y^l)^\top \mu^E \geq 0^{n_R} \quad (\text{PTwithoutIneq dual})
\end{aligned}$$

is unbounded. Let $\bar{\mu}^E$ be an unbounded ray, i.e. $-(\bar{\mu}^E)^\top G_c^E(y^u, y^l) < 0$ and $G_R^E(y^u, y^l)^\top \bar{\mu}^E \geq 0$. Then $(x^{l,k} = 0, \mu^I = 0, \bar{\mu}^E)$ is an unbounded ray for (PTEq W-dual). In combination with the feasible solution $(\bar{x}^{l,k}, \bar{\mu}^I, 0)$ for (PTEq W-dual), we can construct feasible points $(\bar{x}^{l,k}, \bar{\mu}^I, \lambda \bar{\mu}^E)$ with arbitrarily small objective value as $\lambda \rightarrow \infty$. \square

Therefore, we also have the result of Lemma 3.8 for the nonlinear case. Note that we may assume $\bar{x}^{l,k} = 0$ in the above proof, which will allow us to reuse the primal variable copies $x^{l,k}$ in the optimality package for $y^{l,k}$.

Choosing M_μ works similar to the situation in Section 3. For example, we can choose M_μ to be equal to the optimal objective function value of the dual of (PTEq) with inequalities only, combined with the HPR constraint set and variables of (BLP). It still holds that this problem always has a finite optimum and admits a feasible solution for (PTEq dual) with $\mu^E = 0$.

5.3 Implications and Optimality Packages for Nonlinear Follower

Assumption 2 ensures that we can still express optimality of the follower's continuous decisions via strong duality. As before, we denote the lower-level dual variables in iteration k by π^k . OPTIMALITY PACKAGE consists of primal and dual feasibility constraints for the lower level as well as the strong duality equation for fixed integer variables $y^{l,k}$:

$$\begin{aligned}
g(x^u, y^u, x^l, y^l) &\geq g(x^u, y^u, x^{l,k}, y^{l,k}) \\
G(y^u, x^{l,k}, y^{l,k}) &\leq 0 \\
\nabla_{x^l} G(y^u, x^{l,k}, y^{l,k})^\top \pi^k &\geq \nabla_{x^l} g(x^u, y^u, x^{l,k}, y^{l,k})^\top \\
\nabla_{x^l} g(x^u, y^u, x^{l,k}, y^{l,k}) x^{l,k} &= -\pi^k{}^\top G(y^u, x^{l,k}, y^{l,k}) + \pi^k{}^\top \nabla_{x^l} G(y^u, x^{l,k}, y^{l,k}) x^{l,k} \\
x^{l,k} &\in \mathbb{R}_+^{n_R}, \pi^k \in \mathbb{R}_+^s.
\end{aligned} \tag{24}$$

Note that we consider the gradient $\nabla_{x^l} g$ to be a row vector. This is for reasons of consistency with the Jacobian matrix $\nabla_{x^l} G$, which has gradients as its rows. Recall that our formulation needs to implement the following implications:

$$\begin{aligned}
[\psi^k = 1] &\implies \text{OPTIMALITY PACKAGE for } y^{l,k} \text{ is } \textit{active}, \text{ i.e. (24) is added to (MP)} \\
[\psi^k = 0] &\implies \text{OPTIMALITY PACKAGE for } y^{l,k} \text{ is } \textit{inactive}, \\
&\text{i.e. no additional restrictions are imposed on the master problem.}
\end{aligned} \tag{25}$$

In order to do this we again use a big-M formulation, adding (or, respectively, subtracting) a term $(1 - \psi^k)M_\pi$ to (from) each inequality in (24).

Just as described in Section 3.2, a sufficiently large M_π can then be found by solving auxiliary problems for the maximal constraint violations under the given bounds for x^u, y^u, x^l, y^l , while the package-specific variable copies $x^{l,k}$ and π^k are fixed to 0. This way, bounds for the dual variables of the lower level (which in general are not available) are not required for realizing the desired implications.

With similar arguments as in Lemma 3.12, the OPTIMALITY PACKAGE can be simplified to

$$\begin{aligned}
g(x^u, y^u, x^l, y^l) &\geq g(x^u, y^u, x^{l,k}, y^{l,k}) - \nabla_{x^l} g(x^u, y^u, x^{l,k}, y^{l,k}) \\
&\quad - \pi^k{}^\top G(y^u, x^{l,k}, y^{l,k}) x^{l,k} + \pi^k{}^\top \nabla_{x^l} G(y^u, x^{l,k}, y^{l,k}) x^{l,k} \\
&\quad - (1 - \psi^k) M_\pi \\
\nabla_{x^l} G(y^u, x^{l,k}, y^{l,k})^\top \pi^k &\geq \nabla_{x^l} g(x^u, y^u, x^{l,k}, y^{l,k})^\top - (1 - \psi^k) M_\pi \mathbb{1}_{n_R} \\
x^{l,k} &\in \mathbb{R}^{n_R}, \pi^k \in \mathbb{R}_+^s,
\end{aligned} \tag{26}$$

