INDUSTRIAL AND SYSTEMS ENGINEERING

BOBILib: Bilevel Optimization (Benchmark) Instance Library

Johannes Thürauf

Department Liberal Arts and Social Sciences, University of Technology Nuremberg (UTN)

Thomas Kleinert

Quantagonia München, Germany

Ivana Ljubić

ESSEC Business School

Ted K. Ralphs

Department of Industrial and Systems Engineering, Lehigh University

Martin Schmidt

Department of Mathematics, Trier University

COR@L Technical Report 25T-027

LEHIGH UNIVERSITY

COR@L COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

BOBILib: Bilevel Optimization (Benchmark) Instance Library

JOHANNES THÜRAUF *[1], THOMAS KLEINERT †[2], IVANA LJUBIĆ ‡[3], TED K. RALPHS §[4], AND MARTIN SCHMIDT ¶[5]

[1]Department Liberal Arts and Social Sciences, University of Technology Nuremberg (UTN)
[2] Quantagonia München, Germany
[3]ESSEC Business School
[4] Department of Industrial and Systems Engineering, Lehigh University
[5] Department of Mathematics, Trier University

October 22, 2025

Abstract

In this report, we present the BOBILib, a collection of more than 2600 instances of mixed integer bilevel linear optimization problems (MIBLPs). The goal of this library is to provide a large and well-curated set of test instances freely available for the research community so that new and existing algorithms in bilevel optimization can be tested and compared in a standardized way. The library is sub-divided into instances of different types and also contains different benchmark instance sets. Moreover, we present a new data format for MIBLPs that is less error-prone compared to an older format that will now be deprecated. We provide numerical results for all instances of the library using available bilevel solvers. Based on these numerical results, we select benchmark instance sets, which provide a meaningful basis for experimental comparisons of solution methods in a moderate time. The instances, together with solution files, can be downloaded at https://bobilib.org.

Keywords: Mixed integer bilevel linear optimization, Benchmarking, Instance library, Problem instances, Computational optimization

*johannes.thuerauf@utn.de
†thomas.kleinert@quantagonia.com
‡ljubic@essec.edu
§ted@lehigh.edu
¶martin.schmidt@uni-trier.de

1 Introduction

Computational mixed integer bilevel optimization is a rather young field of research. The first computational studies using small continuous linear-linear or linear-quadratic bilevel optimization problems were conducted in the 1980s; see, e.g., Fortuny-Amat and McCarl (1981), Bialas and Karwan (1984), or Bard and Moore (1990). A short time later, the first branch-and-bound algorithm for mixed integer bilevel linear optimization problems (MIBLPs) was proposed by Moore and Bard (1990). However, all computational tests were conducted on very few and very small academic instances. After almost 20 years without much computational progress, DeNegre and Ralphs (2009) published the first general-purpose branch-and-cut algorithm for pure integer linear bilevel optimization problems, which can be seen as an extension of the work by Moore and Bard (1990). Moreover, in his dissertation, DeNegre (2011) conducted a more detailed computational analysis of the proposed branch-and-cut algorithm. These works can be seen as tipping points in the history of computational (integer) bilevel optimization. Since then, a number of works have appeared that tackle MIBLPs; see the survey by Kleinert et al. (2021) for a rather comprehensive overview of approaches.

In the last decade, computational research on several special classes of bilevel problems became increasingly popular. However, the community is still suffering from a lack of well-curated and actively used instance libraries for testing new methods and for comparing existing algorithms. Such libraries are rather standard in more established fields of computational optimization; see, e.g., MIPLIB (Bixby et al. 1998; Koch et al. 2011; Gleixner et al. 2021) for mixed integer linear optimization, QPLIB (Furini et al. 2019) for quadratic optimization, MINLPLIB (MINLPLib 2022) for mixed integer nonlinear optimization, or GLOBALLib (Floudas et al. 1999) for global optimization (integrated into MINLPLIB in the last years).

In the much younger field of bilevel optimization, first attempts have already been made to publish curated instance libraries. There is BASBLib (Paulavicius and Adjiman 2017), see also `https://github.com/basblsolver/BASBLib` on GitHub, as well as BOLIB (Zhou et al. 2018), see also `https://biopt.github.io/bolib`. Both collections contain only continuous bilevel instances and both are coupled to proprietary software, since BASBLib is based on AMPL and BOLIB is based on Matlab. Moreover, the solver package published by Jungen et al. (2023) also contains a rather large set of test instances that are available at `https://git.rwth-aachen.de/avt-svt/public/libdips`.

In addition to these existing libraries, there are the test problem generator for linear (and continuous) bilevel optimization problems presented in Moshirvaziri et al. (1996); the test problem generator for quadratic and linear-quadratic bilevel optimization problems discussed in Calamai and Vicente (1993) and Calamai and Vicente (1994); and the generator introduced in DeNegre (2011) and discussed further in Section 3.2.2, which was used to generate some of the instances in the BOBILib.

First attempts for collections of mixed integer bilevel instances have been made by Ralphs (2016) and Sinnl (2020). With this report and the associated library, our aim is to streamline and extend the existing attempts to collect and curate (mixed integer) bilevel optimization instances and provide a detailed documentation of many bilevel instances from the literature. All instances are publicly available for download at

https://bobilib.org.

This library contains more than 2600 instances together with detailed statistics on the instances, as well as (best known) solutions for those instances for which we were able to compute a feasible or provably optimal solution using available solvers for MIBLPs.

The remainder of this report is structured as follows. In Section 2, we provide some basics on bilevel optimization and fix some notation. In Section 3, we describe several classes of instances that are part of the BOBILib. The data format of all the instances is introduced and explained in Section 4, where we also present a new format for solution files. Afterward, we briefly present numerical results for all instances of the BOBILib in Section 5. For compiling these results, we use two solvers for MIBLPs that are available today. The computational details are discussed in Section 5.1. In Section 5.2, we discuss the implementation of the necessary feasibility checks before we briefly present the numerical results for the overall collection in Section 5.3. In particular, based on these numerical results, we also provide different benchmark instance sets that we describe in Section 5.4. We present a separate benchmark set for interdiction problems as well as for mixed and pure integer problems. In Section 6, we give a short overview of the content of the companion website, before we discuss our future plans for the library in Section 7. Furthermore, we provide detailed statistics about each instance set in the appendix.

2 Bilevel Optimization in a Nutshell

We consider MIBLPs of the form

$$\min_{x,y} \quad c_u^\top x + d_u^\top y \tag{MIBLP-a}$$

$$\text{s.t.} \quad Ax + By \geq a, \tag{MIBLP-b}$$

$$x_i \in \mathbb{Z} \cap [x_i^-, x_i^+] \quad \text{for all } i \in I_u \subseteq \{1, \dots, n_x\}, \tag{MIBLP-c}$$

$$y \in S(x), \tag{MIBLP-d}$$

where for a fixed x, $S(x)$ is the set of optimal solutions to the problem

$$\min_{y} \quad d_l^\top y \tag{LL-a}$$

$$\text{s.t.} \quad Cx + Dy \geq b, \tag{LL-b}$$

$$y_i \in \mathbb{Z} \cap [y_i^-, y_i^+] \quad \text{for all } i \in I_l \subseteq \{1, \dots, n_y\}. \tag{LL-c}$$

Here, the problem data is given by the objective vectors $c_u \in \mathbb{Q}^{n_x}$, $d_u, d_l \in \mathbb{Q}^{n_y}$; the matrices $A \in \mathbb{Q}^{m_u \times n_x}$, $B \in \mathbb{Q}^{m_u \times n_y}$, $C \in \mathbb{Q}^{m_l \times n_x}$, $D \in \mathbb{Q}^{m_l \times n_y}$; and the right-hand side vectors $a \in \mathbb{Q}^{m_u}$ and $b \in \mathbb{Q}^{m_l}$.

Table 1: Overview of MILP-MILP instance classes w.r.t. number of variables and constraints (B = binary, I = integer, MI = mixed integer).

	Total	UL Variables			LL Variables			UL Constraints		LL Constraints	
		Min	Max	Type	Min	Max	Type	Min	Max	Min	Max
interdiction											
assignment	24	25	25	B	25	25	B	1	1	45	45
edge clique	220	19	1593	B	8	1653	B	1	1	28	3363
generalized	90	40	50	B	40	50	MI	20	20	30	50
knapsack	599	10	500	B	10	500	I	1	1	11	501
multidimensional-knapsack	954	10	500	B	10	500	B	1	29	11	529
network	72	22	79	B	44	158	B	1	1	41	974
general-bilevel											
mixed integer	336	10	399 808	MI	10	399 808	MI	0	480 585	4	961 170
pure integer	359	1	714 549	I	1	714 549	I	0	84 788	3	169 576

Problem (1) is called the *upper-level* or *leader's* problem, whereas Problem (2) is called the *lower-level* or *follower's* problem with respect to a given x. We consider bilevel problems in which there may be both integer and continuous variables at both levels. For discussing the instances in our test set, we make use of the following two notions. First, we call upper-level constraints in (MIBLP-b) *coupling constraints* if they involve lower-level variables y. Second, upper-level variables that appear in the lower-level constraints (LL-b) are called *linking variables* and the respective lower-level constraints in which they appear are called *linking constraints*.

We note that in the typical case in which there are alternative optimal solutions to (2) for a given x ($|S(x)| > 1$), different assumptions can be made regarding the follower's behavior in choosing a member of $S(x)$. By putting both x and y below the "min" in (1), we are implicitly assuming that the follower selects among alternative optima one that is of greatest benefit to the leader. This is the so-called *optimistic* version of the bilevel problem. The primary alternative is the *pessimistic* bilevel problem (see, e.g., the seminal textbook by Dempe (2002) for more details), where the assumption is the opposite: the follower chooses among the alternatives the one that is worst for the leader. Although the instances in the library are appropriate for benchmarking solution methods for either of these cases, we only consider the optimistic case in the analysis in this paper. Moreover, the library only contains instances that are deterministic, i.e., all problem data is certain.

Finally, we refer to Moore and Bard (1990), Vicente et al. (1996), and Köppe et al. (2010) for the study of existence of solutions, as well to Hansen et al. (1992) and Jeroslow (1985) for studies on the formal worst-case computational complexity of MIBLPs.

3 Description of the Instances

In this section, we describe the two main classes of MIBLPs that currently appear in the library: (mixed integer) interdiction problems (of which there are a number of subclasses) and general mixed integer instances, which are further subdivided into the mixed and pure integer cases. Table 1 presents a general overview. Note that in some places below, we refer to existing sets of instances that have previously appeared in the literature using the tag by which the set is referred to on the BOBILib website, described in Section 6.

3.1 Interdiction Instances

We start by discussing the interdiction instances; see, e.g., Kleinert et al. (2021) for a general discussion of this problem class.

3.1.1 Assignment Interdiction

This problem class contains instances in which the goal of the upper-level decision maker is to maximize the minimum cost achievable by the lower-level player by fixing a subset of the lower-level variables to zero. Each interdiction decision is associated with a cost. The upper level contains a single knapsack constraint that represents the interdiction budget. The lower-level problem is an assignment problem. DeNegre (2011) generated 25 instances from bicriteria assignment problems contained in the *Multiple Criteria Decision Making Numerical Instances Library* by Figueira, which, unfortunately, is no longer available on the web. The first objective function of the original problem is used as the lower-level objective function and the second objective function serves as the budget constraint of the upper level. Each instance consists of $n_x = n_y = 25$ upper- and lower-level variables, a single upper-level interdiction constraint, and 45 lower-level inequality constraints. From the 25 original instances, we excluded the invalid instance 2AP05-12, which results in 24 instances in the set called inter-assig.

