INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

Improving Directions in Mixed Integer Bilevel Linear Optimization

Federico Battista and Ted K. Ralphs

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

COR@L Technical Report 25T-018-R1

LEHIGH UNIVERSITY

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Improving Directions in Mixed Integer Bilevel Linear Optimization

Federico Battista *[1] and Ted K. Ralphs †[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Original Publication: October 30, 2025
Last Revised: December 30, 2025

Abstract

We consider the central role of improving directions in solution methods for mixed integer bilevel linear optimization problems (MIBLPs). Current state-of-the-art methods for solving MIBLPs employ the branch-and-cut framework originally developed for solving mixed integer linear optimization problems. This approach relies on oracles for two kinds of subproblems: those for checking whether a candidate pair of leader's and follower's decisions is bilevel feasible, and those required for generating valid inequalities. Typically, these two types of oracles are managed separately, but in this work, we explore their close connection and propose a solution framework based on solving a single type of subproblem: determining whether there exists a so-called *improving feasible direction* for the follower's problem. Solution of this subproblem yields information that can be used both to check feasibility *and* to generate strong valid inequalities. Building on prior works, we expose the foundational role of improving directions in enforcing the follower's optimality condition and extend a previously known hierarchy of optimality-based relaxations to the mixed-integer setting, showing that the associated relaxed feasible regions coincide exactly with the closure associated with intersection cuts derived from improving directions. Numerical results with an implementation using a modified version of the open source solver MibS show that this approach can yield practical improvements.

1 Introduction

Bilevel optimization problems are a class arising from the recasting of game-theoretic equilibrium problems, such as that of finding a subgame perfect Nash equilibrium in the classic Stackelberg game, as mathematical optimization problems. Stackelberg games are two-player sequential games in which the players, called the *leader* and the *follower*, make one move each, with each move consisting of deciding the values of a set of associated decision variables. The leader chooses values for their decision variables first and then the follower selects an optimal reaction by solving an optimization problem called the *follower's (reaction) problem*, with the leader's solution as an input

*feb223@lehigh.edu
†ted@lehigh.edu

parameter. In this paper, we focus on the well-studied special case of a general bilevel optimization problem known as a *mixed integer bilevel linear optimization problem* (MIBLP) in which the decision variables of both the leader and the follower are constrained by linear inequalities and the values of some variables are required to be integer.

The goal of solving a bilevel optimization problem is to determine the optimal decision of the leader. Doing so, however, requires understanding the functional dependence of the follower's reaction on the leader's decision. From the leader's standpoint, the casting of the problem as a mathematical optimization problem can be viewed as introducing constraints involving the so-called *value function* of the follower's problem into a standard mathematical optimization problem. The value function encodes optimality conditions for the follower's problem that are parameterized on the leader's decision.

The introduction of these additional nonconvex constraints is the main reason for both the theoretical and practical difficulty of solving problems in this class. In fact, even the simplest case, in which all constraints are linear and all variables are continuous, known as the *bilevel linear optimization problem* (BLP), is strongly NP-hard [Hansen et al. (1992); Buchheim (2023)]. Such bilevel optimization problems are usually tackled by reformulating them as single-level mathematical optimization problems by incorporating optimality conditions for the follower's problem derived from, e.g., the KKT conditions, as constraints; see, e.g., [Dempe and Zemkoho (2020); Zare et al. (2019)].

MIBLPs are more difficult to solve than BLPs, both in theory and in practice. From a theoretical computational complexity standpoint, a decision version of MIBLP is complete for the class Σ_2^P [Stockmeyer (1976); Jeroslow (1985)], the second level of the so-called polynomial hierarchy. Roughly speaking, this means that even if we had a constant-time oracle for solving NP-complete subproblems (e.g., MILPs), solving MIBLPs would still remain as difficult as solving MILPs.

Despite the intractability indicated by the problem's worst-case complexity, the fact that we can reliably solve MILPs of small- to medium-scale in practice means that practical algorithms based on oracle computations should be possible. Indeed, the most successful approach for solving MIBLPs to date is the branch-and-cut algorithm, which takes a solution approach similar to that for solving MILPs but relies crucially on the solution of MILP subproblems both for checking feasibility and for generating the valid inequalities needed to strengthen the weak initial relaxation.

To date, state-of-the-art branch-and-cut solvers [Fischetti, Ljubić, et al. (2017); Tahernejad, Ralphs, and DeNegre (2020)] have treated the MILP subproblems arising when generating cuts as separate from those arising when checking feasibility of a solution, despite their theoretical equivalence. We argue that this separation overlooks an opportunity for significant algorithmic gains. By adopting a more integrated, "gray-box" perspective—where these oracles are not isolated but instead allowed to share internal information—we can leverage the strong overlap between separation and feasibility checking. We show that such a shift reduces redundant oracle computations and paves the way for more effective solution methods.

1.1 Definitions and Notation

Before outlining the contribution of this paper, we lay out the formal definitions and notation, as well as briefly review known results used in the remainder of the paper. In the literature, a number of equivalent ways of formulating MIBLPs have been presented. Here, we use the so-called value function formulation:

$$\min \left\{ cx + d^1 y \;\middle|\; x \in X, y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y,\ d^2 y \le \phi(b^2 - A^2 x) \right\}, \tag{MIBLP}$$

where

- $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$ and $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$, respectively, reflect the integrality requirements on the values of the leader's and follower's variables;

- $\mathcal{P}_1(x) = \left\{ y \in \mathbb{R}_+^{n_2} \;\middle|\; G^1 y \ge b^1 - A^1 x \right\}$ is the set of values of the followers variables satisfying the *leader's constraints*;

- $\mathcal{P}_2(x) = \left\{ y \in \mathbb{R}_+^{n_2} \;\middle|\; G^2 y \ge b^2 - A^2 x \right\}$ is the set of values of the follower's variables satisfying the *follower's constraints*,

while the input data are $c \in \mathbb{Q}^{n_1}$; $d^1, d^2 \in \mathbb{Q}^{n_2}$; $A^1 \in \mathbb{Q}^{m_1 \times n_1}$; $G^1 \in \mathbb{Q}^{m_1 \times n_2}$; $b^1 \in \mathbb{Q}^{m_1}$; $A^2 \in \mathbb{Q}^{m_2 \times n_1}$; $G^2 \in \mathbb{Q}^{m_2 \times n_2}$; and $b^2 \in \mathbb{Q}^{m_2}$. As discussed further below, this formulation assumes the follower behaves in an *optimistic* fashion (see Dempe [2002] for discussion of other formulations).

The function ϕ is the aforementioned *value function* of the follower's problem, defined as

$$\phi(\beta) = \min \left\{ d^2 y \;\middle|\; G^2 y \ge \beta, y \in Y \right\} \quad \forall \beta \in \mathbb{R}^{m_2}. \tag{VF}$$

The role of the value function ϕ is to encode the optimality conditions for the follower's problem. Specifically, for $\hat{x} \in X, \hat{y} \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y$, if

$$d^2 \hat{y} \le \phi(b^2 - A^2 \hat{x}), \tag{OPT}$$

then $\hat{y}$ is contained in the *rational reaction set* w.r.t. to $\hat{x}$, defined formally as

$$\mathcal{R}(\hat{x}) = \left\{ y \in \mathcal{S}(\hat{x}) \;\middle|\; d^2 y \le d^2 \bar{y},\ \forall \bar{y} \in \mathcal{S}(\hat{x}) \right\}, \tag{RS}$$

where

$$\mathcal{S}(\hat{x}) = \{ y \in Y \mid \mathcal{P}_1(\hat{x}) \cap \mathcal{P}_2(\hat{x}) \}$$

is the set of feasible solutions for the follower's problem, given the leader's decision. In this case, we say that $(\hat{x}, \hat{y})$ is *bilevel feasible*. The set of all bilevel feasible solutions is the *bilevel feasible region*, denoted by

$$\mathcal{F} = \{ (x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid x \in X, y \in \mathcal{R}(x) \}.$$

When $|\mathcal{R}(\hat{x})| > 1$, the lower-level problem has multiple optimal solutions and one can make different assumptions about the procedure for selecting among alternative optima. The formulation (MIBLP) that we employ here specifies the aforementioned *optimistic* assumption in which the follower selects the response that is most favorable for the leader's objective function.

The standard relaxations of an MIBLP are either the linear optimization problem (LP) relaxation with the polyhedral feasible region

$$\mathcal{P} = \left\{ (x, y) \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{n_2} \;\middle|\; y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \right\},$$

or the MILP relaxation with feasible region

$$\mathcal{S} = \mathcal{P} \cap (X \times Y).$$

Throughout the paper, we make the following standard assumptions.

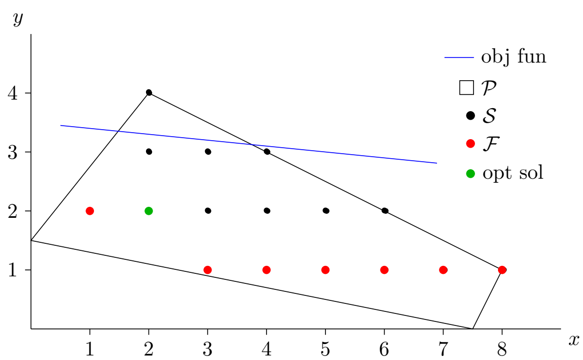
Assumption 1. *$\mathcal{P}$ is bounded.*

Assumption 2. *All first-level variables with at least one non-zero coefficient in the second-level problem (so-called linking variables) are integer.*

The first assumption guarantees the boundedness of (MIBLP), but is made primarily for the sake of presentation and can be relaxed. The second assumption ensures that the optimal solution value of (MIBLP) is attainable whenever the optimal solution value is finite [Vicente et al. (1996)]. Finally, because the validity of some inequalities we discuss relies on integrality of the input parameters, we make the following simplifying assumptions in the remainder of the paper.

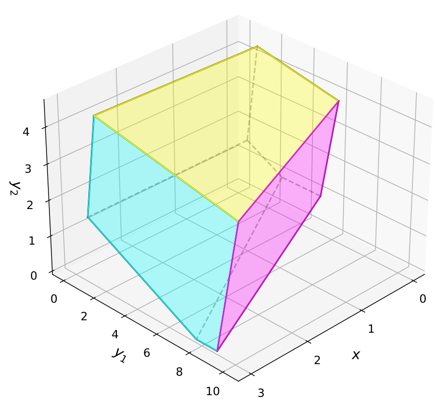
Assumption 3. *$A^2x + G^2y - b^2 \in \mathbb{Z}^{m_2}$ for all $(x, y) \in \mathcal{S}$ and $d^2 \in \mathbb{Z}^{n_2}$.*

Example 1. *For the reminder of this paper, we make use of the well-known bilevel problem from Moore and Bard [1990] as a running example. Figure 1 shows the bilevel feasible region and optimal solution, along with the standard relaxations $\mathcal{P}$ and $\mathcal{S}$. While a two-dimensional example is illustrative, it cannot capture some of the complexities we address in this work. Hence, we also present a three-dimensional bilevel problem, as shown in Figure 2.*



$$
\begin{aligned}
\min_{x \in \mathbb{Z}_+} \quad & -x - 10y \\
\text{s.t.} \quad & y \in \operatorname{argmin}\{y : \\
& \qquad -5x + 4y \leq 6 \\
& \qquad x + 2y \leq 10 \\
& \qquad 2x - y \leq 15 \\
& \qquad 2x + 10y \geq 15 \\
& \qquad y \in \mathbb{Z}_+ \}
\end{aligned}
$$

Figure 1: The feasible region, the LP and MILP relaxation, and the optimal solution of the example from Moore and Bard [1990].



$$
\begin{aligned}
\min_{x \in \{0,1,2,3\}} \quad & -x - 2y_1 - 5y_2 \\
\text{s.t.} \quad & y \in \operatorname{argmin}\{y_2 : \\
& \qquad x + 3y_2 \geq 3 \\
& \qquad -16x + 12y_1 - 4y_2 \leq 59 \\
& \qquad -x + 9y_1 - 2y_2 \geq 2 \\
& \qquad -2x + 6y_1 + 23y_2 \geq 40 \\
& \qquad -x + y_1 + 10y_2 \leq 45 \\
& \qquad y \in \mathbb{Z}^2_+ \}
\end{aligned}
$$

Figure 2: A three-dimensional MIBLP and the feasible region of its LP relaxation.

1.2 Improving Directions

We now introduce the concept of improving directions.

Definition 1. *A vector $w \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}$ is an* improving direction *(ID) if $d^2 w < 0$ and we denote the set of all IDs as $\mathcal{W} := \{w \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2} \mid d^2 w < 0\}$. With respect to a given $(\hat{x}, \hat{y}) \in \mathcal{P}$, an ID w is an* improving feasible direction *(IFD) if $\hat{y} + w \in \mathcal{P}_2(\hat{x})$. The set of all improving feasible directions with respect to $(\hat{x}, \hat{y})$ is*

$$\mathcal{W}(\hat{x}, \hat{y}) = \{w \in \mathcal{W} \mid \hat{y} + w \in \mathcal{P}_2(\hat{x})\}. \tag{1}$$

Informally, improving feasible directions with respect to a bilevel infeasible solution are those that point towards the bilevel feasible region $\mathcal{F}$. Note that although this definition requires that improving directions must themselves satisfy the integrality requirements of the follower's problem, an improving direction can nevertheless be said to be feasible with respect to any point in $\mathcal{P}$, not only points in $\mathcal{S}$.

The most obvious use of improving feasible directions is as a certificate of bilevel infeasibility for points in $\mathcal{S}$. We present the following fundamental result that shows the relationship between the existence of an improving direction and bilevel feasibility of a given point.

Proposition 1. *Let $(\hat{x}, \hat{y}) \in \mathcal{S}$. Then we have $(\hat{x}, \hat{y}) \in \mathcal{F} \iff \mathcal{W}(\hat{x}, \hat{y}) = \emptyset$.*

Proof. ($\Rightarrow$) Assume $(\hat{x}, \hat{y}) \in \mathcal{F}$ and $\mathcal{W}(\hat{x}, \hat{y}) \neq \emptyset$ for sake of contradiction. Then $d^2 \hat{y} = \phi(b^2 - A^2 \hat{x})$. Now let $w \in \mathcal{W}(\hat{x}, \hat{y})$, then by definition $\hat{y} + w \in \mathcal{P}_2(\hat{x})$. Moreover, we have

$$d^2(\hat{y} + w) < d^2 \hat{y} = \phi(b^2 - A^2 \hat{x}).$$

This implies that either $d^2 \hat{y} > \phi(b^2 - A^2 \hat{x})$ or $\hat{y} \notin \mathcal{P}_2(\hat{x})$, then $\hat{y} \notin \mathcal{R}(\hat{x})$. This contradicts the bilevel feasibility of $(\hat{x}, \hat{y})$.

($\Leftarrow$) We prove the contrapositive. Let $(\hat{x}, \hat{y}) \notin \mathcal{F}$ be given. Then $\exists\ \bar{y} \in \mathcal{S}(\hat{x})$ such that $d^2 \bar{y} < d^2 \hat{y}$. Now consider $w := \bar{y} - \hat{y}$. Note that $\hat{y} + w \in \mathcal{P}_2(\hat{x})$ and $d^2 w < 0$ by construction. Then $w \in \mathcal{W}(\hat{x}, \hat{y})$ and the statement is proven. □

This result leads to an alternative method to check bilevel feasibility, which is to check whether $\mathcal{W}(\hat{x}, \hat{y}) = \emptyset$. By Proposition 1, when $(\hat{x}, \hat{y}) \in \mathcal{S}$, emptiness of $\mathcal{W}(\hat{x}, \hat{y})$ is equivalent to bilevel feasibility. On the other hand, $\mathcal{W}(\hat{x}, \hat{y})$ may be empty if $(\hat{x}, \hat{y}) \in \mathcal{P} \setminus \mathcal{S}$, even if $(\hat{x}, \hat{y}) \notin \text{conv}(\mathcal{F})$ (Proposition 1 does not apply). From a computational perspective, this is quite important. As a side note, Proposition 1 also indicates that checking feasibility of a given solution is a problem in co-NP, which is interesting, though not unexpected.

Given a candidate pair $(\hat{x}, \hat{y}) \in \mathcal{S}$ and a direction $w \in \mathcal{W}(\hat{x}, \hat{y})$, the follower's solution can be augmented to form an *improving solution*, i.e., a new candidate pair $(\hat{x}, \hat{y} + w)$ that remains in $\mathcal{S}$ but improves the follower's objective, i.e., $d^2(\hat{y} + w) < d^2 \hat{y}$. In fact, when $(\hat{x}, \hat{y})$ satisfies integrality requirements, checking whether $\mathcal{W}(\hat{x}, \hat{y})$ is empty is formally equivalent to determining the existence of an improving solution to the follower's problem, as both can be formulated as MILPs. However, as discussed in Section 4.2, elements of $\mathcal{W}(\hat{x}, \hat{y})$ can be generated using a variety of objective functions, enabling the computation of directions with different desirable properties. This makes it a seemingly more flexible approach to certifying bilevel infeasibility.