explicitly enforcing only dual feasibility and exploiting strong duality. Recall that we were able to eliminate all primal variables from (20), which, unfortunately, is not possible

in general for (26). This, however, does not impede the reduction step to (26). One should just be aware that $x^{l,k}$ are not necessarily optimal solutions of the primal follower problem with fixed $y^l = y^{l,k}$. Note that we can use the same primal variable copies $x^{l,k}$ as in (PT W-dual). This is because setting $x^{l,k}$ to any optimal follower response given fixed $y^{l,k}$ and y^u will work for both (PT W-dual) and (26) at the same time if $y^u \in \text{Proj}_{(y^u)} P_{\text{lin}}(y^{l,k})$, i.e. if the optimality package for $y^{l,k}$ is supposed to be active. Otherwise, $x^{l,k} = 0$ is always possible in both problems without imposing any relevant implications.

6 Conclusions

In this work, we proposed an exact algorithm for solving problems from the challenging class of mixed-integer nonlinear bilevel optimization problems where both leader and follower integer variables are present on both levels. Our method is based on recent work of Yue, Gao, Zeng and You [YGZY18], following the same projection-based scheme described in Algorithm 1. We turned it into an exact method under an additional Assumption 4, which bans continuous upper-level variables from lower-level constraints. In conjunction with the other assumptions it therefore guarantees that a bilevel optimum is attained if the problem is feasible in the first place—a fact for which our algorithm also contributes a constructive proof. Assumption 4 is relatively mild compared to other assumptions with the above-mentioned effect that are commonly made in the literature. The key enhancement of our algorithm is to separately realize implication (11) for an open subset U of the relevant projections, chosen to be as large as possible, and the remaining boundary cases—as outlined on p. 15.

Furthermore, we extend the algorithm from [YGZY18], which had originally been designed for bilevel problems with a linear HPR, to a more general problem class allowing nonlinearities. The limiting requirements are given by Assumptions 2-3, which essentially ensure that optimality of the continuous follower decisions can be expressed via strong duality, and that the HPR with these optimality conditions can be handled by an off-the-shelf solver. The nonlinear version of our method as described in Section 5 may be particularly attractive if primal variable copies can be eliminated from the Wolfe dual, though this is not a strict requirement for using it.

Proof-of-concept computational results have been presented for the case in which the lower level is linear in the follower variables, but products of lower- and upper-level variables may be present in the objective functions of both levels. Therefore our implementation covers a problem class for which currently no solver exists to the best of the authors' knowledge. Our method is able to solve many bilevel library instances that have been modified to also include continuous variables and nonlinear terms on both levels (unfortunately, no established library exists for this problem class yet). Still, there are clear limitations in terms of instance size, which is not surprising given the extremely chal-

lenging problems class. We think that the experiments are quite encouraging, especially considering the number of optimality packages that still allow the master problem to be solved. It is important to note that our framework relies on established MINLP solvers, so any performance improvements in the underlying MINLP solver will also automatically further benefit our approach. However, the master problem growing due to additional optimality packages is not the only limiting factor. We also observed instances for which optimum solutions of the master problem regularly ended up being boundary cases, i.e. in the relevant projection but not in U . Hence, they required a large number of no-good cuts, showing that our modification for exactness in general comes at a price. Moreover, this observation highlights the importance of avoiding such situations and consequently of our equation-handling modifications from Sections 3.1.1 and 5.2 for solving problems with equations on the lower level.

In future work, further performance gains might be achieved using cutting planes specifically designed for the structure of the master problem as it evolves during the solution process and/or by warmstarting. Perturbation of the follower problem could increase the chance for the master problem solution being in U . However, problem-specific knowledge will be necessary in order to not run into the very same problem that is illustrated in Example 2.9. In preliminary computations, we have observed that having good bounds for the dual variables involved can help the solver immensely. While such bounds cannot be given in general, dual variables often have a nice interpretation in applications, which might allow for a suitable estimation.

Acknowledgements

We thank Martin Schmidt for valuable suggestions on software and literature.

The first author thanks the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for their support within GRK 2297 MathCoRe. The research of the second author was performed partially within the Energie Campus Nürnberg and as such was supported by funding of the Bavarian State Government.

References

- [BABBL92] Omar Ben-Ayed, Charles E Blair, David E Boyce, and Larry J LeBlanc. Construction of a real-world bilevel linear programming model of the highway network design problem. *Annals of Operations Research*, 34(1):219–254, 1992.
- [Bar98] Jonathan F. Bard. *Practical bilevel optimization*, volume 30 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1998. doi:10.1007/978-1-4757-2836-1. Algorithms and applications.