3.1.2 Edge Clique Interdiction

In these instances, the follower solves a maximum cardinality clique problem on an undirected graph $G = (V, E)$ with the set of nodes V and the set of edges E. The leader can interdict (i.e., remove) at most k edges from the graph G with the goal to minimize the size of the maximum clique in the remaining graph. Tang et al. (2015) introduced instances using graphs with $|V| \in \{8, 10, 12, 15\}$ and density $d \in \{0.7, 0.9\}$, which leads to $|E| = \lfloor d|V|(|V| - 1)/2 \rfloor$ many edges. Every potential edge has the same probability of being created and the interdiction budget is set to $k = \lceil |E|/4 \rceil$. For each of these parameter configurations, 10 instances have been created resulting in a total of 80 instances in the set bcpins. For these instances, Fischetti et al. (2018b) consider an extended formulation of the lower-level problem with an additional family of valid inequalities that strengthen the LP relaxation of the lower-level problem. These new instances are collected in the set plusbcpins with again 80 instances. In the same way, Fischetti et al. (2018b) generated 60 larger instances with $|V| \in \{40, 50, 60\}$ but without the additional constraints; see the instance set clique. Due to the problem structure, every instance has a single upper-level constraint and several lower-level constraints. In addition, all upper- and lower-level variables are binary variables.

3.1.3 Generalized Interdiction

Fischetti et al. (2018b) introduced randomly generated generalized interdiction instances. Here, the upper-level player can interdict certain non-negative variables of the lower-level player by setting their upper bound to zero. The instances are constructed in the following way. We start with a first set of upper-level variables x and lower-level variables y that are binary. All coefficients of the objective functions and constraints are taken uniformly random as integers in $[-50, 50]$. Every instance has 20 upper-level constraints on x and y and 20 lower-level constraints on y, all in $\leq$ form. The right-hand side of each constraint is given by $\lfloor \frac{\alpha}{100} \Sigma \rfloor$, where α is an integer taken uniformly random in $[25, 75]$ and Σ is either the sum of all positive or negative coefficients of the currently considered row, both with 50 % probability. In addition, for every upper-level constraint additional upper-level binary slack variables and for every lower-level constraint additional lower-level continuous slack variables are introduced to avoid feasibility problems. Hence, the upper-level variables are all binary, whereas the lower-level problem is a mixed integer optimization problem. Finally, the lower level has N interdiction constraints. Each such interdiction constraint may set the upper bound of a lower-level variable y to zero via a binary variable x of the upper level. Fischetti et al. (2018b) generated 10 instances for every feasible combination of $n_x \in \{20, 30\}$, $n_y \in \{20, 30\}$, and $N \in \{10, 20, 30\}$. Note that $N \leq \min\{n_x, n_y\}$ must hold. In addition to the N interdiction constraints, the upper and lower level contain 20 more constraints each. In total, 90 instances are generated and are summarized in the instance set generalized.

3.1.4 Knapsack Interdiction

There are several sets of interdiction instances, in which the follower has a single knapsack constraint, or in which both, the leader and the follower may have multiple knapsack constraints.

DeNegre (2011) introduced such instances based on bicriteria instances of the *Multiple Criteria Decision Making Numerical Instances Library* by Figueira. The first objective of the bicriteria problem is used to define the lower-level objective function, while the second objective defines the left-hand side of the interdiction budget constraint of the upper level. The interdiction budget equals $\lceil \sum_{i=1}^{n_y} a_i/2 \rceil$, where a_i denotes the costs of interdicting the ith lower-level variable. The instances have $n_x = n_y = n \in \{10, 20, 30, 40, 50\}$ variables and $n + 2$ constraints. There are 20 instances per size n, which makes a total of 100 instances. In the library, 99 of these 100 instances are included in the set inter-kp since we excluded the instance K5010W01 that is not available for us in the used file format.

Caprara et al. (2016) introduced knapsack interdiction instances for which the follower's problem is generated by the knapsack generator of Martello et al. (1999). The profits and weights of the knapsack are taken from the interval $[0, 100]$ in an uncorrelated way. For each number of items $n \in \{35, 40, 45, 50, 55\}$, the authors generated 10 instances;

see the instance set cclw. In the same way, Fischetti et al. (2018b) generated 90 instances for each number of items $n \in \{100, 200, 300, 400, 500\}$ giving a total of 450 instances, which are all part of the instance set kp. Thus, we do not exclude 9 "trivial" instances as in Fischetti et al. (2018b).

3.1.5 Multidimensional Knapsack Interdiction

Fischetti et al. (2019) introduced multidimensional knapsack interdiction instances that are based on the SAC-94 library that contains multidimensional knapsack instances; see Khuri et al. (1995). The original instances have 2 to 30 knapsack constraints and 10 to 90 items. Thus, in the converted interdiction instances, the dimensions $n_x = n_y$ match the number of items. For each item, there is a lower-level interdiction-type constraint with which the upper-level player can set a lower-level variable to zero. For each instance, the knapsack constraints are distributed in three different ways: (i) the first knapsack constraint as an upper-level constraint and the remaining knapsack constraints as follower constraints, (ii) the first 50 % of the constraints (rounded up) as upper-level constraints and the remaining ones as lower-level constraints, and (iii) all but the last constraint as upper-level constraints. In the cases (i) and (ii), the follower problem is a multidimensional knapsack problem, while in case of (iii) the lower level has a single knapsack constraint. When the underlying multidimensional knapsack instance has just two constraints, all three transformations give the same instance with one leader and one follower constraint. This results in 45 instances of type (i), 54 instances of type (ii), and 45 instances of type (iii), i.e., we have a total of 144 instances in the set imkp.

The same authors also use multidimensional knapsack instances from Chu and Beasley (1998) to construct interdiction variants thereof. The original single-level instances have $n \in \{100, 250, 500\}$ variables per level. In addition, there are $m \in \{5, 10, 30\}$ knapsack constraints. The coefficients for the knapsack constraints were drawn uniformly from $[0, 1000]$. The right-hand side of each knapsack constraint was computed as the sum of all coefficients weighted by $\alpha \in \{0.25, 0.5, 0.75\}$. For each combination of n, m, and α, Chu and Beasley (1998) created 10 instances, which makes 270 instances. Fischetti et al. (2019) converted these instances to interdiction problems by adding $n_x = n$ upper-level variables that can set the $n_y = n$ lower-level variables to zero. This results in n lower-level interdiction constraints. In addition, the m knapsack constraints are distributed in three different ways: (i) all but one of the m knapsack constraint belong to the upper level, (ii) the knapsack constraints are distributed equally among the two levels (rounded up in favor of the upper level), and (iii) all but one knapsack constraints belong to the lower level. This makes a total of 810 interdiction instances in the set or.

3.1.6 Network Interdiction

Baggio et al. (2021) propose 72 randomly generated instances arising from a trilevel context, in which one has to defend a given network against possible attacks. More precisely, the original trilevel instances are of max-min-max form and consider a defender-attacker-defender structure. The defender is not only able to adopt preventive strategies but also to defend the network after an attack takes place. The networks considered in the instances are trees and general graphs with $n \in \{25, 50, 80\}$ nodes. The instances that we consider for this library are the second- and third-level problems, i.e., the lower-bilevel problems of the original trilevel problems as they are used in Fischetti et al. (2017). These instance are collected in the set inter-fire.

3.2 General Bilevel Instances

3.2.1 Mixed Integer Instances

We consider an instance to be mixed integer if there is at least one continuous and one integer variable present at either level. Xu and Wang (2014) proposed a set of MIBLPs with $n_x = n_y = n \in \{10, 60, 110, 160, 210, 260, 310, 360, 410, 460\}$ variables. The upper-level variables are constrained to be integer but some of the lower-level variables are continuous. The number of upper-level as well as lower-level constraints is $0.4n$. All matrices, vectors, right-hand sides, etc. are uniformly distributed integers. The constraint matrices A, B, C, and D have entries in $[0, 10]$, the objective vectors c_u, d_u, and d_l have entries in $[-50, 50]$, the upper-level right-hand side vector a has entries in $[30, 130]$, and the lower-level right-hand side vector b has entries in $[10, 110]$. There are 10 instances for every value of n, which gives a total of 100 instances in the set xuwang. Fischetti et al. (2017) used the exact same procedure to construct larger instances of the same structure. For each size $n_x = n_y = n \in \{500, 600, 700, 800, 900, 1000\}$, the authors generated 10 instances, which gives a total of 60 additional instances. The latter can be found in the instance set xularge.

Kleinert and Schmidt (2021) converted the 87 benchmark instances of MIPLIB2010 (Koch et al. 2011) and the 240 benchmark instances of MIPLIB2017 (Gleixner et al. 2021) into MIBLPs. For every single-level instance, three bilevel variants have been generated: (i) all constraints belong to the upper level, (ii) the first 50 % (rounded up) of the constraints belong to the upper level, the rest to the lower level, and (iii) all constraints belong to the lower level. In all three variants, the first 50 % of the variables (rounded up) are upper-level variables. Further, the original objective function is the upper-level objective function, while the lower-level objective function is set to the original objective function multiplied by -1. From these 261 instances belonging to MIPLIB2010 and 720 instances belonging to MIPLIB2017, we excluded all instances that are based on infeasible original instances, that contain range constraints, that have a zero objective function in the lower level, or that contain no linking variables. In addition, we excluded all instances in the instance set based on MIPLIB2017 that are also included in the instance set based on MIPLIB2010. This results in 102 bilevel instances derived from MIPLIB2010, which consist of 60 pure integer and 42 mixed integer instances. Further, we have 227 instances derived from MIPLIB2017, which split into 93 pure integer and 134 mixed integer instances. The resulting bilevel instance sets are called miplib2010 and miplib2017, respectively.

3.2.2 Pure Integer Instances

The set denegre contains 110 randomly generated instances. These instances were created using a publicly available generator[1]. 50 out of these 110 instances are part of a benchmark set introduced by DeNegre (2011) and have 20 lower-level constraints, no leader constraints, and all coefficients in the objective functions and constraints are random integers in the range $[-50, 50]$. Moreover, the number of upper-level and lower-level variables varies within $\{5, 10, 15\}$. We created the remaining 60 additional instances using the same instance generator, which are particularly suitable for a benchmark set. These instances have 30 lower-level constraints, all coefficients are within $[-30, 30]$, the number of lower-level variables is in $\{5, 10\}$, and the number of upper-level variables is in $\{25, 30\}$. For details about how the instances were generated, please consult the source code of the publicly available instance generator.

Fischetti et al. (2016) and Fischetti et al. (2018a) introduced instances that are based on 19 binary instances of the MIPLIB3.0; see Bixby et al. (1998). The instances have been transformed into bilevel problems by considering the first $k\,\%$ of the variables as lower-level variables (rounded up) and the remaining ones as upper-level variables. For each instance, three variants have been generated with $k \in \{10, 50, 90\}$. All constraints are assumed to be lower-level constraints. The original objective function is set as the leader's objective, while the follower's objective function coefficients are set to $d_l := -d_u$. This makes a total of 57 instances. In addition, the provided instance set miplib3 also contains 3 more instances based on another instance of the MIPLIB3.0, which are analogously created. We also obtain 60 pure integer instances from the previously described miplib2010 instance set and additional 93 pure integer instances from the miplib2017 set; see the last subsection as well.