In the context of traditional, single-level integer linear optimization, the problem of determining whether there exists a so-called *augmenting vector* has been previously studied and it was shown to be oracle-polynomial-time equivalent to solving an MILP by Schulz [2009], who called it the *augmentation problem*. Similarly, bilevel feasibility of a given candidate solution in the MIBLP context can be checked by determining whether an IFD exists. It follows that an algorithm for solving MIBLPs based only on generating IFDs is possible.

Example 2. *In Figure 3 we show three bilevel infeasible points and possible improving feasible directions moving them toward points in $\mathcal{F}$ of the Moore and Bard [1990] example. Note that for points in $\mathcal{S} \setminus \mathcal{F}$, there must exist at least one IFD (see, e.g., $\hat{y}^1$ and $\hat{y}^2$). However, IFDs may also exist for points in $\mathcal{P}$ (e.g., $\hat{y}^3$). Moreover, the same improving direction (e.g., w_1) may be feasible for more than one point. It is easy to verify that given any of these points (say, $\hat{y}^1$), we can move along any of the IFDs (say, w_1), to obtain an improving feasible solution.*

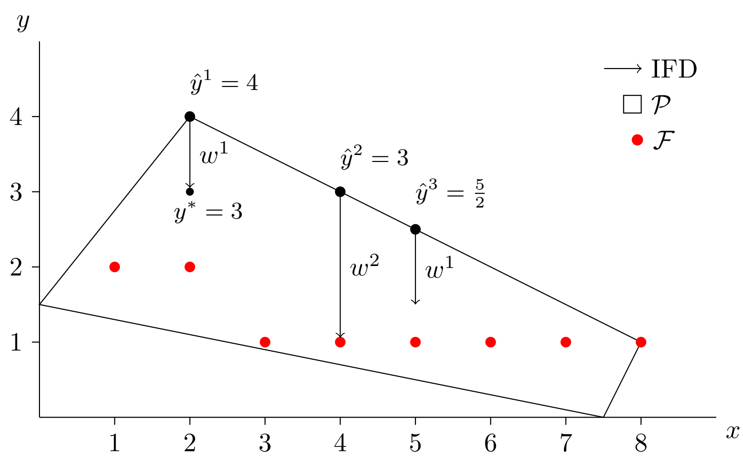


Figure 3: The improving feasible directions moving points in $\mathcal{S}$ (or $\mathcal{P}$) towards points in $\mathcal{F}$.

1.3 Valid Inequalities

We briefly review the basic concepts of valid inequalities.

Definition 2 (Valid Inequality)**.** *A* valid inequality *for $\mathcal{F}$ is a triple $(\alpha^x, \alpha^y, \beta) \in \mathbb{Q}^{n_1} \times \mathbb{Q}^{n_2} \times \mathbb{Q}$ such that*

$$\mathcal{F} \subseteq \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid \alpha^x x + \alpha^y y \geq \beta\}.$$

The goal of generating valid inequalities in a branch-and-cut algorithm can be expressed in several related ways. Most obviously, the goal is to strengthen the relaxation, leading to an improved dual bound. This is accomplished by removing solutions to the relaxation that are infeasible to the original problem through the addition of valid inequalities violated by those solutions. This shrinks the feasible region of the relaxation, resulting in a better approximation of conv($\mathcal{F}$).

Another way of viewing the same goal is as that of implicitly enforcing constraints that were relaxed, in this case the optimality conditions. Although the constraints that were relaxed are nonlinear, their effect can be replicated using linear inequalities because the objective function is linear, which allows us to convexify the feasible region. Thus, while the goal in the MILP setting is to expose points satisfying integrality conditions as extreme points of the relaxed feasible region, here we need to additionally ensure that the exposed solutions satisfy second-level optimality conditions. In other words, the exposed integer points must also lie on the boundary of the epigraph of the value function (roughly speaking). The fact that the exposed points must simultaneously satisfy these two different properties results in complex interactions when generating valid inequalities that require careful algorithmic control. More details of the theory underlying the generation of valid inequalities in the MIBLP case, as well as detailed discussions of control mechanisms are provided in Tahernejad and Ralphs [2025].

The general recipe by which most of the known classes of valid inequalities for MIBLPs are constructed is to first identify a set $\mathcal{C}$ containing no (improving) bilevel feasible solutions in its interior and then generate an inequality valid for $\operatorname{conv}(\overline{\operatorname{int}(\mathcal{C})} \cap \mathcal{P})$. Such an inequality is valid for $\mathcal{F}$, since $\mathcal{F} \subseteq \overline{\operatorname{int}(\mathcal{C})} \cap \mathcal{P}$. Because this recipe is strictly a generalization of the one first proposed by Balas [1972] in the context of MILPs, these inequalities are sometimes referred to broadly as *intersection cuts* (ICs). By employing different "solution free" sets and by replacing $\mathcal{F}$ with the feasible region of a relaxation, we can derive a wide range of different classes of valid inequality.

The most common way in which ICs are generated in practice is the method proposed in the original paper of Balas [1972]. That is, we replace $\mathcal{P}$ with a simplicial radial cone that contains $\mathcal{F}$ and whose single extreme point lies in the interior of a convex set $\mathcal{C}$ containing no feasible points in its interior, as described above. Then the hyperplane defined by the points of intersection of the rays of the simplicial cone with the set $\mathcal{C}$ is a hyperplane that separates the cone's extreme point from $\mathcal{F}$. When we reference the term *intersection cut* in the remainder of the paper, we are referring to this specific type of IC, defined as follows.

Definition 3 (Bilevel Free Set)**.** *A* bilevel free set *(BFS) is a set* $\mathcal{C} \subseteq \mathbb{R}^{n_1+n_2}$ *such that* $\operatorname{int}(\mathcal{C}) \cap \mathcal{F} = \emptyset$.

Definition 4 (Intersection Cut)**.** *Let* $\mathcal{C} \subseteq \mathbb{R}^{n_1+n_2}$ *be a convex BFS with a given point* $(\hat{x}, \hat{y}) \in \mathbb{R}^{n_1+n_2}$ *in its interior. Let* $\mathcal{V}(\hat{x}, \hat{y})$ *be a simplicial radial cone containing* $\mathcal{F}$ *with vertex* $(\hat{x}, \hat{y})$*. Then, if the triple* $(\alpha^x, \alpha^y, \beta) \in \mathbb{Q}^{n_1+n_2+1}$ *is such that the set* $\{(x, y) \in \mathbb{R}^{n_1+n_2} \mid \alpha^x x + \alpha^y y = \beta\}$ *is the unique hyperplane containing the points of intersection of* $\mathcal{C}$ *with the extreme rays of* $\mathcal{V}(\hat{x}, \hat{y})$*, we have* $(\alpha^x, \alpha^y, \beta)$ *is an inequality valid for* $\mathcal{F}$ *and violated by* $(\hat{x}, \hat{y})$*. Such inequality is called an* intersection cut*.*

The simplicial cone is often taken to be that described by a linearly independent set of inequalities binding at some basic feasible solution to the initial LP relaxation, but we consider cones derived in other ways later in the paper. In Section 3.2, we define a notion of rank similar to the standard notion from the theory of valid inequalities for MILPs. Beginning with inequalities derived from (bilevel infeasible) extreme points of $\mathcal{P}$, which are the inequalities of rank 1, one can then iteratively apply this procedure to derive inequalities of higher rank.

In Fischetti, Ljubić, et al. [2017] and Fischetti, Ljubić, et al. [2018], a number of classes of ICs were introduced, but two in particular play a central role in what follows—those arising from the existence of *improving solutions* and *improving directions*, respectively. In [Tahernejad and Ralphs

(2025)], the authors refer to these two classes as *improving solution intersection cuts* (ISICs) and *improving direction intersection cuts* (IDICs).

As observed by Fischetti, Ljubić, et al. [2017], the strength of an IC is directly related to the "size" of the set $\mathcal{C}$: a "larger" such set should result in a stronger inequality (see discussion in Fischetti, Ljubić, et al. [2017]; Fischetti, Ljubić, et al. [2018]). Because of the specific forms of convex sets utilized for the two classes of ICs, an improving direction/solution that will result in a large bilevel free set for one class will not necessarily result in a large bilevel free set for the other class.

1.4 Contribution

Improving directions arise naturally in bilevel optimization and the use of an oracle for finding such directions in an algorithm for solving bilevel optimization problems is not new. As observed in Tahernejad and Ralphs [2025], ICs generated from improving directions are the strongest inequalities known from an empirical standpoint. Beyond separation, such oracles have also been used for branching [Wang and Xu (2017)] and for strengthening relaxations [Xueyu et al. (2022)].

The contributions of this paper are two-fold. From the theoretical standpoint, we highlight the fundamental role of improving directions in restoring the follower's optimality condition. To this end, we generalize the hierarchy of optimality-based relaxations presented in Xueyu et al. [2022] for bilevel linear problems with only binary variables to the mixed-integer context and show that the convex hulls of the feasible regions defined by this hierarchy can be exactly characterized using inequalities generated from improving directions.

From a computational standpoint, our contribution is to show that unifying the oracle computations for the separation of bilevel infeasible points with those for checking bilevel feasibility and leveraging the equivalence of the underlying MILP subproblems can lead to significant improvements in the practical performance of the branch-and-cut algorithm. Given that the problem of finding an improving direction serves both to generate strong valid inequalities and check bilevel feasibility, it naturally arises as a suitable candidate for the purpose. Motivated by this, we propose and implement a branch-and-cut algorithm that follows the basic outline described in Tahernejad, Ralphs, and DeNegre [2020] but avoids checking feasibility of solutions by solving the follower's problem and instead solves the problem of determining whether there exists an improving feasible direction. We argue that solving the leader's problem to provable optimality without ever explicitly evaluating the follower's value function is not just possible but has the potential to improve empirical performance.

1.5 Outline

The remainder of this paper is organized as follows: Section 2 introduces a hierarchy of optimality-based relaxations that arise from improving directions. Section 3 discusses the generation of inequalities valid for bilevel feasible points derived from improving directions. Section 4 introduces the algorithm and presents methods to generate these improving feasible directions. Finally, Section 5 presents the computational results.

2 The k-opt Hierarchy

Both of the standard relaxations of MIBLPs discard the optimality conditions for the follower's problem completely, which results in weak relaxations in general. In zero-sum problems, for example, the follower's problem is implicitly being solved with an objective function that is precisely the opposite of the follower's true objective.

Rather than completely discard the optimality condition, it is possible to impose a weaker condition that improves the bound yielded by the linear relaxation, yet whose computation remains tractable. In general, this can be achieved by replacing the reaction set $\mathcal{R}(x)$ with a suitable superset. A straightforward way to define such a set is to replace the follower's problem with a relaxation. In Xie et al. [2025], for example, the optimality condition is relaxed by allowing lower-level solutions within a specified optimality gap. Here, we follow the ideas of Xueyu et al. [2022] and instead require the follower's reactions to satisfy local rather than global optimality conditions. In [Xueyu et al. (2022)], this was done for MIBLPs with only binary variables in the follower's problem. We extend the method by generalizing the definition of *k-neighborhood*. For a given $y \in Y$ and $k \in \mathbb{Z}_+$, the k-neighborhood of y is

$$\mathcal{N}_k(y) = \{\bar{y} \in Y \mid \|\bar{y} - y\|_1 \leq k\}.$$

For $y \in Y$, the elements in the k-neighborhood are the points in Y that can be reached by following any direction $w \in Y$ with $\|w\|_1 \leq k$. Therefore, we can modify the reaction set (RS) to one requiring only local optimality by replacing $\mathcal{R}(x)$ with

$$\mathcal{R}(x;k) = \left\{y \in \mathcal{S}(x) \;\middle|\; d^2 y \leq d^2 \bar{y},\ \forall \bar{y} \in \mathcal{N}_k(y) \cap \mathcal{S}(x)\right\}. \tag{k-RS}$$

for some $k \in \mathbb{Z}_+$. Note that for $k = 0$ we have that $\mathcal{N}_0(y) = \{y\}$ and $\mathcal{R}(x;0) = \mathcal{S}(x)$, whereas for

$$\bar{k} := \sum_{i=1}^{r_2} \left(\left\lfloor \max_{(x,y)\in\mathcal{P}} y_i \right\rfloor - \left\lceil \min_{(x,y)\in\mathcal{P}} y_i \right\rceil \right) + \sum_{i=r_2+1}^{n_2} \left\lceil \max_{(x,y)\in\mathcal{P}} y_i - \min_{(x,y)\in\mathcal{P}} y_i \right\rceil,$$

which is finite by Assumption 1, we have $\mathcal{N}_{\bar{k}}(y) = Y$ and $\mathcal{R}(x;\bar{k}) = \mathcal{R}(x)$, for all $x \in X$.

In Xueyu et al. [2022], (k-RS) is referred to as the *k-optimal reaction set* and any $y \in \mathcal{R}(x;k)$ as a *k-optimal reaction*. Moreover, the authors show that for $Y = \{0,1\}^{n_2}$ and for any fixed $k \in \mathbb{Z}_+$, the set of points in (k-RS) is MILP-representable with a description of polynomial size.

The MIBLP with k-optimal follower is then formally defined as follows:

$$\begin{aligned} \min\ & cx + d^1 y \\ \text{s.t. } & (x,y) \in \mathcal{S} \\ & y \in \mathcal{R}(x;k), \end{aligned} \tag{BP$_k$}$$

for all $k \in \mathbb{Z}_+$. The set of feasible points of (BP$_k$) is

$$\mathcal{F}(k) = \{(x,y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid x \in X, y \in \mathcal{R}(x;k)\}. \tag{2}$$

Following an approach similar to Xueyu et al. [2022], we can characterize membership in $\mathcal{R}(x;k)$ for $x \in X$ by a set of necessary and sufficient conditions. For this purpose, we denote the set of all improving directions with a 1-norm no bigger than k as

$$\mathcal{W}^k = \{w \in \mathcal{W} \mid \|w\|_1 \leq k\},$$

and those that are feasible with respect to a given $(\hat{x}, \hat{y})$ as

$$\mathcal{W}(\hat{x}, \hat{y}; k) = \left\{ w \in \mathcal{W}^k \mid \hat{y} + w \in \mathcal{P}_2(\hat{x}),\ \|w\|_1 \leq k \right\}.$$

Then we have the following result.

Proposition 2. *Let* $(\hat{x}, \hat{y}) \in \mathcal{S}$. *Then* $(\hat{x}, \hat{y}) \in \mathcal{F}(k) \Longleftrightarrow \mathcal{W}(\hat{x}, \hat{y}; k) = \emptyset$.

Proof. Let $k \in [\bar{k}]$. Then the proof is divided into two parts.
($\Rightarrow$) We prove the contrapositive. Let $\mathcal{W}(\hat{x}, \hat{y}; k) \neq \emptyset$ be given. Then there exists $w \in \mathcal{W}(\hat{x}, \hat{y}; k)$ such that $\bar{y} := \hat{y} + w \in \mathcal{N}_k(\hat{y}) \cap \mathcal{S}(\hat{x})$ with $d^2\bar{y} < d^2\hat{y} \Rightarrow \hat{y} \notin \mathcal{R}(\hat{x}; k) \Rightarrow (\hat{x}, \hat{y}) \notin \mathcal{F}(k)$.

($\Leftarrow$) Again, we prove the contrapositive. Let $(\hat{x}, \hat{y}) \notin \mathcal{F}(k)$ be given. Then $\hat{y} \notin \mathcal{R}(\hat{x}; k) \Rightarrow \exists\ \bar{y} \in \mathcal{N}_k(\hat{y}) \cap \mathcal{S}(\hat{x})$ with $d^2\bar{y} < d^2\hat{y}$. Now consider $w := \bar{y} - \hat{y}$. Note that $\|w\|_1 \leq k$, $w \in \mathcal{P}_2(\hat{x})$ and $d^2 w < 0$ by construction. Then $w \in \mathcal{W}(\hat{x}, \hat{y}; k)$ and this proves the statement. □

In other words, just as we can check the feasibility of $(\hat{x}, \hat{y})$ with respect to the constraints of (MIBLP) by checking emptiness of $\mathcal{W}(\hat{x}, \hat{y})$, we can check feasibility with respect to (BP_k) by checking emptiness of $\mathcal{W}(\hat{x}, \hat{y}; k)$.