- [BBF⁺16] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. On handling indicator constraints in mixed integer programming. *Comput. Optim. Appl.*, 65(3):545–566, December 2016. doi:10.1007/s10589-016-9847-8.
- [BLM⁺15] Andreas Bärman, Frauke Liers, Alexander Martin, Maximilian Merkert, Christoph Thurner, and Dieter Weninger. Solving network design problems via iterative aggregation. *Mathematical Programming Computation*, 7(2):189–217, 2015. doi:10.1007/s12532-015-0079-1.
- [BLTW15] Pierre Bonami, Andrea Lodi, Andrea Tramontani, and Sven Wiese. On mathematical programming with indicator constraints. *Mathematical Programming*, 151(1):191–223, Jun 2015. doi:10.1007/s10107-015-0891-4.
- [BOW16] Michel Baes, Timm Oertel, and Robert Weismantel. Duality for mixed-integer convex minimization. *Mathematical Programming*, 158(1-2):547–564, 2016.
- [Chv83] Vašek Chvátal. *Linear programming*. W.H.Freeman, 1983.
- [Dem02] Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.
- [Dem18] Stephan Dempe. Bilevel optimization: theory, algorithms and applications. *Optimization Online*, 2018. URL http://www.optimization-online.org/DB_HTML/2018/08/6773.html.
- [DFLL10] Claudia D’Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi. On interval-subgradient and no-good cuts. *Operations Research Letters*, 38(5):341–345, 2010. doi:10.1016/j.orl.2010.05.010.
- [DGM18] Hatim Djelassi, Moll Glass, and Alexander Mitsos. Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. *Journal of Global Optimization*, pages 1–52, 2018.
- [DK17] Stephan Dempe and F Mefo Kue. Solving discrete linear bilevel optimization problems using the optimal value reformulation. *Journal of Global Optimization*, 68(2):255–277, 2017.
- [DR09] Scott T. DeNegre and Ted K. Ralphs. A branch-and-cut algorithm for integer bilevel linear programs. In *Operations research and cyber-infrastructure*, pages 65–78. Springer, 2009.

- [FLMS16a] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. bilevel (solver for mixed-integer bilevel linear problems), 2016. URL <https://msinnl.github.io/pages/bilevel.html>.
- [FLMS16b] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. Intersection cuts for bilevel optimization. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 77–88. Springer, 2016.
- [FLMS17] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017. doi:10.1287/opre.2017.1650.
- [FLMS18] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172(1-2):77–103, 2018.
- [GAB⁺20] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The scip optimization suite 7.0. Technical Report 20-10, ZIB, Takustr. 7, 14195 Berlin, 2020.
- [GO20] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. URL <http://www.gurobi.com>.
- [GOS⁺20] Veronika Grimm, Galina Orlinskaya, Lars Schewe, Martin Schmidt, and Gregor Zöttl. Optimal design of retailer-prosumer electricity tariffs using bilevel optimization. *Omega*, page 102327, 2020. doi:10.1016/j.omega.2020.102327.
- [HO03] J.N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003. doi:10.1007/s10107-003-0375-9.
- [HS18] Daniel Huppmann and Sauleh Siddiqui. An exact solution method for binary equilibrium problems with compensation and the power market uplift problem. *European Journal of Operational Research*, 266(2):622–638, 2018.
- [KA14] Polyxeni-Margarita Kleniati and Claire S Adjiman. Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development. *Journal of Global Optimization*, 60(3):425–458, 2014.

- [KA15] Polyxeni-Margarita Kleniati and Claire S Adjiman. A generalization of the branch-and-sandwich algorithm: from continuous to mixed-integer nonlinear bilevel problems. *Computers & Chemical Engineering*, 72:373–386, 2015.
- [KAA⁺11] Thorsten Koch, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E. Bixby, Emilie Danna, Gerald Gamrath, Ambros M. Gleixner, Stefan Heinz, Andrea Lodi, Hans Mittelmann, Ted Ralphs, Domenico Salvagnin, Daniel E. Steffy, and Kati Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011. doi:10.1007/s12532-011-0025-9.
- [KLPS19] Thomas Kleinert, Martine Labbé, Fränk Plein, and Martin Schmidt. There’s no free lunch: On the hardness of choosing a correct big-m in bilevel optimization. 2019.
- [LMS98] Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management science*, 44(12-part-1):1608–1622, 1998.
- [LV13] Martine Labbé and Alessia Violin. Bilevel programming and price setting problems. *4OR*, 11(1):1–30, 2013.
- [Mit10] Alexander Mitsos. Global solution of nonlinear mixed-integer bilevel programs. *Journal of Global Optimization*, 47(4):557–582, 2010.
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, second edition, 2006. doi:10.1007/978-0-387-40065-5.
- [PM19] Salvador Pineda and Juan Miguel Morales. Solving linear bilevel problems using big-ms: not all that glitters is gold. *IEEE Transactions on Power Systems*, 2019.
- [RA16] Ted K. Ralphs and Elspeth Adams. Bilevel optimization problem library, 2016. URL <https://coral.ise.lehigh.edu/data-sets/bilevel-instances/>.
- [Ral15] Ted K. Ralphs. Mibs (mixed integer bilevel solver), 2015. URL <https://github.com/tkralphs/MibS>.
- [Sah20] N. V. Sahinidis. BARON user manual v. 2020.10.16, 2020. URL <https://www.minlp.com/downloads/docs/baron%20manual.pdf>.
- [Sta52] Heinrich von Stackelberg. *Theory of the market economy*. 1952.

