Closing the Gap in Linear Bilevel Optimization: A New Valid Primal-Dual Inequality

Thomas Kleinert[1,2], Martine Labbé[3,4], Fränk Plein[3,4], and Martin Schmidt[5]

Abstract. Linear bilevel optimization problems are often tackled by replacing the linear lower-level problem with its Karush–Kuhn–Tucker (KKT) conditions. The resulting single-level problem can be solved in a branch-and-bound fashion by branching on the complementarity constraints of the lower-level problem's optimality conditions. While in mixed-integer single-level optimization branch-and-cut has proven to be a powerful extension of branch-and-bound, in linear bilevel optimization not too many bilevel-tailored valid inequalities exist. In this paper, we briefly review existing cuts for linear bilevel problems and introduce a new valid inequality that exploits the strong duality condition of the lower level. We further discuss strengthened variants of the inequality that can be derived from McCormick envelopes. In a computational study, we show that the new valid inequalities can help to close the optimality gap very effectively on a large test set of linear bilevel instances.

1. The Difficulty in Closing the Optimality Gap

Roughly speaking, branch-and-bound algorithms solve mathematical optimization problems by successively finding lower and upper bounds on the optimal objective function value. This procedure progressively decreases the optimality gap, i.e., the difference of the two bounds, until it is closed and the lower and upper bound meet. For minimization problems, every primal feasible solution provides a valid upper bound on the objective function value. Lower bounds in turn are computed by solving relaxations of the original problem. While modern branch-and-bound algorithms may find good primal solutions quickly, proving optimality by closing the optimality gap might be very challenging. It is not unusual to observe solution processes similar to the dashed line in Figure 1, which shows an exemplary evolution of the lower and upper bounds over the number of visited nodes provided by a branch-and-bound implementation. An almost optimal solution is found right at the beginning, but the lower bound improves only slowly. As a result, many branch-and-bound nodes need to be visited until the gap is closed and optimality is proved.

In mixed-integer programming, the discussed obstacle has been tackled by subsequently adding valid inequalities that cut off integer-infeasible points. In many cases, this yields tighter relaxations and ultimately delivers stronger lower bounds. Such branch-and-cut algorithms are now state-of-the-art in solving mixed-integer problems.

Linear bilevel problems, in which some variables of a linear upper-level problem need to constitute an optimal solution of a second linear optimization problem (the lower-level problem), are no exception to the behavior discussed above in general.

Date: September 16, 2020.

2010 *Mathematics Subject Classification.* 90Cxx, 90-08, 90C11, 90C46.

Key words and phrases. Bilevel optimization, Valid inequalities, Branch-and-cut, Computational analysis.

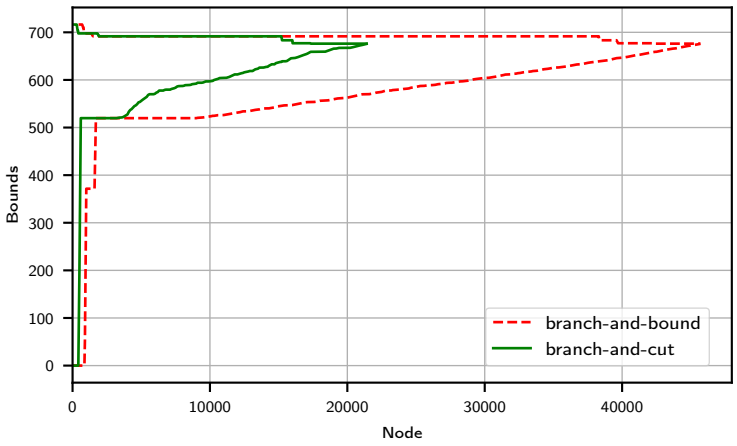

FIGURE 1. Exemplary evolution of lower and upper bounds in dependence of visited nodes for a branch-and-bound (dashed) and a branch-and-cut (solid) algorithm.

While bilevel-feasible points, i.e., points that satisfy all upper-level constraints and lower-level optimality, can often be found quickly [17], proving optimality is much more difficult. In fact, the dashed lines in Figure 1 is based on a simple branch-and-bound code for linear bilevel problems applied to an exemplary instance. Similarly to mixed-integer programming, valid inequalities could be used to provide tighter relaxations of bilevel problems by cutting off bilevel-infeasible points, i.e., points that violate optimality of the lower-level problem. However, for linear bilevel problems not many tailored valid inequalities are known.

In this paper, we derive such a valid inequality for linear bilevel problems by exploiting the strong-duality condition of the lower-level problem. This primal-dual inequality turns out to be very effective for some instances. Indeed, applying it to the same instance that was used for the dashed plot in Figure 1 yields much faster convergence; see the solid plot in Figure 1. The lower bound increases much quicker, which results in around 20 000 visited nodes compared to roughly 45 000 nodes when the inequality is not used. We will analyze the benefit gained by the proposed valid inequality in detail in a computational study later in the paper.