With the next result, we show that (BP_k) defines a hierarchy of relaxations for (MIBLP).

Theorem 1. $\mathcal{S} = \mathcal{F}(0) \supseteq \mathcal{F}(1) \supseteq \mathcal{F}(2) \supseteq \ldots \supseteq \mathcal{F}(\bar{k}) = \mathcal{F}$.

Proof. To show that $\mathcal{F}(k) \supseteq \mathcal{F}(k+1)$ for $k = 0, 1, \ldots, \bar{k} - 1$, it is sufficient to observe that $\mathcal{W}(x, y; k') \subseteq \mathcal{W}(x, y; k'')$, for all $k', k'' \in [\bar{k}]$, with $k' \leq k''$, then it follows from Proposition 2. For $k = 0$ we have that $\mathcal{N}_0(y) = \{y\}$ and $\mathcal{R}(x; 0) = \mathcal{S}(x)$. Furthermore, if $k = \bar{k}$ then $\mathcal{N}_{\bar{k}}(y) = Y$ and $\mathcal{R}(x; \bar{k}) = \mathcal{R}(x)$. □

The next question we address explores the theoretical computational complexity of computing the dual bound of (BP_k) for a fixed value of k. As a natural extension of the result in Xueyu et al. [2022], the following theorem shows that the decision version of (BP_k) is NP-complete.

Theorem 2. *The decision version of* (BP_k) *is* NP-*complete for any fixed integer* $k \geq 1$.

Proof. Given any fixed integer $k \geq 1$, we consider the feasibility problem associated with (BP_k), which is one form of decision version of (BP_k). That is, we consider the problem of deciding whether $\exists (x, y) \in \mathcal{F}(k)$. First, we prove that this decision problem is in NP. Then, we show that the problem of determining whether a given MILP is feasible can be reduced to that of deciding whether (BP_k) is feasible.

To show that the feasibility problem associated with (BP_k) is in NP, we show that when $\mathcal{F}(k) \neq \emptyset$, then there exists a certificate that can be verified in polynomial time. When $\mathcal{F}(k) \neq \emptyset$, there must exist $(\hat{x}, \hat{y}) \in \mathcal{F}(k)$. By Proposition 2, $(\hat{x}, \hat{y}) \in \mathcal{F}(k)$ if and only if $\mathcal{W}(\hat{x}, \hat{y}; k) = \emptyset$. Then, a verifier is deciding whether $\forall w \in \mathcal{W}^k, w \notin \mathcal{W}(\hat{x}, \hat{y}; k)$. Given any $w \in \mathcal{W}^k$, verifying that $w \in \mathcal{W}(\hat{x}, \hat{y}; k)$ can be done in polynomial time by deciding the membership of $\hat{y} + w$ to the polytope $\mathcal{P}_2(\hat{x})$. Moreover, the set $\mathcal{W}^k$ has a cardinality of $\mathcal{O}(n^k)$, which is polynomial in n for any fixed integer $k \geq 1$. Therefore, the certificate can be verified in time polynomial in n.

To show that the problem of deciding feasibility of an MILP can be reduced to that of deciding feasibility of (BP$_k$), let $\mathcal{X} := \{x \in X \mid A^1 x \geq b^1\}$ and consider the MILP feasibility problem of determining whether $\mathcal{X} \neq \emptyset$. The certificate for this problem is any $x \in \mathcal{X}$. We show how to construct such a certificate from the certificate of an instance of (BP$_k$). As such, let an instance of (BP$_k$) feasibility be defined as follows. We let $Y = \mathbb{R}^{n_2}_+$, $G^1 = 0_{m_1 \times n_2}$, $A^2 = 0_{m_2 \times n_1}$, $G^2 = I_{n_2}$ (the identity matrix of order n_2), $b^2 = \mathbf{0}_{n_2}$ and $d^2 = \mathbf{1}_{n_2}$ (the all-zeros and all-ones vectors of order n_2, respectively). Note that by construction, $\mathcal{R}(x; k) = \mathcal{R}(x) = \{\mathbf{0}_{n_2}\}$ for all $x \in \mathcal{X}$. It is easy to verify that $x \in \mathcal{X} \iff (x, \mathbf{0}_{n_2}) \in \mathcal{F}(k)$. Then any certificate of feasibility for the constructed instance of (BP$_k$) can be mapped to a certificate for the feasibility of the MILP. □

In the context of bilevel problems with only binary variables at the second level, Xueyu et al. [2022] show that optimizing over (BP$_k$) for "small" values of k already yields much stronger dual bounds for (MIBLP). In this work, the relevance of this hierarchy of relaxations is mainly theoretical. We show in Section 3 that the feasible solutions of (BP$_k$) can be described by adding to the MILP relaxation $\mathcal{S}$ a specific (finite) class of valid linear inequalities arising from improving directions (or equivalently, improving solutions). From a theoretical standpoint, this gives us some insight regarding the strength of such a class of inequalities.

Example 3. *Figure 4 shows slices of $\mathcal{P}$ of the three-dimensional example for the four integer values of x. For each slice, we illustrate the feasible points of (BP$_k$) for different values of k. In particular, we can observe in Figure 4b ($\hat{x} = 1$) that*

$$\begin{aligned} &\mathcal{R}(\hat{x}, 1) \setminus \mathcal{R}(\hat{x}) = \{(\hat{x}, 3, 2), (\hat{x}, 7, 3), (\hat{x}, 2, 2), (\hat{x}, 1, 2)\} \supset \\ \supset\ &\mathcal{R}(\hat{x}, 2) \setminus \mathcal{R}(\hat{x}) = \{(\hat{x}, 2, 2), (\hat{x}, 1, 2)\} \supset \\ \supset\ &\mathcal{R}(\hat{x}, 3) \setminus \mathcal{R}(\hat{x}) = \{(\hat{x}, 1, 2)\}, \end{aligned}$$

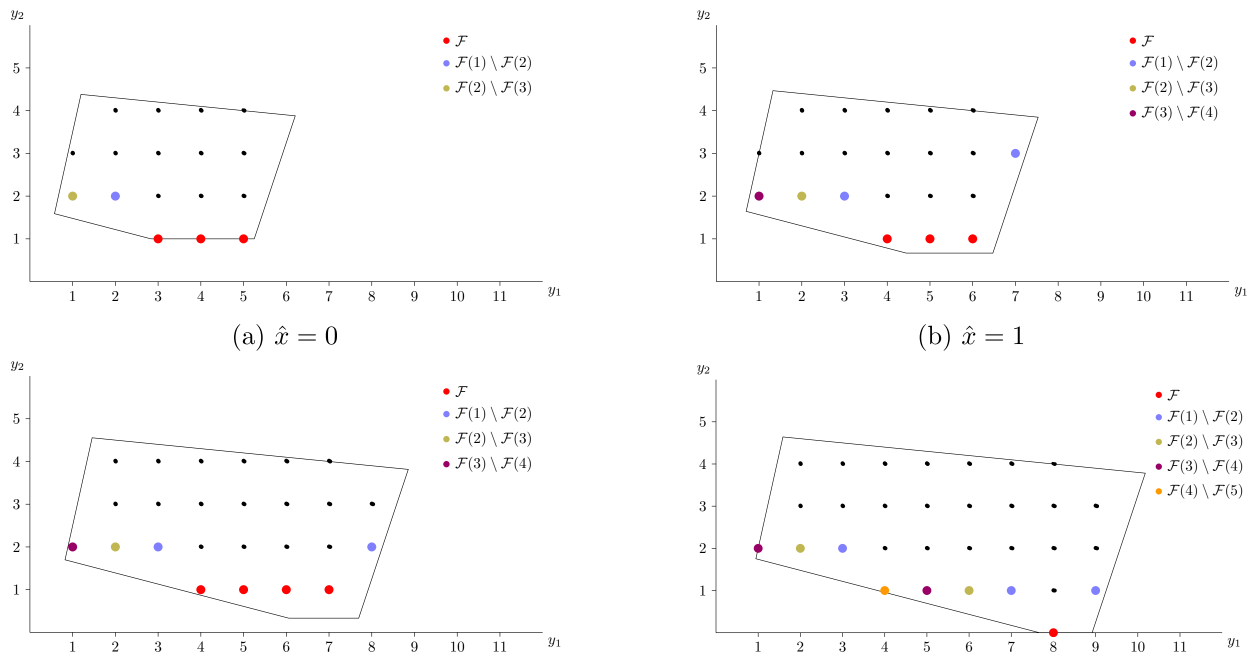
and that

$$\mathcal{F}(1) \supset \mathcal{F}(2) \supset \mathcal{F}(3) \supset \mathcal{F}(4) = \mathcal{F} \cup \{(3, 4, 1)\}.$$

Note that $(3, 4, 1)$ is the only element in $\mathcal{F}(4)$ that is not also in $\mathcal{F}$, since for its unique IFD $w = (4, -1)$ we have $\|w\|_1 = 5$. Therefore, for all $k \geq 5$, $\mathcal{F}(k) = \mathcal{F}$.

3 Valid Inequalities from IDs

In this section, we discuss the generation of valid inequalities from improving directions. Our goal is to formalize what seems to be an intuitive connection between IDICs and the k-opt hierarchy introduced in Section 2, both of which are derived from the concept of improving directions. The connection we aim to establish is that, roughly speaking, $\mathcal{F}(k)$ can be described using IDICs derived only from BFSs obtained from improving directions $w \in \mathcal{W}$ and such that $\|w\|_1 \leq k$. After briefly reviewing the definitions related to ICs in Section 3.1 and introducing a notion of rank analogous to that for valid inequalities in MILPs in Section 3.2, we prove the main theoretical result in Section 3.3.



(a) $\hat{x} = 0$ (b) $\hat{x} = 1$

(c) $\hat{x} = 2$ (d) $\hat{x} = 3$

Figure 4: Slices of the feasible region of (BP_k).

3.1 Intersection Cuts

As already mentioned, we focus on two related types of intersection cuts that can be derived from improving directions and improving solutions, respectively. Specifically, the classes differ in the definition of bilevel free set.

Improving Direction Intersection Cuts. We first describe the BFS used for generating IDICs.

Theorem 3 (Fischetti, Ljubić, et al. [2018]). *Let $(\hat{x}, \hat{y}) \in \mathbb{R}^{n_1+n_2}$ be the extreme point of a simplicial radial cone $\mathcal{V}(\hat{x}, \hat{y})$ containing $\mathcal{F} \neq \emptyset$ and $w \in \mathcal{W}(\hat{x}, \hat{y})$. Then we have that*

$$\alpha^x x + \alpha^y y \geq \beta \quad \forall (x, y) \in \mathcal{F},$$

where this inequality is the IC generated from the bilevel free set

$$\mathcal{C}_{\text{ID}}(w) = \left\{ (x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \;\middle|\; A^2 x + G^2(y + w) \geq b^2 - 1, y + w \geq -1 \right\}. \tag{IDIC}$$

Furthermore, $\alpha^x \hat{x} + \alpha^y \hat{y} < \beta$.

To see why $\mathcal{C}_{\text{ID}}(w)$ is a BFS for any $w \in \mathcal{W}$, observe that for $(\hat{x}, \hat{y}) \in \mathcal{S} \cap \text{int}(\mathcal{C}_{\text{ID}}(w))$, $\hat{y} + w$ is feasible for the associated follower's problem, which means that w is an IFD with respect to $(\hat{x}, \hat{y})$. Hence, $(\hat{x}, \hat{y})$ must be bilevel infeasible. Thus, the set $\mathcal{C}_{\text{ID}}(w)$ contains all points with respect to which w is an improving feasible direction.

Note that the definition of BFS given in (IDIC) is independent of the point $(\hat{x}, \hat{y})$. As long as w is an ID, $\mathcal{C}_{\text{ID}}(w)$ is a BFS—it need not be feasible with respect to any particular point (although the BFS associated with a given ID could be empty). The reason we may want to construct a direction that is an IFD w.r.t. a specific point is that this ensures the point lies in the interior of $\mathcal{C}_{\text{ID}}(w)$, which in turn ensures that the point will violate the generated IC.

To summarize, separation of $(\hat{x}, \hat{y}) \in \mathcal{P} \setminus \mathcal{F}$ by some IC can be guaranteed if

(*i*) $\mathcal{W}(\hat{x}, \hat{y}) \neq \emptyset$;

(*ii*) we can construct a simplicial radial cone $\mathcal{V}(\hat{x}, \hat{y}) \supseteq \mathcal{F}$ with $(\hat{x}, \hat{y})$ as its extreme point; and

(*iii*) $\mathcal{V}(\hat{x}, \hat{y}) \nsubseteq \mathcal{C}_{\text{ID}}(w)$ $(\mathcal{F} \neq \emptyset)$.

By Proposition 1, condition (*i*) is automatically satisfied whenever $(\hat{x}, \hat{y}) \in \mathcal{S} \setminus \mathcal{F}$. In practical computation, the case that $\mathcal{W}(\hat{x}, \hat{y}) = \emptyset$ may arise and is an important consideration, as discussed further in Tahernejad and Ralphs [2025]. In such a case, we cannot separate $(\hat{x}, \hat{y}) \in \mathcal{P}$ from $\mathcal{F}$ with an IDIC, although we can still do so with an inequality valid for $\text{conv}(\mathcal{S})$. Condition (*ii*) is typically easy to satisfy, since when $(\hat{x}, \hat{y})$ is an extreme point of $\mathcal{P}$, the simplicial cone arises naturally from an associated LP basis. Violation of condition (*iii*) means $\mathcal{F} = \emptyset$, which typically only happens after branching constraints have been applied in the context of a branch-and-cut algorithm.

Improving Solution Intersection Cuts. Let us now consider, in constrast, the BFSs used to generate ISICs.

Theorem 4 (Fischetti, Ljubić, et al. [2017]; Fischetti, Ljubić, et al. [2018])**.** *Let* $(\hat{x}, \hat{y}) \in \mathbb{R}^{n_1+n_2}$ *be the extreme point of a simplicial radial cone* $\mathcal{V}(\hat{x}, \hat{y})$ *containing* $\mathcal{F}$ *such that* $d^2\hat{y} > d^2y^*$ *for some* $y^* \in \mathcal{P}_2(\hat{x}) \cap Y$*. Then, under the stated assumptions, we have*

$$\alpha^x x + \alpha^y y \geq \beta \quad \forall (x, y) \in \mathcal{F},$$

where this inequality is the IC associated with the bilevel free set

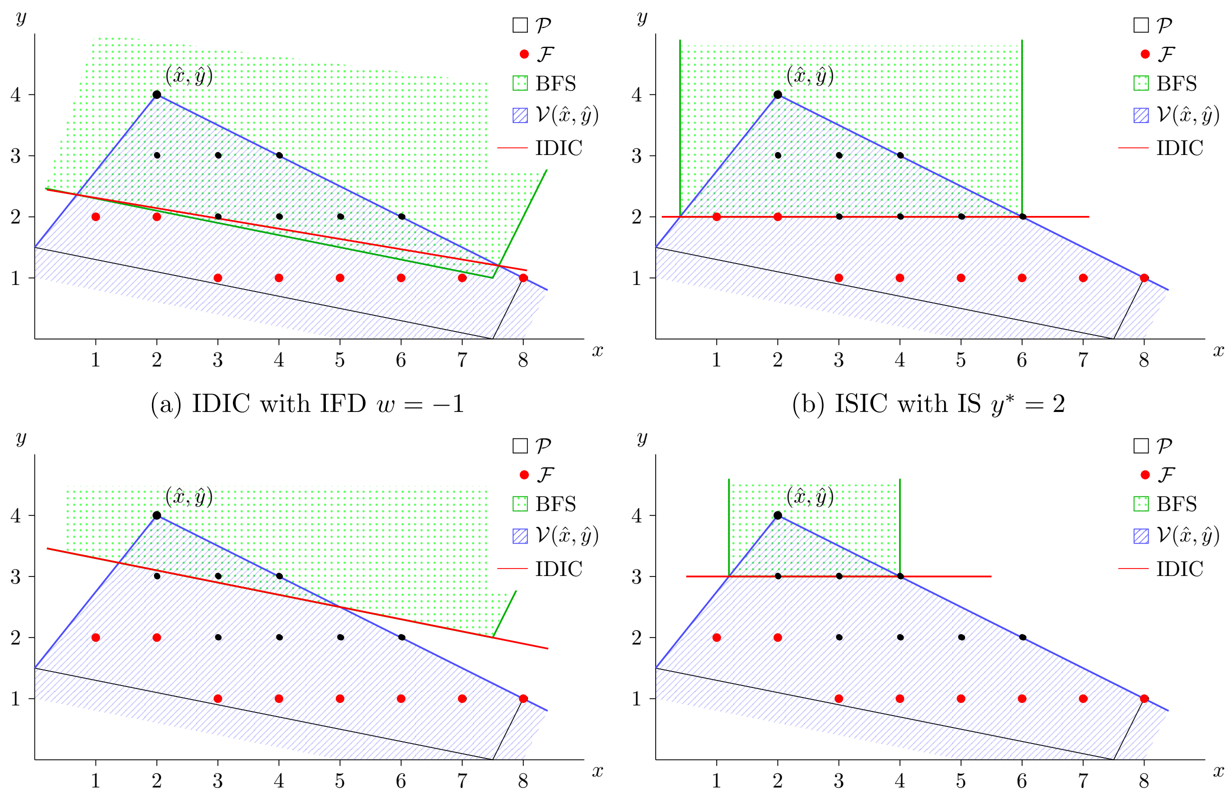
$$\mathcal{C}_{\text{IS}}(y^*) = \left\{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \;\middle|\; d^2y \geq d^2y^*, A^2x \geq b^2 - G^2y^* - 1\right\}, \tag{ISIC}$$

Furthermore, $\alpha^x\hat{x} + \alpha^y\hat{y} < \beta$*.*

The convex set (ISIC) includes all points (x, y) for which x satisfies the follower's constraints for the *fixed* improving solution $y^* \in Y$ and for which y has a second-level objective no better than that of y^*. As with IDICs, the BFS does not depend on the point $(\hat{x}, \hat{y})$. There is a BFS associated with each $y^* \in Y$ (though again, some could be empty). The reason we may desire a y^* such that $d^2y^* < d^2\hat{y}$ is to guarantee that $(\hat{x}, \hat{y})$ can be separated.