We further consider 30 instances of pure integer bilevel problems introduced as Testbed 1 in Ozaltin and Zhang (2017). These instances contain between 50 to 90 upper-level variables and 70 to 110 lower-level variables. They can be found in the instance set zhang. Moreover, we included 6 instances of the literature on computational bilevel optimization; see the set misc.[2] In particular, we added the instances moore90 and moore90_2 that model the Examples 1 and 2 of Moore and Bard (1990) with the adaption of additional variable bounds.

[1] https://github.com/tkralphs/MIBLPInstanceGenerator

[2] We note that we are not able to conclusively determine the origin for all of these instances.

Table 2: Keywords in the auxiliary file.

Keyword	Meaning
@NUMVARS	Next line contains number of the lower-level variables
@NUMCONSTR	Next line contains number of the lower-level constraints
@VARSBEGIN	Marks the beginning of the variables section
@VARSEND	Marks the end of the variables section
@CONSTRBEGIN	Marks the beginning of the constraints section
@CONSTREND	Marks the end of the constraints section
@NAME	Next line contains the name of the instance (optional)
@MPS	Next line contains the name of the MPS file corresponding to this auxiliary file
@LP	Next line contains the name of the LP file corresponding to this auxiliary file

4 Data Formats

We now discuss the data format used to represent the bilevel instances in Section 4.1, followed by specifying a corresponding solution format in Section 4.2.

4.1 Instance Files

Every instance is given as a pair of files. The first file describes the MILP obtained by omitting the requirement of lower-level optimality. In other words, it is the relaxation comprised of all upper- and lower-level variables, all upper- and lower-level constraints, and the upper-level objective function. This is the relaxation that is sometimes called the "high-point relaxation" in the literature, but that we simply refer to as the "MILP relaxation" in what follows. This first file can be either in lp or mps format and is called the *instance file* in the following. For the library, we use .mps.gz files created using CPLEX 22.1 (IBM ILOG CPLEX Optimizer 2024). For more details regarding the mps format, we refer to Nazareth (1987).

Let us emphasize that, in order to increase consistency, all problems in the library have been converted to min-min bilevel problems (and then to the new format as described above from an older pre-existing format that is deprecated). The format presented here supports only min-min problems. Obviously, this is without loss of generality and is consistent with the original mps format, which only allowed for minimization.

The second file is the *auxiliary file* (.aux) that specifies which variables and constraints are associated with the lower-level problem, as well as the coefficients of the lower-level objective function. The auxiliary file is divided into sections demarked by keywords that start with the "@" symbol. All keywords are summarized in Table 2. Typically, an auxiliary file starts with the keyword @NUMVARS followed by a line specifying the number of lower-level variables. Analogously, the number of lower-level constraints is given below the @NUMCONSTR keyword. The beginning and end of the variable section is indicated by @VARSBEGIN and @VARSEND. In between these two keywords, each line consists of the name of one of the lower-level variables followed by its objective coefficient in the lower-level problem (separated by white space). The variable names must match the corresponding variable names of the associated instance file. Analogously, the constraints section is bracketed by the keywords @CONSTRBEGIN and @CONSTREND. Each row in between consists of the name of one of the constraints from the instance file that is designated a lower-level constraint. Note that in general, bounds on the lower-level variables can be interpreted as constraints either of the upper- or lower-level problem. In our format, such bounds belong to the lower-level problem by default. To indicate that they are upper-level, the bounds should be imposed explicitly as named constraints in the instance and then excluded from the list of lower-level constraints. The instance file itself can be either in MPS or LP format, so exactly one of the keywords @MPS or @LP appears in the auxiliary file. Specifying a name using the @NAME keyword is optional. Note that in the presented format, the number of variables (@NUMVARS) and the number of constraints (@NUMCONSTR) has to be specified before the section of variable and constraint names begins.

As an example, Figure 1 shows the mps and aux files for the famous Example 2 in Moore and Bard (1990) with additional variables bounds, which after reformulation as a min-min instance is given by

$$\begin{aligned}(-)\min_{x\in\mathbb{Z},y} \quad & F(x,y) = x + 2y \\ \text{s.t.} \quad & x \in [0,3],\ y \in S(x),\end{aligned}$$

where, for a fixed $x \in \mathbb{R}$, $S(x)$ denotes the set of optimal solutions of the integer linear problem

$$\begin{aligned}(-)\min_{y\in\mathbb{Z}} \quad & f(x,y) = -y \\ \text{s.t.} \quad & -x + 2.5y \leq 3.75, \\ & x + 2.5y \geq 3.75, \\ & 2.5x + y \leq 8.75, \\ & y \in [1,2].\end{aligned}$$

The corresponding aux and mps file (after reformulating all constraints as $\leq$) is given in Figure 1.

4.2 Solution Files

For each instance for which a feasible solution is known, we also provide a solution file given in json format, with allowable keywords and values are specified in Table 3. The

```
* ENCODING=ISO-8859-1
NAME          moore90_2
ROWS
 N  R0004
 L  R0001
 L  R0002
 L  R0003
COLUMNS
    MARK0000  'MARKER'                 'INTORG'
    C0001     R0004                          1
    C0001     R0001                         -1
    C0001     R0002                         -1
    C0001     R0003                        2.5
    C0002     R0004                          2
    C0002     R0001                        2.5
    C0002     R0002                       -2.5
    C0002     R0003                          1
    MARK0001  'MARKER'                 'INTEND'
RHS
    rhs       R0001                       3.75
    rhs       R0002                      -3.75
    rhs       R0003                       8.75
BOUNDS
 UP bnd       C0001                          3
 LO bnd       C0002                          1
 UP bnd       C0002                          2
ENDATA
```

```
@NUMVARS
1
@NUMCONSTRS
3
@VARSBEGIN
C0002  -1.
@VARSEND
@CONSTRSBEGIN
R0001
R0002
R0003
@CONSTRSEND
@NAME
moore90_2
@MPS
moore90_2.mps
```

Figure 1: The mps (top) and aux (bottom) file of the modified example by Moore and Bard (1990)

Table 3: Keywords, respectively keys, of a solution file, which is given in a json format.

Keywords	Value
name	name of the instance
bilevel_type	optimistic or pessimistic
status	optimal, infeasible, feasible, or open
difficulty	easy or hard
objective_value	objective value
upper_level_decisions	values of the upper-level decisions
lower_level_decisions	values of the lower-level decisions

solution file starts by specifying the name of the instance and the bilevel type for which the solution is computed, i.e., it either equals optimistic or pessimistic. We note that the instance data (instance and auxiliary file) do not depend on the bilevel type and we currently only provide solutions for the optimistic case.

On the website described in Section 6, we report the current status of the instance, which indicates if an optimal solution (optimal) or only a feasible solution without proof of optimality (feasible) is provided. If infeasibility of the instance is proven by the used solvers, we set the status to infeasible. Moreover, if none of the three previous cases applies, we set the status to open, which means that no feasibility status was reported within the set time limit. Note that when there are no coupling constraints and the MILP relaxation is feasible, this is technically enough to conclude the instance is feasible, even when no feasible point has been determined, but neither of the solvers used for analysis reports this information. We further classify the difficulty of an instance as easy if it can be solved by each of the used solvers (see the next section for more details) within 180 s. Otherwise, the instance is considered as hard. Finally, we provide the objective value (objective_value), the upper-level decisions (upper_level_decisions), and the lower-level decisions (lower_level_decisions). If the status of the instance is optimal or feasible, the latter values contain the best known feasible solution together with its objective function value. Otherwise, these values are null. Finally, we note that the solution file can be in either a dense format (including zero values of variables) or in a sparse format (omitting the zero values in the solution).

As an example, we state the solution file for the previously presented example by Moore and Bard (1990) in Figure 2.

5 Numerical Results

In order to provide feasible or optimal solutions to the instances of the library and construct benchmark sets, we attempted to solve all instances in the collection using two currently available bilevel solvers (described below). Based on the obtained results,

```
{
    "name": "moore90_2",
    "bilevel_type": "optimistic",
    "status": "optimal",
    "difficulty": "easy",
    "objective_value": 5.0,
    "upper_level_decisions": {
        "C0001": 3.0
    },
    "lower_level_decisions": {
        "C0002": 1.0
    }
}
```


Figure 2: The solution file moore90_2.res.json of the modified example by Moore and Bard (1990)

we compile three benchmark sets for interdiction, mixed integer, and pure integer instances. These sets provide a basis for conducting computational comparisons of different solution methods in a reasonable time. We first outline the used bilevel solvers and the corresponding computational setup in Section 5.1. Afterward, we discuss the procedure to verify bilevel feasibility of computed points in Section 5.2. We give a short overview regarding the numerical results w.r.t. the entire library in Section 5.3. Finally, we discuss the benchmark instance sets and present corresponding numerical results in Section 5.4.

5.1 Solvers and Computational Setup

Compared to the multiple mature solvers available for single-level mixed integer linear optimization (Gleixner et al. 2021), the development of general-purpose solvers for mixed integer bilevel problems is still in its infancy. For our computational study, we use two different bilevel solvers.

The first one is the open-source solver MibS 1.2.2 that is freely available under an open source license; see DeNegre et al. (2024). MibS can solve general MIBLPS of the form (1). For the mathematical details we refer to DeNegre and Ralphs (2009) and Tahernejad and Ralphs (2020). Different single-level solvers can be included in MibS to solve the mixed integer linear subproblems that occur in the course of the solution process. By default, MibS uses the open-source mixed integer linear solver SYMPHONY; see Ralphs and Güzelsoy (2005). However, in our computational study, we use CPLEX 22.1.0.0 (IBM ILOG CPLEX Optimizer 2024) as the solver for the MILP subproblems arising during the solution process.

The second bilevel solver we use is the mixed integer bilevel linear solver by Fischetti et al. (2024), which is available as a pre-compiled binary and can be used after requesting a license file from the authors. This solver uses CPLEX 12.7.1 as the underlying MILP solver. Moreover, the latter solver only supports the older and deprecated format for the auxiliary files.[3] For this solver, different settings are available that allow to solve instances of different types. We do not conduct a specific tuning of these settings for each instance. Instead, we start with the default setting 4 (MIX++) and if this is not applicable, we apply the setting 99 (HC++). Moreover, we set the option randomseed to -1 to ensure that the results are deterministically reproducible.

The computations are carried out on a single node of a server[4] with Intel Xeon Gold 6326 CPUs. For both solvers and each instance, we use a time limit of 1 h, we set a memory limit of 32 GB, and limit the number of threads to 1.