The remainder of the paper is structured as follows. In Section 2 we formally introduce linear bilevel problems and review existing valid inequalities. Afterward, we develop a new valid inequality based on the strong-duality condition of the lower-level problem in Section 3 and also propose some tighter variants. In Section 4, we evaluate the effectiveness of the inequalities in a computational study. Finally, we conclude in Section 5.

2. Linear Bilevel Problems and Valid Inequalities

In this paper, we consider linear bilevel problems of the form

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \quad c^\top x + d^\top y \quad \text{s.t.} \quad Ax + By \geq a,\ y \in \mathcal{S}(x), \tag{1}$$

where $\mathcal{S}(x)$ denotes the set of optimal solutions of the parameterized linear program

$$\max_{\bar{y}} \quad f^\top \bar{y} \quad \text{s.t.} \quad D\bar{y} \leq b - Cx, \tag{2}$$

with $c \in \mathbb{R}^n$, $d, f \in \mathbb{R}^m$, $A \in \mathbb{R}^{k\times n}$, $B \in \mathbb{R}^{k\times m}$, $a \in \mathbb{R}^k$, $C \in \mathbb{R}^{\ell\times n}$, $D \in \mathbb{R}^{\ell\times m}$, and $b \in \mathbb{R}^\ell$. The upper-level player (or leader) optimizes the upper-level problem (1) by anticipating the optimal reaction y of the lower-level player (or follower). Whenever

the follower is indifferent for a given x, the set of optimal solutions $\mathcal{S}(x)$ is not a singleton. In this case, the formulation in (1) establishes the so-called optimistic solution, i.e., the leader may select any solution $y \in \mathcal{S}(x)$ that is the most favorable one for the upper-level problem; see [5]. Furthermore, throughout the paper, we make the following standard assumption (see, e.g., [1–3]) that is necessary in Section 3 for the derivation of a valid inequality for Problem (1).

Assumption 1. *The shared constraint set*

$$\Omega := \{x \in \mathbb{R}^n, y \in \mathbb{R}^m \colon Ax + By \geq a,\ Cx + Dy \leq b\}$$

is nonempty and bounded.

In general, bilevel problems are intrinsically nonconvex due to their hierarchical structure and even linear bilevel problems are known to be strongly NP-hard [14]. In addition, even checking local optimality is NP-hard; see [23]. For many real-world problems that require a bilevel or even multilevel modeling, application-specific solution techniques have been developed. This includes but is not limited to fields such as energy markets [8, 13, 15], pricing problems [18, 19], or network interdiction problems [4, 10]. In a more general setting in which no problem-specific structure can be exploited, most solution techniques resort to an equivalent single-level reformulation. For linear bilevel problems, this is typically done by replacing the lower-level problem (2) by its necessary and sufficient Karush–Kuhn–Tucker (KKT) conditions, which yields a mathematical program with complementarity constraints:

$$\min_{x,y,\lambda} \quad c^\top x + d^\top y \tag{3a}$$

$$\text{s.t.} \quad (x, y) \in \Omega, \tag{3b}$$

$$\lambda \in \Omega_D := \{\lambda \geq 0 \colon D^\top \lambda = f\}, \tag{3c}$$

$$\lambda^\top (b - Cx - Dy) \leq 0. \tag{3d}$$

This reformulation was first mentioned in [12], which also contains two solution approaches exploiting the disjunctive nature of the complementarity constraints (3d). The first one is a mixed-integer linear reformulation of the KKT complementarity constraints, which requires additional binary variables and sufficiently large big-M constants. The problem can then be solved by standard mixed-integer solvers. However, big-Ms that are chosen too small can yield suboptimal or infeasible solutions [21] and verifying the correctness of a big-M constant is as hard as solving the original bilevel problem; see [16]. From today's point of view, this method should only be used if correct big-Ms can be obtained via problem-specific knowledge. The second approach mentioned in [12] overcomes this obstacle by branching directly on the complementarity constraints: for all $j = 1, \ldots, \ell$, either the primal lower-level constraint is binding, i.e., $(b - Cx - Dy)_j = 0$, or $\lambda_j = 0$ holds. This approach is evaluated in more detail in [3] and improving branching rules have been proposed in [14].

One drawback of this complementarity-based branch-and-bound approach (as well as of the mixed-integer approach using big-Ms) is a weak root relaxation. The problem that is solved in the root node is Problem (3) without the complementarity constraints (3d). In this setting, dual feasibility of the lower level (3c) is completely decoupled from the primal upper- and lower-level constraints (3b). In the original problem (3), these two sets of constraints are solely coupled by the complementarity constraints (3d)—the exact same constraints are initially relaxed and branched on in a bilevel branch-and-bound algorithm. In this view, the coupling is brought back subsequently via branching. It is thus desirable to extend such bilevel branch-and-bound approaches to branch-and-cut algorithms by adding cuts that resolve the

missing coupling, either already at the root node or later in the branch-and-bound tree. However, up to now, not too many bilevel-specific valid inequalities are known.

