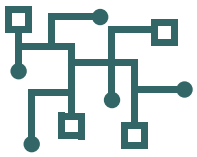


ISE

Industrial and
Systems Engineering

A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation

Sahar Tahernejad and Ted K. Ralphs

Department of Industrial and System Engineering, Lehigh University, Bethlehem, PA

Scott T. DeNegre

Hospital for Special Surgery, New York, New York

COR@L Technical Report 16T-015-R5

A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation

Sahar Tahernejad[*1], Ted K. Ralphs[†1], and Scott T. DeNegre[‡2]

[1]Department of Industrial and System Engineering, Lehigh University, Bethlehem, PA
[2]Hospital for Special Surgery, New York, New York

Original Publication: Dec 30, 2016
Last Revised: Sep 23, 2019

Abstract

In this paper, we describe a comprehensive algorithmic framework for solving mixed integer bilevel linear optimization problems (MIBLPs) using a generalized branch-and-cut approach. The framework presented merges features from existing algorithms (for both traditional mixed integer linear optimization and MIBLPs) with new techniques to produce a flexible and robust framework capable of solving a wide range of bilevel optimization problems. The framework has been fully implemented in the open-source solver `MibS`. The paper describes the algorithmic options offered by `MibS` and presents computational results evaluating the effectiveness of the various options for the solution of a number of classes of bilevel optimization problems from the literature.

1 Introduction

This paper describes an algorithmic framework for the solution of *mixed integer bilevel linear optimization problems* (MIBLPs) and `MibS`, its open-source software implementation. MIBLPs comprise a difficult class of optimization problems that arise in applications in which multiple, possibly competing decision-makers (DMs), make a sequence of decisions over time. For an ever-increasing array of such applications, the traditional framework of mathematical optimization, which assumes a single DM with a single objective function making a decision at a single point in time, is inadequate.

The motivation for the development of `MibS`, which was begun a decade ago, is both to serve as an open test bed for new algorithmic ideas and to provide a platform for solution of the wide

[*]sat214@lehigh.edu
[†]ted@lehigh.edu
[‡]DeNegreS@hss.edu

variety of practical problems of this type currently coming under scientific study. The modeling framework underlying `MibS` is that of *multilevel optimization*. Multilevel optimization problems model applications in which decisions are made in a sequence, with decisions made earlier in the sequence affecting the options available later in the sequence. Under the assumption that all DMs are rational and have complete information about their own and each other's models and input data (there is no stochasticity), we can describe such problems formally using the language of mathematical optimization. To do so, we consider a set of decision variables, partitioned into subsets associated with individual DMs. Conceptually, these sets of decision variables are ordered according to the sequence in which decisions are to be made. We use the term *level* (or *stage*, in some contexts) to denote each DM's position in the sequence. The term is intended to conjure up a hierarchy in which decisions flow from top to bottom, so decisions earlier in the sequence are said to be "at a higher level." The decisions made by higher-level DMs influence those of lower-level DMs through a parametric dependence of the lower-level decision problems on higher-level decisions. Thus, higher-level decisions must be made taking into account the effect those decisions will have on second-level decisions, which in turn impact the objective value of the first-level solution.

The decision hierarchy of a national government provides an archetypal example. The national government makes decisions about tax rates or subsidies that in turn affect the decisions of state and local governments, which finally affect the decisions of individual taxpayers. Since later decisions affect the degree to which the national government achieves its original objective, the decisions at that higher level must be made in light of the reactions of lower-level DMs to those decisions [Bard, 2013].

Thanks to the steady improvement in solution methodologies for traditional optimization problems and the availability of open-source solvers [COIN-OR], the development of practical solution methods for MIBLPs has become more realistic. The availability of practical solution methods has in turn driven an accompanying increase in demand for such methodologies. The literature is now replete with applications requiring the solution of such models [Salmeron et al., 2004, Bienstock and Verma, 2010, Zhang et al., 2016, Gao and You, 2017]. In the remainder of this section, we introduce the basic framework of bilevel optimization.

1.1 Mixed Integer Bilevel Optimization

In this section, we formally introduce MIBLPs, the class of multilevel optimization problem in which we have only two levels and two associated DMs. The decision variables of an MIBLP are partitioned into two subsets, each conceptually controlled by one of the DMs. We generally denote by x the variables controlled by the *first-level DM* or *leader* and require that the values of these variables be contained in the set $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$ representing integrality constraints. We denote by y the variables controlled by the *second-level DM* or *follower* and require the values of these variables to be in the set $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$. Throughout the paper, we refer to a sub-vector of $x \in \mathbb{R}^{n_1}$ indexed by a set $K \subseteq \{1, \dots, n_1\}$ as x_K (and similarly for sub-vectors of $y \in \mathbb{R}^{n_2}$).

The general form of an MIBLP is

$$\min_{x \in X} \{cx + \Xi(x)\}, \tag{MIBLP}$$

where the function Ξ is a *risk function* that encodes the part of the objective value of x that depends on the response to x in the second level. This function may have different forms, depending on

the precise variant of the bilevel problem being solved. In this paper, we focus on the so-called *optimistic* case [Loridan and Morgan, 1996], in which

$$\Xi(x) = \min \left\{ d^1 y \;\middle|\; y \in \mathcal{P}_1(x), y \in \operatorname{argmin}\{ d^2 y \;\middle|\; y \in \mathcal{P}_2(x) \cap Y \} \right\}, \tag{RF-OPT}$$

where

$$\mathcal{P}_1(x) = \left\{ y \in \mathbb{R}^{n_2}_+ \;\middle|\; G^1 y \geq b^1 - A^1 x \right\}$$

is a parametric family of polyhedra containing points satisfying the linear constraints of the first-level problem with respect to a given $x \in \mathbb{R}^{n_1}$ and

$$\mathcal{P}_2(x) = \left\{ y \in \mathbb{R}^{n_2}_+ \;\middle|\; G^2 y \geq A^2 x \right\}$$

is a second parametric family of polyhedra containing points satisfying the linear constraints of the second-level problem with respect to a given $x \in \mathbb{R}^{n_1}$. The input data is $A^1 \in \mathbb{Q}^{m_1 \times n_1}$, $G^1 \in \mathbb{Q}^{m_1 \times n_2}$, $b^1 \in \mathbb{Q}^{m_1}$, $A^2 \in \mathbb{Q}^{m_2 \times n_1}$ and $G^2 \in \mathbb{Q}^{m_2 \times n_2}$. As is customary, we define $\Xi(x) = \infty$ in the case of either $x \notin X$ or infeasibility of the problem on the right-hand side of (RF-OPT).

Note that it is typical in the literature on bilevel optimization for the parametric right-hand side of the second-level problem to include a fixed component, i.e., to be of the form $b^2 - A^2 x$ for some $b^2 \in \mathbb{Q}^{m^2}$. The form introduced here is more general, since adding a constraint $x_1 = 1$ to the upper level problem results in a problem equivalent to the usual one. The advantage of the form here is that it results in a risk function that is subadditive in certain important cases (though not in general), a desirable property for reasons that are beyond the scope of this paper.

As we mentioned above, other variants of the risk function are possible and the algorithm we present can be adapted to these cases (see Section 3.3). A pessimistic risk function can be obtained simply by considering

$$\Xi(x) = \max \left\{ d^1 y \;\middle|\; y \in \mathcal{P}_1(x), y \in \operatorname{argmin}\{ d^2 y \;\middle|\; y \in \mathcal{P}_2(x) \cap Y \} \right\}. \tag{RF-PES}$$

Note that according to this definition, the second-level solution y is required to be a member of $\mathcal{P}_1(x)$ (i.e., be feasible for the upper-level constraints), though there could exist $y \in \operatorname{argmin}\{d^2 y \mid y \in \mathcal{P}_2(x) \cap Y\} \setminus \mathcal{P}_1(x)$. Allowing such second-level solutions to be chosen is also possible and can be accommodated within the framework of (RF-PES) by the addition of dummy variables at the first level. The details are beyond the scope of this paper.

It should be pointed out that we explicitly allow the second-level variables to be present in the first-level constraints. This allowance is rather non-intuitive and leads to many subtle algorithmic complications. Nevertheless, there are applications in which it is necessary to allow this and since `MibS` does allow this possibility, we felt it appropriate to express the results in this full generality. The reader should keep in mind, however, that many of the ideas discussed herein can be simplified in the more usual case that $G^1 = 0$ and we endeavor to point this out in particular cases.

Note that the formulation (MIBLP) does not explicitly involve the second-level variables. The reason for expressing the formulation in this way is to emphasize two things. First, it emphasizes that the goal of solving (MIBLP) is to determine the optimal values of the first-level variables only. Thus, we would ideally like to "project out" the second-level variables. In contrast to the Benders method for traditional optimization problems, however, the second-level variables are re-introduced in solving the relaxation that we employ in our branch-and-cut algorithm (see Section 2.1). Second,

it is necessary at certain points in the algorithm to evaluate $\Xi(x)$, and it is convenient to give it an explicit form here to make this step clearer.

MIBLPs also have an alternative formulation that employs the *value function* of the second-level problem and explicitly includes the second-level variables. This formulation is given by

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

where ϕ is the so-called *value function* of the second-level problem, which is a standard mixed integer linear optimization problem (MILP). The value function returns the optimal value of second-level problem for a given right-hand side, defined by

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

From this alternative formulation, it is evident that if the values of the first-level variables are fixed in (MIBLP-VF) (i.e., the constraints imply that their values must be constant), then the problem of minimizing over the remaining variables is an MILP. In fact, it is not difficult to observe that only the first-level variables having non-zero coefficients in the second-level constraints need be fixed in order for $\phi(A^2 x)$ to be rendered a constant and for (MIBLP-VF) to reduce to an MILP. This result is stated formally in Section 2.2. Because of their central importance in what follows, we formally define the concept of *linking variables*.

Definition 1 (Linking Variables)**.** *Let*

$$L = \left\{ i \in \{1, \ldots, n_1\} \;\middle|\; A_i^2 \neq 0 \right\},$$

be the set of indices of first-level variables with non-zero coefficients in the second-level problem, where A_i^2 denotes the i^{th} column of matrix A^2. We refer to such variables as linking variables*.*

Per our earlier notation, x_L is the sub-vector of $x \in \mathbb{R}^{n_1}$ corresponding to the linking variables. By assuming all linking variables are integer variables, we assure the existence of an optimal solution [Vicente et al., 1996].

Assumption 1. *$L = \{1, \ldots, k_1\}$ for $k_1 \leq r_1$.*

We note here that it can in fact be assumed without loss of generality that $k_1 = r_1$ by simply moving the non-linking variables to the second level. While this is conceptually inconsistent with the intent of the original model, it is not difficult to see that the resulting model is *mathematically* equivalent.

In what follows, it will be convenient to refer to the set

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

of all points satisfying the non-negativity and linear inequality constraints at both levels and the set

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

of points in $\mathcal{P}$ that also satisfy integrality restrictions.

Assumption 2. *$\mathcal{P}$ is bounded.*

This assumption is made to simplify the exposition, but is easy to relax in practice. Corresponding to each $x \in \mathbb{R}^{n_1}$ with $x_L \in \mathbb{Z}^L$, we have the *rational reaction set*, which is defined by

$$\mathcal{R}(x) = \operatorname{argmin}\left\{d^2 y \;\middle|\; y \in \mathcal{P}_2(x) \cap Y\right\}.$$

This set may be empty either because $\mathcal{P}_2(x) \cap Y$ is itself empty or because there exists $r \in \mathbb{R}^{n_2}_+$ such that $G^2 r \geq 0$ and $d^2 r < 0$ (in which case the second-level problem is unbounded no matter what first-level solution is chosen). The latter case can be easily detected in a pre-processing step, so we assume w.l.o.g. that this does not occur (note that this case cannot occur when $G^1 = 0$, since Assumption 2 would then imply that $\left\{r \in \mathbb{R}^{n_2}_+ \setminus 0 \;\middle|\; G^2 r \geq 0\right\} = \emptyset$).

Assumption 3. $\left\{r \in \mathbb{R}^{n_2}_+ \;\middle|\; G^2 r \geq 0, d^2 r < 0\right\} = \emptyset$.

Under our assumptions, the *bilevel feasible region* (with respect to the first- and second-level variables in (MIBLP-VF)) is

$$\mathcal{F} = \{(x, y) \in X \times Y \mid y \in \mathcal{P}_1(x) \cap \mathcal{R}(x)\}$$

and members of $\mathcal{F}$ are called *bilevel feasible solutions*. Although the ostensible goal of solving (MIBLP) is to determine $x \in X$ minimizing $cx + \Xi(x)$, this is equivalent to finding a member of $\mathcal{F}$ that optimizes the first-level objective function $cx + d^1 y$. Observe, however, that by this definition of feasibility, we may have $x^* \in X$ that is optimal for (MIBLP), while for some $\hat{y} \in Y$, $(x^*, \hat{y}) \in \mathcal{F}$ but $\Xi(x^*) < d^1 \hat{y}$.

Because we consider the problem to be that of determining the optimal first-level solution, it is also useful to denote the feasible set with respect to first-level variables only as

$$\mathcal{F}_1 = \operatorname{proj}_x(\mathcal{F}).$$

For $x \in \mathbb{R}^{n_1}$, we have that

$$x \in \mathcal{F}_1 \Leftrightarrow x \in \operatorname{proj}_x(\mathcal{F}) \Leftrightarrow \mathcal{R}(x) \cap \mathcal{P}_1(x) \neq \emptyset \Leftrightarrow \Xi(x) < \infty$$

and we say that $x \in \mathbb{R}^{n_1}$ is feasible if $x \in \mathcal{F}_1$. We can then interpret the conditions for bilevel feasibility of (x, y) as consisting of the following two properties of the first- and second-level parts of the solution independently.

Feasibility Condition 1. $x \in \mathcal{F}_1$.

Feasibility Condition 2. $y \in \mathcal{P}_1(x) \cap \mathcal{R}(x)$.

We exploit this notion of feasibility later in our algorithm.

A related set (see Figure 1) is the set of feasible solutions to the *bilevel linear optimization problem* (BLP) that results from discarding the integrality restrictions, defined as

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

where

$$\mathcal{R}_{\mathrm{LP}}(x) = \operatorname{argmin}\left\{d^2 y \;\middle|\; y \in \mathcal{P}_2(x)\right\}.$$

Note that we do *not* have in general that $\mathcal{F} \subseteq \mathcal{F}_{\mathrm{LP}}$, as discussed later in Section 1.3.

1.2 Special Cases

There are a number of cases for which `MibS` has specialized methods. One of the most important special cases is the *zero sum* case in which $d^1 = -d^2$. The *mixed integer interdiction problem* (MIPINT) is a specific subclass of zero sum problem in which the first-level variables are binary and are in one-to-one correspondence with the second-level variables ($n = n_1 = n_2$). When a first-level variable is fixed to one, this prevents the associated variable in the second-level problem from taking a non-zero value. The interdiction problem can be formulated as [Israeli, 1999, DeNegre, 2011]

$$\min \left\{ d^1 y \;\middle|\; x \in \mathcal{P}_1^{INT} \cap \mathbb{B}^n, y \in \operatorname{argmax}\{ d^1 y \;|\; y \in \mathcal{P}_2^{INT}(x) \cap Y \} \right\}, \tag{MIPINT}$$

where

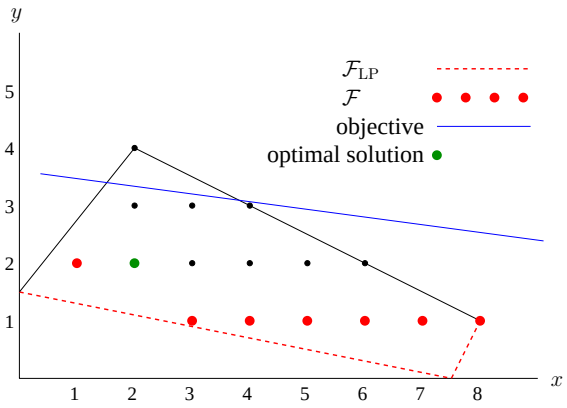
$$\mathcal{P}_1^{INT} = \left\{ x \in \mathbb{R}^n_+ \;\middle|\; A^1 x \geq b^1 \right\},$$
$$\mathcal{P}_2^{INT}(x) = \left\{ y \in \mathbb{R}^n_+ \;\middle|\; G^2 y \geq b^2, y \leq \operatorname{diag}(u)(e - x) \right\},$$

e represents an n-dimensional vector of ones and $u_i \in \mathbb{R}$ denotes the upper bound of y_i for $i = 1, ..., n$. The special structure of MIPINTs can be employed to develop specialized methods for these problems.