5.2 Verifying Feasibility

Checking the bilevel feasibility of a given point consists of multiple steps. First of all, a feasible solution has to satisfy all upper- and lower-level constraints. However, for bilevel feasibility, it is also necessary that the part of the solution associated with the follower is optimal for the follower's problem (2) with the part of the solution associated with the leader fixed to the given values. For the provided bilevel feasible solutions, respectively optimal solutions, we ensure that these conditions are satisfied by applying the following procedure in which we use Gurobi 11.0.3 (Gurobi Optimization, LLC 2023) to solve the occurring optimization problems.

i. We solve the MILP relaxation obtained by omitting the requirement of lower-level optimality and stop after the first feasible solution is found. This step serves as a simple check for infeasibility of the considered bilevel problem. Note that the MILP relaxation is contained in the instance file and its infeasibility directly proves infeasibility of the bilevel problem.

ii. Next, we verify that the point given in the solution file, consisting of upper-level and lower-level decisions, satisfies all constraints and integer restrictions w.r.t. the variables, as well as that the upper-level objective value matches. To this end, we again consider the MILP relaxation of Step (i), which is an MILP, and apply the feasibility checker (version 1.0.3) of the MIPLIB2017 (Gleixner et al. 2021) with the default tolerance of 10^{-4} for the linear constraints and for the integer restrictions. We note that this checker only checks for feasibility and not for optimality of any solution. For the technical details of this feasibility checker, we refer to Koch et al. (2011) and Gleixner et al. (2021).

[3]The instances of the library are also available to the authors in the older format to which this solver was then applied.

[4]https://doc.nhr.fau.de/clusters/woody/

iii. Afterward, we check that the lower-level problem is solved to optimality by the given point. To this end, we consider the lower-level problem in which the upper-level variables are fixed to the corresponding values of the provided point. We then solve this lower-level problem to global optimality. Afterward, we compare the obtained objective value (φ^{check}) with the lower-level objective value (φ^{sol}) corresponding to the given point. We check for optimality up to a tolerance of 10^{-4} by evaluating $|\varphi^{\text{sol}} - \varphi^{\text{check}}|/(10^{-10} + \varphi^{\text{check}}) \leq 10^{-4}$. The left-hand side is derived from the relative MILP gap definition of CPLEX.[5]

5.3 Overview of the Numerical Results for the Entire Collection

We now give a brief overview of the numerical results for the entire collection (detailed results for each instance set can be found in the appendix). In the following, we discarded the results of a particular solver in a small number of cases in which the solver produced a solution that did not pass the feasibility check of Section 5.2. Moreover, we consider a so-called "virtual best solver" that for each instance returns the best available result and the corresponding fastest runtime of the two considered bilevel solvers.

First, we provide some statistical properties of the collection in Table 4. This overview shows that the instance set is rather diverse and contains instances with different types of variables and different sizes. In particular, the collection contains instances with and without coupling constraints and instances with and without continuous linking variables.

Table 5 summarizes the results of solving all instances with both solvers, with the instances classified according to the statuses described earlier in Section 4.2. Roughly 39 % of the instances can be solved to global optimality by at least one solver within 1 h. In addition, for another nearly 44 % of the instances, at least one solver provides a feasible solution without optimality proof. Of the remaining instances, approximately 1 % are proven to be infeasible and the status of the rest is open, which means that the output of the solvers did not indicate a feasibility status.

In Figure 3 and Table 6 and 7, we summarize the runtimes of the virtual best solver w.r.t. all instances solved to global optimality. The overall picture shows that the majority of the instances can be solved with a moderate runtime. In particular, Table 7 shows that a large number of the instances can be solved rather quickly (within 10 s). Instances whose solution time exceeds 10 s are evidently more difficult for current bilevel solvers to solve to optimality. Nevertheless, for a large number of instances, a feasible solution is known (status is feasible), but there is no proof of optimality; see Table 5 again. Thus, it is proving optimality (i.e., improving the dual bound) that seems to be the main challenge. As a result, the library contains primarily instances that are either easy or difficult and there is a need to increase the number of instances of moderate difficulty in the future. Moreover, we note that all instances with continuous linking

[5] https://www.ibm.com/docs/en/icos/22.1.1?topic=parameters-relative-mip-gap-tolerance

Table 4: Statistics for the number of variables and constraints in the entire collection.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	1	59	200	500	714 549
Integer	0	0	0	0	77 626
Binary	0	40	100	400	636 923
Continuous	0	0	0	0	399 808
LL Variables	1	60	200	500	714 549
Integer	0	0	0	70	40 180
Binary	0	0	58	250	674 369
Continuous	0	0	0	0	399 608
Linking Variables	1	59	200	500	714 549
Integer	0	0	0	0	77 626
Binary	0	31	100	400	636 923
Continuous	0	0	0	0	394 447
UL Constraints	0	1	1	9	480 585
LL Constraints	3	66	201	501	961 170
Coupling Constraints	0	0	0	0	356 461
Linking Constraints	3	50	184	500	490 038

Table 5: Number of solved and open problems for the entire collection with time limit of 1 h.

Total	Optimal	Infeasible	Feasible	Open
2654	1027	33	1175	419

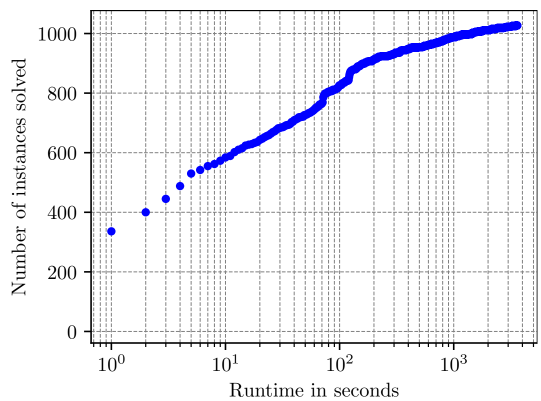

Figure 3: Number of solved instances within 3600 seconds w.r.t. the virtual best solver and the entire collection.

Table 6: Statistics about the runtimes (s) of the virtual best solver for the entire collection (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	0.46	4.62	71.38	3483.85

variables are open because neither of the solvers currently supports this type of linking variables.

5.4 Benchmark Instance Sets

For developing optimization methods and software, it is useful to have curated sets of benchmark instances that serve as a meaningful basis for experimental comparison. Motivated by this and the success of the benchmark sets of MIPLIB2010 and MIPLIB2017, we also selected three subsets of instances that we refer to as the interdiction, pure integer, and mixed integer benchmark instance sets. To this end, we include the corresponding instances of the library that satisfy all of the following properties:

i. Both solvers produced a final result within 1400 s;

ii. The instance requires at least 10 s for each solver to produce the result;

Table 7: Number of solved instances of the entire collection within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
579	245	164	39

iii. The result is either that the instance is proved infeasible or has a provably finite optimum;

iv. The results of both solvers are consistent and the solution (if feasible) passes the feasibility check; see Section 5.2.

Condition (i) ensures that each benchmark instance can be solved in a reasonable time by today's solvers. We exclude instances that are too easy (≤ 10 s) for both solvers by Condition (ii). Moreover, we only consider instances for which both solvers either prove infeasibility or compute a globally optimal solution; see Condition (iii). Finally, we only include instances for which both solvers provide consistent results, i.e., both solvers have the same status (infeasible or optimal) and the corresponding optimal objective values F^1 and F^2 satisfy $|F^1 - F^2|/(10^{-10} + |F^2|) \leq 10^{-4}$; similar to the feasibility check in Section 5.2. One further goal while compiling the benchmark instance sets was to have rather balanced sets regarding the different types of the instances. Applying the described conditions to all instances of the library leads to 235 instances, which we then partition into interdiction, pure integer, and mixed integer benchmark instance sets, i.e., each set only contains those benchmark instances that belong to the corresponding class of bilevel problems.

5.4.1 Interdiction Benchmark Set

The interdiction benchmark set consists of 69 instances in total, 31 of which are classified as easy (they can be solved by both solvers within 180 s) and 38 of which are classified as hard. We note that the virtual best solver can solve 60 of the instances within 180 s. Consequently, 9 instances are hard for both of the solvers. Moreover, all of these instances are feasible and solved to global optimality within 1400 s by both solvers. Note that these interdiction instances were solved as general MIBLPs, not using methods specialized for interdiction problems.

We now discuss and analyze the interdiction benchmark instance set. In Figure 4 and Table 8 and 9, we display the runtimes of the virtual best solver w.r.t. the benchmark instances. A majority of the instances can be solved within 180 s. Furthermore, the runtimes of the instances that are solved in the interval between 10 s and 180 s are rather evenly distributed. In the remaining time interval, 9 more instances can be solved by the virtual best solver. Consequently, the benchmark set contains instances that are rather easy to solve for the current bilevel solvers as well as some more challenging instances that can be still solved in reasonable time.

Next, we consider the distribution of the benchmark instances regarding the different instance sets of the library. To this end, we give an overview of the classes of bilevel problems that are part of the benchmark set in Table 10, which shows that different types of interdiction problems are present in the benchmark set. We provide the detailed Table 18 in the appendix that gives an overview to which instance set each single benchmark instance belongs to.

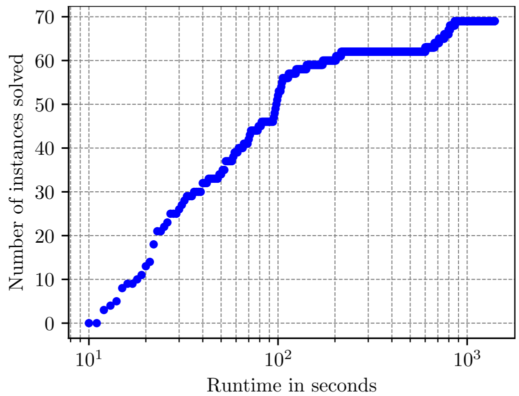

Figure 4: Number of solved interdiction instances in [10, 1400] seconds w.r.t. the virtual best solver and the interdiction benchmark instance set.

Table 8: Statistics about the runtimes (s) of the virtual best solver for the interdiction benchmark instance set.

Min	1st Quartile	Median	3rd Quartile	Max
11.29	21.92	50.98	99.27	860.25

Table 9: Number of interdiction benchmark instances solved within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
0	52	17	0

Table 10: Overview of the classes of bilevel problems that are part of the interdiction benchmark set.

Sets	BOBILib	Benchmark Set
collection	2654	69
interdiction	1959	69
generalized	90	30
assignment	24	0
knapsack	599	16
multidimensional-knapsack	954	3
clique	220	0
network	72	20

Table 11: Statistics for the number of variables and constraints of the interdiction benchmark instances.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	27	40	49	50	400
Integer	0	0	0	0	0
Binary	27	40	49	50	400
Continuous	0	0	0	0	0
LL Variables	27	40	50	98	400
Integer	0	0	10	20	400
Binary	0	0	20	94	156
Continuous	0	0	0	20	20
Linking Variables	10	10	30	49	400
Integer	0	0	0	0	0
Binary	10	10	30	49	400
Continuous	0	0	0	0	0
UL Constraints	1	1	1	20	20
LL Constraints	29	30	40	138	401
Coupling Constraints	0	0	0	20	20
Linking Constraints	10	10	30	49	400

Finally, we provide some statistics of bilevel properties of the interdiction benchmark instances such as the number of upper- and lower-level variables and constraints in Table 11. This overview particularly shows that the chosen interdiction benchmark instances cover different classes of bilevel problems such as problems with and without coupling constraints.