In [1], the complementarity conditions (3d) are used to derive *disjunctive cuts* that can be applied to the root node problem. For each violated complementarity constraint, solving a linear optimization problem (LP) yields such a cut. In a very small example, the usefulness of the cut is demonstrated. It is also shown that sometimes this cut couples constraints (3b) and (3c) and sometimes it does not.

In [2], three root node cuts are presented that can be derived from the solution of the root node problem. The first one is a *Gomory-like cut*. For each violated complementarity constraint of the lower level, two inequalities can be derived. One of them is acting on the primal upper- and lower-level variables and the other one on the dual lower-level variables. At least one of the two inequalities must be valid and is actually a cut. Since the valid one is not known, both inequalities are added to the problem and a binary switching variable is used to select the valid inequality. In this light, the two inequalities add a rather implicit coupling of the constraints (3b) and (3c). Another variant are so-called *extended cuts* that, similar to the Gomory-like cuts, also involve binary switching variables. However, it is noted that these cuts are deeper than the Gomory-like cuts. One can also derive two cuts that do not involve a switching variable. These cuts are called *simple cuts* in [2]. Again, the combination of both cuts implicitly couples the primal upper as well as lower level with the dual lower level. In a small numerical study it is shown that applying a cut generation phase at the root node that adds cuts of either one of the three types, outperforms pure branch-and-bound.

To the best of our knowledge no other general-purpose valid inequalities dedicated to linear bilevel problems have been published so far.

3. A New Valid Primal-Dual Inequality

All cuts reviewed in the last section have in common that they exploit the explicit disjunctive structure of the complementarity conditions. They are all derived from a single violated complementarity condition and it is not clear which violated one should be chosen to separate a cut. In this section, we derive a valid inequality for Problem (1) based on the aggregated complementarity conditions (3d). Using dual feasibility (3c), we can substitute $\lambda^\top D$ with f in (3d) to obtain

$$\lambda^\top b - \lambda^\top C x - f^\top y \leq 0. \tag{4}$$

This is exactly the strong-duality condition of the lower-level problem (2), as shown in the following. For a fixed upper-level decision x, the dual to the lower-level problem (2) is given by

$$\min_{\lambda \in \Omega_D} \quad \lambda^\top (b - Cx). \tag{5}$$

For every primal-dual feasible point (y, λ), weak duality

$$\lambda^\top b - \lambda^\top C x - f^\top y \geq 0$$

holds. Thus, every primal-dual feasible point satisfying Inequality (4) fulfills the strong-duality equation and is primal-dual optimal for the lower level. An alternative formulation of the single-level reformulation (3) can hence be obtained by replacing the KKT complementarity condition (3d) with the strong duality condition (4). The main drawback of this approach is the bilinear term $\lambda^\top C x$ of primal upper-level and dual lower-level variables. When considering only integer linking variables, as, e.g., in [25], linearizations can be applied yielding mixed-integer linear reformulations. Here, however, we study purely continuous bilevel problems. Thus, this bilinear term cannot be reformulated in a mixed-integer linear way as opposed to the KKT complementarity condition (3d).

Still, the strong duality inequality can be used to derive a valid inequality for Problem (3). A straightforward idea is to relax the nonconvex term $\lambda^\top Cx$ by replacing each term $C_i.x$ in (4) with an upper bound $C_i^+ \geq C_i.x$, where $C_i.$ denotes the ith row of C. This yields the inequality

$$\lambda^\top b - \lambda^\top C^+ - f^\top y \leq 0, \tag{6}$$

where C^+ denotes the vector of upper bounds C_i^+. The rationale behind this inequality is very simple and the inequality is obviously valid. Despite, or even because of its simplicity, this inequality can be very useful. It explicitly couples the primal lower-level variable y to the dual lower-level variable λ—a coupling that is missing in the root node problem of branch-and-bound approaches. The bounds C_i^+ can be obtained, e.g., from variable bounds on x. While this approach is cheap from a computational point of view, it may result in weak inequalities depending on the tightness of the bounds on x. Stronger bounds C_i^+ can be computed with the auxiliary LPs

$$C_i^+ := \max_{x,y,\lambda} \quad C_i.x \quad \text{s.t.} \quad (x,y,\lambda) \in \Omega \times \Omega_D,\ (x,y,\lambda) \in \mathcal{C}, \tag{7}$$