1.3 Computational Challenges

From the point of view of both theory and practice, bilevel problems are difficult to solve. From the perspective of complexity theory, the problem is NP-hard even for the case in which all variables are continuous ($r_1 = r_2 = 0$). The general case is in the class Σ_2^p-hard [Jeroslow, 1985], which is the class of problems that can be solved in non-deterministic polynomial time, given an NP oracle. To put it in terms that are a little less formal, these are problems for which even the problem of checking feasibility of a given point in $X \times Y$ is complete for NP in general. Alternatively, simply computing $\Xi(x)$ for $x \in X$ (determining its objective function value), is also NP-hard.

Because determining whether a given solution is feasible is an NP-complete problem, solution of MIBLPs is inherently more difficult than solution of the more familiar case of MILP (equivalent to the case of $L = \emptyset$). Search algorithms for MILPs rely on a certain amount of algorithmically guided "luck" to find high-quality solutions quickly. This works reasonably well in this simpler case because checking feasibility of any given solution is efficient. In the case of MIBLPs, we cannot rely on this property and even if we are lucky enough to get a high-quality first-level solution, checking its feasibility is still a difficult problem. Furthermore, some of the properties whose exploitation we depend on in the case of MILP do not straightforwardly generalize to the MIBLP case. For example, removing the integrality requirement for all variables does not result in a relaxation, since relaxing the second-level problem may make solutions that were previously feasible infeasible. In fact, as we mentioned earlier, the feasible region $\mathcal{F}_{\text{LP}}$ of this BLP does not necessarily even contain the feasible region $\mathcal{F}$ of (MIBLP). Thus, even if the solution to this BLP is in $X \times Y$, it is not necessarily optimal for (MIBLP). These properties can be seen in Figure 1, which displays a well-known example originally from Moore and Bard [1990] that is well-suited for illustrating these concepts.



$$\begin{aligned} \min_{x \in \mathbb{Z}_+} \quad & -x - 10y \\ \text{s.t.} \quad & y \in \operatorname{argmin} \{ y : \\ & -25x + 20y \leq 30 \\ & x + 2y \leq 10 \\ & 2x - y \leq 15 \\ & 2x + 10y \geq 15 \\ & y \in \mathbb{Z}_+ \} \end{aligned}$$

Figure 1: The feasible region of IBLP [Moore and Bard, 1990].

1.4 Previous Work

Bilevel optimization has its roots in game theory and the general class of problems considered in this paper are a type of *Stackelberg game*. The Stackelberg game was studied in a seminal work by Von Stackelberg [1934]. The first bilevel optimization formulations in the form presented here were introduced and the term was coined in the 1970s by Bracken and McGill [1973], but computational aspects of related optimization problems have been studied since at least the 1960s (see, e.g., Wollmer [1964]). Study of algorithms for the case in which integer variables appear is generally acknowledged to have been launched by Moore and Bard [1990], who initiated working on general MIBLPs and discussed the computational challenges of solving them. They proposed a branch-and-bound algorithm for solving MIBLPs which is guaranteed to converge if all first-level variables are integer or all second-level variables are continuous.

Until recently, most computational work focused on special cases with exploitable structure. Bard and Moore [1992], Wen and Huang [1996] and Faísca et al. [2007] studied the bilevel problems with binary variables. Bard and Moore [1992] proposed an exact algorithm for *integer bilevel optimization problems* (IBLPs) in which all variables are binary. Wen and Huang [1996], on the other hand, considered MIBLPs with binary first-level variables and continuous second-level ones and suggested a *tabu search heuristic* for generating solutions to these problems. Faísca et al. [2007] concentrated on MIBLPs in which all discrete variables are constrained to be binary. They reformulated the second-level problem as a multi-parametric problem with the first-level variables as parameters.

DeNegre and Ralphs [2009] and DeNegre [2011] generalized the existing framework of branch and cut, which is the standard approach for solving MILPs, to the case of IBLPs by introducing the *integer no-good cut* to separate bilevel infeasible solutions from the convex hull of bilevel feasible solutions (see Section 2.4). They also initiated development of the open-source solver `MibS`. Xu and Wang [2014] focused on problems in which all first-level variables are discrete and suggested a multi-way branch-and-bound algorithm in which branching is done on the slack variables of the second-level problem. A decomposition algorithm based on column-and-constraint generation was employed by Zeng and An [2014] for solving general MIBLPs. Their method finds the optimal

solution if it is attainable, otherwise it finds an ϵ-optimal solution. Caramia and Mari [2015] introduced a non-linear cut for IBLPs and suggested a method of linearizing this cut by the addition of auxiliary binary variables. Furthermore, they introduced a branch-and-cut algorithm for IBLPs which employs the integer no-good cut from [DeNegre and Ralphs, 2009]. Knapsack interdiction problems were studied by Caprara et al. [2016] and they proposed an exact algorithm for these problems. Hemmati and Smith [2016] formulated competitive prioritized set covering problem as an MIBLP and proposed a cutting plane algorithm for solving it. Wang and Xu [2017] proposed the *watermelon algorithm* for solving IBLPs, which removes bilevel infeasible solutions that satisfy integrality constraints by a non-binary branching disjunction. Lozano and Smith [2017] employed a single-level value function reformulation for solving the MIBLPs in which all first-level variables are integer. Fischetti et al. [2017a] suggested a branch-and-cut algorithm for MIBLPs employing a class of *intersection cuts* valid for MIBLPs under mild assumptions and developed a new family of cuts for the MIBLPs with binary first-level variables. In [Fischetti et al., 2017b], they extended their algorithm by suggesting new types of intersection cuts for IBLPs and introduced the so-called *hypercube intersection cut*, valid for MIBLPs in which the linking variables are discrete. Finally, Mitsos [2010] and Kleniati and Adjiman [2014a,b] considered the more general case of mixed integer bilevel non-linear optimization.

1.5 Overview of Branch and Cut

Branch and cut (a term coined by Padberg and Rinaldi [1987, 1991]) is a variant of the well-known branch-and-bound algorithm of Land and Doig [1960], the most widely-used algorithm for solving many kinds of non-convex optimization problems. For purposes of illustration, we consider here the solution of the general optimization problem

$$\min_{x \in \mathcal{X}} f(x), \tag{GO}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is the *feasible region* and $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*.

The approach of both the branch-and-bound and the branch-and-cut algorithms is to search the feasible region by partitioning it and then recursively solving the resulting subproblems. Implemented naively, this results in an inefficient complete enumeration. This potential inefficiency is avoided by utilizing upper and lower bounds computed for each subproblem to intelligently "prune" the search. The recursive partitioning process can be envisioned as a process of searching a rooted tree, each of whose nodes corresponds to a subproblem. The root of the tree is the node corresponding to the original problem (sometimes called the *root problem*) and the nodes adjacent to it are its *children*. The parent-child relationship applies to other nodes connected along paths in the tree in the obvious way. Nodes with no children are called *leaf nodes*. Henceforth, we denote the set of all leaf nodes in the current search tree by T.

Although it is not usually described this way, we consider the algorithm as a method for iteratively removing parts of the feasible region of the given relaxation that can be proven not to contain any *improving* feasible solutions (feasible solutions that are better than the best solution found so far) until no more such regions can be found. Although most sophisticated branch-and-bound algorithms for optimization do implicitly discard parts of the feasible region that can be shown to contain only suboptimal solutions, the algorithm we present does this more *explicitly*. The removal

is done either by *branching* (partitioning the remaining set of improving solutions to define smaller subproblems) or *cutting* (adding inequalities satisfied by all improving solutions).

A crucial element of both branching and cutting is the identification of *valid disjunctions*. The notion of valid disjunction defined here, which we refer to as an *improving* valid disjunction in order to distinguish it from the more standard definition, refers to a collection of sets that contain all *improving* feasible solutions (while the standard valid disjunction should include all feasible solutions). In the remainder of the paper, we drop the word "improving" when referring to such disjunctions.

Definition 2 (Improving Valid Disjunction)**.** *An* (improving) valid disjunction *for* (GO) *with respect to a given* $x^* \in \mathcal{X}$ *is a disjoint collection*

$$X_1, X_2, \ldots, X_k$$

of subsets of $\mathbb{R}^n$ *such that*

$$\{x \in \mathcal{X} \mid f(x) < f(x^*)\} \subseteq \bigcup_{1 \leq i \leq k} X_i.$$

Imposing such disjunctions is our primary method of eliminating suboptimal solutions and improving the strength of the relaxations used to produce bounds on the optimal value. They play a crucial role in defining methods of both branching and cutting, as we discuss in what follows. We now briefly describe the individual components of branch and cut that we refer to in Section 2.

Bounding. The most important factor needed to improve the efficiency of the algorithm is a tractable method of producing strong upper and lower bounds on the optimal solution value of each subproblem. Typically, lower bounds (assuming minimization) are obtained by solving a relaxation of the original (sub)problem. In most cases, a convex relaxation is chosen in order to ensure tractability, though with problems as difficult as MIBLPs, even a non-convex relaxation may be tractable enough to serve the desired purpose. Upper bounds are obtained by producing a solution feasible to the subproblem, either by heuristic methods or by showing that the solution of the relaxation is feasible to the original problem. We denote the lower and upper bounds associated with node t by L^t and U^t, respectively. Whenever we cannot obtain a feasible solution to a given subproblem t, we set $U^t = \infty$. Similarly, if the relaxation at node t is infeasible, we have $L^t = \infty$.

The bounds on the individual subproblems can be aggregated to obtain global bounds on the optimal solution value to the root problem. Upper bounds of all leaf nodes are aggregated to obtain the global upper bound

$$U = \min_{t \in T} U^t,$$

which represents the objective value of the best solution found so far (known as the *incumbent*). Note that this formula is a bit simplified for the purposes of presentation, since it is possible for bilevel feasible solutions that are *not* feasible for the current subproblem to be produced during the bounding step. While these do not technically update the upper bound for the subproblem, they do contribute to improvement of the global upper bound. Similarly, lower bounds for the leaf nodes are aggregated to form the global lower bound

$$L = \min_{t \in T} L^t.$$

These global upper and lower bounds are updated as the algorithm progresses and when they become equal, the algorithm terminates.

Pruning. Any node t for which $L^t \geq U$ can be discarded (*pruned*), since the feasible region of such a node cannot contain a solution with objective value lower than the current incumbent. This pruning rule implicitly subsumes two special cases. The first is when the feasible region of subproblem t is empty in which case $L^t = \infty$. The second is when $L^t = U^t \geq U$, which typically arises when solving the relaxation of the subproblem associated with node t produces a solution feasible for the original problem.

Since the global bounds are updated dynamically, this means that whether a node can be pruned or not is not a fixed property—a node that previously could not have been pruned may be pruned later when a better incumbent is found. We discuss bounding techniques in more detail in Sections 2.1 and 2.5.

Branching. When bounds have been computed for a node and it cannot be pruned, we then partition the feasible region by identifying and imposing a valid disjunction of the form specified in Definition 2. The subproblems resulting from partitioning the feasible region $\mathcal{X}^t$ of node t are optimization problems of the form

$$\min_{x \in \mathcal{X}^t \cap X_i} f(x),\ 1 \leq i \leq k,$$

which can then be solved recursively using the same algorithm, provided that the collection $\{X_i\}_{1 \leq i \leq k}$ is chosen so as not to change the form of the optimization problem.

Typically, the disjunction is chosen so that $\bigcup_{1 \leq i \leq k} X_i$ does not contain the solution to the current relaxation, but we will see that this is not always possible in the case of MIBLP. When $\mathcal{X} \subseteq \mathbb{Z}^r \times \mathbb{R}^{n-r}$, the disjunction

$$X_1 = \{x \in \mathbb{R}^n \mid \pi x \leq \pi_0\} \quad X_2 = \{x \in \mathbb{R}^n \mid \pi x \geq \pi_0 + 1\} \tag{GD}$$

is always valid when $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ and $\pi_i = 0$ for $i > r$, since we must then have $\pi x \in \mathbb{Z}$ for all $x \in \mathcal{X}$. If the solution $\hat{x}$ to the relaxation is such that $\pi \hat{x} \notin \mathbb{Z}$, then we have $\hat{x} \notin X_1 \cup X_2$. When π is the i^{th} unit vector and $\pi_0 = \lfloor \hat{x}_i \rfloor$ (assuming $\hat{x}_i \notin \mathbb{Z}$), then the disjunction (GD) is called a *variable disjunction*. In the case of a standard MILP, at least one such disjunction must be violated whenever the solution to the relaxation is not feasible to the original problem (assuming the relaxation is a linear optimization problem (LP)). Thus, such disjunctions are all that is needed for a convergent algorithm. The situation is slightly different in the case of MIBLPs.

An important question for achieving good performance is how to choose the branching disjunctions effectively, as it is typically easy to identify many such disjunctions. A measure often used to judge effectiveness is the resulting increase in the lower bound observed after imposing the disjunction. We discuss branching strategies in more detail in Section 2.3.

Searching. The order in which the subproblems are considered is critically important, since, as we have already noted, the global bounds are evolving and the order in which the subproblems are considered affects their evolution. Typical options are *depth-first* (prioritize the "deepest"

nodes in the tree, which tends to emphasize improvement of the upper bound by locating better incumbents) and *best-first* (prioritize those with the smallest lower bound, which tends to emphasize improvement of the lower bound). In practice, one usually employs hybrids that balance these two goals.

Cutting. The main way in which the branch-and-cut algorithm differs in its basic strategy from the branch-and-bound algorithm is that it dynamically strengthens the relaxation by adding *valid inequalities*. As with our notion of valid disjunction, we allow here for a definition slightly different than the standard one in order to allow for inequalities that remove feasible, non-improving solutions. We drop the term "improving" when discussing such inequalities in the remainder of the paper.

Definition 3 (Improving Valid Inequality)**.** *The pair* $(\alpha, \beta) \in \mathbb{R}^{n+1}$ *is an* (improving) valid inequality *for* (GO) *with respect to* $x^* \in \mathcal{X}$ *if*

$$\{x \in \mathcal{X} \mid f(x) < f(x^*)\} \subseteq \{x \in \mathbb{R}^n \mid \alpha x \geq \beta\}.$$

As with branching, we typically aim to add inequalities that are violated by the solution to the current relaxation, thus removing the solution from the feasible region of the relaxation, along with some surrounding polyhedral region that contains no (improving) solutions to the original problem (or subproblem), from consideration.

Furthermore, as with branching, it is generally possible to derive a large number of valid inequalities and a choice must be made as to which ones to add to the current relaxation (adding too many can negatively impact the effectiveness of the relaxation). Also as with branching, the goal of cutting is to improve the lower bound, although selecting directly for this measure is problematic for a number of reasons. We discuss strategies for generating valid inequalities in more detail in Section 2.4. We refer the reader to [Achterberg, 2007] for more detailed description of the branch-and-cut algorithm.

1.6 Outline

The remainder of the paper is organized as follows. In Section 2, we discuss different components of the basic branch-and-cut algorithm implemented in `MibS`. Section 3 describes the overall algorithm in `MibS` and the implementation of this software. In Section 4, we discuss computational results. Finally, in Section 5, we conclude the paper with some final thoughts.

2 A Branch-and-cut Algorithm

The algorithm implemented in `MibS` is based on the algorithmic framework originally described by DeNegre and Ralphs [2009], but with many additional enhancements. The framework utilizes a variant of the branch-and-cut algorithm and is similar in basic outline to state-of-the-art algorithms currently used for solving MILPs in practice. The case of MIBLP is different from the MILP case in a few important ways that we discuss here at a high level before we discuss various specific components.

As we have described earlier in Section 1.5, one way of viewing the general approach taken by the branch-and-cut algorithm is as a method for iteratively removing parts of the feasible region of the given relaxation that can be proven not to contain any improving feasible solutions. In MILP, this is done primarily by either removing *lattice-point free* polyhedral regions (usually by imposing valid disjunctions or by generating associated valid inequalities) or by maintaining an *objective cutoff* that removes all non-improving solutions through the usual pruning process. In the MILP case, we do not generally need to explicitly track or remove solutions that are feasible to the original MILP. Such a solution can only be feasible for the relaxation at one leaf node in the search tree and is either suboptimal for the corresponding subproblem (in which case the solution must have been generated by an auxiliary heuristic) or optimal to that subproblem (in which case the node will be immediately pruned). In either case, it is unlikely the solution will arise again and there is no computational advantage to tracking it explicitly.