As with IDICs, it may be possible to generate an inequality when $(\hat{x}, \hat{y}) \notin \mathcal{S}$. Conditions (*ii*) and (*iii*) for separation by an IDIC must also be satisfied for separation by an ISIC, but instead of condition (*i*), we must have $\mathcal{P}_2(\hat{x}) \cap Y \neq \emptyset$. By Proposition 1, this is assured when $(\hat{x}, \hat{y}) \in \mathcal{S} \setminus \mathcal{F}$.

Example 4. *Let us consider the example from Moore and Bard [1990]. The optimal solution* $(\hat{x}, \hat{y})$ *to the LP relaxation satisfies integrality requirements, but is bilevel infeasible. In this case, it is straightforward to see that* $w = -1$ *is an improving feasible direction and* $y^* = 2$ *is an improving solution. Figure 5 depicts all possible IDICs and ISICs obtained by combining the direction and solution with both BFSs. For the sake of this example, the BFSs reported here are defined by the original constraints rather than the relaxed right-hand sides. Although separation of the current solution is guaranteed under broad assumptions, for deeper cuts, we want* $\mathcal{C}_{\text{ID}}(w)$ *to be as large as possible and this means choosing a "short" directions* w*. On the other hand, larger set* $\mathcal{C}_{\text{IS}}(y^*)$ *arise from solutions* $y^* \in Y$ *that are "further" from* $\hat{y}$*, i.e., for which* $\|y^* - \hat{y}\|$ *is "larger".*



(a) IDIC with IFD $w = -1$

(b) ISIC with IS $y^* = 2$

(c) IDIC with IFD $y^* - \hat{y}$

(d) ISIC with IS $\hat{y} + w$

Figure 5: Illustration of IS and IDICs using different improving solutions/directions on Moore and Bard example [Moore and Bard (1990)]

Connections between ISICs and IDICs. For any given $(\hat{x}, \hat{y}) \in \mathcal{S} \setminus \mathcal{F}$, it is easy to verify that if we are given any improving direction $w \in \mathcal{W}(\hat{x}, \hat{y})$, we can use it to obtain an improving solution $y^* = \hat{y} + w \in \mathcal{P}_2(\hat{x}) \cap Y$ and vice versa. However, this equivalence does not always hold for points in $\mathcal{P} \setminus \mathcal{F}$. Consider, for instance, the point $(1, 2.2) \in \mathcal{P} \setminus \mathcal{S}$ from the Moore and Bard [1990] example. Figure 5a shows that $(1, 2.2) \notin \mathcal{C}_{\text{ID}}(-1)$, implying $\mathcal{W}(1, 2.2) = \emptyset$, even though Figure 5c confirms that $(1, 2.2) \in \mathcal{C}_{\text{IS}}(2)$. Thus, there may exist points in $\mathcal{P} \setminus \mathcal{S}$ that admit improving solutions, but no improving feasible directions. In this case, it occurs because the direction $w = -0.2$ does not satisfy the integrality requirements. As a consequence, the separation of fractional extreme points may fail even when $d^2\hat{y} > \phi(b^2 - A^2\hat{x})$, whereas this cannot happen with ISICs.

Another important aspect is the contrast of the two BFSs. The set $\mathcal{C}_{\text{ID}}(w)$ is defined by a direction w, not a fixed second-level solution. In contrast, $\mathcal{C}_{\text{IS}}(y^*)$ is defined with respect to a fixed $y^* \in Y$. Intuitively, to enlarge $\mathcal{C}_{\text{IS}}(y^*)$, y^* should be a *high quality* solution to the second-level problem—corresponding to directions of larger magnitude—which is in opposition to the goal of choosing w such that $\mathcal{C}_{\text{ID}}(w)$ is large. For this reason, depending on what kind of IC we are generating, different directions must be considered desirable. This issue is addressed in depth in Section 4.2.

3.2 Closures and Rank for Intersection Cuts

In the context of MILPs, one way of characterizing the overall strength of a specific class of valid inequalities is by analyzing the so-called *closure*, which is the convex set obtained by adding all inequalities in the class to the initial LP relaxation. Before making the connections between IDICs and the k-opt hierarchy more formal, we first show how to apply the standard notions of closure and rank to the ICs discussed in Section 3.1.

As with MILPs, when the set of non-dominated inequalities in a class is finite, the closure is a polyhedron and is itself a relaxation of the original problem. The inequalities defining this first closure are defined to have rank 1. Taking the closure again with respect to the relaxation defined by the first closure yields the rank 2 closure and this process can be iterated. In general, the closure of rank r is the closure with respect to that of rank $r-1$. The rank of a given valid inequality with respect to this hierarchy is the smallest value of r such that the inequality is valid for the rank r closure, but not the rank $r-1$ closure [Cornuéjols (2008)].

To apply these concepts to ICs, an obvious approach would be to consider all cuts that can be derived from the procedure of Definition 4, taking $(\hat{x}, \hat{y})$ to be one of the extreme points of $\mathcal{P}$. Given an extreme point $(\hat{x}, \hat{y})$, the definition requires specifying a simplicial radial cone pointed at $(\hat{x}, \hat{y})$. Such a cone is easily obtained from a basis of the LP relaxation with respect to which $(\hat{x}, \hat{y})$ is the associated basic feasible solution. Different bases yield different cuts, so a closure could be derived by considering all cones arising from all bases for all extreme points of $\mathcal{P}$ and combining these with all possible BFSs.

A simpler construction is one in which we enumerate all possible BFSs and consider the convex hull of the complement of the interior for each. By taking the intersection of all such complements with $\mathcal{P}$, we obtain a similar closure. More formally, for each $w \in \mathcal{W}$, let $D^1(w) := \text{conv}(\mathcal{P} \setminus \text{int}(\mathcal{C}_{\text{ID}}(w)))$. Then the first closure of $\mathcal{P}$ with respect to the class of ICs associated with BFSs (IDIC) is called the *rank 1 IDIC closure* and defined as

$$\mathcal{P}^1_{\text{ID}} = \bigcap_{w \in \mathcal{W}} D^1(w).$$

Although the set of all IDs $\mathcal{W}$ is not finite in general, it can be replaced by a finite subset consisting of IDs with 1-norm at most $\bar{k}$ (as defined earlier) in the above. The *rank r IDIC closure*, denoted by $\mathcal{P}^r_{\text{ID}}$, is recursively defined as the rank 1 closure of $\mathcal{P}^{r-1}_{\text{ID}}$ as follows

$$\mathcal{P}^r_{\text{ID}} = \bigcap_{w \in \mathcal{W}} D^r(w),$$

where $D^r(w) := \text{conv}(\mathcal{P}_{\text{ID}}^{r-1} \setminus \text{int}(\mathcal{C}_{\text{ID}}(w)))$, for all $w \in \mathcal{W}$, and $\mathcal{P}_{\text{ID}}^0 := \mathcal{P}$. Hence, the closure yields a natural hierarchy of relaxations over these classes of valid inequalities, ranging from low-rank (weaker) to high-rank (stronger) cuts.

As noted before, since there are points in $\mathcal{P} \setminus \text{conv}(\mathcal{F})$ that cannot be separated by IDICs at all, the facet-defining inequalities of $\text{conv}(\mathcal{F})$ cannot all be expected to have finite IDIC rank. The next example shows a case in which there exists an r^* such that $\mathcal{P}_{\text{ID}}^{r^*+1} = \mathcal{P}_{\text{ID}}^{r^*}$ and $\mathcal{P}_{\text{ID}}^{r^*} \cap \mathcal{S} \neq \mathcal{F}$.

Example 5. *Figure 6 depicts a polytope $\mathcal{P} \supset \mathcal{F}$ of the Moore and Bard [1990] example. As illustrated, all extreme points $(x, y) \in \text{ext}(\mathcal{P})$ are fractional, i.e., $(x, y) \notin \mathcal{S}$. However, none of these extreme points lie in $\text{int}(\mathcal{C}_{\text{ID}}(-1))$ (depicted in green), and thus cannot be separated by an IDIC. Consequently, we have that $\mathcal{P}_{\text{ID}}^r = \mathcal{P}$, for all $r > 0$. Moreover, $\mathcal{P}_{\text{ID}}^r \cap \mathcal{S} \neq \mathcal{F}$, as the point $(3, 2) \in (\mathcal{P}_{\text{ID}}^r \cap \mathcal{S}) \setminus \mathcal{F}$.*

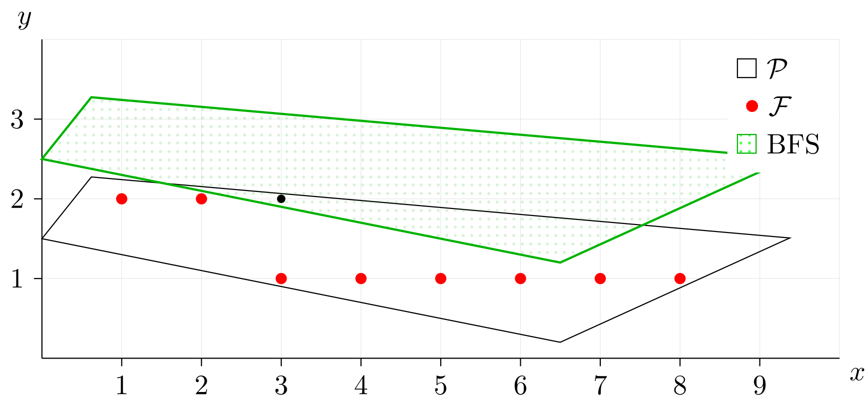


Figure 6: Illustration of a polytope where all extreme points are fractional and have an empty set of IFDs.

On the other hand, it is possible to separate any point in $\mathcal{S} \setminus \mathcal{F}$ that is an extreme point of $\mathcal{P}$ from $\text{conv}(\mathcal{F})$ with an IDIC, so a pure cutting plane method using a combination of IDICs and MILP cuts is possible. Next, we show that $\mathcal{P}_{\text{ID}}^r$ is a polyhedron for any $r > 0$.

Proposition 3. *For all $w \in \mathcal{W}$, $D^1(w)$ is a polyhedron.*

Proof. We show that the set of extreme points of $\text{conv}(\mathcal{P} \setminus \text{int}(\mathcal{C}_{\text{ID}}(w)))$ is finite. Note that $\mathcal{P} \cap \mathcal{C}_{\text{ID}}(w)$ is a polyhedron. Hence, $\text{ext}(\mathcal{P} \cap \mathcal{C}_{\text{ID}}(w))$ is a finite set. Then, by construction we have that

$$\text{ext}(\text{conv}(\mathcal{P} \setminus \text{int}(\mathcal{C}_{\text{ID}}(w)))) \subseteq \text{ext}(\mathcal{P}) \cup \text{ext}(\mathcal{P} \cap \mathcal{C}_{\text{ID}}(w)).$$

Since both sets on the right-hand side are finite, so must be the left-hand side. □

Theorem 5. *Let $r > 0$ be given. Then $\mathcal{P}_{\text{ID}}^r$ is a polyhedron.*

Proof. Note that for all $\hat{w} \in \mathcal{W} \setminus \mathcal{W}^{\bar{k}}$, we have $\text{conv}(\mathcal{P} \setminus \text{int}(\mathcal{C}_{\text{ID}}(\hat{w}))) = \mathcal{P}$. Then

$$\mathcal{P}_{\text{ID}}^r = \bigcap_{w \in \mathcal{W}} D^r(w) = \bigcap_{w \in \mathcal{W}^{\bar{k}}} D^r(w),$$

for all $r > 0$. First, we prove the statement for $r = 1$. For all $w \in \mathcal{W}^{\bar{k}}$, $D^1(w)$ is polyhedral by Proposition 3. It follows that $\mathcal{P}^1_{\text{ID}}$ is obtained as the intersection of a finite number of polyhedra. Assuming the statement holds for $r-1$, we prove the statement for $r > 1$. Since $\mathcal{P}^{r-1}_{\text{ID}} \subseteq \mathcal{P}$ is polyhedral, by Proposition 3 we have that $D^r(w)$ is still polyhedral, for all $w \in \mathcal{W}^{\bar{k}}$. Hence, $\mathcal{P}^r_{\text{ID}}$ is the result of the intersection of a finite number of polyhedra. □

The same analysis can be applied to the ISICs defined by (ISIC) to obtain a hierarchy defined by closures $\mathcal{P}^r_{\text{IS}}$.

3.3 Intersection Cuts and k-optimality

The hierarchy of relaxations presented in the previous section provides a framework to classify the strength of IDICs as a function of their IDIC rank, which is the natural counterpart of the similar hierarchies in the theory of valid inequalities for MILPs (Chvátal-Gomory rank, etc.). Our goal in this section, however, is to analyze a different and more practical hierarchy that classifies the strength of IDICs as a function of the points in $\mathcal{S}$ they can separate, with a close relationship to the k-opt hierarchy introduced earlier.

The analysis in this section was the original motivation for undertaking the work presented in this paper, driven by the desire of computing the strong bounds produced by the k-opt relaxation. As our main result in this section shows, these bounds can be obtained with only a minimal modification to the existing branch-and-cut framework of Tahernejad, Ralphs, and DeNegre [2020] by restricting the generation of valid inequalities to the subset of IDICs that are valid for the feasible points of the k-opt hierarchy, that is, those derived from IFDs with 1-norm at most k. To formalize this result, we first introduce the notion of a *k-IDIC*, an IDIC valid for (BP$_k$).

Definition 5. *For $k \in [\bar{k}]$, a k-IDIC is an IC generated from a BFS $\mathcal{C}_{\text{ID}}(w)$ such that $w \in \mathcal{W}^k$.*

Before getting into the formalities, we first outline the intuition behind the proof. Consider any bilevel infeasible point $(x, y) \in \mathcal{S} \setminus \mathcal{F}$ and let $k^* = \min_{w \in \mathcal{W}(x,y)} \|w\|_1$ be the smallest 1-norm of any IFD w.r.t. (x, y). Proposition 2 and Theorem 1 together imply that $(x, y) \in \mathcal{F}(k)$ if and only if $k \leq k^* - 1$. In particular, we must have $\mathcal{W}(x, y; k^*) \neq \emptyset$ and hence, there must be at least one IFD $w \in \mathcal{W}(x, y; k^*)$. From this IFD w, given an appropriate simplicial cone (which we show below always exists), we can always derive a k^*-IDIC separating (x, y) from $\mathcal{F}(k^*)$ (and hence also from $\mathcal{F}(k)$ for all $k^* \leq k \leq \bar{k}$).

To state the same logic in other terms, Proposition 2 and Theorem 1 together specify a partition of $\mathcal{S} \setminus \mathcal{F}$ into *levels* that correspond to levels of the k-opt hierarchy. A point (x, y) is on *level k* if it is in $\mathcal{F}(k) \setminus \mathcal{F}(k+1)$. A point on level k can be separated from $\mathcal{F}(k+1)$ by a $(k+1)$-IDIC.