where $\mathcal{C}$ is a constraint set containing already added valid inequalities of type (6) and might be empty. This problem is bounded due to Assumption 1, such that finite bounds C_i^+ exist. In addition to the root node, Inequality (6) can also be added at any node u deeper in the branch-and-bound tree, where the bound C_i^+ is potentially tighter due to branching or previously added inequalities of type (6). This yields tighter inequalities that are locally valid for the subtree rooted at node u. Besides already added (locally) valid inequalities, the set $\mathcal{C}$ then also contains branching decisions, and $\mathcal{C}$ and C_i^+ in (7) both depend on the current branch-and-bound node u. For the ease of presentation, we omit an index u for $\mathcal{C}$ and C_i^+, because this dependence will always be clear from the context. We discuss implementation details such as the timing of the generation of valid inequalities (6) or the derivation of the bounds C_i^+ in Section 4, where we also demonstrate the effectiveness of the inequalities in a numerical study.

Before, let us emphasize that Inequality (6) can also be derived from another perspective. Consider a general bilinear term $z = vw$ with bounds $v^- \leq v \leq v^+$ and $w^- \leq w \leq w^+$. Then, McCormick envelopes [20] provide linear under- and overestimators for $z = vw$:

$$z \geq v^+ w + v w^+ - v^+ w^+, \qquad z \geq v^- w + v w^- - v^- w^-, \tag{8a}$$

$$z \leq v^- w + v w^+ - v^- w^+, \qquad z \leq v^+ w + v w^- - v^+ w^-. \tag{8b}$$

This can be applied to the strong-duality condition (4). We can decompose the bilinear products $\lambda^\top Cx = \sum_{i=1}^{\ell} z_i$ to obtain terms $z_i = v_i w_i$ with $v_i = \lambda_i$ and $w_i = C_i.x$. Due to the sign in the strong-duality condition (4), only the overestimators (8b) can be used:

$$\lambda^\top b - \sum_{i=1}^{\ell} z_i - f^\top y \leq 0, \tag{9a}$$

$$z_i \leq \lambda_i^- C_i.x + \lambda_i C_i^+ - \lambda_i^- C_i^+ \quad \text{for all } i = 1,\dots,\ell, \tag{9b}$$

$$z_i \leq \lambda_i^+ C_i.x + \lambda_i C_i^- - \lambda_i^+ C_i^- \quad \text{for all } i = 1,\dots,\ell. \tag{9c}$$

If we apply the initial bounds $\lambda_i^- = 0$ for all $i = 1,\dots,\ell$, then (9b) simplifies to

$$z_i \leq \lambda_i C_i^+. \tag{10}$$

Obviously, Inequality (6) is fulfilled if (9a) and (10) are satisfied. Contrary, when Inequality (6) is feasible, then $z_i = \lambda_i C_i^+$ is feasible for (9a) and (10) Thus, (9a)

together with (10) is equivalent to Inequality (6). However, whenever tighter (local) bounds $\lambda_i^- > 0$ are available, e.g., after presolve or branching, (9a) and (9b) provide a tightening of (6). The second overestimator (9c) involves bounds $C_i^- \leq C_i.x$, which can again be obtained by variable bounds on x or by minimizing instead of maximizing in Problem (7). However, it also involves upper bounds λ_i^+ for the initially unbounded dual variables λ_i. In general, such dual upper bounds are not available so that the overestimator (9c) cannot be used. Yet, whenever a (maybe locally valid) bound for λ_i is available by chance, e.g., due to a combination of branching and node presolve, the overestimator (9c) can be used to potentially tighten the valid inequality (6). In this light, the derivation via McCormick envelopes (8) may indeed provide tighter versions of Inequality (6). While the applicability of the tighter variants of the inequality solely depends on the availability of bounds, the basic inequality (6) can always be derived. We will discuss the applicability of the tightened variants in Section 4.

Furthermore, one could also relax $\lambda^\top Cx$ in the strong-duality inequality (4) by replacing each term $\lambda^\top C_{\cdot j}$ with an upper bound $C_j^+ \geq \lambda^\top C_{\cdot j}$, where $C_{\cdot j}$ denotes the jth column of C. We then obtain the inequality

$$\lambda^\top b - \sum_{i=j}^{n} C_j^+ x_j - f^\top y \leq 0. \tag{11}$$

This inequality couples all three types of variables x, y, and λ and can also be derived from the McCormick envelopes (8) by decomposing $\lambda^\top Cx = \sum_{j=1}^n z_j$ with $z_j = v_j w_j$, $v_j = \lambda^\top C_{\cdot j}$, and $w_j = x_j$. However, Inequality (11), respectively both overestimators (8b), involve finding lower or upper bounds $C_j^\pm$ for $\lambda^\top C_{\cdot j}$. This means that every problem

$$\min_{x,y,\lambda} \quad \lambda^\top C_{\cdot j} \quad \text{s.t.} \quad (x,y,\lambda) \in \Omega \times \Omega_D,\ (x,y,\lambda) \in \mathcal{C}, \tag{12}$$

needs to be bounded to obtain finite coefficients for each x_j. The lower-level problem (2) is bounded due to Assumption 1. Thus, the feasible set Ω_D of the dual lower-level problem (5) is bounded in the direction $b - Cx$ of the dual objective function. However, this is not necessarily the case for the optimization directions $C_{\cdot j}$. In fact, preliminary computational tests revealed that no instance in our test set has the property that all problems (12) are bounded. We thus refrain from using Inequality (11) and its variants that can be derived by McCormick envelopes. Finally, note that (6) and (11) are also valid for the pessimistic version of the bilevel problem,