In MIBLP, in contrast, we may need to track and remove discrete sets of solutions. This is mainly to avoid the duplication of effort in evaluating the value function ϕ for a specific value of the linking variables. It is possible that the same computation will arise in more than one node in the search tree and re-computation should be avoided. In particular, for $x^1, x^2 \in X$, we have

$$x^1_L = x^2_L \Rightarrow \phi(A^2x^1) = \phi(A^2x^2).$$

Thus, tracking which sub-vectors of values for linking variables have been seen before in a pool (called the *linking solution pool*) can be computationally advantageous. We discuss the mechanism for doing this in Section 3.

2.1 Bounding

Lower Bound. Perhaps the most important step in the branch-and-bound algorithm is that of deriving a lower bound on the optimal solution value. As we have already described, relaxing integrality restrictions does not provide an overall relaxation. However, since $\mathcal{F} \subseteq \mathcal{S} \subseteq \mathcal{P}$, relaxing the optimality condition $d^2y \leq \phi(A^2x)$ in the formulation (MIBLP-VF) results in two alternative relaxations to the original problem. The first relaxation (known as *high-point problem* [Moore and Bard, 1990]) is

$$\min_{(x,y)\in\mathcal{S}} cx + d^1y, \tag{R}$$

in which only the optimality condition is relaxed, while the second is

$$\min_{(x,y)\in\mathcal{P}} cx + d^1y, \tag{LR}$$

in which the integrality constraints are also relaxed. Although both are rather weak bounds, they can be strengthened by the addition of the valid inequalities described in Section 2.4. Because of the enhanced tractability of (LR) and because of the advantages in warm-starting the computation during iterative computation of the bound, we use (LR) as our relaxation of choice. While it is clear that this is a relaxation of (MIBLP-VF), to see that it is a relaxation of (MIBLP), note that we have

$$d^1y^* \leq \Xi(x^*),$$

where (x^*, y^*) is a solution to (LR).

At nodes other than the root node, we can use a similar bounding strategy. Since the branching strategy we describe shortly employs only changes to the bounds of variables, the subproblems that arise have feasible regions of the form

$$\mathcal{F}^t = \left\{(x, y) \in \mathcal{F} \;\middle|\; l_x^t \leq x \leq u_x^t, l_y^t \leq y \leq u_y^t\right\},$$

where t is the index of the node/subproblem, and (l_x^t, u_x^t) and (l_y^t, u_y^t) represent vectors of upper and lower bounds for the first- and second-level variables, respectively. The subproblem at node t is thus the optimization problem

$$\min_{(x,y)\in\mathcal{F}^t} \quad cx + d^1 y, \tag{MIBLP^t}$$

which one can easily show is itself an MIBLP. Thus, we can apply a similar relaxation to obtain the bound

$$L^t = \min_{(x,y)\in\mathcal{P}^t} cx + d^1 y, \tag{LR^t}$$

where $\mathcal{P}^t = \{(x, y) \in \mathcal{P} \mid l_x^t \leq x \leq u_x^t, l_y^t \leq y \leq u_y^t\}$. If it can be verified that conditions for pruning are met (see Section 2.2), then node t should be discarded. Otherwise, the next step is to check feasibility of

$$(x^t, y^t) \in \operatorname*{argmin}_{(x,y)\in\mathcal{P}^t} \; cx + d^1 y,$$

the solution obtained when solving the relaxation.

Feasibility Check. Recall that the upper bound for a given node is derived by exhibiting a feasible solution. Unlike in the case of MILP, checking whether a solution to (LR^t) is feasible for (MIBLP-VF) may itself be a difficult computational problem. Such check involves verifying that (x^t, y^t) satisfies the constraints that were relaxed: integrality conditions and second-level optimality conditions. In other words, (x^t, y^t) is bilevel feasible if and only if the following conditions are satisfied.

Feasibility Condition 3. $x^t \in X$.

Feasibility Condition 4. $y^t \in \mathcal{R}(x^t)$.

Condition 3 ensures that the integrality constraints for the first-level variables are satisfied, while Condition 4 guarantees that (x^t, y^t) satisfies the optimality constraint of second-level problem. Satisfaction of these two conditions (as opposed to the more general Conditions 1 and 2) is enough to ensure $(x^t, y^t) \in \mathcal{F}$, given that (x^t, y^t) is also a solution to (LR^t).

Verifying Condition 3 is straightforward, so this is done first. If Condition 3 is satisfied, then we consider verification of Condition 4. This involves checking both whether $y^t \in Y$ and whether $d^2 y^t = \phi(A^2 x^t)$. Checking whether $y^t \in Y$ is inexpensive so we do this first. The latter check is more expensive and is only required under conditions detailed later in Section 3. We may thus defer it until later, depending on the values of parameters described in Section 3.2.

If we decide to undertake this latter check, we first evaluate $\phi(A^2 x^t)$ to either obtain

$$\hat{y}^t \in \operatorname{argmin}\left\{d^2 y \;\middle|\; y \in \mathcal{P}_2(x^t) \cap Y\right\}, \tag{SL-MILP}$$

or determine $\mathcal{P}_2(x^t) \cap Y = \emptyset$. The MILP (SL-MILP) is solved using an auxiliary MILP solver (more details on this in Section 3.4). If (SL-MILP) has a solution (as it must when $y^t \in Y$), we check whether $d^2 y^t = d^2 \hat{y}^t$. If so, (x^t, y^t) is bilevel feasible and must furthermore be an optimal solution to the subproblem at node t. In this case, $U^t = L^t$ and the current global upper bound U can be set to U^t if $U^t < U$. If (SL-MILP) has no solution, we have that $x^t \notin \mathcal{F}_1$ and x^t can be eliminated from further consideration either by branching or cutting. In fact, in this case, we can eliminate not only x^t, but any first-level solution for which $x_L = x_L^t$. We thus add x_L^t to the linking solution pool in this case.

Although it is not necessary for correctness, (SL-MILP) may be solved even when $y^t \notin Y$ (and even if $x_i^t \notin \mathbb{Z}$ for some $i \notin L$), since this may still lead either to the discovery of a new bilevel feasible solution or a proof that $x^t \notin \mathcal{F}_1$. In Section 3, we discuss when the problem (SL-MILP) should necessarily be solved and when solving it is optional. Even in the case of infeasibility of (x^t, y^t), $(x^t, \hat{y}^t)$ is bilevel feasible (though not necessarily optimal to (MIBLP^t)) whenever $x^t \in X$ and $\hat{y}^t \in \mathcal{P}_1(x^t)$. The second condition is satisfied vacuously in the usual case of $G^1 = 0$. If $(x^t, \hat{y}^t)$ is feasible, then we can update the global upper bound if $cx^t + d^1 \hat{y}^t < U$.

Observe that solving (SL-MILP) reveals more information than the simple value of $\phi(A^2 x^t)$. Note that we have

$$\phi(A^2 x) \leq \phi(A^2 x^t) \; \forall x \in X \text{ such that } \mathcal{P}_2(x) \ni \hat{y}^t, \tag{VFB}$$

which means that $\phi(A^2 x^t)$ reveals an upper bound on the second-level problem for any other first-level solutions that admits $\hat{y}^t$ as a feasible reaction. This upper bound can be exploited in the generation of certain valid inequalities (see DeNegre [2011], Fischetti et al. [2017a]) and is also the basis for an algorithm by Mitsos [2010]. Furthermore, we have

$$\phi(A^2 x) = \phi(A^2 x^t) \; \forall x \in \mathbb{R}^{n_1} \text{ such that } x_L = x_L^t,$$

which means we get the *exact* value of the second-level problem with respect to certain other first-level solutions "for free". Because (i) it may improve the global upper bound, (ii) can reveal that $x^t \notin \mathcal{F}_1$ (and, along with other solutions sharing the same values for the linking variables, should no longer be considered), and (iii) may also provide bounds on the second-level value function for other first-level solutions, (SL-MILP) may be solved in `MibS` even when (x^t, y^t) does not satisfy integrality requirements (see Section 3 for details).

Upper Bound. As we have just highlighted, the feasibility check may produce a bilevel feasible solution $(x^t, \hat{y}^t)$, but $\mathcal{R}(x^t) \cap \mathcal{P}_1(x^t)$ may not be a singleton and there may exist a rational reaction $\bar{y} \in \mathcal{R}(x^t) \cap \mathcal{P}_1(x^t)$ to x^t with $d^1 \bar{y} < d^1 \hat{y}^t$, which results $d^1 \hat{y}^t > \Xi(x^t)$ (note that the selected policy is the optimistic one). To compute the "true" objective function value $cx^t + \Xi(x^t)$ associated with x^t and produce the best possible upper bound, we may explicitly compute $\Xi(x^t)$. Once this computation is done and the associated solution and bound are stored, we can safely eliminate x^t from the feasible region by either branching or cutting, as we describe below, in order to force consideration of alternative solutions. This computation is required under certain conditions and may be optionally undertaken in others, depending on the parameter settings described in detail in Section 3. $\Xi(x^t)$ is obtained by solving

$$\Xi(x^t) = \min \left\{ d^1 y \;\middle|\; y \in \mathcal{P}_1(x^t) \cap \mathcal{P}_2(x^t) \cap Y, d^2 y \leq \phi(A^2 x^t) \right\},$$

which is an MILP since we have already computed $\phi(A^2x^t)$ in the feasibility check. Solving this MILP to evaluate $\Xi(x^t)$ yields the globally valid upper bound $cx^t + \Xi(x^t)$.

Observe that even the problem of minimizing $\Xi(x)$ over $\{x \in X \mid x_L = x_L^t\}$ is still an MILP and can reveal an upper bound across a range of first-level solutions. This result is formalized in Theorem 1 below.

2.2 Pruning

As is the case with all branch-and-bound algorithms, pruning of node t occurs whenever $L^t \geq U$. This again subsumes two special cases. First, if checking Conditions 3 and 4 verifies that (x^t, y^t) is bilevel feasible, then we must have $L^t = U^t \geq U$. Second, when the relaxation is infeasible, we have $L^t = \infty$ and we can again prune the node.

There is one additional case that is particular to MIBLPs and that is when all linking variables are fixed. In this case, we utilize the property of MIBLPs illustrated in the next theorem.

Theorem 1 ([Fischetti et al., 2017a]). *For $\gamma \in \mathbb{Z}^L$, we have*

$$\mathcal{F} \cap \{(x,y) \in X \times Y \mid x_L = \gamma\} = \mathcal{S} \cap \left\{(x,y) \in X \times Y \;\middle|\; d^2y \leq \phi(A^2x), x_L = \gamma\right\}.$$

Proof. From the definitions of sets $\mathcal{S}$ and $\mathcal{F}$ provided in Section 1.1, it follows that

$$\mathcal{F} = \mathcal{S} \cap \left\{(x,y) \in X \times Y \;\middle|\; d^2y \leq \phi(A^2x)\right\}.$$

The result follows. □ □

Corollary 1. *For $\gamma \in \mathbb{Z}^L$, we have*

$$\min\{cx + d^1y \mid (x,y) \in \mathcal{F}, x_L = \gamma\} = \min\{cx + d^1y \mid (x,y) \in \mathcal{S}, d^2y \leq \phi(A^2x), x_L = \gamma\}. \tag{UB}$$

What Theorem 1 tells us is that when the values of all linking variables are fixed at node t ($l^t_{x_L} = u^t_{x_L}$, either because of branching constraints or otherwise), then optimizing over $\mathcal{F}^t$ is equivalent to optimizing over $\mathcal{S}^t$ while additionally imposing an upper bound on the objective value of the second-level problem. In other words, we have the following.

Corollary 2. *Whenever $\mathcal{F}^t \subseteq \{(x,y) \in \mathcal{F} \mid x_L = \gamma\}$ for some $\gamma \in \mathbb{Z}^L$, then*

$$\min_{(x,y)\in\mathcal{F}^t} cx + d^1y = \min\left\{cx + d^1y \;\middle|\; (x,y) \in \mathcal{S}^t, d^2y \leq \phi(A^2x), x_L = \gamma\right\}, \tag{UB^t}$$

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

Therefore, the optimal solution value for the subproblem at node t can be obtained by solving problem (UB^t) with $\gamma = x_L^t$. Furthermore, solving problem (UB) instead of (UB^t) provides a bilevel feasible solution which is at least as good as the optimal solution of node t since it is a relaxation (it is explained in detail in Section 4.3 that there may be more than one node with $x_L = \gamma$). Hence, node t can be pruned after solving either (UB^t) or (UB). Note that these

problems may be infeasible, in which case node t is infeasible. Although solving problem (UB) may be more difficult than solving problem (UB^t), solving problem (UB) provides useful information about all bilevel feasible solutions with $x_L = x_L^t$ (not only those feasible for node t), so it is generally recommended to solve (UB) instead of (UB^t) (see Sections 3.2 and 3.3).

Even if the values of linking variables are not fixed at node t ($l_{x_L}^t \neq u_{x_L}^t$), it may be advantageous to solve (UB) with $\gamma = x_L^t$ whenever $x_L^t \in \mathbb{Z}^L$ in order to improve the upper bound. In Section 3, we describe the parameters of `MibS` that control when problem (UB) is solved during the solution process. The cases in which, this problem must be solved, regardless of the values of parameters, are also discussed.

2.3 Branching

As previously described, the role of the branching procedure is to remove regions of the feasible set of the relaxation that contain no improving bilevel feasible solutions. This is accomplished by imposing a valid disjunction, which we typically choose such that it is violated by the solution to the current relaxation (i.e., removes it from the feasible region). When the branching procedure is invoked for node t, we have that either

- the solution $(x^t, y^t) \notin \mathcal{F}$ because
  - $(x^t, y^t) \notin X \times Y$ or
  - $d^2 y^t > \phi(A^2 x^t)$ (we have previously solved (SL-MILP) with $\gamma = x_L^t$);

  or
- we have $(x^t, y^t) \in X \times Y$ and we are not sure of its feasibility status because we chose not to solve (SL-MILP) (see Section 3 for an explanation of the conditions under which we may make this choice).

These alternatives make it clear that we may sometimes want to branch, even though $(x^t, y^t) \in X \times Y$, necessitating a branching scheme that is more general than the one traditionally used in MILP.

Branching on Linking Variables. Due to Theorem 1, we know that once the values of linking variables are fixed completely at a given node t ($l_{x_L}^t = u_{x_L}^t$), the node can be pruned after solving (UB). One strategy for branching is therefore to think of the search as being over the set of possible values of the linking variables and to prefer a branching disjunction that partitions *this* set of values. In such a scheme, we only consider branching on linking variables as long as any such variables remain unfixed. We show in Section 4 that the apparent logic in such a scheme is effective empirically in some cases, but this scheme also raises issues that do not generally arise in branching on variables in MILP. In particular, situations may arise in which there are no linking variables with fractional values ($x_L^t \in \mathbb{Z}^L$) and yet we would still like to (or need to) impose a branching disjunction.

Naturally, when there *does* exist $i \in L$ such that $x_i^t \notin \mathbb{Z}$, we can impose a variable disjunction

$$X_1 = \left\{(x, y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \leq \lfloor x_i^t \rfloor\right\} \qquad X_2 = \left\{(x, y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \geq \lfloor x_i^t \rfloor + 1\right\},$$

as usual. When $x_L^t \in \mathbb{Z}^L$, however, it is less obvious what to do. If we either have $x_i^t \notin \mathbb{Z}$ for $i \notin L$ or we have that $y^t \notin Y$, then there exists the option of breaking with the scheme and imposing a standard variable disjunction on a non-linking variable (those not in set L). Computational experience has shown that this may not always be a good idea empirically.