It is important to emphasize that there is no relationship in general between the IDIC *rank* of an inequality (as defined in Section 3.2) and the level of the points it can separate. In fact, for any fixed k, a k-IDIC can have any IDIC rank. There exists an entire rank hierarchy for k-IDICs for any fixed k that mirrors that derived in the previous section. Applying Definition 5 to the extreme points of $\mathcal{P}$, for example yields k-IDICs of "rank 1" and we can iterate to derive k-IDICs of higher rank.

Next, we state the main result of this section which comprises two parts. The first establishes that

k-IDICs are valid for $\mathcal{F}(k)$ by showing that for all $w \in \mathcal{W}^k$, $\mathcal{C}_{\text{ID}}(w)$ is "$\mathcal{F}(k)$-free". The second shows that any point in $\mathcal{S}$ that is not in $\mathcal{F}(k)$ can be separated by a k-IDIC.

Theorem 6. *For all $k \in [\bar{k}]$, the following statements hold:*

(1) *all k-IDICs are valid for $\mathcal{F}(k)$; and*

(2) *for all $(x, y) \in \mathcal{S} \setminus \mathcal{F}(k)$, there exists a k-IDIC violated by (x, y).*

In other words, k-IDICs can separate all and only *points in $\mathcal{S} \setminus \mathcal{F}(k)$.*

Proof. (1) Let $k \in [\bar{k}]$ and $w \in \mathcal{W}^k$ be given. We show that $\text{int}(\mathcal{C}_{\text{ID}}(w)) \cap \mathcal{F}(k) = \emptyset$. By definition of (IDIC), $w \in \mathcal{W}(x, y; k)$ for all $(x, y) \in \text{int}(\mathcal{C}_{\text{ID}}(w)) \cap \mathcal{S}$. Consequently, Proposition 2 implies $(x, y) \notin \mathcal{F}(k)$.

(2) Let $k \in [\bar{k}]$ and $(x, y) \in \mathcal{S} \setminus \mathcal{F}(k)$ be given. First, we show that a suitable simplicial radial cone can be constructed from standard geometric arguments. The polyhedron

$$\mathcal{F}^{(x,y)}(k) = \text{conv}(\mathcal{F}(k) \cup \{(x, y)\})$$

contains $\mathcal{F}(k)$ and has (x, y) as one of its extreme points. Moreover, there exists a simplicial radial cone $\mathcal{V}(x, y)$ pointed at (x, y) whose description is obtained by selecting a set of linearly independent inequalities binding at (x, y) from the description of $\mathcal{F}^{(x,y)}(k)$. Then, by Proposition 2 there exists a $w \in \mathcal{W}(x, y; k)$; and $(x, y) \in \text{int}(\mathcal{C}_{\text{ID}}(w))$ by the definition of (IDIC). Therefore,

$$\text{conv}\left(\mathcal{F}^{(x,y)}(k) \setminus \text{int}(\mathcal{C}_{\text{ID}}(w))\right)$$

is a k-IDIC violated by (x, y). □

To complete the argument, we show that, for any given $k \in [\bar{k}]$, the convexification of the set of points satisfying both integrality requirements and all possible k-IDICs derived from Theorem 6 coincides with that of the feasible region of (BP_k). As such, for all $k \in [\bar{k}]$, let

$$\Pi_{\text{ID}}^k = \bigcap_{(x,y) \in \mathcal{F}(k-1) \setminus \mathcal{F}(k)} \ \bigcap_{w \in \mathcal{W}(x,y;k)} \text{conv}\left(\mathcal{F}^{(x,y)}(k) \setminus \text{int}(\mathcal{C}_{\text{ID}}(w))\right).$$

be the intersection of all the k-IDICs separating from $\mathcal{F}(k)$ points in $\mathcal{F}(k-1) \setminus \mathcal{F}(k)$, those having level $k - 1$.

Corollary 1. *For all $k \in [\bar{k}]$, $\text{conv}(\mathcal{S} \cap \Pi_{\text{ID}}^k) = \text{conv}(\mathcal{F}(k))$.*

From a theoretical standpoint, the previous result connects the strength of IDICs generated from an ID with 1-norm k to the k-opt relaxation. From a practical standpoint, it suggests that the separation of this family of cuts produces dual bounds converging to those associated with the hierarchy (BP_k). As a matter of fact, branch-and-cut and cutting-plane methods can be interpreted as optimizing over the convex hull of a certain feasible region. Therefore, such dual bounds can, in principle, be computed by a pure cutting plane in which integrality is restored with the separation of standard MILP cuts, while k-optimality is enforced by k-IDICs. Equivalently, the same bounds can be produced by a branch-and-cut algorithm exclusively generating k-IDICs, with integrality enforced (convexification) through standard MILP branching.

We end the section with a few complementary results, illustrated through examples. In contrast with the properties established for IDICs, an analogous result does not hold in general for ISICs. Specifically, given a point $(\hat{x}, \hat{y}) \in \mathcal{S} \setminus \text{conv}(\mathcal{F}(k))$ and a $w \in \mathcal{W}(\hat{x}, \hat{y}; k)$, it is not necessarily the case that w is an IFD for every $(x, y) \in \text{int}(\mathcal{C}_{\text{IS}}(\hat{y} + w)) \cap \mathcal{S}$, as we exemplify next.

Example 6. *Figure 7 shows the slice of the three-dimensional example for $\hat{x} = 1$. Let us consider $\hat{y} = (4, 3)$, $w = (0, -1)$ and $y^* = (4, 2)$ as the improving solution. The BFS is defined as $\mathcal{C}_{\text{IS}}(y^*) = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid y_2 \geq 2, 0 \leq x \leq 3\}$. One can verify that $(1, 7, 3) \in \mathcal{C}_{\text{IS}}(y^*)$, but $w \notin \mathcal{W}(1, 7, 3; 1) = \emptyset$.*

Note that, unlike the MILP setting where all points satisfying integrality requirements within the convex hull of $\mathcal{S}$ are feasible, there may exist points $(x, y) \in \text{conv}(\mathcal{F}(k)) \cap \mathcal{S}$ for which $\mathcal{W}(x, y; k) \neq \emptyset$, and thus $(x, y) \notin \mathcal{F}(k)$, as illustrated in the following example.

Example 7. *Figure 7 shows a slice of $\text{conv}(\mathcal{F}(1))$ (in green) of the three-dimensional example, for $\hat{x} = 1$. It is easy to verify that the vector $\hat{w} = (0, -1)$ is an IFD for the point $(4, 2) \in \mathcal{S}(\hat{x})$, i.e., $\mathcal{W}(4, 2; k) \neq \emptyset$. Although $(4, 2) \in \text{int}(\mathcal{C}_{\text{ID}}(\hat{w}))$, there exists no simplicial radial cone pointed at $(4, 2)$ that contains $\mathcal{F}(k)$, since $(4, 2) \in \text{conv}(\mathcal{F}(k))$. Therefore, in this case, separation with k-IDICs is not possible but also unnecessary.*

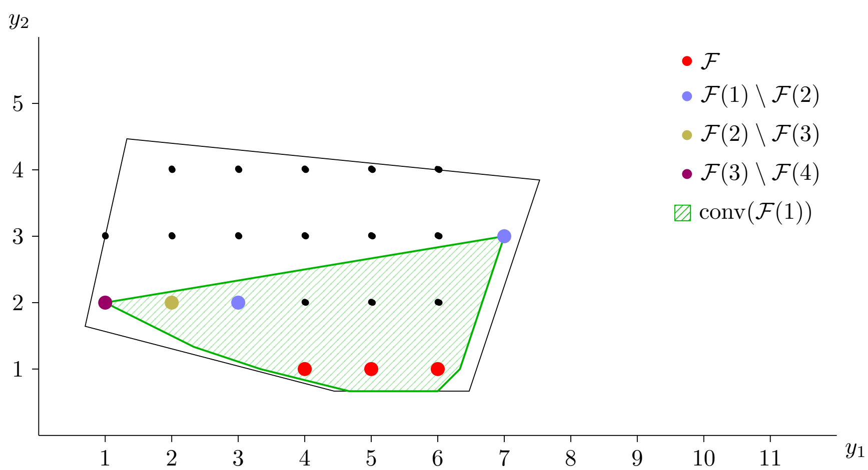


Figure 7: A slice of $\text{conv}(\mathcal{F}(1))$ of the three-dimensional example, for $\hat{x} = 1$

4 Branch-and-Cut Algorithm

In this section, we present a branch-and-cut algorithm based on the overarching framework of [Tahernejad, Ralphs, and DeNegre (2020); Tahernejad and Ralphs (2025)] but in which the feasibility check and the generation of valid inequalities are unified with an oracle for finding an IFD. The implementation is based on the open source solver `MibS` [DeNegre et al. (2024)] and we refer the reader to [Tahernejad, Ralphs, and DeNegre (2020); Tahernejad and Ralphs (2025)] for details. This section describes the details of the main components, focusing on the differences from what is currently implemented in `MibS 1.2`.

4.1 General Framework

The outline of the overall method is in Algorithm 1. Obviously, this is only a general framework

Algorithm 1 Generic Branch-and-Cut Using Improving Direction Oracle

1: Initialize the set Q of unexplored subproblems with the original problem.
2: $U \leftarrow \infty$
3: **while** $Q \neq \emptyset$ **do**
4:     **Select** a node t from Q with feasible region $\mathcal{F}^t$.
5:     **Bound** node t to generate dual bound L^t and candidate solution (x^t, y^t).
6:     **if** $L^t = \infty$ **or** $L^t \geq U$ **then**
7:         **Prune** node t.
8:     Check $\mathcal{W}(x^t, y^t) \stackrel{?}{=} \emptyset$ (optional if $(x^t, y^t) \notin \mathcal{S}$).
9:     **if** $\mathcal{W}(x^t, y^t) \neq \emptyset$ and/or $(x^t, y^t) \notin \mathcal{S}$ **then**
10:         Either

- **separate** (x^t, y^t) from $\text{conv}(\mathcal{F})$ and put t back in Q; or
- **branch** and add new subproblems $t_1, \ldots, t_k$ to Q.

11:     **else**
12:         $(x^t, y^t) \in \mathcal{F}$ by Proposition 1.
13:         $U^t \leftarrow cx^t + d^1 y^t$.
14:         $U \leftarrow \min\{U, U^t\}$.
15:         **Prune** node t.

and the precise way in which various steps in this algorithm are implemented determines crucially how the algorithm will perform in practice. In particular, the choice between separation (generating a violated valid inequality) and branching on line 10 is important. Most of the detailed control mechanisms, including the one for making the decision between branching and cutting, are the same here as in `MibS 1.2` and are discussed in [Tahernejad and Ralphs (2025)]. In the remainder of this section, we focus on the method used for generating IFDs, which is the main innovation.

4.2 Generating Feasible Improving Directions

Line 8 of the algorithm is the step that differentiates our approach from that of Tahernejad, Ralphs, and DeNegre [2020]. We now switch gears and discuss the empirical issues surrounding the framework that has been introduced in the previous sections. In particular, we focus on the practical details of the problem of generating an appropriate IFD, given a candidate solution $(\hat{x}, \hat{y}) \in \mathcal{P}$. As we already mentioned, the goal of finding an IFD is twofold. First, the IFD (or the proof that none exists), certifies the feasibility status of $(\hat{x}, \hat{y})$ by Proposition 1 when $(\hat{x}, \hat{y}) \in \mathcal{S}$ (otherwise, $(\hat{x}, \hat{y})$ is trivially infeasible). Second (and perhaps more crucially), the direction serves to produce an IC violated by $(\hat{x}, \hat{y})$. For this latter purpose, the particular direction produced makes a big difference. In the following two sections, we first discuss relevant objective functions and then both an exact and a heuristic way of generating directions according to these objective functions.

4.2.1 Objective Functions

Although any IFD in $\mathcal{W}(\hat{x}, \hat{y})$ suffices to generate a BFS of either the form (IDIC) or the form (ISIC) that can be used to generate an associated IC violated by $(\hat{x}, \hat{y})$, the particular objective function used serves to guide the selection, ideally leading to BFSs that are "larger" and therefore produce deeper cuts.

Let us consider the BFS (IDIC) first. As previously remarked, a direction $w \in \mathcal{W}(\hat{x}, \hat{y})$ is likely to include more points in $\mathcal{C}_{\text{ID}}(w)$ if its 1-norm is small. Clearly, a possible approach is to consider the following objective function

$$\min \ \|w\|_1 . \tag{3}$$

Fischetti, Ljubić, et al. [2018] proposed an objective function that has a related goal

$$\min \left\{ \sum_{i=1}^{m_2} \max\{g_i^2 w, 0\} + \|w\|_1 \right\}, \tag{4}$$

where g_i^2 is the i^{th} row of G^2. This objective can be linearized in the obvious way and the idea is that row i can be dropped in the definition of set (IDIC) if $g_i^2 w \leq 0$, enlarging the BFS.

For the BFS (ISIC), although it is constructed based on a solution, we can nevertheless consider any $w \in \mathcal{W}(\hat{x}, \hat{y})$ and set $y^* = \hat{y} + w$. Intuitively, a good direction for this IC is obtained by maximizing the improvement in the follower's objective function by using the objective

$$\min d^2 w. \tag{5}$$

Other objectives are also possible. For example, an objective with the philosophy similar to that of (4) is suggested in Fischetti, Ljubić, et al. [2017].

4.2.2 Algorithms

Exact. Given a desired objective function, the most straightforward way of generating an IFD is to take the approach described in [Fischetti, Ljubić, et al. (2018); Tahernejad and Ralphs (2025)] for the generation of IDICs. That is, we describe elements of $\mathcal{W}(\hat{x}, \hat{y})$ as the points in the following mixed-integer set and directly optimize over this set with the given objective using an off-the-shelf method for solving MILPs.

$$\begin{aligned} d^2 w &\leq -1 \\ G^2 w &\geq b^2 - A^2 \hat{x} - G^2 \hat{y} \\ w &\geq -\hat{y} \\ w &\in Y. \end{aligned} \tag{ID}$$

As previously noted, the infeasibility of (ID) certifies emptiness of $\mathcal{W}(\hat{x}, \hat{y})$, while any solution is an IFD with respect to $(\hat{x}, \hat{y})$. From a theoretical standpoint, the question of whether there is a point satisfying (ID) is equivalent to that of whether there is an improving solution to the follower's problem, provided that $(\hat{x}, \hat{y}) \in \mathcal{S}$. From a practical standpoint, there is a difference, since (ID) allows us to specify an objective function that favors certain improving directions (and thus certain improving solutions) over others.

Heuristic. Although optimizing over (ID) is a straightforward approach, the main challenges in finding IFDs this way is the computational burden associated with solving an NP-hard subproblem. Fortunately, to guarantee the correctness of a hypothetical branch-and-cut algorithm using an oracle based on Proposition 1 for checking bilevel feasibility, we only need an exact answer to the associated decision problem when the candidate solution satisfies integrality requirements. In cases where separation is the primary task, a heuristic approach for finding feasible solutions of (ID) would suffice. Furthermore, it may be computationally advantageous to first run heuristics, even when the solution belongs to $\mathcal{S}$, resorting to an exact algorithm only if no IFD is found using the heuristic. Recent work by Gaar et al. [2024] highlights the benefits of heuristic methods for identifying improving solutions in the context of generating valid inequalities for integer bilevel nonlinear problems, further supporting the promise of this approach.

We present two heuristics aimed at producing an IFD with respect to a given $(\hat{x}, \hat{y}) \in \mathcal{P}$ by restricting the feasible region to, e.g., elements of $\mathcal{W}(\hat{x}, \hat{y}; k)$, for some "small" value of k. This choice is supported by some preliminary evidence. First, the example of Figure 4 suggests that $\mathcal{F}(k)$ is a reasonably good approximation of $\mathcal{F}$ even for small values of k. More broadly, the results in Xueyu et al. [2022] also indicate that $k \approx 3$ provides a good trade-off between the quality of the dual bound of (BP_k) and the computational burden required to compute it. Corollary 1 provides a theoretical guarantee that the effect of generating k-IDICs in this context should mirror the effect of solving the k-opt relaxation. Perhaps most importantly, this problem lends itself well to heuristic approaches, such as local searches, since if a direction $w \in \mathcal{W}(\hat{x}, \hat{y}; k)$ exists, then it must direct $\hat{y}$ towards points in its k-neighborhood. Consequently, a promising approach is to search exclusively on directions leading $\hat{y}$ to points in $\mathcal{N}_k(\hat{y})$.

The first method is described in Algorithm 2 and uses a pure local search algorithm. Given a value of k, it examines all possible directions w with $\|w\|_1 \leq k$ and checks if they are improving and feasible. Note that this approach allows to use any (possibly non-linear and/or non-convex) objective function as a measure of "quality" of the multiple IFDs the local search may identify.