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \quad c^\top x + d^\top y \quad \text{s.t.} \quad Ax + By \geq a,\ y \in \mathcal{S}(x),$$

since the lower-level problem is still given by (2). However, in order to streamline the presentation, we will stick to the discussion of the optimistic case.

4. Computational Study

We now evaluate the effectiveness of the valid inequalities derived in Section 3 within a complementarity-based branch-and-bound framework similar to what is described in Section 2. All our experiments are carried out on a single thread using the C interface of CPLEX 12.10 on a compute cluster with Xeon E3-1240 v6 CPUs at 3.7 GHz and 32 GB RAM; see [22] for more details.

Our complementarity-based branch-and-bound algorithm is realized in the following way. We introduce slack variables $s_i = b_i - C_i.x - D_i.y \geq 0$ to the single-level reformulation (3) for every lower-level constraint. We can then rewrite the complementarity constraints (3d) using special-ordered-sets of type 1 (SOS1) for each pair (s_i, λ_i). This way, we could use the SOS1 capabilities of CPLEX to branch

TABLE 1. Test set sizes.

test set	reference	total	solved	easy	remaining
CLIQUE	[11]	60	60	0	60
IMKP	[10]	144	70	17	53
INTER-ASSIG	[6]	24	24	4	20
INTER-CLIQUE	[11]	80	80	0	80
INTER-KP	[6]	99	78	38	40
KP	[11]	450	449	358	91
XU	[9, 24]	160	160	96	64

on the complementarity conditions. However, to have full control and information on the branching (in particular, on the set $\mathcal{C}$), we implemented our own branching and book-keeping using generic CPLEX callbacks. We branch on the most violated complementarity constraint $i \in \{1, \dots, \ell\}$ by setting either $s_i = 0$ or $\lambda_i = 0$, while leaving the node selection to CPLEX. This basic branch-and-bound procedure serves as a benchmark and is called B&B throughout this section. Interestingly, a preliminary computational study revealed that B&B already outperforms the native SOS1 branching of CPLEX.

We extend this setting to a branch-and-cut approach by subsequently adding the valid inequalities described in Section 3 via generic CPLEX callbacks. We therefore use the general formulation (9). This allows to add tighter inequalities whenever the required bounds are available. In a preliminary computational study, we tested various inequalities and strategies of how and when to add the inequalities. It turned out that computing the bounds $C_i^{\pm}$ and $\lambda_i^{\pm}$ with auxiliary LPs, similar to Problem (7), provides significantly better bounds and thus tighter inequalities than using internal global and local bounds provided "for free" by CPLEX. Although time-consuming, we follow the former approach to generate the tightest inequalities possible. Our preliminary experiments also revealed that making use of the McCormick overestimators (9b) and (9c) by tightening λ_i^- and C_i^- is only beneficial for a very small fraction of tested instances and in most cases it even harms the solution process. Hence, in the remainder of this section, we only discuss results for Inequality (6), implemented as the set of inequalities (9a) and (10). In particular, we compare the following parameterizations, where $\ell \in \mathbb{N}$ denotes the number of lower-level primal constraints:

B&B: The branch-and-bound benchmark without additional inequalities.
C&B: The set of inequalities (9a) and (10) is added at the root node if violated.
B&C(5): Inequality (9a) is added at the root and the inequalities (10) are added whenever (6) is violated at a node with depth $d = p\lfloor \ell/5 \rfloor$, $0 \leq p \in \mathbb{N}$.
B&C(10): Like B&C(5) but with $d = p\lfloor \ell/10 \rfloor$.

Obviously, the separation routine is invoked twice as many times in B&C(10) compared to B&C(5).