5.4.2 Mixed Integer Benchmark Set

The mixed integer benchmark set comprises 95 instances, of which 67 are easy and 28 are hard. The virtual best solver can solve 91 instances within 180 s, i.e., 4 instances are hard for both solvers. Consequently, the two solvers show greater performance variability for the mixed integer instances compared to the interdiction benchmark instances. Again, all instances are feasible and can be solved to global optimality.

As we did with the interdiction benchmark set, we display the runtimes of the virtual best solver for the mixed integer benchmark instances in Figure 5, as well as in Tables 12 and 13. The results show that the majority of the instances can be solved by the virtual best solver rather quickly, i.e., within approximately 210 s. However, we note that the difference in run times between the two solvers for individual instances is much higher than with the other benchmark sets.

All of the mixed integer benchmark instances belong to the sets xuwang or xularge; see

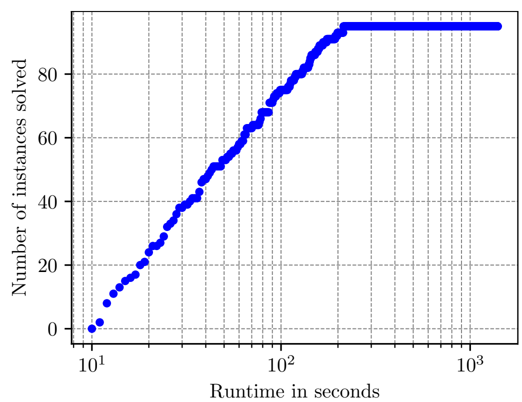

Figure 5: Number of solved mixed integer instances in [10, 1400] seconds w.r.t. the virtual best solver and the mixed integer benchmark instance set.

Table 12: Statistics about the runtimes (s) of the virtual best solver for the mixed integer benchmark instance set.

Min	1st Quartile	Median	3rd Quartile	Max
10.4	20.07	40.16	90.61	213.92

Table 13: Number of mixed integer benchmark instances solved within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
0	75	20	0

Table 14: Statistics for the number of variables and constraints of the mixed integer benchmark instances.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	110	410	600	800	1000
Integer	110	410	600	800	1000
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
LL Variables	110	410	600	800	1000
Integer	63	216	292	402	512
Binary	0	0	0	0	0
Continuous	47	204	298	406	527
Linking Variables	110	410	600	800	1000
Integer	110	410	600	800	1000
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
UL Constraints	44	164	240	320	400
LL Constraints	44	164	240	320	400
Coupling Constraints	44	164	240	320	400
Linking Constraints	44	164	240	320	400

Table 19 in the appendix. The detailed numerical results (given in the appendix) show that the mixed integer instances are either easy to solve or rather challenging, e.g., in particular in the case of those derived from MIPLIB2010 and MIPLIB2017. Consequently, it would improve the overall library to have more mixed integer instances that can be solved within a moderate time limit to global optimality.

As in the previous section, we provide some statistics of bilevel properties of the mixed integer benchmark instances in Table 14, which shows less variety compared to the interdiction benchmark set.

5.4.3 Pure Integer Benchmark Set

The pure integer benchmark set consists of 71 instances with 35 easy and 36 hard instances. The virtual best solver can solve the 68 of these instances within 180 s. Thus, the two solvers show even greater differences in the runtimes than for the mixed integer benchmark instances.

Compared to the interdiction and mixed integer benchmark sets, the runtimes of pure integer instances are distributed less uniformly between the minimal and maximal runtimes as illustrated in Figure 6 and underlined by the Tables 15 and 16. The detailed distribution of the pure integer benchmark instances can be found in Table 20 in the appendix.

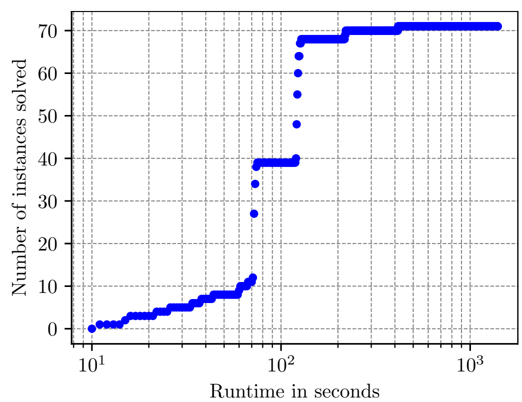

Figure 6: Number of solved pure integer instances in [10, 1400] seconds w.r.t. the virtual best solver and the pure integer benchmark instance set.

Table 15: Statistics about the runtimes (s) of the virtual best solver for the pure integer benchmark instance set.

Min	1st Quartile	Median	3rd Quartile	Max
10.29	71.64	73.53	121.65	415.69

Table 16: Number of pure integer benchmark instances solved within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
0	39	32	0

Table 17: Statistics for the number of variables and constraints of the pure integer benchmark instances.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	5	20	25	25	2258
Integer	0	20	20	25	25
Binary	0	0	0	0	2258
Continuous	0	0	0	0	0
LL Variables	5	5	10	10	2258
Integer	0	5	10	10	110
Binary	0	0	0	0	2258
Continuous	0	0	0	0	0
Linking Variables	5	20	25	25	2258
Integer	0	20	20	25	25
Binary	0	0	0	0	2258
Continuous	0	0	0	0	0
UL Constraints	0	0	0	0	0
LL Constraints	6	30	30	30	1820
Coupling Constraints	0	0	0	0	0
Linking Constraints	6	30	30	30	1820

Again, we provide some statistics of bilevel properties of the pure integer benchmark instances in Table 17, which also shows less variety compared to the interdiction benchmark set.

Overall, the presented benchmark sets are well suited as a (first) meaningful basis for experimental comparisons of different solution methods or solvers. Compared to the interdiction benchmark set, the pure and mixed benchmark sets exhibit less diversity in terms of the origin of the instances. However, this is in line with the current status of the literature because bilevel research on interdiction problems is much more mature than the research on general pure and mixed integer bilevel optimization.

6 The BOBILib Website

Besides this report, we have also set up a website for the BOBILib. On this website, the user can download the overall set of BOBILib instances, as well as all subsets of instances and the benchmark sets described in Section 5.4. All instances are licensed under the Creative Commons CC BY-SA 4.0 license.[6]

Moreover, the website contains four tables (one for the entire collection and three for the respective benchmark sets) of instances that can be filtered and sorted according to

[6] https://creativecommons.org/licenses/by-sa/4.0/

different statistics of the instances, such as the number of upper-/lower-level variables or constraints, the presence of coupling constraints, etc. Using these tables, the user can also reach a separate sub-page for each instance on which we provide more detailed information about the instance (compared to what is given in the table of all instances). Additionally, a solution file (in json format; see Section 4.2) can be downloaded on these sub-pages.

Finally, the website contains a collection of links to other code repositories that provide additional functionality that can be used in combination with the BOBILib instances but which is not part of the library itself. An example is a code to create quadratic matrices with certain properties to turn mixed integer linear into mixed integer quadratic instances. Another example is the feasibility checker described in Section 5.2.

7 Future Plans

With the BOBILib, we compiled more than 2600 instances of MIBLPs. Since such a structured and well-curated library was missing so far in the field of computational mixed integer bilevel optimization, we hope that this initiative helps to further propel this young field of research.

We emphasize that the current library is only a starting point and we intend to actively develop and maintain it over time. In particular, there are at least five aspects that we would like to improve over the next months and years. First, the impact of the library can still be increased significantly by collecting more instances of real-world bilevel problems and, second, more instances that are not interdiction problems. In particular, integrating application-driven instances of bilevel optimization—such as those arising in transportation, machine learning, or energy networks and markets—will lead to an even more diverse collection. Instances based on real-world problems often contain specific structures that can be exploited to develop tailored solution approaches, which may then be generalized to broader classes of problems. Consequently, such instances can foster the algorithmic advances in the field of mixed integer bilevel optimization. We therefore encourage researchers and practitioners to contribute application-driven instances for future releases of the library. Third, we hope to eventually produce a more balanced set of instances, since we currently have disproportionately many instances classified as either easy or hard but too few instances of medium difficulty. Fourth, since our numerical experiments reveal that for almost all infeasible instances, the solvers prove infeasibility at the beginning of the solution process despite the MILP relaxation being feasible, we would also like to include non-trivial infeasible instances. Fifth and finally, the extension of the library towards pessimistic bilevel problems is a reasonable direction for future development.

In light of these goals for growth, we invite submission of new instances from members of the community so that the collection can expand in a positive direction. The website will include a mechanism for submitting new best-known solutions for all instances in the solution format of Section 4.2. The statuses of all instances will be tracked and updated over time so that progress in the field can be easily followed. The respective contact data can be found on the library's website

https://bobilib.org.

We look forward to seeing how the community utilizes this data and also to seeing how the existence of benchmarks helps to move research in this field forward.

Acknowledgements

The authors thank the Deutsche Forschungsgemeinschaft for their support within projects A05 and B08 in the Sonderforschungsbereich/Transregio 154 "Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks". This research was made possible with support from Office of Naval Research Grant N000142212676.

Parts of the computations were executed on the high performance cluster "Elwetritsch" at the TU Kaiserslautern, which is part of the "Alliance of High Performance Computing Rheinland-Pfalz" (AHRP) We kindly acknowledge the support of RHRK.

Other parts were executed on the HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU). The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG).

Furthermore, we thank the ZIMK at Trier University, and in particular Christian Hennen, for their help with many IT related questions that we needed to answer to set up the website.

The authors acknowledge the use of DeepL and OpenAI's ChatGPT for partly editing and polishing the text and figures for spelling, grammar, and stylistic improvements. Additionally, ChatGPT was utilized for support in basic coding tasks.

Moreover, we are very grateful for many helpful discussions with Markus Sinnl and for all people who have contributed instances to the library. Finally, we thank Michelle Mergen for her help in preparing first versions of the required codes to structure and analyze the library as well as to set up the website.

References

Baggio, A., M. Carvalho, A. Lodi, and A. Tramontani (2021). "Multilevel Approaches for the Critical Node Problem". In: *Operations Research* 69.2, pp. 486–508. DOI: 10.1287/opre.2020.2014.

Bard, J. F. and J. T. Moore (1990). "A branch and bound algorithm for the bilevel programming problem". In: *SIAM Journal on Scientific and Statistical Computing* 11.2, pp. 281–292. DOI: 10.1137/0911017.

Bialas, W. F. and M. H. Karwan (1984). "Two-level linear programming". In: *Management Science* 30.8, pp. 1004–1020. DOI: 10.1287/mnsc.30.8.1004.

Bixby, R. E., S. Ceria, C. M. McZeal, and M. W. P. Savelsbergh (1998). *An updated mixed integer programming library: MIPLIB 3.0*. Technical Report. Rice University.

Calamai, P. H. and L. N. Vicente (1993). "Generating Linear and Linear-Quadratic Bilevel Programming Problems". In: *SIAM Journal on Scientific Computing* 14.4, pp. 770–782. DOI: 10.1137/0914049.

Calamai, P. H. and L. N. Vicente (1994). "Generating Quadratic Bilevel Programming Test Problems". In: *ACM Transactions on Mathematical Software* 20.1, pp. 103–119. DOI: 10.1145/174603.174411.

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

Chu, P. C. and J. E. Beasley (1998). "A genetic algorithm for the multidimensional knapsack problem". In: *Journal of Heuristics* 4.1, pp. 63–86. DOI: 10.1023/A:1009642405419.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Springer. DOI: 10.1007/b101970.