The second approach is inspired by the well-known primal heuristic for MILPs, *local branching* [Fischetti and Lodi (2008)], and consists in intersecting the feasible region of (ID) with the k-neighborhood of $\hat{y}$, leading to the following formulation:

$$
\begin{aligned}
d^2 w &\leq -1 \\
G^2 w &\geq b^2 - A^2 \hat{x} - G^2 \hat{y} \\
w &\geq -\hat{y} \\
\|w\|_1 &\leq k \\
w &\in Y.
\end{aligned}
\tag{k-ID}
$$

The resulting MILP tends to be noticeably easier to solve than (ID). Moreover, if (4) is used as objective function, then the same artificial variables can be used to linearize both the objective and the 1-norm constraint.

Algorithm 2 `generateNeighbors(k, (x, y), obj)`

Input: $\texttt{k} \leq \bar{k}$, $\texttt{(x, y)} \in \mathcal{P}$, $\texttt{obj} : \mathbb{Z}^{n_2} \to \mathbb{R}$
Output: $w^* = \text{argmin}_{w \in \mathcal{W}} \texttt{obj}(w)$, with $\mathcal{W} \subseteq \mathcal{W}(\hat{x}, \hat{y}; k)$

1: Let $\mathcal{W} \leftarrow \emptyset$ ▷ Initialize $\mathcal{W}$
2: **for** $w \in \mathbb{Z}^{n_2}$ such that $\|w\|_1 \leq \texttt{k}$ **do**
3:     **if** $d^2 w \not\leq -1$ **then**
4:         Discard w and go to line 2 ▷ w is not improving
5:     **if** $G^2 w \not\geq b^2 - A^2 \hat{x} - G^2 \hat{y}$ **then**
6:         Discard w and go to line 2 ▷ w is not feasible
7:     **if** $\hat{y} + w \not\geq 0$ **then**
8:         Discard w and go to line 2 ▷ w is not feasible
9:     $\mathcal{W} \leftarrow \mathcal{W} \cup \{w\}$ ▷ w is an IFD for $(\hat{x}, \hat{y})$
10: $w^* \leftarrow \text{argmin}_{w \in \mathcal{W}} \texttt{obj}(w)$
11: **return** w^*

5 Computational Results

In this section, we report on the extensive empirical analysis carried out to assess the impact of the modifications to the standard branch-and-cut approach already implemented in `MibS 1.2`. This analysis has several related goals. First and foremost, we want to assess the advantage of unifying the feasibility check with the generation of valid inequalities by use of a single oracle. Second, within the context of Algorithm 1, we want to measure the computational benefit of heuristic methods for finding IFDs. The overall objective is to determine whether the scheme presented has the potential to improve state-of-the-art solvers.

5.1 Implementation Details

We build on the foundation presented in Tahernejad, Ralphs, and DeNegre [2020], which is implemented in the open source solver `MibS`, distributed by the COIN-OR [2018] Foundation's repository of open source projects. For a comprehensive description of the parameters available in `MibS`, we refer the reader to the documentation available at `https://github.com/coin-or/MibS`. All algorithmic variations presented have been implemented on top of version `1.2.1` and their employment can be regulated using the following newly introduced parameters.

- `useImprovingDirectionOracle` controls whether bilevel feasibility is checked using the new oracle for generating improving directions as in Algorithm 1. Otherwise, `MibS`'s default oracle is used and the scheme follows to the one presented in Tahernejad, Ralphs, and DeNegre [2020].

- `improvingDirectionType` controls the method used to find an IFD, either the local search is used, as described in Algorithm 2, or the problem (ID) is solved as an MILP. Note that even when the local search is used, the solver must still solve (ID) when necessary to guarantee the correctness (see discussion in Section 4.2).

- `maxNeighborhoodSize` controls the parameter k in Algorithm 2 (k-ID, resp.) if `improvingDirectionType` is set to 1 (0, resp.).

- `useLocalSearchDepthLb` and `useLocalSearchDepthUb` regulate the use of heuristics for finding an IFD. These parameters restrict the execution of Algorithm 2 or (k-ID) to nodes whose depth in the search tree falls within the specified lower and upper bounds. Otherwise, problem (ID) is solved.

Most of the experiments were done using the automatic default parameters settings in MibS 1.2.1, but when `useImprovingDirectionOracle` is set to `True`, several default behaviors are changed.

- The automatic setting of parameters related to cut generation is disabled and *only* the generation of IDICs is enabled. The automatic defaults in `MibS` may or may not enable generation of IDICs and/or additional classes.

- The `fractional` branching strategy is always used, even for interdiction problems. This is because we conjecture that the `fractional` strategy has a clear potential to be more effective in the context of Algorithm 1, since it fosters the satisfaction of integrality requirements at both levels, allowing a more frequent application of Proposition 1 and, in turn, the discover of primal bounds.

- Finally, in `MibS 1.2`, the bilevel feasibility check precedes the cut generation and is implemented with an oracle that evaluates (VF) by solving the follower's problem to optimality. By default, this check is undertaken when either the current solution satisfies integrality at both levels or all the linking variables are fixed. In the latter case, `MibS` solves an upper bounding problem (one additional MILP) to find the best solution for the given set of linking variables. When `useImprovingDirectionOracle` is `True`, the usual check is bypassed, and the feasibility check is instead undertaken *after* cut generation, since Proposition 1 says that the integer solution is feasible if and only if we are not able to find an IFD (generate an IC).

5.2 Dataset

The selection of a diversified collection of instances is one of the crucial tasks for an insightful empirical analysis since special classes of problems can show very different behavior. Recently, Thürauf et al. [2025] released the Bilevel Optimization (Benchmark) Instance Library (BOBILib) whose intent is to provide access to the community to a large and well-curated set of test instances, in a similar fashion of the MIPLIB [Gleixner et al. (2021)] for MILPs. The instances used in this work are drawn from Tahernejad and Ralphs [2025], which includes some of the instances available in BOBILib, along with additional ones constructed by the authors. Moreover, a subset of instances from Xueyu et al. [2022] is included. Table 1 provides details on each dataset, including the number of instances, types of leader's and follower's variables, the range of variables and constraints at each level, and the alignment of the objective functions of the two levels. There are a total of 399 interdiction problems and 277 IBLPs. The final count of instances is 676.

Class	Data Set	#	Var. Type	Var#	Constr#	Align	Source
Interdiction	INT-DEN	300	B B	10-40 10-40	1 11-41	-1	DeNegre [2011]
	INT-SHI	99	B B	15-30 15-30	1 16-31	-1	Xueyu et al. [2022]
IBLP	DEN	50	I I	5-15 5-15	0 20	Varies	DeNegre [2011]
	DEN2	110	I I	5-10 5-20	0 5-15	Varies	DeNegre [2011]
	ZHANG	30	B I	50-80 70-110	0 6-7	0.6-0.8	Zhang and Ozaltın [2017]
	ZHANG2	30	I I	50-80 70-110	0 6-7	0.6-0.8	Zhang and Ozaltın [2017]
	FIS	57	B B	Varies Varies	Varies Varies	-1	Fischetti, Ljubić, et al. [2018]

Table 1: Summary of the datasets

5.3 Experimental Setup

All experiments were conducted on compute nodes running the Linux (Debian 8.11) operating system with dual AMD Opteron 6128 processors and 32 GB of RAM. All experiments were run sequentially with a time limit of 3600 seconds and a memory limit of 16 GB. Unless stated otherwise, all other parameters of `MibS 1.2.1` were set to their default values.

5.4 Configurations

We compared various configurations using the approach of Algorithm 1 (`useImprovingDirectionOracle` set to `True`) with various configuration of `MibS 1.2.1`. Variations of Algorithm 1 are referred to by names prefixed with `idB&C`. The configurations of `idB&C` differ in the settings for the previously introduced parameters. Because of the numerous policies that have been tested, we group all configurations under the following classes, where "`*`" stands as a placeholder for the actual value of the corresponding parameter:

- `idB&C-MILP`: solves IFD problem exactly as the MILP (ID) using an off-the-shelf solver;
- `idB&C-MILP-k_*`: solves IFD problem (k-ID) exactly for the specified value of k;
- `idB&C-LS-k_*`: employs the local search Algorithm 2 with the specified value of k;
- `idB&C-LS-k_*-dBnd_*_*`: uses the local search Algorithm 2 when the depth of the current node in the tree falls between the specified range. Otherwise, it solves (ID);

This last set of configurations tries to determine whether it is more computationally advantageous to employ heuristics only at the lower levels of the enumeration tree before resorting to solving (ID), or vice versa. Clearly, an exhaustive evaluation of all possible combinations of different ranges and values of k for each configuration would be impractical. Therefore, based on the observations of Section 4.2.2, we limited the range of `maxNeighborhoodSize` to be in $\{2, 3, 4, 5\}$, and considered two settings for the ordered pair `useLocalSearchDepthLb\Ub`: $(0, 10)$ and $(10, +\infty)$. For the case $k = 3$, we also tested additional pairs: $(0, 8)$, $(0, 12)$, $(8, +\infty)$, and $(12, +\infty)$. In total, 21 different `idB&C` configurations were tested.

As previously hinted, for those configurations using heuristics (i.e. all, except `idB&C-MILP`) the solution of (ID) is mandatory if the current solution $(x, y) \in \mathcal{S}$ and no IFD is found by such approximations. This guarantees a correct application of Proposition 1.

The configurations of `MibS` tested are as follows:

- `MibS`: all parameters are set at their default values (the generation of IDICs may be disabled);
- `MibS only IDICs`: *only* the generation of IDICs is enabled and no other classes; IFDs are generated by solving problem (ID) as an MILP using an off-the-shelf solver.
- `MibS IDIC-MILP`: the generation of IDICs is enabled and separation of other classes of cuts is determined automatically by the default mechanism in `MibS`; IFDs are generated by solving problem (ID) as an MILP using an off-the-shelf solver.
- `MibS IDIC-LS-k_*`: the generation of IDICs is enabled and separation of other classes of cuts is determined automatically by the default mechanism in `MibS`; IFDs are generated by first employing Algorithm 2 and then solving problem (ID) as an MILP when no IFD is found using local search.

The branch-and-cut scheme of `MibS only IDICs` differs from `idB&C-MILP` solely in its use of the standard oracle. Therefore, it serves as the most relevant baseline configuration for comparison with `idB&C`. In contrast, due to the use of multiple separation routines and the eventuality of disabling IDIC generation, `MibS` provides a fair comparison mainly with `MibS only IDICs`, `MibS IDIC-MILP`, and `MibS IDIC-LS-k_*`. The specific values of k and the control mechanism of the latter for finding IFDs were selected from the best-performing configuration of `idB&C`.

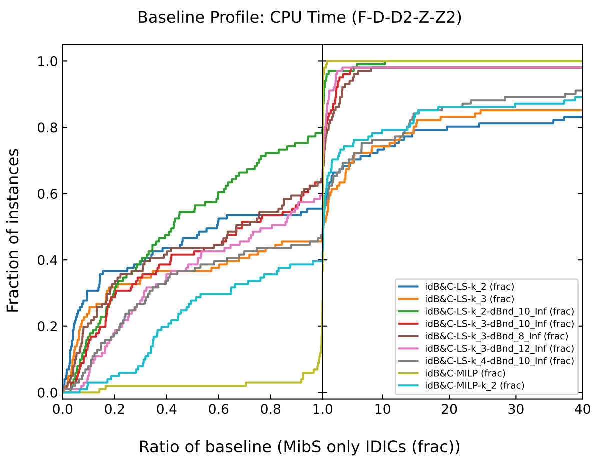
5.5 Results

To summarize the results of the experiment, we present several kinds of plots:

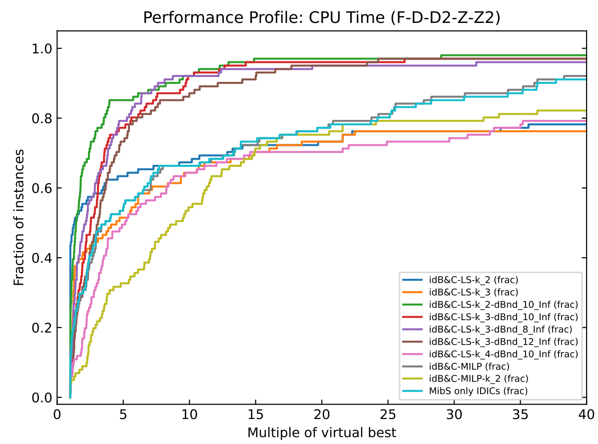
- **Performance profiles** show empirical cumulative distribution functions (CDFs) of ratios of a given performance measure of interest against the "virtual best" Dolan and Moré [2002]. Typical measures are total solution time or the number of nodes in the search tree. These plots are useful to compare the performance of several configurations on the given measure.
- **Baseline profiles** mimics the previous profiles, but present empirical CDFs of ratios of a given performance measure against the performance of a "baseline", i.e. a solver or a specific configuration of it, rather than against the virtual best. These plots show more distinctly the fraction of instances in which a certain configuration outperforms (on the left side) or underperforms (on the right side) the baseline.

- **Cumulative profiles** combine two profiles. On the left side, they show the empirical CDFs of the fraction of instances solved within the time limit. On the right side, they report the empirical CDFs of the fraction of instances that closed a certain final gap within the time limit. Note that the lines on the two sides always connect, since the fraction of instances solved within the time limit equals that of the instances with zero gap at the time limit.

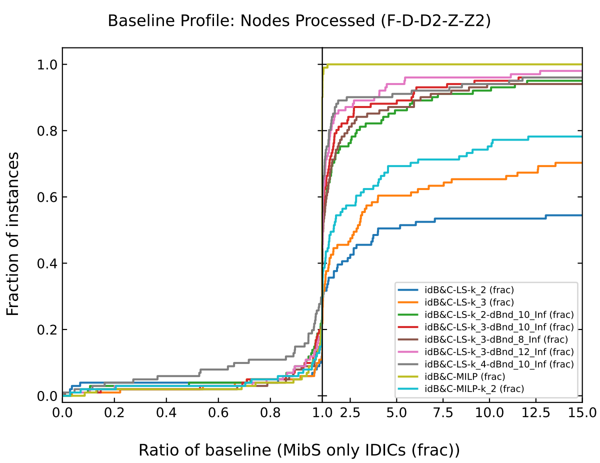
In order to create each plot on (a subset of) the dataset, we first solved all instances with all considered configurations. Instances are then excluded from the plots if (i) they were not solved within the time limit by any of the configurations (except for cumulative profiles) and/or (ii) the solution time is less than 5 seconds for all methods. Note that showing numerous configurations in a single plot may have a negative impact on its readability. For this reason, each profile will plot only the most promising results (highest performing configurations). Those not shown can be considered less effective.