To compare our different methods, we use linear bilevel instances described in [17]. Table 1 summarizes the sizes of different test sets. The column "reference" indicates the origin in the literature of each subset and in the column "total" we state the size of the respective test sets. Further, the column "solved" shows how many instances are solved by at least one of the above methods in a time limit of 1 h, whereas "easy" indicates how many are solved in less then 10 s by all four methods. Finally, the last column displays the remaining number instances for each test set. Note that the test set XU consists of the test sets XUWANG and XULARGE, which are constructed the same way. Furthermore, based on our preliminary computational experiments, we

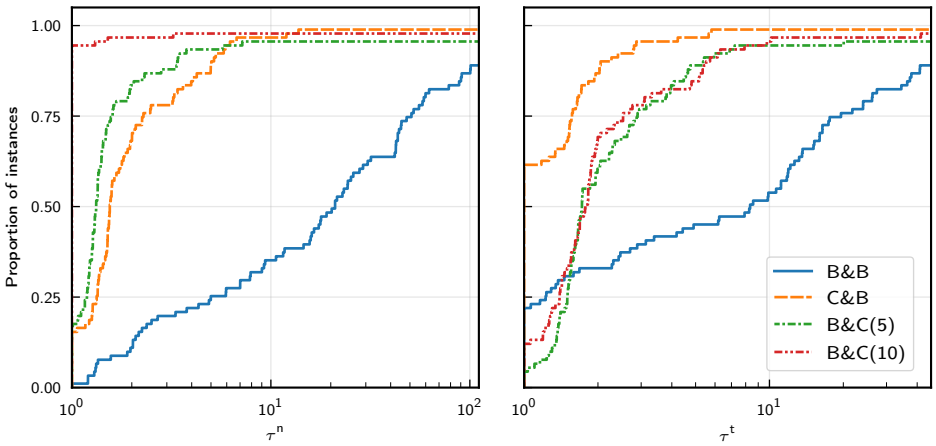

FIGURE 2. Log-scaled performance profiles for branch-and-bound nodes (left) and running times (right) for all remaining KP instances; see Table 1.

completely omit the test sets DENEGRE, GENERALIZED, as well as INT0SUM since they are too easy (i.e., all instances are labeled "easy") and GK, INTER-FIRE, as well as MIPLIB since they are too hard (i.e., hardly any instance is labeled "solved"). We thus obtain a total of 408 instances in Table 1. In the following, we discuss our observations w.r.t. the remaining instances in each of these different test sets. We illustrate the performance of the different parameterizations of our implementation using performance profiles according to [7]. For each instance i and implementation variant s, we compute the performance ratio

$$r^{\mathrm{n}}_{i,s} := \frac{n_{i,s}}{\min\{n_{i,s} \colon s \in S\}}$$

w.r.t. the branch-and-bound node count, where S is the set of all studied implementation variants. This means that $n_{i,s}$ is the node count of variant s on instance i. Every performance profile for node counts in this section shows the proportion of instances for which a given approach lies within a factor $\tau^{\mathrm{n}} \geq 1$ of the best approach. Similarly, we introduce τ^{t} for performance profiles w.r.t. the running times in wall-clock seconds.

It is well known that cuts often work only on a small number of instances and not throughout large and diverse test sets, in particular if they exploit a certain structure. Thus, we first discuss the impact of the valid inequalities for specific subsets of instances. It has already been shown in Figure 1 that the application of our valid inequalities is capable of closing the optimality gap much faster compared to a pure branch-and-bound. This effect is even more pronounced for all instances of the test set CLIQUE. These instances are solved immediately once the valid inequality is added at the root node. In contrast, B&B finds the optimal solution early in the tree in most of the cases but the lower bound does not improve at all. Thus, B&B cannot solve a single instance within the time limit of 1 h. For INTER-CLIQUE, we observe a similar behavior, except that a few instances can also be solved by B&B.

On the other hand, for the test set KP, it is beneficial to also separate inequalities further down in the branch-and-bound tree. Figure 2 shows performance profiles for branch-and-bound node counts (left) and total running times (right) for these instances. We first discuss the node counts and observe that C&B yields a notable improvement over B&B. However, C&B in turn is clearly dominated by B&C(10), which needs the least branch-and-bound nodes for almost every instance. On the

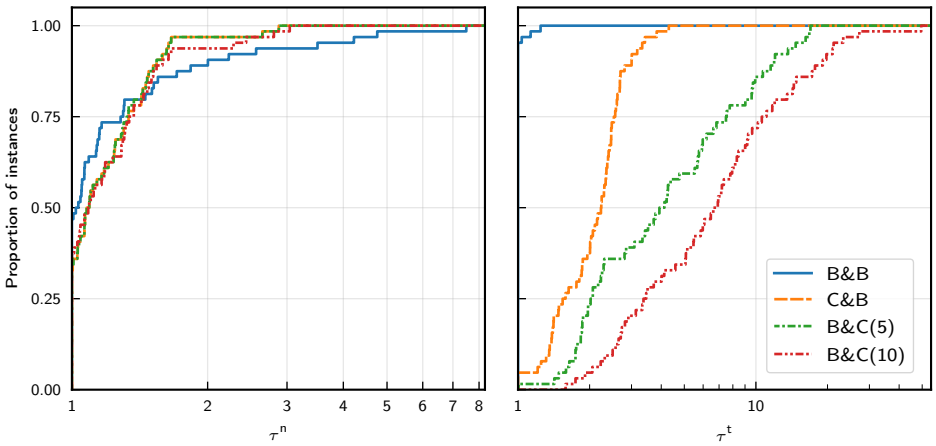

FIGURE 3. Log-scaled performance profiles for branch-and-bound nodes (left) and running times (right) over remaining XU instances.

other hand, this comes at a certain price since the node count improvement is not significant enough to compensate the time needed to separate the additional cuts; see also the right plot in Figure 2. Thus, C&B yields the best performance in terms of running times and dominates every other approach. The results on the test set INTER-ASSIG show similar trends w.r.t. nodes, but in contrast to KP, B&C(10) is also the best performing variant in terms of running times.