DeNegre, S. T. (2011). "Interdiction and discrete bilevel linear programming". PhD thesis. Lehigh University. URL: https://coral.ise.lehigh.edu/~ted/files/papers/ScottDeNegreDissertation11.pdf.

DeNegre, S. T. and T. K. Ralphs (2009). "A branch-and-cut algorithm for integer bilevel linear programs". In: *Operations Research and Cyber-Infrastructure*. Springer, pp. 65–78. DOI: 10.1007/978-0-387-88843-9_4.

DeNegre, S. T., T. K. Ralphs, S. Tahernejad, and Y. Xie (2024). URL: https://github.com/coin-or/MibS (visited on 04/22/2024).

Fischetti, M., I. Ljubic, M. Monaci, and M. Sinnl (2024). URL: https://msinnl.github.io/pages/bilevel.html (visited on 04/22/2024).

Fischetti, M., I. Ljubić, M. Monaci, and M. Sinnl (2016). "Intersection cuts for bilevel optimization". In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer, pp. 77–88. DOI: 10.1007/978-3-319-33461-5_7.

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

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

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

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

Floudas, C. A., P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Schweiger (1999). *Handbook of Test Problems in Local and Global Optimization*. Kluwer Academic Publishers. DOI: 10.1007/978-1-4757-3040-1.

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

Furini, F., E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. V. Sahinidis, S. Vigerske, and A. Wiegele (2019). “QPLIB: a library of quadratic programming instances”. In: *Mathematical Programming Computation* 11.2, pp. 237–265. DOI: 10.1007/s12532-018-0147-4.

Gleixner, A., G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. Christophel, K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin, and Y. Shinano (2021). *MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library*. 3, pp. 443–490. DOI: 10.1007/s12532-020-00194-3.

Gurobi Optimization, LLC (2023). *Gurobi Optimizer Reference Manual*. URL: https://www.gurobi.com.

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

IBM ILOG CPLEX Optimizer (2024). *IBM ILOG CPLEX Optimizer*. URL: https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer.

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

Jungen, D., A. Zingler, H. Djelassi, and A. Mitsos (2023). *libDIPS – Discretization-Based Semi-Infinite and Bilevel Programming Solvers*. URL: https://optimization-online.org/?p=24914.

Khuri, S., T. Baeck, and J. Heitkoetter (1995). *SAC94 Suite: Collection of Multiple Knapsack Problems*. http://www.cs.cmu.edu/Groups/AI/areas/genetic/ga/test/sac/0.html. Accessed: 2020-04-27.

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

Kleinert, T. and M. Schmidt (2021). "Computing Feasible Points of Bilevel Problems with a Penalty Alternating Direction Method". In: *INFORMS Journal on Computing* 33.1, pp. 198–215. DOI: 10.1287/ijoc.2019.0945.

Koch, T., T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter (2011). "MIPLIB 2010: mixed integer programming library version 5". In: *Mathematical Programming Computation* 3.2, pp. 103–163. DOI: 10.1007/s12532-011-0025-9.

Köppe, M., M. Queyranne, and C. T. Ryan (2010). "Parametric Integer Programming Algorithm for Bilevel Mixed Integer Programs". In: *Journal of Optimization Theory and Applications* 146.1, pp. 137–150. DOI: 10.1007/s10957-010-9668-3.

Martello, S., D. Pisinger, and P. Toth (1999). "Dynamic Programming and Strong Bounds for the 0-1 Knapsack Problem". In: *Management Science* 45.3, pp. 414–424. DOI: 10.1287/mnsc.45.3.414.

MINLPLib (2022). *MINLPLib. A Library of Mixed-Integer and Continuous Nonlinear Programming Instances*. Git hash: 2be6d7c0, last access: 2022-11-09. URL: https://www.minlplib.org.

Moore, J. T. and J. F. Bard (1990). "The Mixed Integer Linear Bilevel Programming Problem". In: *Operations Research* 38.5, pp. 911–921. DOI: 10.1287/opre.38.5.911.

Moshirvaziri, K., M. A. Amouzegar, and S. E. Jacobsen (1996). "Test problem construction for linear bilevel programming problems". In: *Journal of Global Optimization* 8.3, pp. 235–243. DOI: 10.1007/BF00121267.

Nazareth, J. L. (1987). *Computer solution of linear programs*. Monographs on Numerical Analysis. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, pp. xvi+231.

Ozaltin, O. Y. and J. Zhang (2017). *A Branch-and-Cut Algorithm for Discrete Bilevel Linear Programs*. Tech. rep. URL: https://optimization-online.org/?p=14558.

Paulavicius, R. and C. S. Adjiman (2017). *BASBLib - a library of bilevel test problems, v2.2 [Data set]*. DOI: 10.5281/zenodo.897966.

Ralphs, T. K. (2016). *Cor@l: Bilevel Optimization Problem Library*. http://coral.ise.lehigh.edu/data-sets/bilevel-instances/. Accessed: 2020-04-30.

Ralphs, T. K. and M. Güzelsoy (2005). "The Symphony Callable Library for Mixed Integer Programming". In: *The Next Wave in Computing, Optimization, and Decision Technologies*. Boston, MA: Springer US, pp. 61–76.

Sinnl, M. (2020). *Bilevel Integer Programming and Interdiction Problems*. Accessed: 2020-04-30. URL: https://msinnl.github.io/pages/bilevel.html.

Tahernejad, S. and T. K. Ralphs (2020). *Valid Inequalities for Mixed Integer Bilevel Linear Optimization Problems*. URL: https://coral.ise.lehigh.edu/~ted/files/papers/MibSCuts20.

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

Vicente, L., G. Savard, and J. Júdice (1996). "Discrete Linear Bilevel Programming Problem". In: *Journal of Optimization Theory and Applications* 89.3, pp. 597–614. DOI: 10.1007/BF02275351.

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

Zhou, S., A. B. Zemkoho, and A. Tin (2018). *BOLIB: Bilevel Optimization LIBrary of test problems*. URL: https://arxiv.org/abs/1812.00230.

Appendix

Detailed Numerical Results: Distribution of the Interdiction Benchmark Instance Set

Table 18: Detailed distribution of the interdiction benchmark instance set. The sets printed in sans-serif style actually contain instances while the other rows of the table are used to structure the overall library.

Sets	BOBILib	Benchmark Set
collection	2654	69
interdiction	1959	69
generalized	90	30
generalized	90	30
assignment	24	0
inter-assig	24	0
knapsack	599	16
cclw	50	6
inter-kp	99	4
kp	450	6
multidimensional-knapsack	954	3
or	810	0
imkp	144	3
clique	220	0
bcpins	80	0
clique	60	0
plusbcpins	80	0
network	72	20
inter-fire	72	20

Detailed Numerical Results: Distribution of the Mixed Integer Benchmark Instance Set

Table 19: Detailed distribution of the mixed integer benchmark instance set. The sets printed in sans-serif style actually contain instances while the other rows of the table are used to structure the overall library.

Sets	BOBILib	Benchmark Set
collection	2654	95
general-bilevel	695	95
mixed integer	336	95
miplib2017	134	0
miplib2010	42	0
xuwang	100	36
xularge	60	59

Detailed Numerical Results: Distribution of the Pure Integer Benchmark Instance Set

Table 20: Detailed distribution of the pure integer benchmark instance set. The sets printed in sans-serif style actually contain instances while the other rows of the table are used to structure the overall library.

Sets	BOBILib	Benchmark Set
collection	2654	71
general-bilevel	695	71
pure integer	359	71
miplib2017	93	1
miplib2010	60	1
misc	6	0
miplib3	60	0
denegre	110	62
zhang	30	7

Detailed Numerical Results: Overview Easy and Hard Instances

Table 21: Detailed overview of the number of easy and hard instances per instance set

Set	Easy	Hard	Total
collection	468	2186	2654
interdiction	211	1748	1959
generalized	25	65	90
generalized	25	65	90
assignment	24	0	24
inter-assig	24	0	24
knapsack	92	507	599
cclw	12	38	50
inter-kp	35	64	99
kp	45	405	450
multidimensional-knapsack	14	940	954
or	0	810	810
imkp	14	130	144
clique	20	200	220
bcpins	10	70	80
clique	0	60	60
plusbcpins	10	70	80
network	36	36	72
inter-fire	36	36	72
general-bilevel	257	438	695
mixed-integer	132	204	336
miplib2017	0	134	134
miplib2010	0	42	42
xuwang	100	0	100
xularge	32	28	60
pure-integer	125	234	359
miplib2017	1	92	93
miplib2010	2	58	60
misc	4	2	6
miplib3	15	45	60
denegre	78	32	110
zhang	25	5	30

Detailed Numerical Results: interdiction

Table 22: Number of solved and open problems for the instance set interdiction with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
1959	631	0	1094	234

Table 23: Statistics for the number of variables and constraints in instance set interdiction.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	55	100	400	1593
Integer	0	0	0	0	0
Binary	10	55	100	400	1593
Continuous	0	0	0	0	0
LL Variables	8	55	154	400	1653
Integer	0	0	0	20	500
Binary	0	8	70	250	1653
Continuous	0	0	0	0	20
Linking Variables	10	55	100	400	1593
Integer	0	0	0	0	0
Binary	10	55	100	400	1593
Continuous	0	0	0	0	0
UL Constraints	1	1	1	4	29
LL Constraints	11	86	201	401	3363
Coupling Constraints	0	0	0	0	20
Linking Constraints	10	55	100	400	1593

Table 24: Statistics about the runtimes (s) of the virtual best solver for instance set interdiction with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.01	0.59	4.25	70.66	3483.85

Table 25: Number of solved instances of the set interdiction within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
380	111	106	34

Detailed Numerical Results: assignment

Table 26: Number of solved and open problems for the instance set inter-assig with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
24	24	0	0	0

Table 27: Statistics for the number of variables and constraints in instance set inter-assig.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	25	25	25	25	25
Integer	0	0	0	0	0
Binary	25	25	25	25	25
Continuous	0	0	0	0	0
LL Variables	25	25	25	25	25
Integer	0	0	0	0	0
Binary	25	25	25	25	25
Continuous	0	0	0	0	0
Linking Variables	25	25	25	25	25
Integer	0	0	0	0	0
Binary	25	25	25	25	25
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	45	45	45	45	45
Coupling Constraints	0	0	0	0	0
Linking Constraints	25	25	25	25	25

Table 28: Statistics about the runtimes (s) of the virtual best solver for instance set inter-assig with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.01	0.07	0.1	0.12	0.18

Table 29: Number of solved instances of the set inter-assig within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
24	0	0	0

Detailed Numerical Results: clique

Table 30: Number of solved and open problems for the instance set clique-class with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
220	160	0	60	0

Table 31: Statistics for the number of variables and constraints in instance set clique-class.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	19	31	59	546	1593
Integer	0	0	0	0	0
Binary	19	31	59	546	1593
Continuous	0	0	0	0	0
LL Variables	8	12	50	586	1653
Integer	0	0	0	0	0
Binary	8	12	50	586	1653
Continuous	0	0	0	0	0
Linking Variables	19	31	59	546	1593
Integer	0	0	0	0	0
Binary	19	31	59	546	1593
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	28	58	105	1326	3363
Coupling Constraints	0	0	0	0	0
Linking Constraints	19	31	59	546	1593