When $x_L^t \in \mathbb{Z}^L$, there is no single variable disjunction that can be imposed on linking variables that would eliminate x^t from (the projection of) the feasible region of both resulting subproblems. Suppose we nevertheless branch on the variable disjunction associated with some variable indexed $i \in L$. Since $x_i^t \in \mathbb{Z}$ (and assuming that $l_{x_i}^t \neq u_{x_i}^t$), we may branch, e.g., either on the valid disjunction

$$X_1 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \leq x_i^t\right\}, \qquad X_2 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \geq x_i^t + 1\right\}$$

or

$$X_1 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \leq x_i^t - 1\right\}, \qquad X_2 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_i \geq x_i^t\right\},$$

preferring the former when $x_i^t \leq u_{x_i}^t - 1$ (otherwise, one of the resulting subproblem will be trivially infeasible and the other will be equivalent to node t). This means x^t satisfies either the first or second term of this disjunction. Although imposing this valid disjunction seems undesirable, since it does not remove (x^t, y^t) from the union of the feasible regions of the resulting subproblems, let us explore the logic of the approach.

Suppose we continue to branch in this way and consider the indices of the set of leaf nodes T^t of the subtree of the search tree rooted at node t. Since this set of leaf nodes represents a partition of $\mathcal{S}^t$, the projection of the feasible region of exactly one of these nodes contains x^t. We have that the values of all linking variables are fixed at this node, since we would otherwise continue branching (we have already noted that such node can be pruned after solving problem (UB)). The projection of the union of the feasible regions of the remaining leaf nodes is thus equal to

$$\text{proj}_x(\mathcal{S}^t) \setminus \left\{x \in X \;\middle|\; x_L = x_L^t\right\}.$$

Hence, this branching rule can be seen as implicitly branching on the disjunction

$$X_1 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_L = x_L^t\right\}, \qquad X_2 = \left\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; x_L \neq x_L^t\right\},$$

which is obviously valid.

As a practical matter, if we implement the scheme of only branching on linking variables using simple variable disjunctions, we inevitably create a number of additional nodes for which x^t remains in the projection of the feasible set (one at each level of the tree down to the leaf node at which all linking variables are fixed). This can easily be detected using the scheme for maintaining a pool of linking solutions that have already been seen and will not cause any significant computational issues. For the remaining nodes, we are guaranteed that x^t is not contained in the projection (though these nodes may also be the root of a subtree in which all linking variables are integer-valued). An alternative would be to branch in a non-binary way using a more complicated disjunction that would directly create the leaf nodes of the subtree rooted at node t, but there seems to be no advantage to such a scheme.

In forming the list of variables that are candidates for branching, we add only linking variables with fractional values if $x_L^t \notin \mathbb{Z}^L$. If $x_L^t \in \mathbb{Z}^L$, then we are forced to add integer-valued linking variables that remain unfixed to the list. If all linking variables are fixed, then we have several options, depending on how parameters are set (see Section 3.2).

- We may prune the node after solving problem (UB), as explained in Section 2.2.
- When $(x^t, y^t) \notin X \times Y$, we may add non-linking variables with fractional values to the candidate set.
- (x^t, y^t) may be removed by generating a cut (after solving (SL-MILP) if $(x^t, y^t) \in X \times Y$, see Section 2.4).

Branching on Fractional Variables. Naturally, we may also consider a more traditional branching scheme in which we only branch on variables with fractional values. In such a scheme, we must consider branching on all variables (first- and second-level), since we may have no other option when $x^t \in X$.

Selecting Candidates. We have so far neglected to address the question of *which* variable to choose as the basis for our branching disjunction when there is more than one candidate. The idea of limiting branching only to the linking variables differs from the usual scheme employed in algorithms for MILP in that we only consider a subset of the integer variables for branching. Nevertheless, in both of the schemes described above, we face a similar decision regarding which of the (possibly many) variables available for branching should actually be chosen. It is well-known that the choice of branching variable can make a substantial difference in practical performance.

In branching procedures for MILP, it is typical to first choose a set of candidates for branching (typically, all integer variables with fractional values are considered). The selection of the "best" branching candidate is then made from this set of candidates based on one of a number of schemes for scoring/predicting the "impact" of choosing a particular branching variable. The details of these scoring mechanisms can be found in any number of references (see, e.g., Achterberg et al. [2005]). Although the method used to define the set of branching candidates differs from the one used in MILP, the method employed in `MibS` for selecting from the candidates is similar to the schemes that can be found in the literature. `MibS` offers the pseudocost, strong branching and reliability branching schemes.

2.4 Cutting

The generation of valid inequalities is an alternative to branching for removing infeasible and/or non-improving solutions. We refer the reader to [Marchand et al., 2002] and [Wolter, 2006] for theoretical background on the separation problem and cutting plane methods in general. In the case of MIBLP, the valid inequalities we aim to generate are those that separate the solution to the current relaxation from the convex hull of the (improving) bilevel feasible solutions. These inequalities are generated iteratively in order to improve the relaxation, as in a standard cutting plane method. DeNegre and Ralphs [2009] showed that such a pure cutting plane algorithm can be used to solve certain classes of MIBLPs.

In order to allow the separation of points that are in conv($\mathcal{F}$) but have already been identified (or can be shown not to be improving), we recall here the notion of improving valid inequality that we introduced in Definition 3, re-stating it in the notation of (MIBLP-VF). A triple $(\alpha^x, \alpha^y, \beta) \in \mathbb{R}^{n_1+n_2+1}$ is said to constitute an *improving valid inequality* for $\mathcal{F}$ if

$$\alpha^x x + \alpha^y y \geq \beta \; \forall (x,y) \in \text{conv}\left(\left\{(x,y) \in \mathcal{F} \;\middle|\; cx + d^1 y < U\right\}\right),$$

where U is the current global upper bound. As mentioned before, we shall drop the word "improving" from the remaining discussion.

In Section 2.1, we stated that the difference between the feasible region of a subproblem and that of the original problem is only in changes to the bounds on variables. This is no longer strictly true when valid inequalities are also being generated and added dynamically to the constraint set of the relaxation. The feasible region $\mathcal{P}^t$ of the relaxation at node t from (LR^t) then becomes

$$\mathcal{P}^t = \left\{(x,y) \in \mathcal{P} \cap \Pi^t \;\middle|\; l_x^t \leq x \leq u_x^t, l_y^t \leq y \leq u_y^t\right\},$$

where Π^t is a polyhedron representing the valid inequalities applied at node t. The set of valid inequalities may include inequalities generated at any of the ancestor nodes in the path to the root node of the search tree.

There are several distinct categories of valid inequality that can be generated, depending on the feasibility conditions violated by the solution (x^t, y^t) that we are trying to separate at node t. Generally speaking, all valid inequalities can be classified as

- Feasibility cuts: Inequalities valid for conv $\left\{(x,y) \in \mathcal{S} \;\middle|\; cx + d^1 y < U\right\}$.
- Optimality cuts: Inequalities valid for conv $\left\{(x,y) \in \mathcal{F} \;\middle|\; cx + d^1 y < U\right\}$.
- Projected optimality cuts: Inequalities valid for conv $(\{(x,y) \in \mathbb{R}^{n_1 \times n_2} \mid x \in \mathcal{F}_1, cx + \Xi(x) < U\})$.

The feasibility cuts include those classes generally employed in solution of MILPs. Inequalities used to remove solutions $(x^t, y^t) \in \mathcal{S}$ at node t may be referred to roughly as *optimality cuts*, while *projected optimality cuts* are those that may remove all solutions with specific first-level values. For example, inequalities that remove the solutions with the same linking component as x^t once the problem (UB) with $\gamma = x_L^t$ has been solved. Clearly, these three classes are not entirely distinct and the categorization is meant only to provide a rough categorization.

`MibS` includes separation routines for a variety of known classes of valid inequalities in the current literature. We give a brief overview here. More details are provided in a companion paper that analyzes the impact of various classes of inequalities in greater detail.

Integer No-Good Cut. This class of inequalities are valid for problems in which $r_1 = n_1$, $r_2 = n_2$ and all problem data are integer (with the possible exception of the objective function). It was introduced by DeNegre and Ralphs [2009] and is derived from a split disjunction obtained by taking combinations of inequalities binding at (x^t, y^t) in a fashion similar to that used in deriving the Chvátal cuts valid for the feasible regions of MILPs. The cut eliminates a single extremal solution that is integral yet not bilevel feasible from the feasible region of the relaxation. It is easy to derive such a cut under the given assumptions.

Generalized No-good Cut. This class of inequalities are valid for problems in which all linking variables are binary. It is a generalization of the *no-good cut* introduced by DeNegre [2011] in the context of MIBLPs. Similar cuts have been used in many other contexts. The idea of a "no-good cut" seems to have first been suggested by Balas and Jeroslow [1972]. The purpose of this cut is to remove all solutions for which the linking variables have certain fixed values.

If at node t, we have $x_L^t \in \mathbb{Z}^L$ and either (i) we choose to solve (SL-MILP) and it is infeasible or (ii) we choose to solve (UB), then we can store x_L^t in the linking solution pool (if applicable) and add a generalized no-good cut with $\gamma = x_L^t$ in order to avoid generating the same linking solution again. This inequality can also be added in other subproblems if/when this linking solution arises again.

Intersection Cut. Fischetti et al. [2017a,b] generalized the well-known concept of an intersection cut [Balas, 1971], used extensively in the MILP setting, to MIBLPs. The various classes introduced differ from each other in the way the underlying "bilevel-free" convex set used for generating the cuts is defined. Three classes of introduced cuts are working for the problems with $G^2y - A^2x \in \mathbb{Z}$ and $d^2y \in \mathbb{Z}$ for all $(x, y) \in \mathcal{S}$. These cuts can be employed for removing the infeasible solution $(x^t, y^t) \notin \mathcal{F}$ from the feasible region, but it is not guaranteed when $(x^t, y^t) \notin \mathcal{S}$. The other class known as *hypercube intersection cut* is valid for the MIBLPs with $x_L \subseteq \mathbb{Z}^L$ and can be added when $x_L^t \in \mathbb{Z}^L$. We solve (UB) and store the solution in the linking solution pool (if applicable) and after doing so, the hypercube intersection cut can be added. This inequality is guaranteed to be violated by (x^t, y^t) and *may* also eliminate *some* other solutions for which $x_L = x_L^t$, but is not guaranteed to eliminate all such solutions.

Increasing Objective Cut. This class of inequalities are valid for problems in which linking variables are binary and $A^2 \geq 0$. Proposed by DeNegre [2011], it is a disjunctive cut derived from the following valid disjunction based on the value function bound (VFB).

$$X_1 = \left\{(x, y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; \sum_{i \in L: x_i^t = 0} x_i = 0, d^2 y \leq d^2 \hat{y}\right\}, \quad X_2 = \left\{(x, y) \in \mathbb{R}^{n_1 \times n_2} \;\middle|\; \sum_{i \in L: x_i^t = 0} x_i \geq 1\right\},$$

where $\hat{y} \in \mathcal{R}(x^t)$. In addition to the way in which this disjunction was exploited by DeNegre [2011], there are other inequalities that could also be derived from this disjunction. This cut can be applied to eliminate $(x^t, y^t) \in X \times Y$ from the feasible region at node t when (x^t, y^t) is found to be infeasible. Note that by solving (UB) following the feasibility check, we could also apply the stronger generalized no-good cut in this situation.

Benders Cut. This class of inequalities are valid for problems in which linking variables are binary, $G^2 \leq 0$, and corresponding to each linking variable x_i, there exists y_i so that $x_i = 1$ results $y_i = 0$. It was originally mentioned by Caprara et al. [2016] in the context of knapsack interdiction problems, but is valid for a broader class of problems. This cut utilizes the value function bound (VFB) and a new such cut can be imposed anytime we find a new feasible solution.

2.5 Primal Heuristics

Just as in the solution of MILPs, the primal heuristics can be employed within a branch-and-cut algorithm for solving MIBLPs in order to enhance the discovery of bilevel feasible solutions. Three different heuristics introduced in [DeNegre, 2011] are implemented in `MibS`, as follows.

Improving Objective Cut Heuristic. Let $(\bar{x}, \bar{y}) \in \mathcal{S}$ and $\hat{y} \in \mathcal{R}(\bar{x})$. As we discussed in Section 2.1, $(\bar{x}, \hat{y})$ is a bilevel feasible solution if it satisfies the first-level constraints. However, $(\bar{x}, \hat{y})$ is not necessarily a good solution with respect to the first-level objective function. The idea of this heuristic is to exploit the information from both $(\bar{x}, \bar{y})$ (which is a good solution with respect to the first-level objective) and $(\bar{x}, \hat{y})$ (which is a likely bilevel feasible solution). To do so, we first determine

$$(\tilde{x}, \tilde{y}) \in \operatorname{argmin} \left\{ cx + d^1 y \;\middle|\; (x, y) \in \mathcal{S}, d^2 y \leq d^2 \hat{y} \right\}.$$

When $(\bar{x}, \hat{y}) \in \mathcal{F}$, such $(\tilde{x}, \tilde{y})$ must exist. Further, we must have $(\tilde{x}, \tilde{y}) \in \mathcal{F}$. A separate feasibility check is thus required and this check is what is expected to finally produce the (potentially) improved solution.

Second-level Priority Heuristic. This heuristic is similar to the improving objective cut heuristic in that it also tries to balance bilevel feasibility with the quality of obtained solution. The MILP solved with this heuristic, however, is

$$\min \left\{ d^2 y \;\middle|\; (x, y) \in \mathcal{S}, cx + d^1 y \leq U \right\}, \tag{PH}$$

where U is the current upper bound. In this problem, the goal of the added constraint is to improve the quality of the solution, while the objective function of this problem increases the chance of generating a bilevel feasible solution (however, it does not guarantee the bilevel feasibility of the produced solution). Note that the upper bound U could be replaced by any chosen "target" to try to ensure improvement on the current incumbent, but then we would not be assured feasibility of (PH).

Weighted Sums Heuristic. This heuristic employs techniques from multi-objective optimization to generate bilevel feasible solutions. The main idea of this heuristic is finding a subset of *efficient* solutions (those for which we cannot improve one of the objectives without degrading the other while maintaining feasibility [Ehrgott and Wiecek, 2005]) of the following problem by using a weighted-sum subproblem [Geoffrion, 1968].

$$\operatorname*{argvmin}_{(x,y)\in\mathcal{S}} \{cx + d^1 y, d^2 y\},$$

where the operator argvmin means finding a solution (or more than one) that is efficient. For more details, see DeNegre [2011].

3 Software Framework

In this section, we describe the implementation of the `MibS` software [DeNegre et al., 2019]. This paper refers to version `1.1`, the latest released version at the time of this writing. The overall

algorithm employed in MibS combines the procedures described in Section 2 in a fashion typical of existing branch-and-cut algorithms for other problems. In fact, MibS is built on top of BLIS, a parallel implementation of branch and cut for the solution of MILPs [Xu et al., 2009]. In Section 3.1, we describe the class structure of the C++ code that encapsulates the implementation. Furthermore, there are a number of important parameters, described in Section 3.2, that determine how the various components described in Section 2 are coordinated. Although there are defaults that are set automatically after analyzing the structure of a particular instance, these parameters can and should be tuned for use in particular applications, Understanding their role in the algorithm is crucial to understanding the overall strategy, described in Section 3.3. Finally, in Section 3.4, we describe important implementational issues surrounding how the second-level problem is solved.

3.1 Design of MibS

We first briefly describe the overall design of the software. MibS is an open-source implementation in C++ of the branch-and-cut algorithm described in Section 3.3. In addition to a core library, MibS utilizes a number of other open-source packages available from the Computational Infrastructure for Operations Research (COIN-OR) repository [COIN-OR]. These packages include the following.

- The COIN-OR High Performance Parallel Search (CHiPPS) Framework [Ralphs et al., 2004], which includes a hierarchy of three libraries.
  - Abstract Library for Parallel Search (ALPS) [Xu and Ralphs, 2019a, Xu et al., 2005]: This project is utilized for managing the global branch and bound.
  - Branch, Constrain, and Price Software (BiCePS) [Ralphs et al., 2004, Xu and Ralphs, 2019b]: This project provides base classes for objects used in MibS.
  - BiCePS Linear Integer Solver (BLIS) [Xu et al., 2009, Xu and Ralphs, 2019c]: This project is the parallel MILP Solver framework from which MibS is derived.
- COIN-OR Linear Programming Solver (CLP) [Forrest, 2017b]: MibS employs this software for solving the LPs arising in the branch-and-cut algorithm.
- SYMPHONY [Ralphs et al., 2019, Ralphs and Güzelsoy, 2005]: This software is used for solving the MILPs required to be solved in the branch-and-cut algorithm.
- COIN-OR Cut Generation Library (CGL) [Cgl]: Both MibS and SYMPHONY utilize this library to generate valid inequalities valid for MILPs.
- COIN-OR Open Solver Interface (OSI) [Osi]: This project is used for interfacing with solvers, such as SYMPHONY, Cbc and CPLEX.