- [TRD20] Sahar Tahernejad, Ted K. Ralphs, and Scott T. DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Computation*, 12(4):529–568, Dec 2020. doi:10.1007/s12532-020-00183-6.
- [VSJ96] L. Vicente, G. Savard, and J. Judice. Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89(3):597–614, June 1996. doi:10.1007/BF02275351.
- [Wen16] Dieter Weninger. *Solving mixed-integer programs arising in production planning*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2016. URL <https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/8226>.
- [WH16] H Paul Williams and JN Hooker. Integer programming as projection. *Discrete Optimization*, 22:291–311, 2016.
- [Wil17] H Paul Williams. The dependency diagram of a mixed integer linear programme. *Journal of the Operational Research Society*, 68(7):829–833, 2017. doi:10.1057/jors.2016.45.
- [WN99] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 1999.
- [Wol61] Philip Wolfe. A duality theorem for non-linear programming. *Quarterly of applied mathematics*, 19(3):239–244, 1961.
- [YGZY18] Dajun Yue, Jiyao Gao, Bo Zeng, and Fengqi You. A projection-based reformulation and decomposition algorithm for global optimization of a class of mixed integer bilevel linear programs. *Journal of Global Optimization*, Jul 2018. doi:10.1007/s10898-018-0679-1.
- [ZA14] Bo Zeng and Yu An. Solving bilevel mixed integer program by reformulations and decomposing. Preprint (Optimization Online), 2014. URL http://www.optimization-online.org/DB_HTML/2014/07/4455.html.

A Appendix

A.1 Example of an Unfavorable Instance

Example A.1. We solve the following bilevel problem employing Algorithm 1, to show that sometimes all HPR-feasible y^l assignments are encountered:

$$\begin{aligned}
& \max_{y^u} && y^u + 2y^l \\
& \text{s.t.} && y^u + y^l \leq 6 \\
& && y^u \in \mathbb{Z}_+ \\
& \max_{y^l} && -y^l \\
& \text{s.t.} && y^u - y^l \leq 0 \\
& && y^l \in \mathbb{Z}_+
\end{aligned}$$

For a graphical illustration we refer to Fig. 1, while relevant details of each iteration are listed in Table 4.

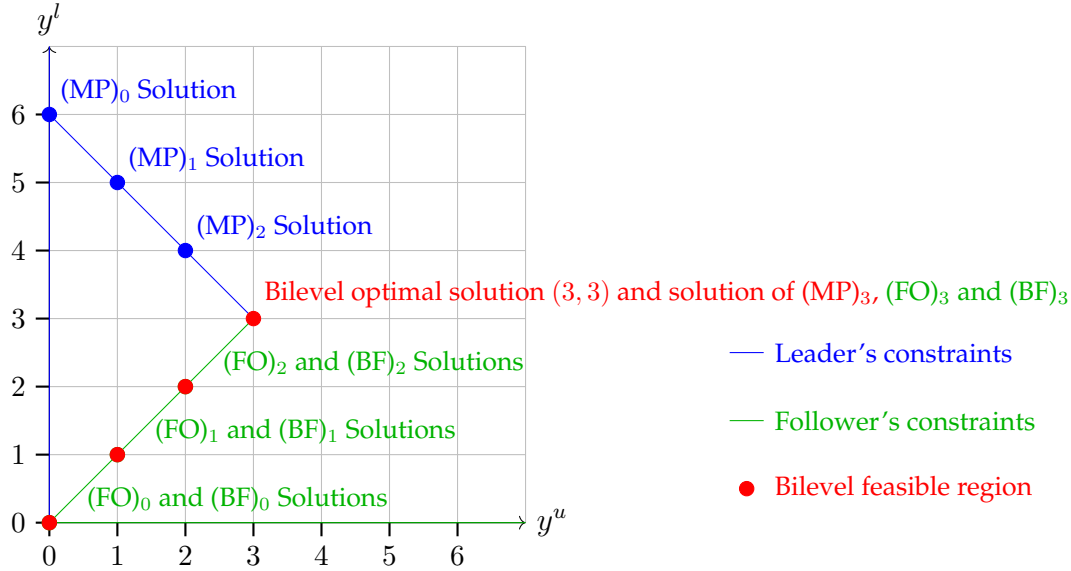


Figure 1: Graphical representation of Example A.1.