Table 32: Statistics about the runtimes (s) of the virtual best solver for instance set clique-class with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.04	0.19	0.64	3.39	380.08

Table 33: Number of solved instances of the set clique-class within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
142	14	4	0

Detailed Numerical Results: bcpins

Table 34: Number of solved and open problems for the instance set bcpins with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
80	80	0	0	0

Table 35: Statistics for the number of variables and constraints in instance set bcpins.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	19	31	46	73	94
Integer	0	0	0	0	0
Binary	19	31	46	73	94
Continuous	0	0	0	0	0
LL Variables	8	10	12	15	15
Integer	0	0	0	0	0
Binary	8	10	12	15	15
Continuous	0	0	0	0	0
Linking Variables	19	31	46	73	94
Integer	0	0	0	0	0
Binary	19	31	46	73	94
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	28	45	66	105	105
Coupling Constraints	0	0	0	0	0
Linking Constraints	19	31	46	73	94

Table 36: Statistics about the runtimes (s) of the virtual best solver for instance set bcpins with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.04	0.14	0.53	2.34	380.08

Table 37: Number of solved instances of the set bcpins within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
73	6	1	0

Detailed Numerical Results: clique (Instance Set)

Table 38: Number of solved and open problems for the instance set clique with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
60	0	0	60	0

Table 39: Statistics for the number of variables and constraints in instance set clique.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	546	702	1102	1239	1593
Integer	0	0	0	0	0
Binary	546	702	1102	1239	1593
Continuous	0	0	0	0	0
LL Variables	586	742	1152	1299	1653
Integer	0	0	0	0	0
Binary	586	742	1152	1299	1653
Continuous	0	0	0	0	0
Linking Variables	546	702	1102	1239	1593
Integer	0	0	0	0	0
Binary	546	702	1102	1239	1593
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	1326	1482	2327	3009	3363
Coupling Constraints	0	0	0	0	0
Linking Constraints	546	702	1102	1239	1593

Detailed Numerical Results: plusbcpins

Table 40: Number of solved and open problems for the instance set plusbcpins with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
80	80	0	0	0

Table 41: Statistics for the number of variables and constraints in instance set plusbcpins.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	19	31	46	73	94
Integer	0	0	0	0	0
Binary	19	31	46	73	94
Continuous	0	0	0	0	0
LL Variables	27	41	58	88	109
Integer	0	0	0	0	0
Binary	27	41	58	88	109
Continuous	0	0	0	0	0
Linking Variables	19	31	46	73	94
Integer	0	0	0	0	0
Binary	19	31	46	73	94
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	53	86	129	208	280
Coupling Constraints	0	0	0	0	0
Linking Constraints	19	31	46	73	94

Table 42: Statistics about the runtimes (s) of the virtual best solver for instance set plusbcpins with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.05	0.22	0.92	5.02	338.31

Table 43: Number of solved instances of the set plusbcpins within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
69	8	3	0

Detailed Numerical Results: generalized

Table 44: Number of solved and open problems for the instance set generalized with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
90	78	0	12	0

Table 45: Statistics for the number of variables and constraints in instance set generalized.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	40	40	50	50	50
Integer	0	0	0	0	0
Binary	40	40	50	50	50
Continuous	0	0	0	0	0
LL Variables	40	40	50	50	50
Integer	10	10	20	20	30
Binary	0	0	10	10	20
Continuous	20	20	20	20	20
Linking Variables	10	10	20	20	30
Integer	0	0	0	0	0
Binary	10	10	20	20	30
Continuous	0	0	0	0	0
UL Constraints	20	20	20	20	20
LL Constraints	30	30	40	40	50
Coupling Constraints	20	20	20	20	20
Linking Constraints	10	10	20	20	30

Table 46: Statistics about the runtimes (s) of the virtual best solver for instance set generalized with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.63	8.22	61.02	264.38	3483.85

Table 47: Number of solved instances of the set generalized within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
22	24	25	7

Detailed Numerical Results: knapsack

Table 48: Number of solved and open problems for the instance set knapsack with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
599	207	0	377	15

Table 49: Statistics for the number of variables and constraints in instance set knapsack.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	100	200	400	500
Integer	0	0	0	0	0
Binary	10	100	200	400	500
Continuous	0	0	0	0	0
LL Variables	10	100	200	400	500
Integer	0	100	200	400	500
Binary	0	0	0	0	55
Continuous	0	0	0	0	0
Linking Variables	10	100	200	400	500
Integer	0	0	0	0	0
Binary	10	100	200	400	500
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	11	101	201	401	501
Coupling Constraints	0	0	0	0	0
Linking Constraints	10	100	200	400	500

Table 50: Statistics about the runtimes (s) of the virtual best solver for instance set knapsack with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.03	1.14	3.57	71.89	3192.16

Table 51: Number of solved instances of the set knapsack within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
124	35	30	18

Detailed Numerical Results: cclw

Table 52: Number of solved and open problems for the instance set cclw with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
50	35	0	15	0

Table 53: Statistics for the number of variables and constraints in instance set cclw.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	35	40	45	50	55
Integer	0	0	0	0	0
Binary	35	40	45	50	55
Continuous	0	0	0	0	0
LL Variables	35	40	45	50	55
Integer	0	0	0	0	0
Binary	35	40	45	50	55
Continuous	0	0	0	0	0
Linking Variables	35	40	45	50	55
Integer	0	0	0	0	0
Binary	35	40	45	50	55
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	36	41	46	51	56
Coupling Constraints	0	0	0	0	0
Linking Constraints	35	40	45	50	55

Table 54: Statistics about the runtimes (s) of the virtual best solver for instance set cclw with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.06	3.35	39.56	284.34	3192.16

Table 55: Number of solved instances of the set cclw within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
13	10	8	4

Detailed Numerical Results: inter-kp

Table 56: Number of solved and open problems for the instance set inter-kp with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
99	74	0	25	0

Table 57: Statistics for the number of variables and constraints in instance set inter-kp.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
LL Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
Linking Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	11	21	31	41	51
Coupling Constraints	0	0	0	0	0
Linking Constraints	10	20	30	40	50

Table 58: Statistics about the runtimes (s) of the virtual best solver for instance set inter-kp with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.03	0.67	11.95	150.12	2934.98

Table 59: Number of solved instances of the set inter-kp within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
37	16	11	10

Detailed Numerical Results: kp

Table 60: Number of solved and open problems for the instance set kp with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
450	98	0	337	15

Table 61: Statistics for the number of variables and constraints in instance set inter-kp.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
LL Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
Linking Variables	10	20	30	40	50
Integer	0	0	0	0	0
Binary	10	20	30	40	50
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	11	21	31	41	51
Coupling Constraints	0	0	0	0	0
Linking Constraints	10	20	30	40	50

Table 62: Statistics about the runtimes (s) of the virtual best solver for instance set kp with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.16	1.14	2.51	8.81	2814.5

Table 63: Number of solved instances of the set kp within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
74	9	11	4

Detailed Numerical Results: multidimensional-knapsack

Table 64: Number of solved and open problems for the instance set multidimensional-knapsack with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
954	92	0	643	219

Table 65: Statistics for the number of variables and constraints in instance set multidimensional-knapsack.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	100	250	500	500
Integer	0	0	0	0	0
Binary	10	100	250	500	500
Continuous	0	0	0	0	0
LL Variables	10	100	250	500	500
Integer	0	0	0	0	0
Binary	10	100	250	500	500
Continuous	0	0	0	0	0
Linking Variables	10	100	250	500	500
Integer	0	0	0	0	0
Binary	10	100	250	500	500
Continuous	0	0	0	0	0
UL Constraints	1	1	4	9	29
LL Constraints	11	102	251	501	529
Coupling Constraints	0	0	0	0	0
Linking Constraints	10	100	250	500	500

Table 66: Statistics about the runtimes (s) of the virtual best solver for instance set multidimensional-knapsack with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.04	4.56	56.79	231.87	3475.39

Table 67: Number of solved instances of the set multidimensional-knapsack within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
30	22	33	7

Detailed Numerical Results: imkp

Table 68: Number of solved and open problems for the instance set imkp with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
144	92	0	52	0

Table 69: Statistics for the number of variables and constraints in instance set imkp.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	30	50	70	105
Integer	0	0	0	0	0
Binary	10	30	50	70	105
Continuous	0	0	0	0	0
LL Variables	10	30	50	70	105
Integer	0	0	0	0	0
Binary	10	30	50	70	105
Continuous	0	0	0	0	0
Linking Variables	10	30	50	70	105
Integer	0	0	0	0	0
Binary	10	30	50	70	105
Continuous	0	0	0	0	0
UL Constraints	1	1	3	4	29
LL Constraints	11	33	52	74	106
Coupling Constraints	0	0	0	0	0
Linking Constraints	10	30	50	70	105

Table 70: Statistics about the runtimes (s) of the virtual best solver for instance set imkp with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.04	4.56	56.79	231.87	3475.39

Table 71: Number of solved instances of the set imkp within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
30	22	33	7

Detailed Numerical Results: or

Table 72: Number of solved and open problems for the instance set or with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
810	0	0	591	219

Table 73: Statistics for the number of variables and constraints in instance set or.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	100	100	250	500	500
Integer	0	0	0	0	0
Binary	100	100	250	500	500
Continuous	0	0	0	0	0
LL Variables	100	100	250	500	500
Integer	0	0	0	0	0
Binary	100	100	250	500	500
Continuous	0	0	0	0	0
Linking Variables	100	100	250	500	500
Integer	0	0	0	0	0
Binary	100	100	250	500	500
Continuous	0	0	0	0	0
UL Constraints	1	1	4	9	29
LL Constraints	101	109	254	501	529
Coupling Constraints	0	0	0	0	0
Linking Constraints	100	100	250	500	500

Detailed Numerical Results: inter-fire

Table 74: Number of solved and open problems for the instance set inter-fire with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
72	70	0	2	0

Table 75: Statistics for the number of variables and constraints in instance set inter-fire.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	22	24	49	77	79
Integer	0	0	0	0	0
Binary	22	24	49	77	79
Continuous	0	0	0	0	0
LL Variables	44	48	98	154	158
Integer	0	0	0	0	0
Binary	44	48	98	154	158
Continuous	0	0	0	0	0
Linking Variables	22	24	49	77	79
Integer	0	0	0	0	0
Binary	22	24	49	77	79
Continuous	0	0	0	0	0
UL Constraints	1	1	1	1	1
LL Constraints	41	63	170	240	974
Coupling Constraints	0	0	0	0	0
Linking Constraints	22	24	49	77	79

Table 76: Statistics about the runtimes (s) of the virtual best solver for instance set inter-fire with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.44	3.08	6.45	98.74	2360.55

Table 77: Number of solved instances of the set inter-fire within specific time ranges (only for instances solved to optimality by the virtual best solver).