MibS is comprised of a number of classes that are either specific to MibS, or are classes derived from a class in one of the libraries of CHiPPS. The main classes of MibS are as follows.

- MibSModel: This class is derived from the BlisModel class. It extracts and stores the model from the input files, which consist of an MPS file and an auxiliary information file. We refer the reader to the README file of MibS for a comprehensive description of the input file format.

- `MibSBilevel`: This class is specific to `MibS` and is utilized for checking the bilevel feasibility of solutions of the relaxation problem. Furthermore, this class finds the bilevel feasible solutions, as described in lines 18, 23 and 28 of Algorithm 1.
- `MibSTreeNode`: This class is derived from the class `BlisTreeNode` and contains the methods for processing the branch-and-bound nodes.
- `MibSCutGenerator`: This class is specific to `MibS` and contains the methods for generating the valid inequalities described in Section 2.4.
- `MibSBranchStrategyXyz`: These classes (one for each strategy `MibS` can use for selecting the final branching candidate: `Pseudo`, `Strong`, and `Reliability`) are derived from the parent classes `BlisBranchStrategyXyz` and contain the implementation used for selecting the final branching candidate.
- `MibSHeuristic`: This class is specific to `MibS` and contains the methods for generating heuristic solutions by employing the primal heuristics illustrated in Section 2.5.
- `MibSSolution`: This is a class derived from `BlisSolution` class and is utilized for storing the bilevel feasible solutions.

Since one important feature of a practical solver is ease of use, we have tried to make `MibS` as user-friendly as possible. Prior to the solution process, `MibS` analyzes the problem to determine its properties, e.g., the type of instance (interdiction or general), the type of variables present at each level (continuous, discrete, or binary) and signs of the coefficients in the constraint matrices. It then checks the parameters set by the user and modifies them if it determines that those values of parameters are not valid for this problem, informing the user of any change in the parameters. For example, `MibS` turns off any cuts selected by the user that are not valid for the instance to be solved.

3.2 Parameters

A wide range of parameters are available for controlling the algorithm.

Branching Strategy. There are several parameters that control the branching strategy. The main one is `branchStrategy`, which controls what variables we allow as branching candidates. The options are

- `linking`: Branch only on linking variables, as long as any such variable remains unfixed, with priority given to such variables with fractional values.
- `fractional`: Branch on any integer first- or second-level variable that has fractional value, as is traditional in solving MILPs.

We also have parameters for controlling the strategy by which the final branching candidate is selected (`strong, pseudocost, reliability`). The default is to use `pseudocost` scoring.

Search Strategy. The search strategy used by `MibS` is controlled by ALPS, the underlying tree search framework. The default is the best-first strategy.

Cutting Strategy. There are parameters for the types of valid inequalities to generate and the strategy for when to generate them (only in the root node, periodically, etc.). Note that we are forced to generate cuts whenever there are no available branching candidates and the current node cannot be pruned. With `branchStrategy` set to `linking`, the parameters for solving problems (SL-MILP) and (UB) (which are described later) can be set so that a pure branch and bound is possible, but this is not possible for the `fractional` case. The default settings for cuts depends on the instance. `MibS` includes an automatic analyzer that determines which classes of cuts are applicable and likely to be effective for a given instance. The frequency of generation is selected automatically by default, based on statistics gathered during early stages, as is standard in many solvers.

Primal Heuristics. Types of primal heuristics to employ and the strategy for how often to employ them (see Section 2.5). Only BLIS (generic MILP) heuristics are turned on by default.

Linking Solution Pool. There is a parameter `useLinkingSolutionPool` that determines whether to maintain a pool $\mathcal{L}$ of linking solutions seen so far, as described earlier, in order to avoid solving identical instances of (SL-MILP) and (UB). When the parameter is set to `True`, we check $\mathcal{L}$ before solving either of (SL-MILP) or (UB).

Each linking solution stored in $\mathcal{L}$ is stored along with both a status tag and, if appropriate, an associated solution. The status tag is one of the following:

- `secondLevelIsInfeasible`: If the corresponding problem (SL-MILP) is solved and it is infeasible.
- `secondLevelIsFeasible`: If the corresponding problem (SL-MILP) is solved and it has an optimal solution, but problem (UB) is not solved.
- `UBIsSolved`: If the corresponding problem (UB) is solved.

All linking solutions are stored in a hash table in order to enable an efficient membership check.

Feasibility Check. The following binary parameters determine the strategy for when to solve the second-level problem (SL-MILP) (for details on solution of the second-level problem, see Section 3.4).

- `solveSecondLevelWhenLVarsFixed`: Whether to solve when $l^t_{x_L} = u^t_{x_L} = x^t_L$ (linking variables fixed).
- `solveSecondLevelWhenLVarsInt`: Whether to solve when $x^t_L \in \mathbb{Z}^L$.
- `solveSecondLevelWhenXVarsInt`: Whether to solve when $x^t \in X$.

- `solveSecondLevelWhenXYVarsInt`: Whether to solve when $(x^t, y^t) \in X \times Y$.

When problem (SL-MILP) is solved, we have the following implications, depending on the result.

- If (SL-MILP) is infeasible, then all solutions (x, y) with $x_L = x_L^t$ are infeasible and can be removed from the feasible region of the relaxation either by generation of a cut or by branching. To avoid solving problems (SL-MILP) for a different solution with the same linking part, the tuple $[x_L^t, \texttt{secondLevelIsInfeasible}]$ can be added to the linking solution pool $\mathcal{L}$.
- If (SL-MILP) has an optimal solution, we can avoid solving problem (SL-MILP) for any (x, y) for which $x_L = x_L^t$ arising in the future by adding the tuple $[x_L^t, \texttt{secondLevelIsFeasible}]$ and the associated solution to (SL-MILP) to the linking solution pool $\mathcal{L}$.

Regardless of parameter settings, we must always solve (SL-MILP) whenever $(x^t, y^t) \in X \times Y$, $x_L^t \notin \mathcal{L}$ (we have not previously solved (SL-MILP) for x_L^t), and there are no branching candidates. Clearly, if we have branching candidates, then we can avoid solution of (SL-MILP) by branching. Similarly, if $(x^t, y^t) \notin X \times Y$, we can generate standard MILP cuts. Otherwise, we must have either

- `branchStrategy` is `fractional`, in which case we must solve (SL-MILP) and then may either remove (x^t, y^t) by generating a valid inequality (if infeasible) or prune the node (if feasible).
- `branchStrategy` is set to `linking` and all linking variables are fixed, in which case we must also solve (UB) (if (SL-MILP) is feasible) and prune the node.

Computing Best UB. The following binary parameters determine when the problem (UB) should be solved in order to compute the best bilevel feasible solution (x, y) with $x_L = x_L^t$ (if such solution exists).

- `computeBestUBWhenLVarsFixed`: Whether to solve when $l_{x_L}^t = u_{x_L}^t = x_L^t$.
- `computeBestUBWhenLVarsInt`: Whether to solve when $x_L^t \in \mathbb{Z}^L$.
- `computeBestUBWhenXVarsInt`: Whether to solve when $x^t \in X$.

After solving (UB), we know that no solution (x, y) with $x_L = x_L^t$ can be improving and thus, all such solutions can be removed either by generation of a cut or by branching. To avoid solving problem (UB) for any solutions (x, y) arising in the future for which $x_L = x_L^t$, the tuple $[x_L^t, \texttt{secondLevelIsFeasible}]$ should be replaced with the tuple $[x_L^t, \texttt{UBIsSolved}]$ and the associated solution stored in the linking solution pool $\mathcal{L}$.

Note that it would be possible to solve (UB^t) rather than (UB) if the goal were only to prune the node. Solving (UB) (which may not be much more difficult) allows us to exploit the solution information globally through the linking solution pool.

Regardless of parameter setting, we must always solve problem (UB) (if it has not been previously solved for x_L^t) when (i) `branchStrategy` is `linking`, (ii) $(x^t, y^t) \in X \times Y$ and is bilevel infeasible and (iii) all linking variables are fixed. This does not mean that in this case, solving problem (UB) (and further fathoming node t) is the only algorithmic option, as explained in Section 2.3.

3.3 Outline of the Algorithm

Algorithm 1 gives the general outline of the node processing loop in this branch-and-cut algorithm. Note that for simplicity, the algorithm as stated assumes that the linking pool is used, though the option of not using this pool is also provided. In almost all cases, use of the linking pool is advantageous (see Section 4.3). At a high level, the procedure consists of the following steps.

1. Solve the relaxation (LR^t) (line 3) and prune the node (lines 5–6) if either
   - (LR^t) is infeasible;
   - $L^t \geq U$; or
   - x_L is fixed and it has been stored in set $\mathcal{L}$ with either `secondLevelIsInfeasible` or `UBIsSolved` tags.
2. If (SL-MILP) was not previously solved with respect to x_L^t, then depending on (x^t, y^t) and the parameter settings, we may next solve it (lines 7–8).
   - (SL-MILP) is infeasible $\Rightarrow x^t \notin \mathcal{F}_1$ (line 9)
     - Add $[x_L^t, \texttt{secondLevelIsInfeasible}]$ to $\mathcal{L}$ (line 10).
     - If x_L is fixed, fathom node t (lines 11–12).
   - (SL-MILP) is feasible $\Rightarrow$ add $[x_L^t, \texttt{secondLevelIsFeasible}]$ to $\mathcal{L}$ (lines 13–14).
3. If (SL-MILP) was solved now or previously and is feasible (line 15), we have either
   - $(x^t, y^t) \in \mathcal{F} \Rightarrow$ update U and fathom node t (lines 17–19).
   - $(x^t, y^t) \notin \mathcal{F}$
     - Update U if $(x^t, \hat{y}^t) \in \mathcal{F}$ (in case of not solving (UB)) (lines 27–28) and eliminate (x^t, y^t) (lines 29–34).
     - If (UB) was not previously solved with respect to x_L^t, then depending on (x^t, y^t) and the parameter settings, we may solve it (lines 20–21).
       * (UB) is feasible $\Rightarrow$ update U (lines 22–23).
       * Remove $[x_L^t, \texttt{UBIsSolved}]$ from set $\mathcal{L}$ and add $[x_L^t, \texttt{UBIsSolved}]$ (line 24).
       * If x_L is fixed, fathom node t (lines 25–26).
4. Finally, we must either branch or remove (x^t, y^t) by adding valid inequalities (lines 29–34).
   - If there are no branching candidates, then we must remove (x^t, y^t) by adding valid inequalities (lines 29–30).
   - If (SL-MILP) was not solved now or previously, we must branch if $(x^t, y^t) \in X \times Y$ (lines 31–32).
   - Otherwise, we have the choice of either adding valid inequalities or branching (lines 33–34).

Algorithm 1: Node processing loop in `MibS`

Input : [Set $\mathcal{L}, U$]
Output: [Set $\mathcal{L}, U, L^t, U^t$]

`branch` $\leftarrow$ `False`, $L^t \leftarrow -\infty$, $U^t \leftarrow \infty$
while `branch` is `False` **do**
  Solve (LR^t)
  $L^t \leftarrow$ The optimal value of (LR^t)
  if (LR^t) is infeasible **or** $L^t \geq U$ **or**
  $(l^t_{x_L} = u^t_{x_L}$ **and** $([x^t_L,$ `secondLevelIsInfeasible`$] \in \mathcal{L}$ **or** $[x^t_L,$ `UBIsSolved`$] \in \mathcal{L}))$ **then**
    Fathom node t
  if $[x^t_L, \cdot] \notin \mathcal{L}$ **and**
  ((`branchStrategy` is `linking` **and** $(x^t, y^t) \in X \times Y$ **and** $l^t_{x_L} = u^t_{x_L}$) **or**
  (`branchStrategy` is `fractional` **and** $(x^t, y^t) \in X \times Y$) **or**
  (`solveSecondLevelWhenXYVarsInt` **and** $(x^t, y^t) \in X \times Y$) **or**
  (`solveSecondLevelWhenXVarsInt` **and** $x^t \in X$) **or**
  (`solveSecondLevelWhenLVarsInt` **and** $x_L \in \mathbb{Z}^L$)**or**
  (`solveSecondLevelWhenLVarsFixed` **and** $l^t_{x_L} = u^t_{x_L}$)) **then**
    Solve (SL-MILP) to find $\phi(A^2 x^t)$
    if (SL-MILP) is infeasible **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup [x^t_L,$ `secondLevelIsInfeasible`$]$
      if $l^t_{x_L} = u^t_{x_L}$ **then**
        Fathom node t
    else
      $\mathcal{L} \leftarrow \mathcal{L} \cup [x^t_L,$ `secondLevelIsFeasible`$]$
  if $[x^t_L,$ `secondLevelIsFeasible`$] \in \mathcal{L}$ **or** $[x^t_L,$ `UBIsSolved`$] \in \mathcal{L}$ **then**
    $\hat{y}^t \leftarrow$ The optimal solution of (SL-MILP)
    if $(x^t, y^t) \in \mathcal{F}$ **then**
      $U^t \leftarrow cx^t + d^1 y^t$, $U \leftarrow \min\{U, U^t\}$
      Fathom node t
    if $[x^t_L,$ `UBIsSolved`$] \notin \mathcal{L}$ **and**
    ((`branchStrategy` is `linking` **and** $(x^t, y^t) \in X \times Y$ **and** $l^t_{x_L} = u^t_{x_L}$) **or**
    (`computeBestUBWhenXVarsInt` **and** $x^t \in X$) **or**
    (`computeBestUBWhenLVarsFixed` **and** $l^t_{x_L} = u^t_{x_L}$) **or**
    (`computeBestUBWhenLVarsInt`)) **then**
      Solve (UB)
      if (UB) is feasible **then**
        $U \leftarrow \min\{U,$ optimal value of (UB)$\}$
      $\mathcal{L} \leftarrow \mathcal{L} \setminus [x^t_L,$ `secondLevelIsFeasible`$]$, $\mathcal{L} \leftarrow \mathcal{L} \cup [x^t_L,$ `UBIsSolved`$]$
      if $l^t_{x_L} = u^t_{x_L}$ **then**
        Fathom node t
    else if $(x^t, \hat{y}^t) \in \mathcal{F}$ **then**
      $U \leftarrow \min\{U, cx^t + d^1 \hat{y}^t\}$
  if `branchStrategy` is `fractional` **and** $(x^t, y^t) \in X \times Y$ **then**
    Remove (x^t, y^t) by generating a cut
  else if $[x^t_L, \cdot] \notin \mathcal{L}$ **and** $(x^t, y^t) \in X \times Y$ **then**
    `branch` $\leftarrow$ `True`
  else
    Remove (x^t, y^t) by generating a cut or `branch` $\leftarrow$ `True`

In the case of not using the linking solution pool, the structure of algorithm is similar to Algorithm 1, but differs in the steps where information from the linking solution pool is exploited or new information is added to this pool. For example, the lines 10, 14 and 24 are eliminated and the lines 5, 7, 15 and 20 are modified.

As mentioned earlier, this algorithm can be adapted for other risk functions. For example, the pessimistic risk function (RF-PES) can be accommodated with a few modifications as follows. The same relaxation is used, but the feasibility check is slightly different. At node t, if $(x^t, y^t) \notin X \times Y$, it is infeasible and should be removed, as usual. Otherwise, we compute $\phi(A^2x^t)$ and check whether $d^2y^t = \phi(A^2x^t)$. If $d^2y^t > \phi(A^2x^t)$, (x^t, y^t) is infeasible, as in the optimistic case, but $d^2y^t = \phi(A^2x^t)$ does not guarantee its feasibility. Feasibility of (x^t, y^t) must be verified by also ensuring that d^1y^t is equal to the optimal value of the MILP

$$\max \left\{ d^1 y \;\middle|\; y \in \mathcal{P}_1(x^t) \cap \mathcal{P}_2(x^t) \cap Y, d^2 y \leq \phi(A^2 x^t) \right\}.$$

The approach for finding the best feasible solution with $x_L = \gamma \in \mathbb{Z}^L$ is also slightly different. To do so, we solve problem (UB) in the optimistic case, but in the pessimistic case, it is equal to $c\bar{x} + d^1\bar{y}$, where $(\bar{x}, \bar{y}) \in X \times Y$ is the optimal solution of the MILP

$$\min\{cx - d^1 y \mid x \in X, y \in \mathcal{P}_1(x) \cap \mathcal{P}_2(x) \cap Y, d^2 y \leq \phi(A^2 x), x_L = \gamma\}.$$

The branching steps are the same as the optimistic case, but some of the valid inequalities may need to be modified. The details are beyond the scope of this paper.