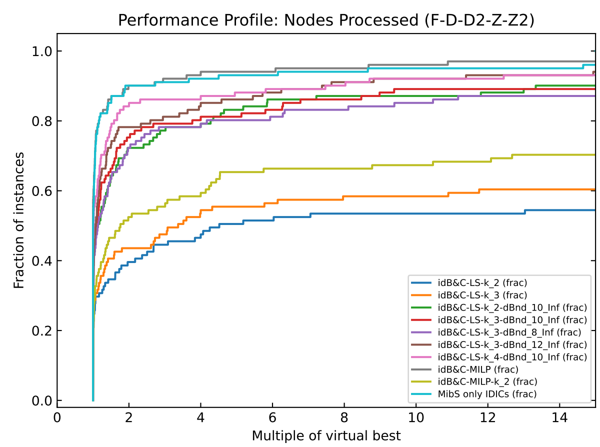
(a) Baseline Profile Solution time



(b) Performance Profile Solution time

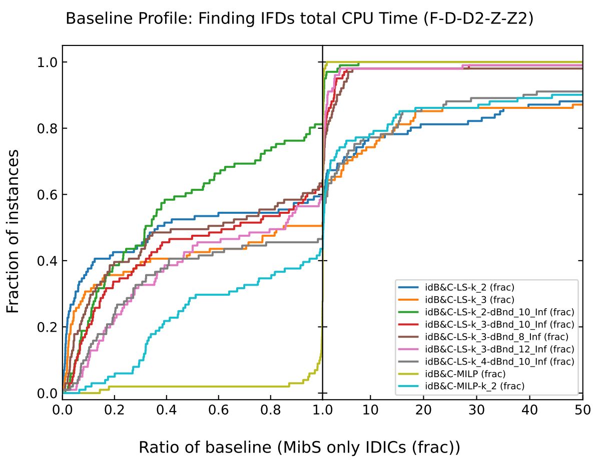


(c) Baseline Profile nodes in the search tree

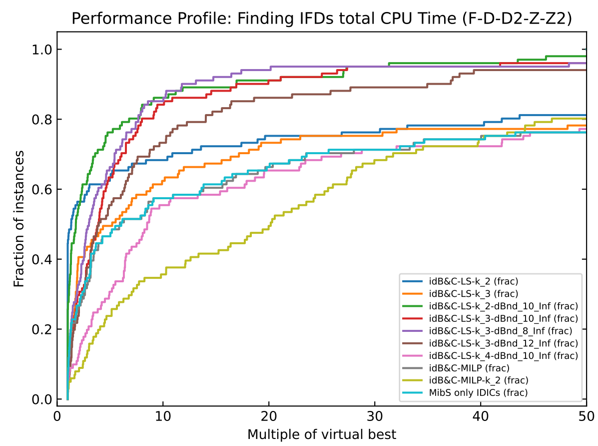


(d) Performance Profile nodes in the search tree

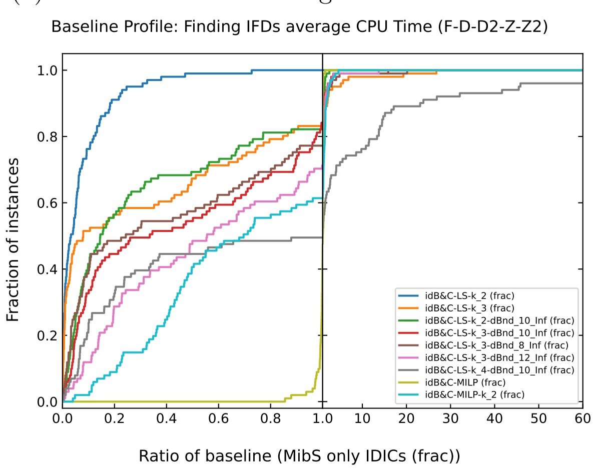
Figure 8: Comparing `idB&C` solution time and nodes in the search tree on IBLPs datasets



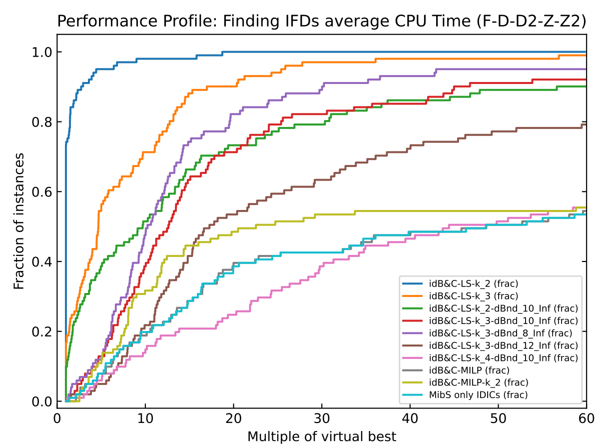
(a) Baseline Profile Finding IFDs Tot. CPU time



(b) Performance Profile Finding IFDs Tot. CPU time



(c) Baseline Profile Finding IFDs Avg. CPU time



(d) Performance Profile Finding IFDs Avg. CPU time

Figure 9: Comparing `idB&C` times for finding IFDs on IBLPs datasets

5.5.1 Improving Direction Oracle

Figures 8–11 show the results of comparisons between `idB&C` in various configurations with the `only IDICs` configuration of `MibS`. This comparison provides the best apples-to-apples comparison of `MibS` and the algorithmic framework proposed here, holding as many things constant as possible. We display baseline and performance profiles using solution CPU time, number of nodes in the search tree, total and average CPU time spent in finding IFDs as measures on all IBLP and Interdiction datasets, comparing all `idB&C` configurations against `MibS only IDICs` (used as the baseline).

The first remarkable result deduced from this experiment is that `idB&C-MILP` shows very similar performances to `MibS only IDICs` over both classes of instances. Recall that `idB&C` never solves the follower's problem explicitly, but instead exploits only Proposition 1 to detect bilevel feasible solutions. Thus, this change by itself does not yield any improvement, but it also does not degrade performance.

From Figure 8c and 10c, we observe the expected tradeoff between the time spent generating a "good" IFD versus the strength of the resulting inequalities. The results show that employing local search for finding IFDs generally comes at a cost in terms of the strength of the separated IDICs. This is expected when replacing an exact method with a heuristic and the result is an increase in the search tree size. In terms of total CPU time, as well as time spent finding IFDs, however, the trade-off strongly favors the use of local search. As a matter of fact, Figures 8a and 9a (10a and 11a, resp.) indicate that local search results in a significant reduction in both the time spent to identify IFDs and total CPU time on a substantial fraction of IBLP (Interdiction, resp.) instances, across all configuration.

In this regard, the strategy of solving (ID) exactly only in higher levels of the tree is key in balancing the drawbacks due to the employment of a heuristic approach with advantages. Interestingly, the experiment emphasize that the best trade-off is achieved for small values of k, i.e., 2 and 3. This result corroborates the related work of Xueyu et al. [2022].

More specifically, consider the solution CPU time on the IBLP dataset. Figures 8a and 8b show that the configurations `idB&C-LS-k_*-dBnd_*_Inf` are the best-performing settings. While both `idB&C-LS-k_*` and `idB&C-MILP-k_2` reduce the solution time on a non-trivial portion of the dataset compared to the baseline, they also exhibit significant slowdowns on the remaining fraction. In contrast, `idB&C-LS-k_2-dBnd_10_Inf` offers more consistent performance, achieving at least a 20% improvement over the baseline for roughly 75% of instances, which increases to at least 50% for over half the dataset.

An even more prominent result is highlighted in Figure 10a on the Interdiction dataset, where Algorithm 2 is particularly effective in finding IFDs on problem with such structure. The more pronounced gains are again obtained by `idB&C-LS-k_*-dBnd_10_Inf`, for $k \in \{2, 3\}$, showing a significant reduction in solution time over more than 90% of all the dataset, and similar time to the baseline on the remaining fraction.

5.5.2 Using Local Search in MibS

In the next experiment, we compares all previously described configurations of MibS in order to measure the effect of using the local search for finding IFDs, holding other elements of the algorithmic strategy of MibS constant. Given the promising outcome of the previous experiment, the parameter settings and the value of k for the best-performing configuration of idB&C (idB&C-LS-k_2-dBnd_10_Inf) were replicated for the configuration of MibS using local search, i.e., MibS IDIC-LS-k_2.

Figure 12 illustrates baseline, performance and cumulative profiles using CPU time and final gap closed as performance measures, with the default MibS configuration serving as the baseline. The CPU time spent in finding IFDs is excluded since MibS might generate different classes of cuts and IDICs may be disabled.

On the one hand, we note from Figure 12e that MibS is the clear winner on the interdiction dataset, primarily due to the use of cuts specialized for these problems. Nevertheless, Figure 12f shows that MibS IDIC-LS-k_2 is the best performing configuration among all other instances. Figure 12a shows that on IBLPs, MibS IDIC-LS-k_2 outperforms MibS by at least 20% on 60% of instances, with solution time reductions reaching up to 60% on about 30% of the dataset. Importantly, by examining the performance of MibS IDIC-MILP one can see that this improvement is attributed not to the use of IDICs alone but to the application of local search specifically. Furthermore, Figures 12c and 12d indicate that MibS is able to close more gap when local search is used. In particular, MibS IDIC-LS-k_2 is able to solve about 12 additional instances to optimality compared to MibS IDIC-MILP and more than 50 with respect to MibS.

Finally, Figure 13 shows baseline and performance profiles plotting the majority of configurations tested in this analysis. Interestingly, plots 13a and 13b reveal that MibS IDIC-LS-k_2 and idB&C-LS-k_2-dBnd_10_Inf have very similar performance, emphasizing the competitiveness of the oracle based on improving directions.

6 Conclusions

Improving directions are a fundamental and versatile tool in mixed-integer bilevel linear optimization, underpinning branching schemes, the formulation of optimality-based relaxations and the generation of strong valid inequalities, and they have been instrumental in advancing the state-of-the-art solution methods for this class of problems.

In this work, we have taken an important step toward explaining and quantifying how improving directions contribute to restoring the follower's optimality condition, by showing that the convex hulls of the feasible regions arising from the optimality-based hierarchy of relaxations are exactly characterized from valid inequalities stemming from improving directions. Moreover, we have shed light on a new role that improving directions play, as they unify the oracle computations for checking bilevel feasibility and generating strong valid inequalities, a perspective that may lead to substantial improvements in empirical performance, as suggested by our promising experiments with our branch-and-cut framework.

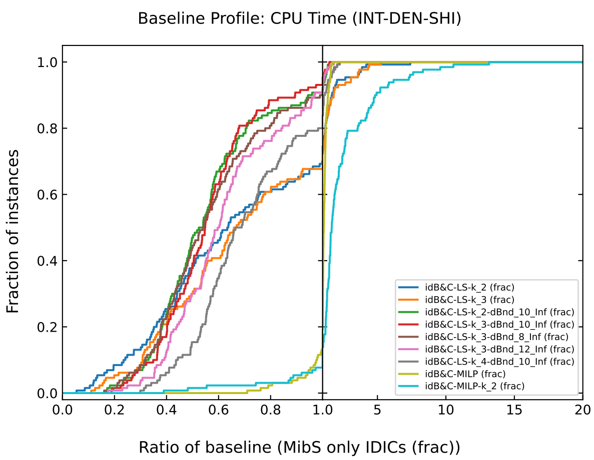
Notably, this novel algorithm is not limited to the generation of intersection cuts discussed here, but can be easily extended to any class of valid inequality for MIBLPs somehow related to the existence of an improving solution or direction. Since our branch-and-cut highlights the centrality of finding an improving direction as NP-complete subproblem, we plan to explore the integration of MILP warm-start capabilities implemented in SYMPHONY [Ralphs et al. (2023)] (and already used in MibS as subsolver), to further enhance performance.

Due to the promising results of the local search shown here, we also plan to develop more refined mechanism for better controlling its dynamic employment and other related enhancements. From a theoretical perspective instead, we will explore connection with more general, yet related, optimality certificates given by test sets for pure integer linear problems as already investigated in, e.g., [Conti and Traverso (1991); Scarf (1997)].

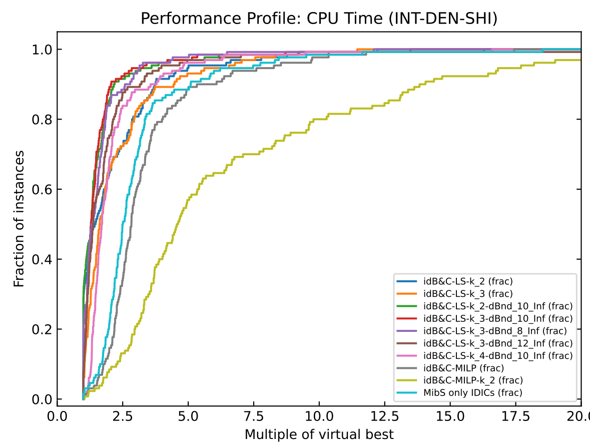
Declarations

Funding This research was made possible with support from Office of Naval Research Grant N000142212676.

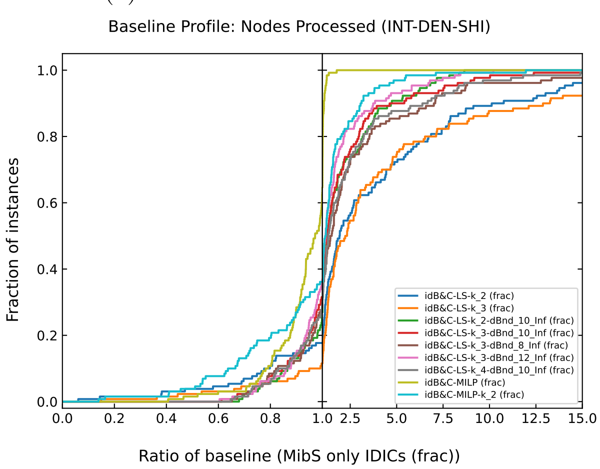
Conflicts of interests Authors have no conflicts of interests to report.



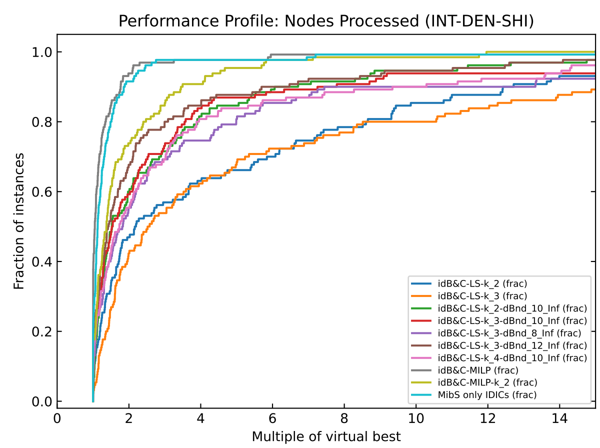
(a) Baseline Profile Solution time



(b) Performance Profile Solution time

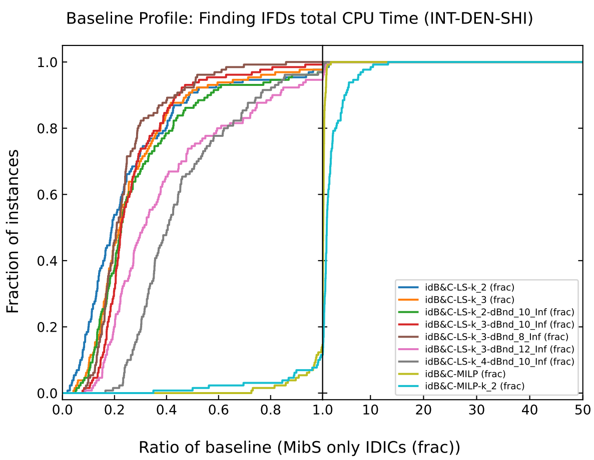


(c) Baseline Profile nodes in the search tree

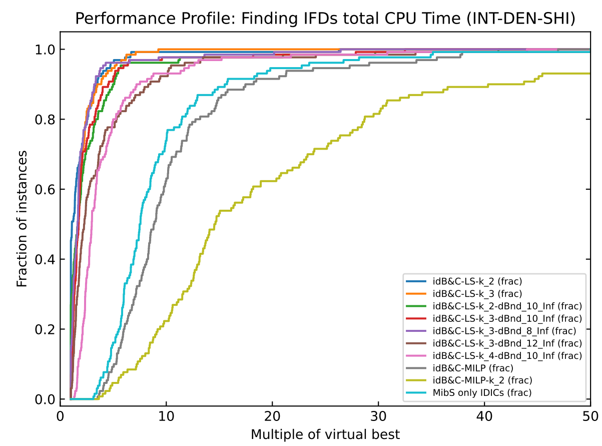


(d) Performance Profile nodes in the search tree

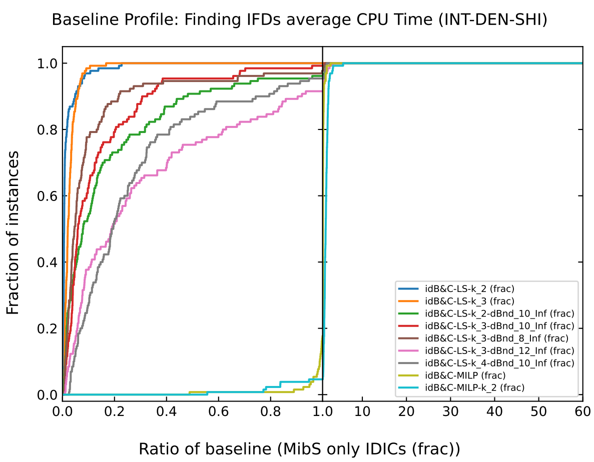
Figure 10: Comparing `idB&C` solution time and nodes in the search tree on interdiction datasets



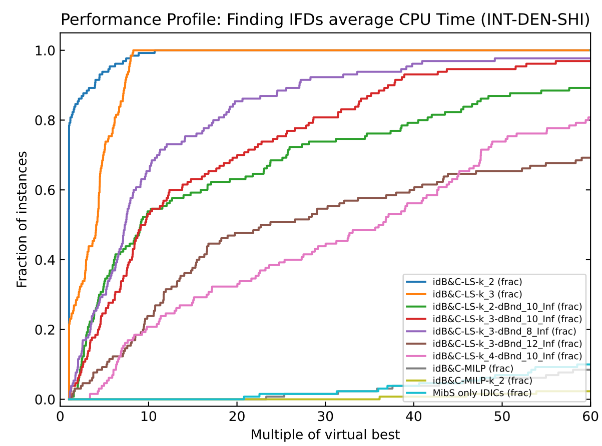
(a) Baseline Profile Finding IFDs Tot. CPU time