While similar trends can also be observed for the node counts for the test sets INTER-KP and IMKP, the decrease in nodes is insufficient to justify a branch-and-cut framework. In other words, B&B is dominated by every other approach in terms of node counts, but the resulting gain in running time is outweighed by cut separation, such that B&B slightly dominates the other variants in terms of running times.

Figure 3 displays performance profiles for nodes and running times restricted to the XU instances. Here, all variants perform pretty similar with respect to the node count. Since cut generation always costs computational time, it is not beneficial regarding running time to use the additional valid inequalities at all. This is especially notable for larger instances with many variables for which a large number of LPs (7) need to be solved to compute the coefficients of the cuts.

Overall, our methods are very useful on the considered instances. Figure 4 shows performance profiles for node counts and running times aggregated for all 408 instances. The branch-and-cut variants solve roughly 30 % more instances than the plain branch-and-bound procedure. All branch-and-cut variants largely outperform B&B, but there is no significant difference between the variants of the branch-and-cut method—neither in terms of node counts nor in terms of running times. To sum up, the C&B approach seems to be the best choice in general but the structure of specific instances might also lead to improved numerical results if the inequalities are added further down in the branch-and-bound tree.

5. Conclusion

In this paper, we derived a new valid primal-dual inequality for linear bilevel problems based on the strong-duality condition of the linear lower-level problem. We further discussed tightened variants of the inequality resulting from McCormick envelopes and tested these inequalities in a computational study. While the latter inequalities are not beneficial in practice, the former simple variant is shown to be crucial for proving optimality for the majority of all tested instances. In fact, for many

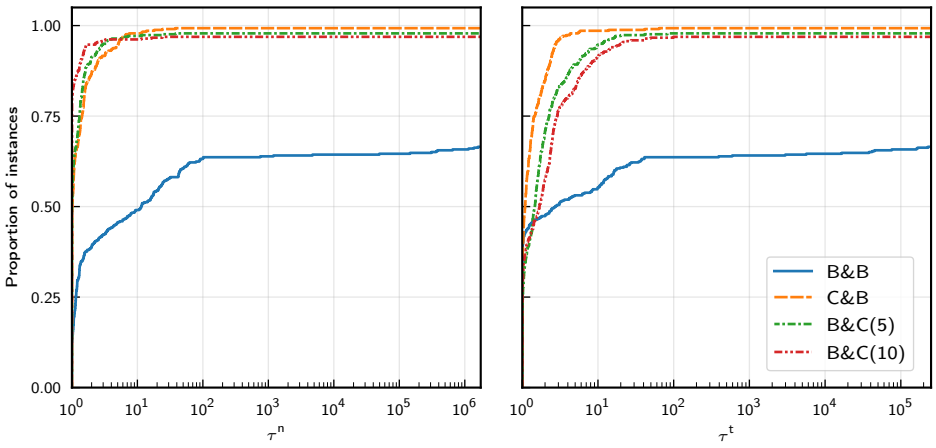

FIGURE 4. Log-scaled performance profiles for branch-and-bound nodes (left) and running times (right) over all remaining instances.

instances, adding a single inequality at the root node is sufficient to immediately close the optimality gap. For other instances, it is shown to be beneficial to add the inequality in a branch-and-cut approach further down in the branch-and-bound tree. Overall, adding the proposed valid inequalities helps to close the optimality gap much faster compared to a pure branch-and-bound algorithm and gives rise to a dedicated branch-and-cut implementation for linear bilevel problems.

While being out of scope of this short paper, we see several enhancements that could be applied within a sophisticated branch-and-cut implementation for linear bilevel problems. First, adding initial valid inequalities already before preprocessing could further improve node counts and running times. Second, in case that the inequality added in the root node does not immediately prove optimality, applying several rounds of adding valid inequalities and bound tightening could be useful. Third, whenever the separation of our inequalities yields bounds $\lambda_i^- > 0$, one could directly fix the corresponding primal lower-level constraint to be active. Finally, although our implemented branching rule already outperforms the SOS1-based branching of CPLEX, other branching and node selection rules may further improve the performance of the overall branch-and-cut implementation.