[0, 10)	[10, 100)	[100, 1000)	[1000, 3600)
38	16	14	2

Detailed Numerical Results: general-bilevel

Table 78: Number of solved and open problems for the instance set general-bilevel with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
695	396	33	81	185

Table 79: Statistics for the number of variables and constraints in instance set general-bilevel.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	1	64	548	4234	714 549
Integer	0	0	0	40	77 626
Binary	0	0	0	1420	636 923
Continuous	0	0	0	0	399 808
LL Variables	1	80	548	4234	714 549
Integer	0	0	5	100	40 180
Binary	0	0	0	2122	674 369
Continuous	0	0	0	172	399 608
Linking Variables	1	60	460	2934	714 549
Integer	0	0	0	25	77 626
Binary	0	0	0	1172	636 923
Continuous	0	0	0	0	394 447
UL Constraints	0	0	0	255	480 585
LL Constraints	3	30	241	2482	961 170
Coupling Constraints	0	0	0	240	356 461
Linking Constraints	3	30	225	1707	490 038

Table 80: Statistics about the runtimes (s) of the virtual best solver for instance set general-bilevel with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	0.25	9.33	71.44	2140.1

Table 81: Number of solved instances of the set general-bilevel within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
199	134	58	5

Detailed Numerical Results: mixed integer

Table 82: Number of solved and open problems for the instance set mixed-integer with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
336	168	7	13	148

Table 83: Statistics for the number of variables and constraints in instance set mixed-integer.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	268	700	3844	399 808
Integer	0	0	10	360	1000
Binary	0	0	0	155	136 568
Continuous	0	0	0	1188	399 808
LL Variables	10	267	700	3843	399 808
Integer	0	0	50	211	26 287
Binary	0	0	0	747	136 571
Continuous	0	18	185	447	399 608
Linking Variables	10	260	600	2681	394 447
Integer	0	0	10	360	1000
Binary	0	0	0	110	136 532
Continuous	0	0	0	737	394 447
UL Constraints	0	0	124	360	480 585
LL Constraints	4	124	320	2472	961 170
Coupling Constraints	0	0	117	320	356 461
Linking Constraints	4	124	320	1806	490 038

Table 84: Statistics about the runtimes (s) of the virtual best solver for instance set mixed-integer with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	2.09	14.62	51.58	213.92

Table 85: Number of solved instances of the set mixed-integer within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
71	77	20	0

Detailed Numerical Results: miplib2010

Table 86: Number of solved and open problems for the instance set miplib2010 with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
102	32	11	17	42

Table 87: Statistics for the number of variables and constraints in instance set miplib2010.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	50	548	1343	4507	64 590
Integer	0	0	0	0	236
Binary	0	100	760	4003	64 590
Continuous	0	0	0	199	5958
LL Variables	50	548	1342	4506	64 590
Integer	0	0	0	0	758
Binary	0	245	1011	4506	64 590
Continuous	0	0	0	9	4392
Linking Variables	25	335	1012	3680	64 590
Integer	0	0	0	0	236
Binary	0	100	450	2357	64 590
Continuous	0	0	0	105	5944
UL Constraints	0	0	16	1832	59 795
LL Constraints	16	618	2482	5996	119 589
Coupling Constraints	0	0	0	747	59 795
Linking Constraints	16	483	1461	4172	113 351

Table 88: Statistics about the runtimes (s) of the virtual best solver for instance set miplib2010 with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.04	0.57	4.38	33.34	580.06

Table 89: Number of solved instances of the set miplib2010 within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
17	13	2	0

Detailed Numerical Results: miplib2017

Table 90: Number of solved and open problems for the instance set miplib2017 with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
227	32	22	37	136

Table 91: Statistics for the number of variables and constraints in instance set miplib2017.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	15	1275	7272	27 578	714 549
Integer	0	0	0	0	77 626
Binary	0	0	1392	12 101	636 923
Continuous	0	0	0	2002	399 808
LL Variables	15	1275	7272	27 578	714 549
Integer	0	0	0	20	40 180
Binary	0	90	2349	16 501	674 369
Continuous	0	0	0	394	399 608
Linking Variables	15	767	5067	23 908	714 549
Integer	0	0	0	0	77 626
Binary	0	0	896	10 058	636 923
Continuous	0	0	0	1493	394 447
UL Constraints	0	0	0	2317	480 585
LL Constraints	6	850	3705	21 340	961 170
Coupling Constraints	0	0	0	1621	356 461
Linking Constraints	6	481	2094	15 128	490 038

Table 92: Statistics about the runtimes (s) of the virtual best solver for instance set miplib2017 with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.02	0.16	0.91	66.42	2140.1

Table 93: Number of solved instances of the set miplib2017 within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
20	6	2	4

Detailed Numerical Results: xularge

Table 94: Number of solved and open problems for the instance set xularge with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
60	60	0	0	0

Table 95: Statistics for the number of variables and constraints in instance set xularge.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	500	600	800	900	1000
Integer	500	600	800	900	1000
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
LL Variables	500	600	800	900	1000
Integer	227	296	375	455	512
Binary	0	0	0	0	0
Continuous	226	305	372	446	527
Linking Variables	500	600	800	900	1000
Integer	500	600	800	900	1000
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
UL Constraints	200	240	320	360	400
LL Constraints	200	240	320	360	400
Coupling Constraints	200	240	320	360	400
Linking Constraints	200	240	320	360	400

Table 96: Statistics about the runtimes (s) of the virtual best solver for instance set xularge with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
5.85	38.83	76.76	129.44	213.92

Table 97: Number of solved instances of the set xularge within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
1	39	20	0

Detailed Numerical Results: xuwang

Table 98: Number of solved and open problems for the instance set xuwang with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
100	100	0	0	0

Table 99: Statistics for the number of variables and constraints in instance set xuwang.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	10	110	260	360	460
Integer	10	110	260	360	460
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
LL Variables	10	110	260	360	460
Integer	3	55	119	184	256
Binary	0	0	0	0	0
Continuous	4	56	114	179	244
Linking Variables	10	110	260	360	460
Integer	10	110	260	360	460
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
UL Constraints	4	44	104	144	184
LL Constraints	4	44	104	144	184
Coupling Constraints	4	44	104	144	184
Linking Constraints	4	44	104	144	184

Table 100: Statistics about the runtimes (s) of the virtual best solver for instance set xuwang with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	0.52	3.93	13.83	78.66

Table 101: Number of solved instances of the set xuwang within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
64	36	0	0

Detailed Numerical Results: pure integer

Table 102: Number of solved and open problems for the instance set pure-integer with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
359	228	26	68	37

Table 103: Statistics for the number of variables and constraints in instance set pure-integer.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	1	20	181	4452	714 549
Integer	0	0	0	15	77 626
Binary	0	0	112	4452	636 923
Continuous	0	0	0	0	0
LL Variables	1	10	180	4452	714 549
Integer	0	0	5	10	40 180
Binary	0	0	90	4452	674 369
Continuous	0	0	0	0	0
Linking Variables	1	20	150	2941	714 549
Integer	0	0	0	15	77 626
Binary	0	0	102	2880	636 923
Continuous	0	0	0	0	0
UL Constraints	0	0	0	0	84 788
LL Constraints	3	20	112	2523	169 576
Coupling Constraints	0	0	0	0	84 788
Linking Constraints	3	20	57	1567	137 938

Table 104: Statistics about the runtimes (s) of the virtual best solver for instance set pure-integer with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	0.08	2.9	72.12	2140.1

Table 105: Number of solved instances of the set pure-integer within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
128	57	38	5

Detailed Numerical Results: denegre

Table 106: Number of solved and open problems for the instance set denegre with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
110	110	0	0	0

Table 107: Statistics for the number of variables and constraints in instance set denegre.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	5	10	20	25	25
Integer	5	10	20	25	25
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
LL Variables	5	5	10	10	15
Integer	5	5	10	10	15
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
Linking Variables	5	10	20	25	25
Integer	5	10	20	25	25
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
UL Constraints	0	0	0	0	0
LL Constraints	20	20	30	30	30
Coupling Constraints	0	0	0	0	0
Linking Constraints	16	20	30	30	30

Table 108: Statistics about the runtimes (s) of the virtual best solver for instance set denegre with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.01	0.05	71.28	120.1	127.12

Table 109: Number of solved instances of the set denegre within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
48	33	29	0

Detailed Numerical Results: miplib3

Table 110: Number of solved and open problems for the instance set miplib3 with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
60	27	0	26	7

Table 111: Statistics for the number of variables and constraints in instance set miplib3.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	3	55	686	3598	78 734
Integer	0	0	0	0	0
Binary	3	55	686	3598	78 734
Continuous	0	0	0	0	0
LL Variables	2	54	686	3597	78 733
Integer	0	0	0	0	0
Binary	2	54	686	3597	78 733
Continuous	0	0	0	0	0
Linking Variables	3	55	686	3598	78 734
Integer	0	0	0	0	0
Binary	3	55	686	3598	78 734
Continuous	0	0	0	0	0
UL Constraints	0	0	0	0	0
LL Constraints	16	112	176	755	4944
Coupling Constraints	0	0	0	0	0
Linking Constraints	4	47	122	426	4815

Table 112: Statistics about the runtimes (s) of the virtual best solver for instance set miplib3 with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.01	0.08	0.85	4.78	1173.55

Table 113: Number of solved instances of the set miplib3 within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
21	2	3	1

Detailed Numerical Results: misc

Table 114: Number of solved and open problems for the instance set misc with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
6	5	0	1	0

Table 115: Statistics for the number of variables and constraints in instance set misc.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	1	4	7	10	10
Integer	0	1	1	10	10
Binary	0	0	0	4	7
Continuous	0	0	0	0	0
LL Variables	1	2	7	10	10
Integer	0	1	2	10	10
Binary	0	0	0	0	7
Continuous	0	0	0	0	0
Linking Variables	1	4	7	10	10
Integer	0	1	1	10	10
Binary	0	0	0	4	7
Continuous	0	0	0	0	0
UL Constraints	0	0	0	1	2
LL Constraints	3	4	4	8	10
Coupling Constraints	0	0	0	0	0
Linking Constraints	3	4	4	7	10

Table 116: Statistics about the runtimes (s) of the virtual best solver for instance set misc with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.0	0.0	0.0	0.01	0.01

Table 117: Number of solved instances of the set misc within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
5	0	0	0

Detailed Numerical Results: zhang

Table 118: Number of solved and open problems for the instance set zhang with time limit of 3600 s.

Total	Optimal	Infeasible	Feasible	Open
30	30	0	0	0

Table 119: Statistics for the number of variables and constraints in instance set zhang.

	Min	1st Quartile	Median	3rd Quartile	Max
UL Variables	50	60	70	80	90
Integer	0	0	0	0	0
Binary	50	60	70	80	90
Continuous	0	0	0	0	0
LL Variables	70	80	90	100	110
Integer	70	80	90	100	110
Binary	0	0	0	0	0
Continuous	0	0	0	0	0
Linking Variables	50	60	70	80	90
Integer	0	0	0	0	0
Binary	50	60	70	80	90
Continuous	0	0	0	0	0
UL Constraints	0	0	0	0	0
LL Constraints	6	6	7	7	7
Coupling Constraints	0	0	0	0	0
Linking Constraints	6	6	7	7	7

Table 120: Statistics about the runtimes (s) of the virtual best solver for instance set zhang with time limit of 3600 s (only for instances solved to optimality).

Min	1st Quartile	Median	3rd Quartile	Max
0.02	0.08	1.35	7.72	219.37

Table 121: Number of solved instances of the set zhang within specific time ranges (only for instances solved to optimality by the virtual best solver).

$[0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, 3600)$
23	5	2	0