3.4 Solving the Second-level Problem

It is evident that in most cases, much (if not most) of the computational effort in executing the algorithm arises from the time required to solve (SL-MILP) and (UB) (See Table 2). A major focus of ongoing work, therefore, is the development of methodology to reduce the time spent solving these problems.

In closely related work on solving two-stage stochastic mixed integer optimization problems, methodology for warm-starting the solution process of an MILP using information derived from previously solved instances has been developed. Hassanzadeh and Ralphs [2014] described a method for solving a sequence of MILPs differing only in the right-hand side. It is shown that a sequence of such solves can be performed within a single branch-and-bound tree, with each solve starting where the previous one left off. Under mild conditions, this method can be used to construct a complete description of the value function ϕ. This method of solving such a sequence of MILPs has been implemented within the SYMPHONY MILP solver [Ralphs et al., 2019, Ralphs and Güzelsoy, 2006, 2005], which is one of several supported solvers that can be used for solution of (SL-MILP) and (UB) within `MibS`. Other supported solvers include CPLEX [CPLEX] and Cbc [Forrest, 2017a]. The parameter `feasCheckSolver` determines which MIP solver to be employed.

4 Computational Results

A number of experiments were conducted to evaluate the impacts of the various algorithmic options provided by `MibS`. The parameters investigated in this section are

- The parameters that determine when the problems (SL-MILP) and (UB) should be solved.
- The parameter that determines the branching strategy.
- The parameter that determines whether to use the linking solution pool or not.
- The parameters that determine whether to use the primal heuristics or not.

Because of space constraints and to allow more in-depth analysis, testing of the effectiveness of various classes of valid inequalities and parameters for controlling them is not included here, but will be the subject of a future study. Three different data sets (171 instances in total) were employed in our experiments as follows.

- `IBLP-DEN`: This set was generated by DeNegre [2011], and contains 50 instances with 15–20 integer variables and 20 constraints, all at the second level.
- `IBLP-FIS`: This is a selected set of 21 of the instances generated by Fischetti et al. [2017a]. These instances originate from `MILPLIB 3.0` [Bixby et al., 1998] and all variables are binary and all constraints are at the second level.
- `MIBLP-XU`: This set was introduced by Xu and Wang [2014] and includes 100 randomly-generated instances. In these problems, all first-level variables are integer with upper bound 10, while the second-level variables are continuous with probability 0.5. The number of first- and second-level variables are equal and n_1 is in the range of $10 - 460$ with an increments of 50. Furthermore, the number of first-level and second-level constraints are equal to $0.4n_1$. To have specific bounds on all integer variables, we added the very loose upper bound 1500 for all integer second-level variables in converting these instances to the `MibS` format.

Table 1 summarizes the properties of the data sets. Note that in the described data sets, all first-level variables are linking.

Table 1: The summary of data sets

Data Set	First-level Vars Num	Second-level Vars Num	First-level Cons Num	Second-level Cons Num	First-level Vars Type	Second-level Vars Type	Size
IBLP-DEN	5-15	5-15	0	20	discrete	discrete	50
IBLP-FIS	4-2481	2-2480	0	16-4944	binary	binary	21
MIBLP-XU	10-460	10-460	4-184	4-184	discrete	continuous, discrete	100

All computational results we report were generated on compute nodes running the Linux (Debian 8.7) operating system with dual AMD Opteron 6128 processors and 32 GB RAM and all experiments were run sequentially. The time limit was 3600 seconds and the pseudocost branching strategy was used to choose the best variable among the branching candidates for all experiments. In all numerical experiments, the generation of generic MILP cuts by the Cut Generation Library [Cgl] was disabled, since these cuts seem unlikely to be effective in addition to the classes specific to MIBLPs

and their integration can cause other algorithmic issues that would first need to be addressed. SYMPHONY was employed as the MILP solver (preprocessing and primal heuristics were turned off) in all experiments, unless otherwise noted. In all experiments, the integer no-good cut was employed for solving the instances of `IBLP-DEN` and `IBLP-FIS` sets, and the problems belonging to the `MIBLP-XU` set were solved by using the hypercube intersection cut. Furthermore, all primal heuristics of `MibS` were disabled in the numerical experiments except as otherwise noted, since these have in general also not proven to be very effective.

All instances were initially solved by all methods described in Sections 4.1–4.3 below, but in plotting performance profiles, we chose the 125 problems that could be solved by at least one method in 3600 seconds and whose solution time exceeds 5 seconds for at least one method. This test set was used for all plots and tables shown in Section 4. The details of the results employed for plotting all of these figures and tables, are shown in the appendix (the reported running times do not include the required time for reading the instances).

4.1 Impact of Strategy for Solving (SL-MILP) and (UB)

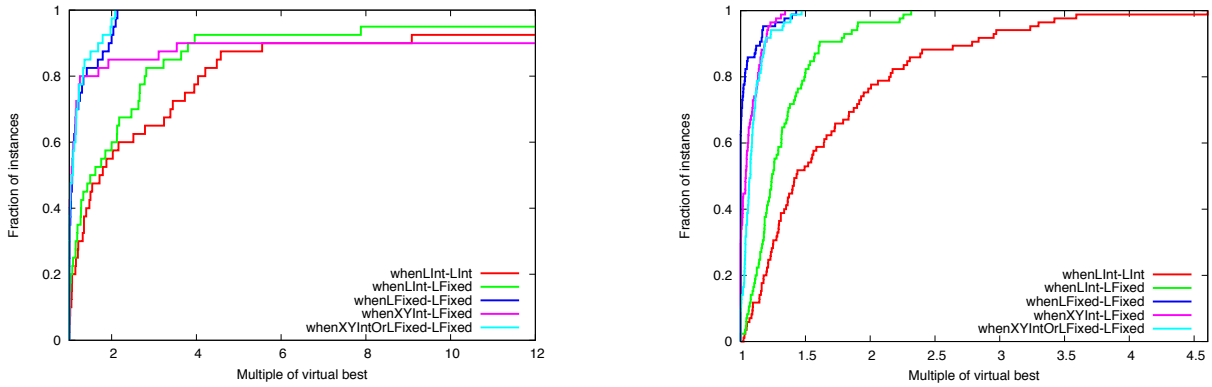
In order to evaluate the impacts of the parameters for solving problems (SL-MILP) and (UB), we employed five different methods:

- `whenLInt-LInt`: Problems (SL-MILP) and (UB) were both solved only when $x_L^t \in \mathbb{Z}^L$, i.e., the parameters `solveSecondLevelWhenLVarsInt` and `computeBestUBWhenLVarsInt` were set to `true`.
- `whenLInt-LFixed`: Problem (SL-MILP) was solved when $x_L^t \in \mathbb{Z}^L$ and problem (UB) was solved when all linking variables are fixed, i.e., the parameters `solveSecondLevelWhenLVarsInt` and `computeBestUBWhenLVarsFixed` were set to `true`.
- `whenLFixed-LFixed`: Problems (SL-MILP) and (UB) were both solved only when linking variables were fixed, i.e., `solveSecondLevelWhenLVarsFixed` and `computeBestUBWhenLVarsFixed` were set to `true`.
- `whenXYInt-LFixed`: Problem (SL-MILP) was solved only when $(x^t, y^t) \in X \times Y$ and problem (UB) was solved only when, in addition, all linking variables were fixed, i.e., the parameters `solveSecondLevelWhenXYVarsInt` and `computeBestUBWhenLVarsFixed` were set to `true`.
- `whenXYIntOrLFixed-LFixed`: Problem (SL-MILP) was only solved when $(x^t, y^t) \in X \times Y$ or all linking variables were fixed and problem (UB) was solved only whenever all linking variables are fixed, i.e., `solveSecondLevelWhenXYVarsInt`, `solveSecondLevelWhenLVarsFixed` and `computeBestUBWhenLVarsFixed` were set to `true`.

In this first set of experiments, the `branchStrategy` was set to `linking` and `useLinkingSolutionPool` was set to `true`. The performance profiles shown in Figure 2 compare the solution time of the five described methods.

Figure 2a shows the results for the `IBLP-DEN` and `IBLP-FIS` sets and Figure 2b shows the results for the `MIBLP-XU` set. These figures show the superiority of `whenLFixed-LFixed` and `whenXYIntOrLFixed-LFixed`

over the other three methods, with roughly the same performance for each. Based on these results, the settings for `whenXYIntOrLFixed-LFixed` have been chosen as the default setting for `MibS` and in the remainder of these experiments unless otherwise noted.



(a) `IBLP-DEN` and `IBLP-FIS` sets

(b) `MIBLP-XU` set

Figure 2: Impact of the parameters for solving problems (SL-MILP) and (UB).

4.2 Impact of Branching Strategy

As mentioned earlier, the `branchStrategy` parameter controls which variables are considered branching candidates and can be set to `fractional` and `linking`. In order to evaluate the effect of the parameter, we compared the performance of these two methods. The linking solution pool was used in both cases.

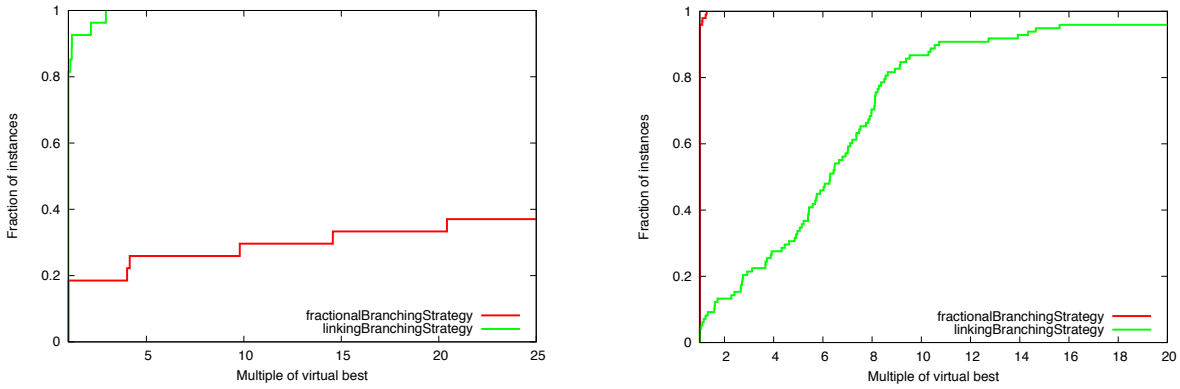
In initial testing, we observed a possible relationship between the number of integer first- and second-level variables and the performance of branching strategies. Hence, we further analyzed the results by dividing them into two separate sets with $r_1 \leq r_2$ (27 instances) and $r_1 > r_2$ (98 instances). Figure 3 shows the performance profiles for these two sets with the solution time as the performance measure.

Figure 3a shows that `linking` generally performed better when $r_1 \leq r_2$, while Figure 3b indicates that the `fractional` branching strategy performed better when $r_1 > r_2$. Based on these results, the default value of `branchStrategy` parameter in `MibS` has been set to `linking` and `fractional` for the instances with $r_1 \leq r_2$ and $r_1 > r_2$, respectively. In general, it would not be easy to predict which branching strategy will behave better for a particular given instance, but it is intuitive that when $r_1 << r_2$, the `linking` strategy would be better.

4.3 Impact of Linking Solution Pool

A set of eight experiments were evaluated to assess the impact of the linking solution pool. The chosen parameters for these experiments were:

- `withoutPoolWhenXYInt-LFixed`: The linking solution pool is not used and `whenXYInt-LFixed` strategy is used for solving problems (SL-MILP) and (UB).



(a) $r_1 \leq r_2$ (b) $r_1 > r_2$

Figure 3: Impact of the `branchStrategy` parameter.

- `withPoolWhenXYInt-LFixed`: The same as `withoutPoolWhenXYInt-LFixed`, but with the use of linking solution pool.
- `withoutPoolWhenXYIntOrLFixed-LFixed`: The linking solution pool is not used and the strategy for solving problems (SL-MILP) and (UB) is `whenXYIntOrLFixed-LFixed`.
- `withPoolWhenXYIntOrLFixed-LFixed`: The same as `withoutPoolWhenXYIntOrLFixed-LFixed`, but with the use of the linking solution pool.

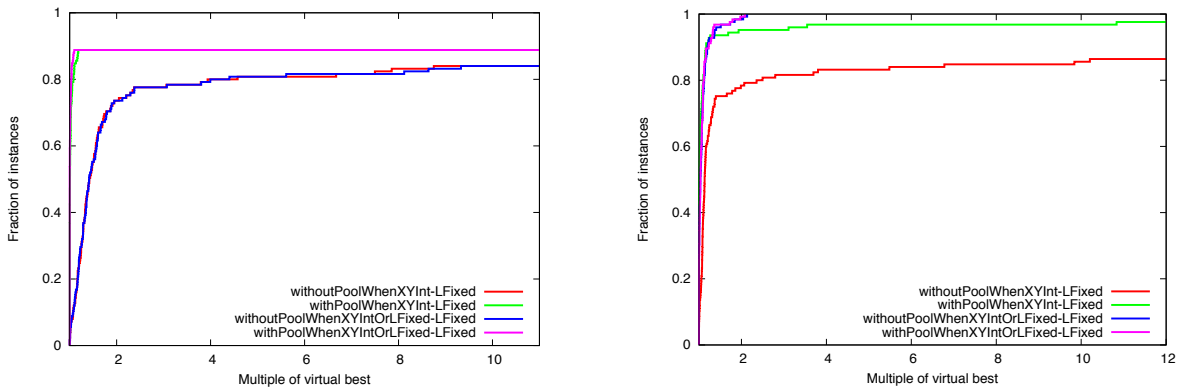
Each of the above four settings was tested with both `fractional` and `linking` branching strategies, although the linking pool was only expected to have a large impact when we allow branching on non-linking variables. This is because when branching only on linking variables, linking solutions can only arise again within the same subtree as they first arose and this can only happen if the solution was not already removed with a valid inequality.

The performance profiles shown in Figure 4 compare the solution time of the described methods. Figure 4a shows the methods which use `fractional` branching strategy, and Figure 4b shows the methods with `linking` branching strategy.

As expected, when branching was not limited to linking variables, the linking solution pool was effective in decreasing the solution time. This can be observed by comparing

- `withPoolWhenXYInt-LFixed` and `withoutPoolWhenXYInt-LFixed` for both the `fractional` and `linking` branching strategies; and
- `withPoolWhenXYIntOrLFixed-LFixed` and `withoutPoolWhenXYIntOrLFixed-LFixed` for the `fractional` branching strategy.

In these cases, branching can also be done on non-linking variables. However, use of the solution pool for `withoutPoolWhenXYIntOrLFixed-LFixed` with `linking` branching strategy does not improve performance because the branching was only done on linking variables. Based on the results achieved in this section, the linking solution pool is used by default in `MibS`.



(a) `fractional` branching strategy

(b) `linking` branching strategy

Figure 4: Impact of the linking solution pool.

Table 2 shows the total number of instances of problems (SL-MILP) and (UB) solved when using the methods `withPoolWhenXYIntOrLFixed-LFixed` and `withoutPoolWhenXYIntOrLFixed-LFixed` with both `linking` and `fractional` branching strategies, as well as the percent of total solution time required. Comparing the columns for the `linking` branching strategy verifies the results shown in Figure 4b. These columns show that using the linking solution pool does not decrease the number of (SL-MILP) and (UB) instances solved, so we do not expect to improve the solution time either. Similarly, the columns for the `fractional` branching strategy verify the results shown in Figure 4a and show that the number of instances of (SL-MILP) and (UB) can be reduced significantly by using the pool in this case, resulting in decreased solution time.

The results with both the `linking` and `fractional` strategies with the linking pool (the first and third columns) show that fewer instances of (SL-MILP) and (UB) are solved in general (and a smaller percentage of time required) with the `fractional` strategy, regardless of whether the overall solution time is smaller with the `linking` strategy or not. This does not mean, however, that using the `fractional` strategy always results in the need to solve fewer problems (SL-MILP) and (UB). It is generally only by taking advantage of the linking solution pool that this is avoided. Comparing `linking` strategy without the pool to the `fractional` strategy without the pool illustrates that the `fractional` strategy sometimes requires solving more problems (SL-MILP). Moreover, Table 2 shows that different data sets do not have the same behavior from the point of required time for solving MILPs.