ACKNOWLEDGMENTS

This research has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. The authors thank the DFG for their support within project A05, B08, and Z01 in CRC TRR 154. Fränk Plein thanks the "Fonds de la Recherche Scientifique" (F.R.S.-FNRS) for financial support. Martine Labbé has been partially supported by the Fonds de la Recherche Scientifique - FNRS under Grant(s) no PDR T0098.18.

REFERENCES

[1] C. Audet, J. Haddad, and G. Savard. "Disjunctive Cuts for Continuous Linear Bilevel Programming." In: *Optimization Letters* 1.3 (2007), pp. 259–267. DOI: 10.1007/s11590-006-0024-3.

[2] C. Audet, G. Savard, and W. Zghal. "New Branch-and-Cut Algorithm for Bilevel Linear Programming." In: *Journal of Optimization Theory and Applications* 134.2 (2007), pp. 353–370. DOI: 10.1007/s10957-007-9263-4.

[3] J. F. Bard and J. T. Moore. "A Branch and Bound Algorithm for the Bilevel Programming Problem." In: *SIAM Journal on Scientific and Statistical Computing* 11.2 (1990), pp. 281–292. DOI: 10.1137/0911017.

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

[5] S. Dempe. *Foundations of Bilevel Programming*. Springer, 2002. DOI: 10.1007/b101970.

[6] S. DeNegre. "Interdiction and discrete bilevel linear programming." PhD thesis. Lehigh University, 2011. URL: https://preserve.lehigh.edu/etd/1226/.

[7] E. D. Dolan and J. J. Moré. "Benchmarking Optimization Software with Performance Profiles." In: *Mathematical Programming* 91.2 (2002), pp. 201–213. DOI: 10.1007/s101070100263.

[8] J. Egerer, V. Grimm, T. Kleinert, M. Schmidt, and G. Zöttl. *The Impact of Neighboring Markets on Renewable Locations, Transmission Expansion, and Generation Investment*. Tech. rep. Jan. 2020. URL: http://www.optimization-online.org/DB_FILE/2019/12/7508.pdf.

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

[10] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. "Interdiction Games and Monotonicity, with Application to Knapsack Problems." In: *INFORMS Journal on Computing* 31.2 (2019), pp. 390–410. DOI: 10.1287/ijoc.2018.0831.

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

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

[13] V. Grimm, L. Schewe, M. Schmidt, and G. Zöttl. "A Multilevel Model of the European Entry-Exit Gas Market." In: *Mathematical Methods of Operations Research* 89.2 (2019), pp. 223–255. DOI: 10.1007/s00186-018-0647-z.

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

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

[16] T. Kleinert, M. Labbé, F. Plein, and M. Schmidt. "There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization." In: *Operations Research* (2019). Forthcoming.

[17] T. Kleinert and M. Schmidt. "Computing Stationary Points of Bilevel Problems with a Penalty Alternating Direction Method." In: *INFORMS Journal on Computing* (2019). DOI: 10.1287/ijoc.2019.0945. Forthcoming.

[18] M. Labbé, P. Marcotte, and G. Savard. "A bilevel model of taxation and its application to optimal highway pricing." In: *Management Science* 44.12-part-1 (1998), pp. 1608–1622. DOI: 10.1287/mnsc.44.12.1608.

[19] M. Labbé and A. Violin. "Bilevel programming and price setting problems." In: *4OR* 11.1 (2013), pp. 1–30. DOI: 10.1007/s10288-012-0213-0.

[20] G. P. McCormick. "Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems." In: *Mathematical Programming* 10.1 (1976), pp. 147–175. DOI: 10.1007%2FBF01580665.

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

[22] Regionales Rechenzentrum Erlangen. *Woodcrest Cluster*. URL: https://www.anleitungen.rrze.fau.de/hpc/woody-cluster/ (visited on 06/03/2020).

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

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

[25] M. H. Zare, J. S. Borrero, B. Zeng, and O. A. Prokopyev. "A Note on Linearized Reformulations for a Class of Bilevel Linear Integer Problems." In: *Annals of Operations Research* 272.1 (2019), pp. 99–117. DOI: 10.1007/s10479-017-2694-x.

(T. Kleinert) [1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany; [2]Energie Campus Nürnberg, Fürther Str. 250, 90429 Nürnberg, Germany

Email address: thomas.kleinert@fau.de

(M. Labbé, F. Plein) [3]Université Libre de Bruxelles, Department of Computer Science, Boulevard du Triomphe, CP212, 1050 Brussels, Belgium; [4]Inria Lille - Nord Europe, Parc scientifique de la Haute Borne, 40, av. Halley - Bât A - Park Plaza, 59650 Villeneuve d'Ascq, France

Email address: {martine.labbe,frank.plein}@ulb.ac.be

(M. Schmidt) [5]Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

Email address: martin.schmidt@uni-trier.de