Table 4: Details for each iteration of Algorithm 1 while solving Example A.1

k	$(MP)_k$ solution	UB	$(FO)_k$ solution	$(BF)_k$ solution	LB	Y_{k+1}^L
0	$(y_0^{u,*}, y_0^{l,*}) = (0, 6)$	12	$\hat{y}_0^1 = 0, \theta_0(y_0^{u,*}) = 0$	$\tilde{y}_0^1 = 0$	0	$\{0\}$
1	$(y_1^{u,*}, y_1^{l,*}) = (1, 5)$	11	$\hat{y}_1^1 = 1, \theta_1(y_1^{u,*}) = -1$	$\tilde{y}_0^1 = 1$	3	$\{0, 1\}$
2	$(y_2^{u,*}, y_2^{l,*}) = (2, 4)$	10	$\hat{y}_2^1 = 2, \theta_2(y_2^{u,*}) = -2$	$\tilde{y}_0^1 = 2$	6	$\{0, 1, 2\}$
3	$(y_3^{u,*}, y_3^{l,*}) = (3, 3)$	9	$\hat{y}_3^1 = 3, \theta_3(y_3^{u,*}) = -3$	$\tilde{y}_0^1 = 3$	9	$\{0, 1, 2, 3\}$

A.2 Toy Example with Nonlinearities

Example A.2. We illustrate Algorithm 2 on a nonlinear toy example, which has bilinear terms in both objective functions and integer variables on both levels:

$$\begin{aligned}
& \max_{x^u, y^{u_1}, y^{u_2}} && x^u \cdot x^l \\
& \text{s.t.} && x^u = y^{u_1} + 2y^{u_2} \\
& && x^u \in \mathbb{R}_+, y^{u_1} \in \{0, 1\}, y^{u_2} \in \{0, 1\} \\
& \max_{x^l, y^l} && -x^u \cdot x^l + 10y^l \\
& \text{s.t.} && -x^l + 2y^l \leq 3(y^{u_1} + 2y^{u_2}) \\
& && x^l + 2y^l \leq 10 \\
& && 2x^l - y^l \leq 15(y^{u_1} + 2y^{u_2}) \\
& && x^l \in \mathbb{R}_+, y^l \in \mathbb{Z}_+
\end{aligned}$$

The relation between leader and follower is quite adversarial since the product term $x^u x^l$ appears with different sign in the two objective functions. The leader controls two binary variables and one continuous variable. The latter is implicitly integer due to the single master constraint and indirectly enters the right-hand side of some follower constraints via this representation. There is one continuous and one integer follower variable.

Table 5: Details for each iteration of Algorithm 2 while solving Example A.2

k	(MP) $_k$ solution	UB	(FO) $_k$ solution	(BF) $_k$ solution	LB	Y_{k+1}^L
0	$(x_0^{u,*}, y_0^{u_1,*}, y_0^{u_2,*}, x_0^{l,*}, y_0^{l,*})$ = (3, 1, 1, 10, 0)	30	$(\hat{x}_0^l, \hat{y}_0^l) = (0, 4)$ $\theta_0(x_0^{u,*}, y_0^{u_1,*}, y_0^{u_2,*}) = 40$	$(\tilde{x}_0^l, \tilde{y}_0^l) = (0, 4)$	0	{4}
1	$(x_1^{u,*}, y_1^{u_1,*}, y_1^{u_2,*}, x_1^{l,*}, y_1^{l,*})$ = (2, 0, 1, 10, 0)	20	$(\hat{x}_1^l, \hat{y}_1^l) = (2, 4)$ $\theta_1(x_1^{u,*}, y_1^{u_1,*}, y_1^{u_2,*}) = 36$	$(\tilde{x}_1^l, \tilde{y}_1^l) = (2, 4)$	4	{4}
2	$(x_2^{u,*}, y_2^{u_1,*}, y_2^{u_2,*}, x_2^{l,*}, y_2^{l,*})$ = (1, 1, 0, 8, 1)	8	$(\hat{x}_2^l, \hat{y}_2^l) = (3, 3)$ $\theta_2(x_2^{u,*}, y_2^{u_1,*}, y_2^{u_2,*}) = 27$	$(\tilde{x}_2^l, \tilde{y}_2^l) = (3, 3)$	4	{3, 4}
3	$(x_3^{u,*}, y_3^{u_1,*}, y_3^{u_2,*}, x_3^{l,*}, y_3^{l,*})$ = (2, 0, 1, 2, 4)	4	$(\hat{x}_3^l, \hat{y}_3^l) = (2, 4)$ $\theta_3(x_3^{u,*}, y_3^{u_1,*}, y_3^{u_2,*}) = 36$	$(\tilde{x}_3^l, \tilde{y}_3^l) = (2, 4)$	4	{3, 4}