4.4 Impact of Primal Heuristics

In order to evaluate the impact of employing primal heuristics implemented in `MibS` (see Section 2.5), we compared four different methods:

- `noHeuristics`: Parameters are set to the default values obtained from the results of previous sections, i.e.,
  - the strategy for solving problems (SL-MILP) and (UB) is `whenXYIntOrLFixed-LFixed`.

Table 2: Analysis of total time and frequency of solving problems (SL-MILP) and (UB)

Data Set	withPoolWhenXYIntOr LFixed-LFixed(Linking)			withoutPoolWhenXYIntOr LFixed-LFixed(Linking)			withPoolWhenXYIntOr LFixed-LFixed(Fractional)			withoutPoolWhenXYIntOr LFixed-LFixed(Fractional)		
	SL Count	UB Count	Time (%)	SL Count	UB Count	Time (%)	SL Count	UB Count	Time (%)	SL Count	UB Count	Time (%)
IBLP-DEN ($r_1 \leq r_2$)	1,027,234	1,023,867	89	1,029,644	1,023,867	89	203,711	48,190	4	5,570,810	316,227	91
IBLP-DEN ($r_1 > r_2$)	1,300,436	1,184,094	53	1,317,993	1,184,094	55	820,832	2,333	17	2,571,199	2,355	39
IBLP-FIS ($r_1 \leq r_2$)	699	695	84	699	695	84	201	29	0	1,336,226	31,058	43
IBLP-FIS ($r_1 > r_2$)	91,280	56,014	13	93,013	56,014	13	19,103	0	2	27,980	0	3
MIBLP-XU	26,593	8,791	4	26,597	8,795	4	5,972	5,618	18	10,654	10,239	38

  - `branchStrategy` is set to `linking` and `fractional` for the instances with $r_1 \leq r_2$ and $r_1 > r_2$, respectively.
  - the linking solution pool is used.

- `impObjectiveCut`: The same as `noHeuristics`, but the improving objective cut heuristic is turned-on.
- `secondLevelPriority`: The same as `noHeuristics`, but the second-level priority heuristic is turned-on.
- `weightedSums`: The same as `noHeuristics`, but the weighted sums heuristic is turned-on.

Figure 5 shows the performance profiles for these four methods with the solution time as the performance measure. The frequency of using heuristics was set to 100.

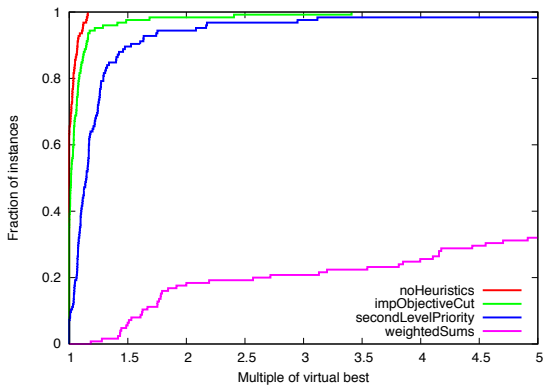


Figure 5: Impact of the primal heuristics

Based on Figure 5, the primal heuristics implemented so far are not effective in improving the solution time. However, it may be possible to improve their performance by parameter tuning. No serious effort has been made so far to do this.

4.5 Impact of MIP Solver

SYMPHONY was employed as the MIP solver in all experiments described in previous sections. In order to investigate the impact of the employed MIP solver in solving MIBLPs, we repeated some of these experiments using CPLEX (with its default setting) as the MIP solver. We observed that when the total time for solving the required MIPs was not large and the MIPs were easy to solve, the impact of the change in MIP solver was not considerable. In cases where solution of the MIPs required more effort, CPLEX did reduce the time for solving the MIPs by roughly half on average (but the exact amount of reduction depends highly on the instance).

5 Conclusions

We have presented a generalized branch-and-cut algorithm, which is able to solve a wide range of MIBLPs. The components of this algorithm have been explained in detail and we have discussed precisely how the specific features of MIBLPs can be utilized to solve various classes of problems that arise in practical applications. Moreover, we have introduced `MibS`, which is an open-source solver for MIBLPs. This solver has been implemented based on the branch-and-cut algorithm described in the paper and provides a comprehensive and flexible framework in which a variety of algorithmic options are available. We have demonstrated the performance of `MibS` and shown that it is a robust solver capable of solving generic MIBLPs of modest size. In future papers, we will describe additional details of the methodology in `MibS`, including the cut generation, which we have not discussed here. Our future plans for `MibS` include the implementation of additional techniques for generating valid inequalities and the possible addition of alternative algorithms.

Acknowledgements

This research was made possible with support from National Science Foundation Grants CMMI-1435453, CMMI-0728011, and ACI-0102687.

References

T. Achterberg. *Constraint Integer Programming*. PhD thesis, 01 2007.

T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

E. Balas. Intersection cuts-a new type of cutting planes for integer programming. *Operations Research*, 19(1):19–39, 1971.

E. Balas and R. Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69, 1972. ISSN 00361399. URL http://www.jstor.org/stable/2099623.

J. F. Bard. Practical bilevel optimization: algorithms and applications. *Springer Science & Business Media*, 30, 2013.

J. F. Bard and J. T. Moore. An algorithm for the discrete bilevel programming problem. *Naval Research Logistics*, 39(3):419–435, 1992.

D. Bienstock and A. Verma. The nk problem in power grids: New models, formulations, and numerical experiments. *SIAM Journal on Optimization*, 20(5):2352–2380, 2010.

R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. Savelsbergh. An updated mixed integer programming library: Miplib 3.0. Technical report, 1998.

J. Bracken and J. T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.

A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing*, 28(2):319–333, 2016.

M. Caramia and R. Mari. Enhanced exact algorithms for discrete bilevel linear problems. *Optimization Letters*, 9(7):1447–1468, 2015.

Cgl. Cut generator library, 2017. URL https://projects.coin-or.org/Cgl.

COIN-OR. Computational Infrastructure for Operations Research. 2018. URL https://www.coin-or.org.

CPLEX. Ibm ilog cplex optimizer, 2017. URL https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

S. DeNegre and T. Ralphs. A Branch-and-Cut Algorithm for Bilevel Integer Programming. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pages 65–78, 2009. doi: 10.1007/978-0-387-88843-9_4. URL http://coral.ie.lehigh.edu/~ted/files/papers/BILEVEL08.pdf.

S. DeNegre, T. Ralphs, and S. Tahernejad. MiBS version 1.1. 2019. doi: 10.5281/zenodo.1439384. URL https://github.com/coin-or/MibS.

S. T. DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. Phd, Lehigh University, 2011. URL http://coral.ie.lehigh.edu/{~}ted/files/papers/ScottDeNegreDissertation11.pdf.

M. Ehrgott and M. M. Wiecek. Multiobjective programming. In M. Ehrgott, J. Figueira, and S. Greco, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 667–722. Springer, Berlin, Germany, 2005.

N. P. Faísca, V. Dua, B. Rustem, P. M. Saraiva, and E. N. Pistikopoulos. Parametric global optimisation for bilevel programming. *Journal of Global Optimization*, 38:609–623, 2007.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, pages 1–27, 2017a.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017b.

J. J. Forrest. Coin-or branch and cut, 2017a. URL https://projects.coin-or.org/Cbc.

J. J. Forrest. Coin-or lp solver, 2017b. URL https://projects.coin-or.org/Clp.

J. Gao and F. You. Design and optimization of shale gas energy systems: Overview, research challenges, and future directions. *Computers & Chemical Engineering*, 106:699–718, 2017.

A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630, 1968.

A. Hassanzadeh and T. Ralphs. A Generalized Benders' Algorithm for Two-Stage Stochastic Program with Mixed Integer Recourse. Technical report, COR@L Laboratory Report 14T-005, Lehigh University, 2014. URL http://coral.ie.lehigh.edu/~ted/files/papers/SMILPGenBenders14.pdf.

M. Hemmati and C. Smith. A mixed integer bilevel programming approach for a competitive set covering problem. Technical report, Clemson University, 2016.

E. Israeli. *System Interdiction and Defense*. PhD thesis, Naval Postgraduate School, 1999.

R. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32:146–164, 1985.

P.-M. Kleniati and C. S. Adjiman. Branch-and-sandwich: A deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development. *Journal of Global Optimization*, 60:425–458, 2014a.

P.-M. Kleniati and C. S. Adjiman. Branch-and-sandwich: A deterministic global optimization algorithm for optimistic bilevel programming problems. part ii: Convergence analysis and numerical results. *Journal of Global Optimization*, 60:459–481, 2014b.

A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

P. Loridan and J. Morgan. Weak via strong stackelberg problem: New results. *Journal of Global Optimization*, 8(3):263–287, 1996.

L. Lozano and J. C. Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.

H. Marchand, A. Martin, R. Weismantel, and L. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123(1):397–446, 2002.

A. Mitsos. Global solution of nonlinear mixed integer bilevel programs. *Journal of Global Optimization*, 47:557–582, 2010.

J. T. Moore and J. F. Bard. The mixed integer linear bilevel programming problem. *Operations research*, 38(5):911–921, 1990.

Osi. Open solver interface, 2017. URL https://projects.coin-or.org/Osi.

M. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, 6(1):1–7, 1987.

M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale traveling salesman problems. *SIAM Review*, 33:60–100, 1991.

T. Ralphs and M. Güzelsoy. The SYMPHONY Callable Library for Mixed Integer Programming. In *Proceedings of the Ninth INFORMS Computing Society Conference*, pages 61–76, 2005. doi: 10.1007/0-387-23529-9_5. URL http://coral.ie.lehigh.edu/~ted/files/papers/SYMPHONY04.pdf.

T. Ralphs and M. Güzelsoy. Duality and Warm Starting in Integer Programming. In *The Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference*, 2006. URL http://coral.ie.lehigh.edu/~ted/files/papers/DMII06.pdf.

T. Ralphs, L. Ladányi, and M. Saltzman. A Library Hierarchy for Implementing Scalable Parallel Search Algorithms. *Journal of Supercomputing*, 28:215–234, 2004. doi: 10.1023/B:SUPE.0000020179.55383.ad. URL http://coral.ie.lehigh.edu/~ted/files/papers/JSC02.pdf.

T. Ralphs, M. Güzelsoy, and A. Mahajan. SYMPHONY version 5.6. 2019. doi: 10.5281/zenodo.2656802. URL https://github.com/coin-or/SYMPHONY/.

J. Salmeron, K. Wood, and R. Baldick. Analysis of electric grid security under terrorist threat. 2004.

L. Vicente, G. Savard, and J. Júdice. Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89(3):597–614, 1996.

H. Von Stackelberg. *Marktform und Gleichgewicht*. Julius Springer, 1934.

L. Wang and P. Xu. The watermelon algorithm for the bilevel integer linear programming problem. *SIAM Journal on Optimization*, 27(3):1403–1430, 2017.

U.-P. Wen and A. Huang. A simple tabu search method to solve the mixed-integer linear bilevel programming problem. *European Journal of Operational Research*, 88:563–571, 1996.

R. Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964.

K. Wolter. Implementation of Cutting Plane Separators for Mixed Integer Programs. Master's thesis, Technische Universität Berlin, 2006.

P. Xu and L. Wang. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & operations research*, 41:309–318, 2014.

Y. Xu and T. Ralphs. ALPS version 2.0. 2019a. doi: 10.5281/zenodo.2576602. URL https://github.com/coin-or/CHiPPS-ALPS.

Y. Xu and T. Ralphs. BiCEPs version 0.99. 2019b. doi: 10.5281/zenodo.2595520. URL https://github.com/coin-or/CHiPPS-BiCePS.

Y. Xu and T. Ralphs. BLIS version 0.94. 2019c. doi: 10.5281/zenodo.2595523. URL https://github.com/coin-or/CHiPPS-BLIS.

Y. Xu, T. Ralphs, L. Ladányi, and M. Saltzman. ALPS: A Framework for Implementing Parallel Tree Search Algorithms. In *The Proceedings of the Ninth INFORMS Computing Society Conference*, volume 29, pages 319–334, 2005. doi: 10.1007/0-387-23529-9_21. URL http://coral.ie.lehigh.edu/~ted/files/papers/ALPS04.pdf.

Y. Xu, T. Ralphs, L. Ladányi, and M. Saltzman. Computational Experience with a Software Framework for Parallel Integer Programming. *The INFORMS Journal on Computing*, 21:383–397, 2009. doi: 10.1287/ijoc.1090.0347. URL http://coral.ie.lehigh.edu/~ted/files/papers/CHiPPS-Rev.pdf.

B. Zeng and Y. An. Solving bilevel mixed integer program by reformulations and decomposition. *Optimization online*, pages 1–34, 2014.

Y. Zhang, L. Snyder, T. Ralphs, and Z. Xue. The Competitive Facility Location Problem Under Disruption Risks. *Transportation Research Part E: Logistics and Transportation Review*, 93, 2016. ISSN 13665545. doi: 10.1016/j.tre.2016.07.002. URL http://coral.ie.lehigh.edu/~ted/files/papers/CFLPD16.pdf.

Appendices

Tables 3–10 present detailed results applied for plotting Figures 2 –5.