(b) Performance Profile Finding IFDs Tot. CPU time

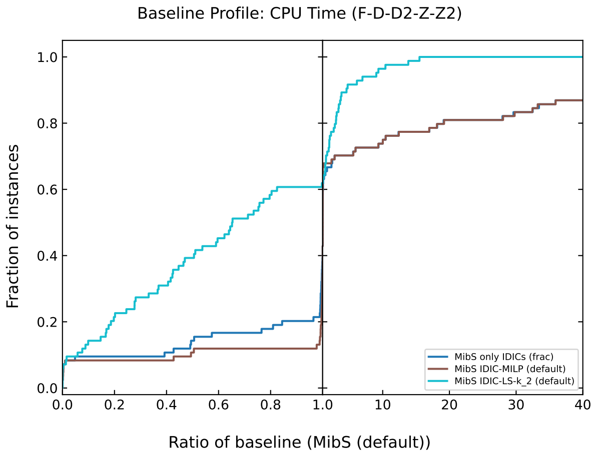


(c) Baseline Profile Finding IFDs Avg. CPU time

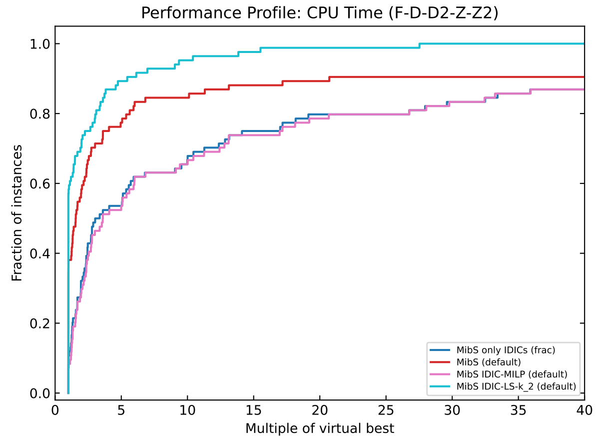


(d) Performance Profile Finding IFDs Avg. CPU time

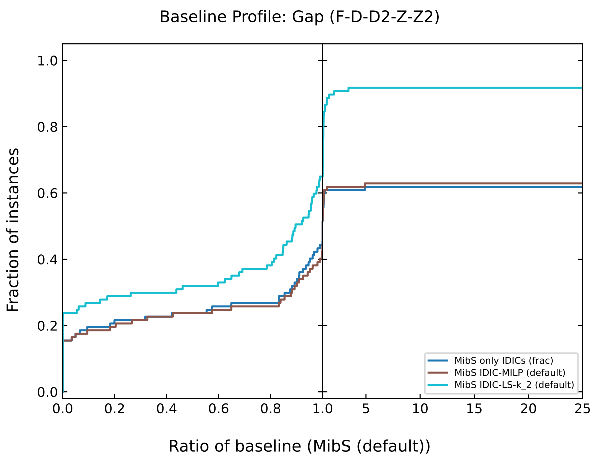
Figure 11: Comparing `idB&C` times for finding IFDs on interdiction datasets



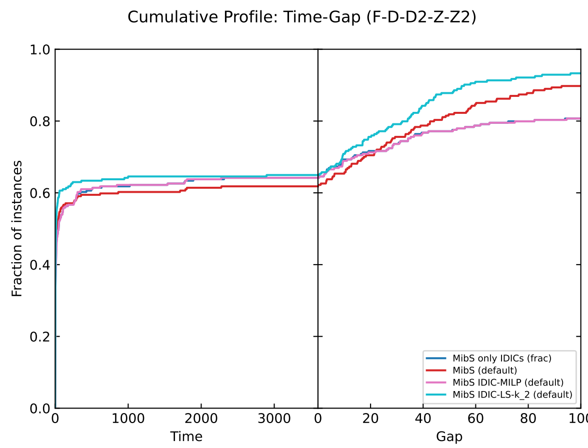
(a) Baseline Profile Solution time



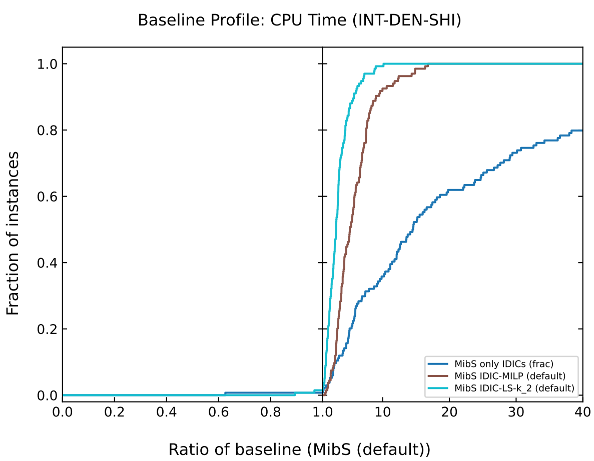
(b) Performance Profile Solution time



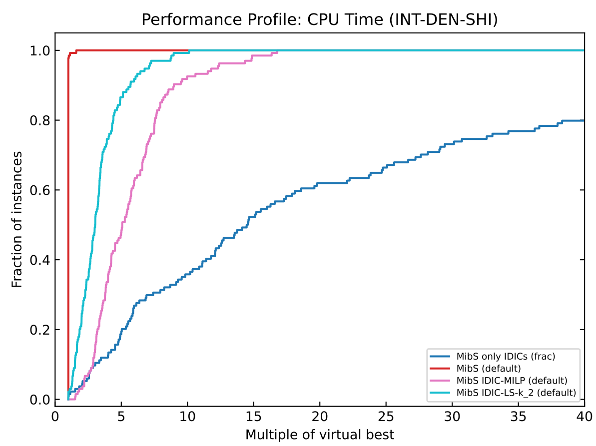
(c) Baseline Profile Gap



(d) Cumulative Profile

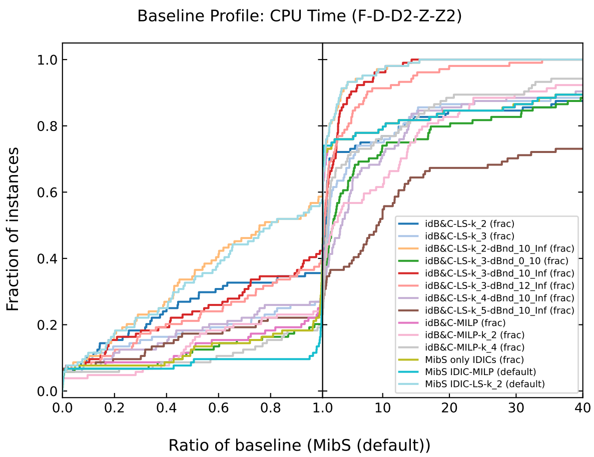


(e) Baseline Profile Solution time

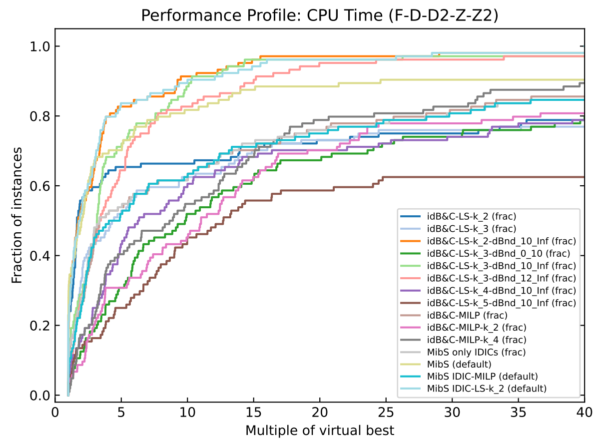


(f) Performance Profile Solution time

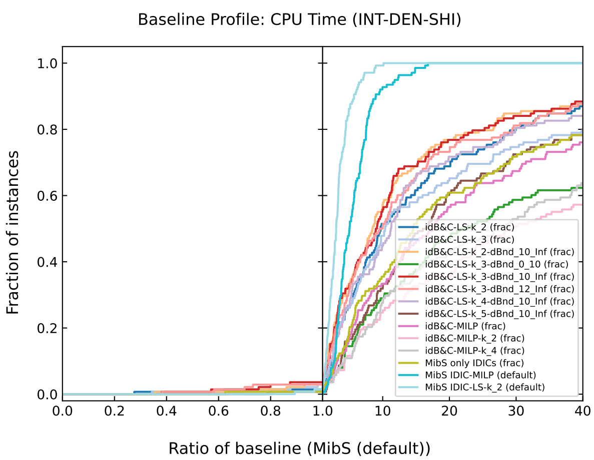
Figure 12: Comparing `MibS` solution times on IBLP and Interdiction datasets



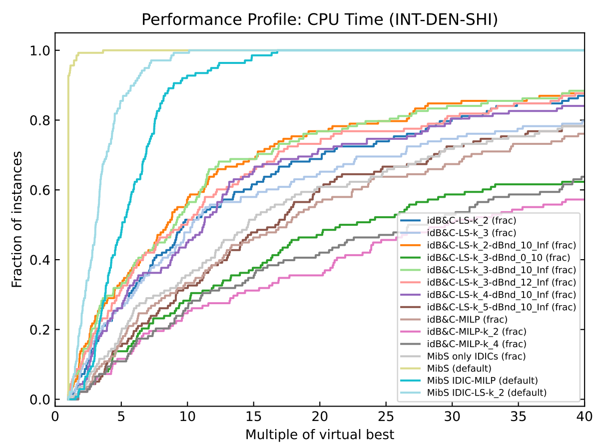
(a) Baseline Profile Solution time



(b) Performance Profile Solution time



(c) Baseline Profile Solution time



(d) Performance Profile Solution time

Figure 13: Comparing all configurations on all the datasets

References

Balas, Egon (1972). "Integer Programming and Convex Analysis: Intersection Cuts from Outer Polars". In: *Mathematical Programming* 2.1, pp. 330–382.

Buchheim, Christoph (2023). "Bilevel Linear Optimization Belongs to NP and Admits Polynomial-Size KKT-based Reformulations". In: *Operations Research Letters* 51.6, pp. 618–622. ISSN: 0167-6377. DOI: 10.1016/j.orl.2023.10.006. (Visited on 11/16/2024).

COIN-OR (2018). *Computational Infrastructure for Operations Research.*

Conti, Pasqualina and Carlo Traverso (1991). "Buchberger Algorithm and Integer Programming". In: *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*. Springer, pp. 130–139.

Cornuéjols, Gérard (Mar. 2008). "Valid Inequalities for Mixed Integer Linear Programs". In: *Mathematical Programming* 112.1, pp. 3–44. ISSN: 1436-4646. DOI: 10.1007/s10107-006-0086-0. (Visited on 07/08/2025).

Dempe, Stephan (2002). *Foundations of Bilevel Programming.* Springer Science & Business Media.

Dempe, Stephan and Alain Zemkoho (2020). "Bilevel Optimization". In: *Springer Optimization and Its Applications*. Vol. 161. Springer.

DeNegre, S.T., T.K. Ralphs, and S. Tahernejad (2024). "MibS Version 1.2". In: DOI: 10.5281/zenodo.1439384. URL: https://github.com/coin-or/MibS.

DeNegre, Scott (2011). *Interdiction and Discrete Bilevel Linear Programming.* Lehigh University.

Dolan, Elizabeth D and Jorge J Moré (2002). "Benchmarking Optimization Software with Performance Profiles". In: *Mathematical programming* 91, pp. 201–213.

Fischetti, Matteo, Ivana Ljubić, Michele Monaci, and Markus Sinnl (2017). "A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs". In: *Operations Research* 65.6, pp. 1615–1637.

— (2018). "On the Use of Intersection Cuts for Bilevel Optimization". In: *Mathematical Programming* 172.1-2, pp. 77–103.

Fischetti, Matteo and Andrea Lodi (2008). "Repairing MIP Infeasibility through Local Branching". In: *Computers & operations research* 35.5, pp. 1436–1445.

Gaar, Elisabeth, Jon Lee, Ivana Ljubić, Markus Sinnl, and Kübra Tanınmış (July 2024). "On SOCP-based Disjunctive Cuts for Solving a Class of Integer Bilevel Nonlinear Programs". In: *Mathematical Programming* 206.1, pp. 91–124. ISSN: 1436-4646. DOI: 10.1007/s10107-023-01965-1. (Visited on 06/02/2025).

Gleixner, Ambros, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp Christophel, Kati Jarck, Thorsten Koch, Jeffrey Linderoth, Marco Luebbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Yuji Shinano, and Domenico Salvagnin (2021). "MIPLIB 2017: Data-driven Compilation of the 6th Mixed-Integer Programming Library". In: *Mathematical Programming Computation* 13, pp. 443–490. DOI: 10.1007/s12532-020-00194-3. URL: http://www.optimization-online.org/DB_FILE/2019/07/7285.pdf.

Hansen, Pierre, Brigitte Jaumard, and Gilles Savard (1992). "New Branch-and-Bound Rules for Linear Bilevel Programming". In: *SIAM Journal on scientific and Statistical Computing* 13.5, pp. 1194–1217.

Jeroslow, Robert G. (June 1985). "The Polynomial Hierarchy and a Simple Model for Competitive Analysis". In: *Mathematical Programming* 32.2, pp. 146–164. ISSN: 1436-4646. DOI: 10.1007/BF01586088. (Visited on 11/16/2024).

Moore, James T. and Jonathan F. Bard (1990). "The Mixed Integer Linear Bilevel Programming Problem". In: *Operations research* 38.5, pp. 911–921.

Ralphs, T.K., M. Güzelsoy, and A. Mahajan (2023). "SYMPHONY Version 5.7". In: DOI: 10.5281/zenodo.7926908. URL: https://github.com/coin-or/SYMPHONY/.

Scarf, Herbert E (1997). "Test Sets for Integer Programs". In: *Mathematical Programming* 79, pp. 355–368.

Schulz, Andreas S. (2009). "On the Relative Complexity of 15 Problems Related to 0/1-Integer Programming". In: *Research Trends in Combinatorial Optimization*. Ed. by William Cook, László Lovász, and Jens Vygen. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 399–428. ISBN: 978-3-540-76795-4 978-3-540-76796-1. DOI: 10.1007/978-3-540-76796-1_19. (Visited on 11/16/2024).

Stockmeyer, Larry J (1976). "The Polynomial-Time Hierarchy". In: *Theoretical Computer Science* 3.1, pp. 1–22.

Tahernejad, S. and T.K. Ralphs (2025). *Valid Inequalities for Mixed Integer Bilevel Linear Optimization Problems*. Technical Report 20T-013. COR@L Laboratory, Lehigh University. URL: https://arxiv.org/abs/2510.02998.

Tahernejad, S., T.K. Ralphs, and S.T. DeNegre (2020). "A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation". In: *Mathematical Programming Computation* 12, pp. 529–568. DOI: 10.1007/s12532-020-00183-6. URL: https://arxiv.org/abs/2104.09010.

Thürauf, J., T. Kleinert, I. Ljubić, T.K. Ralphs, and M. Schmidt (2025). "BOBILib: Bilevel Optimization (Benchmark) Instance Library". In: *Mathematical Programming Computation (Accepted for Publication)*. URL: https://optimization-online.org/2024/07/bobilib-bilevel-optimization-benchmark-instance-library/.

Vicente, Luis Nunes, Gilles Savard, and Joaquim J. Judice (1996). "Discrete Linear Bilevel Programming Problem". In: *Journal of optimization theory and applications* 89, pp. 597–614.

Wang, Lizhi and Pan Xu (2017). "The Watermelon Algorithm for the Bilevel Integer Linear Programming Problem". In: *SIAM Journal on Optimization* 27.3, pp. 1403–1430.

Xie, Y., T.K. Ralphs, and O.A. Prokopyev (2025). *A Branch-and-Cut Algorithm for Mixed Integer Bilevel Optimization with Bounded Rationality*. Working Paper. COR@L Laboratory, Lehigh University.

Xueyu, Shi, Oleg A. Prokopyev, and Ted K. Ralphs (2022). "Mixed Integer Bilevel Optimization with a K-Optimal Follower: A Hierarchy of Bounds". In: *Mathematical Programming Computation* 15, pp. 1–51. DOI: 10.1007/s12532-022-00227-z. URL: http://coral.ie.lehigh.edu/~ted/files/papers/HierarchyOfBounds20.pdf.

Zare, M Hosein, Juan S Borrero, Bo Zeng, and Oleg A Prokopyev (2019). "A Note on Linearized Reformulations for a Class of Bilevel Linear Integer Problems". In: *Annals of Operations Research* 272, pp. 99–117.

Zhang, Junlong and Osman Y Ozaltın (2017). "A Branch-and-Cut Algorithm for Discrete Bilevel Linear Programs". In: *Optimization Online*.