Table 4 summarizes the steps of Algorithm 2. As there are only four different possible upper-level solutions, i.e. $(x^u, y^{u_1}, y^{u_2}) \in \{(0, 0, 0), (1, 1, 0), (2, 0, 1), (3, 1, 1)\}$, optimality of the solution found with Algorithm 2 can easily be verified.

Iteration 0: The first master problem is the High Point Relaxation. It has the unique optimum $(x_0^{u,*}, y_0^{u_1,*}, y_0^{u_2,*}, x_0^{l,*}, y_0^{l,*}) = (3, 1, 1, 10, 0)$ with objective function value 30. However, solving

(FO) shows that $(x_0^l, y_0^l) = (10, 0)$ is not an optimal response to $(y_0^{u1}, y_0^{u2}, x_0^l) = (3, 1, 1)$. A rational follower instead plays $(x_0^l, y_0^l) = (0, 4)$, which leads to a lower-level objective value of 40 (instead of -30 according to the HPR solution). Problem (BF) confirms that $(x_0^{u,*}, y_0^{u1}, y_0^{u2}, x_0^l, y_0^l) = (3, 1, 1, 0, 4)$ represents a bilevel-feasible solution with objective value 0.

We create PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for $y^{l,0} = 4$. This amounts to introducing two sets $\mu^0 = (\mu_1^0, \mu_2^0, \mu_3^0)^\top$ and $\pi^0 = (\pi_1^0, \pi_2^0, \pi_3^0)^\top$ of continuous dual variables for (PT dual) and (22), respectively, to the master problem, together with the binary variable ψ^0 . In addition we need the constraint (13a), and also add the no-good cut (15) upfront.

Iteration 1: We solve the master problem (MP) with the newly added variables and constraints, obtaining a solution with $(x_1^{u,*}, y_1^{u1,*}, y_1^{u2,*}, x_1^{l,*}, y_1^{l,*}) = (2, 0, 1, 10, 0)$ and $\psi^{0,*} = 0$, i.e. the optimality package that has just been added is inactive. We solve (FO) and see that the follower replies $(x^l, y^l) = (2, 4)$, which is confirmed as being bilevel-feasible together with $(x^u, y^{u1}, y^{u2}) = (2, 0, 1)$ by (BF). This might be surprising at first since it means

$$(0, 1) \in \text{Proj}_{((y^{u1}, y^{u2}))} P_{\text{lin}}(y^{l,0}).$$

However, setting $\psi^{0,*} = 0$ was still possible in the master problem. We see at closer inspection that $(0, 1)$ is not in U , but represents a boundary case. We thus add a corresponding no-good cut ensuring $[(y^{u1}, y^{u2}) = (0, 1)] \implies [\psi^0 = 1]$. Bounds are updated to $\text{UB} = 20$ and $\text{LB} = 4$.

Iteration 2: Solving (MP) yields a solution with $(x_2^{u,*}, y_2^{u1,*}, y_2^{u2,*}, x_2^{l,*}, y_2^{l,*}) = (1, 1, 0, 8, 1)$ and $\psi^{0,*} = 0$. It has objective value 8, which is the new upper bound. This time, $y^l = 4$ is indeed impossible for the follower. Instead, (FO) has the optimal solution $(x^l, y^l) = (3, 3)$, again confirmed by (BF). The corresponding bilevel-feasible solution assigns objective values of 3 to the leader and 27 for the follower, respectively. Since the former is worse than the incumbent value of 4 for LB, there is no update on the lower bound.

We create PROJECTION TEST, IMPLICATION and OPTIMALITY PACKAGE for $y^{l,0} = 3$, introducing further variables $\mu^1 = (\mu_1^1, \mu_2^1, \mu_3^1)^\top$, $\pi^1 = (\pi_1^1, \pi_2^1, \pi_3^1)^\top$, and ψ^1 . With $M_\mu = 49$ and $M_\pi = 30$ being precomputed safe choices for big-M parameters, the full master problem now reads:

$$\begin{aligned} & \max_{\substack{x^u, y^{u1}, y^{u2}, x^l, y^l, \\ \psi^0, \psi^1, \mu^0, \mu^1, \pi^0, \pi^1}} & & x^u \cdot x^l \\ & \text{s.t.} & & x^u = y^{u1} + 2y^{u2} \\ & & & -x^l + 2y^l \leq 3(y^{u1} + 2y^{u2}) \\ & & & x^l + 2y^l \leq 10 \\ & & & 2x^l - y^l \leq 15(y^{u1} + 2y^{u2}) \\ & & & -\mu_1^0 + \mu_2^0 + 2\mu_3^0 \geq 0 \\ & & & \mu_1^0 + \mu_2^0 + \mu_3^0 = 1 \\ & & & (3y^{u1} + 6y^{u2} - 8)\mu_1^0 + 2\mu_2^0 \\ & & & + (15y^{u1} + 30y^{u2} + 4)\mu_3^0 \leq \psi^0 M_\mu \end{aligned}$$

$$\begin{aligned}
& -\mu_1^1 + \mu_2^1 + 2\mu_3^1 \geq 0 \\
& \mu_1^1 + \mu_2^1 + \mu_3^1 = 1 \\
& (3y^{u_1} + 6y^{u_2} - 6)\mu_1^1 + 4\mu_2^1 \\
& \quad + (15y^{u_1} + 30y^{u_2} + 3)\mu_3^1 \leq \psi^1 M_\mu \\
& 1 + (y^{u_1} - 1) + (y^{u_2} - 1) \leq \psi^0 \\
& 1 - y^{u_1} + (y^{u_2} - 1) \leq \psi^0 \\
& 1 + (y^{u_1} - 1) - y^{u_2} \leq \psi^1 \\
& -x^u x^l + 10y^l \geq (3y^{u_1} + 6y^{u_2} - 8)\pi_1^0 + 2\pi_2^0 \\
& \quad + (15y^{u_1} + 30y^{u_2} + 4)\pi_3^0 \\
& \quad + 40 - (1 - \psi^0)M_\pi \\
& -\pi_1^0 + \pi_2^0 + 2\pi_3^0 \geq -x^u - (1 - \psi^0)M_\pi \\
& -x^u x^l + 10y^l \geq (3y^{u_1} + 6y^{u_2} - 6)\pi_1^1 + 4\pi_2^1 \\
& \quad + (15y^{u_1} + 30y^{u_2} + 3)\pi_3^1 \\
& \quad + 30 - (1 - \psi^1)M_\pi \\
& -\pi_1^1 + \pi_2^1 + 2\pi_3^1 \geq -x^u - (1 - \psi^1)M_\pi \\
& x^u, x^l \in \mathbb{R}_+, y^{u_1}, y^{u_2} \in \{0, 1\}, y^l \in \mathbb{Z}_+ \\
& \psi^0, \psi^1 \in \{0, 1\}, \mu^0, \mu^1, \pi^0, \pi^1 \in \mathbb{R}_+^3.
\end{aligned}$$

Iteration 3: Solving (MP) now yields a solution with $(x_3^{u,*}, y_3^{u_1,*}, y_3^{u_2,*}, x_3^{l,*}, y_3^{l,*}) = (2, 0, 1, 2, 4)$ and objective value 4. Feasible values for the remaining variables are $\psi^0 = \psi^1 = 1$, $(\mu_1^0, \mu_2^0, \mu_3^0) = (0, 1, 0)$, $(\mu_1^1, \mu_2^1, \mu_3^1) = (0, 0, 1)$, $(\pi_1^0, \pi_2^0, \pi_3^0) = (2, 0, 0)$, $(\pi_1^1, \pi_2^1, \pi_3^1) = (0, 0, 0)$. Note that this is the same solution as obtained in Iteration 1 from (BF) with respect to $(x^u, y^{u_1}, y^{u_2}, x^l, y^l)$. Therefore, we now have $LB = UB$. After completing the iteration, the algorithms terminates with the bilevel-optimal solution $(x^{u,*}, y^{u_1,*}, y^{u_2,*}, x^{l,*}, y^{l,*}) = (2, 0, 1, 2, 4)$.

A.3 Detailed Computational Results

Table 6: Computational results for MINLP instances (without upper-level constraints)

instance number	m_Z	m_R	n_Z	n_R	gap in %	runtime in s	completed iterations	optimality packages
1	5	0	10	0		2.58	4	4
2	5	0	10	0		581.88	19	11
3	5	0	10	0		39.26	15	11

4	5	0	10	0		1.99	5	4
5	5	0	10	0		2.11	5	4
6	5	0	10	0		7.04	9	8
7	5	0	10	0		1.64	5	1
8	5	0	10	0		2.82	6	6
9	5	0	10	0		131.49	22	20
10	5	0	10	0		2.63	6	5
1	5	5	5	5		3.05	6	5
2	5	5	5	5		119.86	18	7
3	5	5	5	5		3.84	9	7
4	5	5	5	5		10.02	12	11
5	5	5	5	5		1.78	5	3
6	5	5	5	5	0.09		3426	7
7	5	5	5	5		1.40	4	1
8	5	5	5	5		4.95	8	6
9	5	5	5	5		5.10	6	5
10	5	5	5	5		1.12	3	2
1	5	10	5	5		2.49	6	5
2	5	10	5	5		122.46	19	7
3	5	10	5	5		3.78	9	7
4	5	10	5	5		11.81	12	11
5	5	10	5	5		1.75	5	2
6	5	10	5	5	0.04		3417	7
7	5	10	5	5		1.46	4	1
8	5	10	5	5		3.69	7	6
9	5	10	5	5		5.22	6	5
10	5	10	5	5		1.28	3	2
1	5	20	5	5		2.45	6	5
2	5	20	5	5		125.46	18	7
3	5	20	5	5		4.40	9	7
4	5	20	5	5		11.51	12	11
5	5	20	5	5		10.32	30	3
6	5	20	5	5	0.02		3428	7
7	5	20	5	5		1.36	4	1
8	5	20	5	5		3.09	7	6
9	5	20	5	5		5.55	6	5
10	5	20	5	5		1.21	3	2
1	5	5	8	2		2.09	5	4
2	5	5	8	2		410.66	19	10
3	5	5	8	2		4.10	9	7

4	5	5	8	2	14.41	13	12
5	5	5	8	2	2.38	6	5
6	5	5	8	2	4.14	7	6
7	5	5	8	2	1.17	4	1
8	5	5	8	2	7.88	9	8
9	5	5	8	2	4.45	6	5
10	5	5	8	2	1.23	3	2
1	5	10	8	2	2.18	5	4
2	5	10	8	2	433.01	20	10
3	5	10	8	2	4.28	9	7
4	5	10	8	2	13.80	13	12
5	5	10	8	2	2.36	6	5
6	5	10	8	2	3.97	7	6
7	5	10	8	2	1.36	4	1
8	5	10	8	2	9.23	9	8
9	5	10	8	2	4.50	6	5
10	5	10	8	2	1.29	3	2
1	5	20	8	2	1.92	5	4
2	5	20	8	2	410.72	20	10
3	5	20	8	2	4.34	9	7
4	5	20	8	2	13.22	13	12
5	5	20	8	2	2.50	6	5
6	5	20	8	2	4.54	7	6
7	5	20	8	2	1.25	4	1
8	5	20	8	2	8.35	9	8
9	5	20	8	2	4.21	6	5
10	5	20	8	2	1.31	3	2
1	3	2	5	5	6.27	13	12
2	3	2	5	5	8.03	8	6
3	3	2	5	5	5.27	11	10
4	3	2	5	5	4.85	10	9
5	3	2	5	5	12.14	38	5
6	3	2	5	5	0.02	6091	3
7	3	2	5	5	1.54	4	3
8	3	2	5	5	2.55	6	5
9	3	2	5	5	0.35	6278	1
10	3	2	5	5	1.09	3	2
1	4	1	5	5	3.91	6	5
2	4	1	5	5	47.04	16	6
3	4	1	5	5	5.00	11	6

4	4	1	5	5	3.27	8	7
5	4	1	5	5	1.70	5	3
6	4	1	5	5	1.68	4	3
7	4	1	5	5	0.71	2	1
8	4	1	5	5	2.30	6	5
9	4	1	5	5	3.67	6	5
10	4	1	5	5	1.10	3	2

Table 7: Computational results for IP instances with upper-level constraints

instance number	m	n	feasible	gap in %	runtime in s	completed iterations	optimality packages
1	10	10	True		3.35	7	7
2	10	10	True		140.41	25	9
3	10	10	True		108.24	24	19
4	10	10	True		1.62	5	4
5	10	10	True		0.97	3	2
6	10	10	True		4988.74	29	29
7	10	10	True		1.66	4	2
8	10	10	True		230.68	18	17
9	10	10	True		691.41	33	31
10	10	10	True	151.35		35	25
1	16	4	True		3.39	8	7
2	16	4	False		51.07	14	13
3	16	4	False		34.2	22	20
4	16	4		∞		36	36
5	16	4	True		9.95	20	18
6	16	4		∞		35	34
7	16	4	True		1.04	3	2
8	16	4	False		3.74	8	8
9	16	4	False		28.92	23	20
10	16	4	True		2.06	6	5
1	18	2	True		2.71	8	7
2	18	2	False		4.51	11	10
3	18	2	False		5.64	14	13
4	18	2	False		14.76	15	15
5	18	2	False		2.72	8	5
6	18	2	False		5.74	10	10
7	18	2	False		0.91	2	2
8	18	2	False		4.65	10	10
9	18	2		∞		2423	34
10	18	2	False		528.38	36	35