Table 3: Detailed results of Figure 2a

	whenLInt-LInt			whenLInt-LFixed			whenLFixed-LFixed			whenXYInt-LFixed			whenXYIntOr LFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
miblp-20-15-50-0110-10-10	-206.00	1.82	423	-206.00	1.08	423	-206.00	0.97	1414	-206.00	1.06	423	-206.00	1.70	423
miblp-20-15-50-0110-10-2	-398.00	12.25	2590	-398.00	9.18	2450	-398.00	11.12	27625	-398.00	9.09	2450	-398.00	11.52	2590
miblp-20-15-50-0110-10-3	-42.00	0.74	267	-42.00	0.61	267	-42.00	0.71	1343	-42.00	0.62	267	-42.00	0.70	267
miblp-20-15-50-0110-10-6	-246.00	12.16	340	-246.00	4.47	340	-246.00	2.19	853	-246.00	4.56	340	-246.00	8.67	340
miblp-20-15-50-0110-10-9	-635.00	26.62	2387	-635.00	14.06	2380	-635.00	7.86	6457	-635.00	14.05	2380	-635.00	25.35	2387
miblp-20-20-50-0110-10-10	-441.00	1322.97	134901	-441.00	684.67	134585	-441.00	526.56	639146	-441.00	692.00	134585	-441.00	1121.84	134901
miblp-20-20-50-0110-10-1	-359.00	264.46	96136	-359.00	244.20	94019	-359.00	260.31	423149	-359.00	228.67	96141	-359.00	248.95	96136
miblp-20-20-50-0110-10-2	-659.00	12.72	3261	-659.00	3.93	3191	-659.00	2.78	7547	-659.00	3.69	3191	-659.00	10.56	3261
miblp-20-20-50-0110-10-3	-618.00	40.06	21622	-618.00	13.04	20780	-618.00	10.74	38188	-618.00	13.18	20788	-618.00	29.85	21622
miblp-20-20-50-0110-10-4	-604.00	>3600	654402	-604.00	3127.12	830908	-604.00	3405.57	6452671	-604.00	3145.16	830808	-604.00	>3600	714877
miblp-20-20-50-0110-10-7	-683.00	3233.20	3069661	-683.00	2046.70	2978073	-683.00	3184.99	11502091	-683.00	1887.37	3003967	-683.00	2511.36	3069661
miblp-20-20-50-0110-10-8	-667.00	182.80	12868	-667.00	87.14	12856	-667.00	40.78	30873	-667.00	80.45	12857	-667.00	148.40	12868
miblp-20-20-50-0110-10-9	-256.00	45.39	35295	-256.00	21.06	31244	-256.00	23.97	76055	-256.00	20.99	31245	-256.00	33.92	35295
miblp-20-20-50-0110-15-1	-450.00	60.74	3516	-450.00	60.84	3506	-450.00	65.16	49137	-450.00	59.38	3506	-450.00	60.89	3516
miblp-20-20-50-0110-15-2	-645.00	73.76	17251	-645.00	50.01	17251	-645.00	96.13	346065	-645.00	50.14	17251	-645.00	63.85	17251
miblp-20-20-50-0110-15-3	-593.00	100.53	3083	-593.00	70.36	3081	-593.00	65.25	42877	-593.00	71.15	3081	-593.00	92.89	3083
miblp-20-20-50-0110-15-4	-441.00	66.29	1625	-441.00	42.88	1625	-441.00	36.99	29904	-441.00	43.00	1625	-441.00	55.23	1625
miblp-20-20-50-0110-15-5	-379.00	860.42	16715	-379.00	632.14	16715	-379.00	615.16	205025	-379.00	651.28	16715	-379.00	730.86	16715
miblp-20-20-50-0110-15-6	-596.00	23.24	1657	-596.00	17.26	1657	-596.00	17.87	29923	-596.00	17.31	1657	-596.00	22.10	1657
miblp-20-20-50-0110-15-7	-471.00	133.26	13405	-471.00	110.80	13405	-471.00	125.24	241285	-471.00	111.02	13405	-471.00	127.42	13405
miblp-20-20-50-0110-15-8	-370.00	41.05	21589	-370.00	39.51	21589	-370.00	138.34	579309	-370.00	39.12	21589	-370.00	40.56	21589
miblp-20-20-50-0110-15-9	-584.00	2.58	588	-584.00	2.03	582	-584.00	1.96	4072	-584.00	2.00	582	-584.00	2.34	588
miblp-20-20-50-0110-5-13	-519.00	>3600	4221171	-519.00	2002.06	4392628	-519.00	2307.90	10196998	-519.00	2464.25	7128138	-519.00	>3600	5830754
miblp-20-20-50-0110-5-15	-617.00	>3600	3325012	-617.00	1122.66	2749914	-617.00	1219.33	6630921	-617.00	1310.67	4018058	-617.00	2953.78	4018081
miblp-20-20-50-0110-5-16	-833.00	44.15	18278	-833.00	6.48	17680	-833.00	4.86	19013	-833.00	6.60	17680	-833.00	38.30	18278
miblp-20-20-50-0110-5-17	-944.00	16.80	18200	-944.00	4.62	17712	-944.00	3.99	21541	-944.00	4.78	17679	-944.00	10.64	18200
miblp-20-20-50-0110-5-19	-431.00	104.93	79279	-431.00	26.75	77256	-431.00	27.52	147268	-431.00	26.56	77256	-431.00	65.55	79279
miblp-20-20-50-0110-5-1	-548.00	52.43	42629	-548.00	14.31	42364	-548.00	12.98	64067	-548.00	14.50	42364	-548.00	36.55	42629
miblp-20-20-50-0110-5-20	-438.00	48.38	60990	-438.00	14.95	50531	-438.00	15.81	76091	-438.00	17.30	60944	-438.00	31.77	60990
miblp-20-20-50-0110-5-6	-1061.00	202.25	224425	-1061.00	62.75	223494	-1061.00	58.74	284550	-1061.00	63.70	222566	-1061.00	127.84	224425
lseu-0.100000	1120.00	657.06	1132617	1120.00	235.84	734286	1120.00	248.54	1071409	1120.00	270.78	1003976	1120.00	626.90	1132617
lseu-0.900000	5838.00	13.91	1023	5838.00	14.41	1023	5838.00	1063.02	4718749	5838.00	14.10	1023	5838.00	13.83	1023
p0033-0.500000	3095.00	12.65	20855	3095.00	10.44	20695	3095.00	6.24	33614	3095.00	10.48	20695	3095.00	11.55	20855
p0033-0.900000	4679.00	0.06	27	4679.00	0.05	27	4679.00	0.65	3455	4679.00	0.06	27	4679.00	0.04	27
p0201-0.900000	15025.00	7.05	2481	15025.00	6.66	2481	15025.00	20.58	18801	15025.00	6.62	2481	15025.00	6.68	2481
stein27-0.500000	19.00	6.21	12115	19.00	6.24	14362	19.00	7.36	21515	19.00	5.88	12115	19.00	5.91	12115
stein27-0.900000	24.00	0.01	15	24.00	0.02	15	24.00	1.25	4445	24.00	0.02	15	24.00	0.02	15
stein45-0.100000	30.00	51.06	89035	30.00	96.92	61518	30.00	50.47	89035	30.00	50.19	89035	30.00	50.17	89035
stein45-0.500000	32.00	554.15	640308	32.00	1088.96	1014908	32.00	635.22	952123	32.00	519.59	640308	32.00	520.73	640308
stein45-0.900000	40.00	0.16	63	40.00	0.16	63	40.00	85.92	103661	40.00	0.14	63	40.00	0.15	63

Table 4: Detailed results of Figure 2b

	whenLInt-LInt			whenLInt-LFixed			whenLFixed-LFixed			whenXYInt-LFixed			whenXYIntOrLFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-110-10	-177.67	336.74	75952	-177.67	150.88	75141	-177.67	149.10	93801	-177.67	153.82	75499	-177.67	223.95	75952
bmilplib-110-1	-181.67	9.23	2806	-181.67	5.35	3629	-181.67	5.42	2878	-181.67	5.35	2806	-181.67	6.34	2806
bmilplib-110-2	-110.67	9.16	4183	-110.67	6.53	4439	-110.67	6.50	4272	-110.67	6.48	4174	-110.67	7.46	4183
bmilplib-110-3	-215.16	10.15	3484	-215.16	6.07	3529	-215.16	5.42	3556	-215.16	5.58	3471	-215.16	7.13	3484
bmilplib-110-4	-197.29	16.30	1917	-197.29	3.66	1740	-197.29	3.54	1801	-197.29	3.69	1751	-197.29	8.20	1917
bmilplib-110-6	-148.25	25.96	8857	-148.25	14.57	8735	-148.25	14.14	9392	-148.25	15.06	8770	-148.25	19.35	8857
bmilplib-110-7	-160.86	11.92	2221	-160.86	4.66	2167	-160.86	4.01	2281	-160.86	4.94	2189	-160.86	7.45	2221
bmilplib-110-8	-155.00	39.85	11835	-155.00	20.66	11916	-155.00	20.65	12498	-155.00	20.78	11663	-155.00	27.93	11835
bmilplib-110-9	-192.92	9.14	2155	-192.92	3.46	2099	-192.92	3.28	2171	-192.92	3.66	2151	-192.92	5.84	2155
bmilplib-160-10	-189.82	87.08	9130	-189.82	30.70	8676	-189.82	31.24	9344	-189.82	33.42	9071	-189.82	55.24	9130
bmilplib-160-1	-165.00	63.76	7908	-165.00	25.00	8028	-165.00	24.18	8145	-165.00	25.70	7901	-165.00	38.00	7908
bmilplib-160-2	-178.24	87.14	8680	-178.24	29.51	8612	-178.24	29.64	8710	-178.24	30.59	8635	-178.24	47.52	8680
bmilplib-160-3	-174.94	196.12	15164	-174.94	62.78	14998	-174.94	54.62	15427	-174.94	63.37	15002	-174.94	104.06	15164
bmilplib-160-4	-135.83	77.00	15005	-135.83	48.60	14708	-135.83	50.86	15354	-135.83	51.02	14772	-135.83	60.24	15005
bmilplib-160-5	-140.78	62.61	4981	-140.78	19.78	4842	-140.78	19.35	5178	-140.78	20.52	4948	-140.78	36.78	4981
bmilplib-160-6	-111.00	32.32	8685	-111.00	27.50	10149	-111.00	25.08	8864	-111.00	24.66	8676	-111.00	27.50	8685
bmilplib-160-7	-96.00	65.43	16589	-96.00	51.07	16804	-96.00	50.97	17430	-96.00	51.20	16560	-96.00	58.05	16589
bmilplib-160-8	-181.40	22.24	3464	-181.40	12.47	3417	-181.40	9.32	3540	-181.40	10.49	3444	-181.40	14.93	3464
bmilplib-160-9	-207.50	34.05	4715	-207.50	16.51	5169	-207.50	14.90	4728	-207.50	15.85	4607	-207.50	23.34	4715
bmilplib-210-10	-130.59	84.71	12226	-130.59	58.89	12010	-130.59	61.00	12497	-130.59	61.81	12176	-130.59	72.22	12226
bmilplib-210-1	-136.80	77.67	7429	-136.80	37.71	7182	-136.80	40.04	7598	-136.80	41.96	7429	-136.80	56.23	7429
bmilplib-210-2	-117.80	153.16	22441	-117.80	111.87	22597	-117.80	109.56	23096	-117.80	113.38	22294	-117.80	131.02	22441
bmilplib-210-3	-130.80	74.50	8897	-130.80	44.80	8523	-130.80	47.44	9165	-130.80	48.19	8848	-130.80	59.95	8897
bmilplib-210-4	-162.20	62.23	4738	-162.20	36.19	5094	-162.20	25.92	4806	-162.20	26.81	4687	-162.20	41.52	4738
bmilplib-210-5	-134.00	156.90	21193	-134.00	106.25	21560	-134.00	105.03	21552	-134.00	108.22	21058	-134.00	130.51	21193
bmilplib-210-6	-125.43	269.65	38538	-125.43	198.77	38367	-125.43	197.88	39941	-125.43	201.40	38443	-125.43	227.81	38538
bmilplib-210-7	-169.73	146.39	13960	-169.73	76.42	13960	-169.73	77.62	14305	-169.73	77.84	13960	-169.73	104.02	13960
bmilplib-210-8	-101.46	70.64	11105	-101.46	56.97	11324	-101.46	58.14	11347	-101.46	61.79	11105	-101.46	62.56	11105
bmilplib-210-9	-184.00	1822.79	143571	-184.00	790.46	144603	-184.00	859.46	143665	-184.00	879.61	142294	-184.00	1184.18	143571
bmilplib-260-10	-151.73	785.34	62063	-151.73	503.08	62143	-151.73	502.94	63026	-151.73	546.55	61767	-151.73	633.15	62063
bmilplib-260-1	-139.00	155.24	10994	-139.00	83.01	11369	-139.00	81.74	11242	-139.00	85.81	10980	-139.00	119.94	10994
bmilplib-260-2	-82.62	117.18	15300	-82.62	156.75	15292	-82.62	109.75	15712	-82.62	109.75	15287	-82.62	114.50	15300
bmilplib-260-3	-144.25	148.68	8777	-144.25	72.67	9563	-144.25	69.26	8859	-144.25	71.54	8769	-144.25	94.43	8777
bmilplib-260-4	-117.33	294.39	32964	-117.33	245.58	32696	-117.33	260.01	33931	-117.33	259.48	32787	-117.33	272.70	32964
bmilplib-260-5	-121.00	201.31	21557	-121.00	167.30	22480	-121.00	173.32	22121	-121.00	166.03	21542	-121.00	183.46	21557
bmilplib-260-6	-124.00	258.94	25420	-124.00	191.68	25826	-124.00	209.79	26023	-124.00	197.04	25362	-124.00	223.31	25420
bmilplib-260-7	-137.80	472.66	40528	-137.80	304.36	40400	-137.80	318.76	41180	-137.80	312.12	40274	-137.80	383.16	40528
bmilplib-260-8	-119.89	112.17	10094	-119.89	73.72	9961	-119.89	76.70	10332	-119.89	87.01	10025	-119.89	96.85	10094
bmilplib-260-9	-160.00	510.37	33496	-160.00	257.08	33928	-160.00	271.18	34236	-160.00	273.66	33468	-160.00	394.98	33496
bmilplib-310-10	-141.86	129.77	9904	-141.86	110.26	9896	-141.86	114.38	10033	-141.86	121.07	9900	-141.86	127.87	9904
bmilplib-310-1	-117.00	521.12	28360	-117.00	301.34	28373	-117.00	329.80	29055	-117.00	329.71	28330	-117.00	372.68	28360
bmilplib-310-2	-105.00	523.84	43448	-105.00	457.17	44736	-105.00	457.69	44841	-105.00	497.66	43399	-105.00	539.20	43448
bmilplib-310-3	-127.52	943.53	66426	-127.52	907.46	72139	-127.52	783.89	67441	-127.52	777.74	66410	-127.52	916.69	66426
bmilplib-310-4	-147.78	701.76	47718	-147.78	510.81	48131	-147.78	572.76	48950	-147.78	569.43	47711	-147.78	652.54	47718
bmilplib-310-5	-161.45	557.76	31838	-161.45	339.05	31553	-161.45	382.13	32454	-161.45	366.90	31782	-161.45	488.65	31838
bmilplib-310-6	-141.18	1448.71	102047	-141.18	1159.42	110836	-141.18	1264.60	103214	-141.18	1191.51	101904	-141.18	1324.84	102047
bmilplib-310-7	-142.00	1443.70	102030	-142.00	1057.98	104876	-142.00	1199.47	104166	-142.00	1129.24	102000	-142.00	1396.34	102030
bmilplib-310-8	-115.34	143.13	11375	-115.34	110.74	11067	-115.34	114.80	11293	-115.34	127.94	11109	-115.34	142.93	11375
bmilplib-310-9	-115.65	423.77	20552	-115.65	231.20	21846	-115.65	248.24	20838	-115.65	255.27	20490	-115.65	281.90	20552
bmilplib-360-10	-108.59	257.34	13727	-108.59	209.20	13671	-108.59	242.94	14064	-108.59	234.92	13697	-108.59	258.65	13727
bmilplib-360-1	-133.00	2416.81	75780	-133.00	1239.50	77206	-133.00	1297.32	77066	-133.00	1353.62	75421	-133.00	1500.96	75780
bmilplib-360-2	-138.44	1187.29	53004	-138.44	829.40	52868	-138.44	999.50	54354	-138.44	965.52	52919	-138.44	1169.80	53004
bmilplib-360-3	-131.00	834.51	40671	-131.00	728.80	40353	-131.00	624.78	41302	-131.00	832.88	40487	-131.00	731.63	40671
bmilplib-360-4	-119.00	371.93	18350	-119.00	286.84	18870	-119.00	338.49	18813	-119.00	332.90	18293	-119.00	361.79	18350
bmilplib-360-5	-164.26	593.68	30001	-164.26	420.56	29790	-164.26	484.66	30352	-164.26	618.86	29947	-164.26	600.63	30001
bmilplib-360-6	-110.12	1169.00	68863	-110.12	1005.68	69453	-110.12	1018.82	70283	-110.12	1181.02	68863	-110.12	1210.13	68863
bmilplib-360-7	-105.00	538.80	31092	-105.00	457.98	32124	-105.00	517.46	31884	-105.00	634.17	30900	-105.00	542.81	31092
bmilplib-360-8	-98.25	399.32	22995	-98.25	343.65	22787	-98.25	416.97	23337	-98.25	362.89	22857	-98.25	386.45	22995
bmilplib-360-9	-127.22	815.44	40383	-127.22	622.13	40733	-127.22	746.66	41235	-127.22	736.16	40329	-127.22	819.68	40383
bmilplib-410-10	-153.37	3527.64	101673	-153.37	2725.83	103359	-153.37	2879.03	103400	-153.37	2729.77	101447	-153.37	2976.52	101673
bmilplib-410-1	-103.50	582.60	20790	-103.50	535.58	20612	-103.50	589.62	21088	-103.50	553.76	20634	-103.50	534.12	20790
bmilplib-410-2	-108.59	803.44	31603	-108.59	734.82	31602	-108.59	748.06	32251	-108.59	840.56	31603	-108.59	882.69	31603
bmilplib-410-3	-96.24	1275.98	33041	-96.24	791.92	33963	-96.24	777.04	33473	-96.24	871.58	32882	-96.24	973.12	33041
bmilplib-410-4	-119.50	1582.86	38489	-119.50	930.82	39270	-119.50	1041.97	39371	-119.50	1031.12	38489	-119.50	1101.03	38489
bmilplib-410-5	-119.22	700.45	23865	-119.22	567.85	23747	-119.22	681.99	24163	-119.22	678.49	23852	-119.22	696.17	23865
bmilplib-410-6	-151.31	404.52	11130	-151.31	261.83	10795	-151.31	333.36	11193	-151.31	322.88	11130	-151.31	337.71	11130
bmilplib-410-7	-123.00	624.95	22208	-123.00	519.43	22843	-123.00	601.78	22628	-123.00	560.34	22146	-123.00	560.38	22208
bmilplib-410-8	-125.78	1717.64	62336	-125.78	1484.95	61784	-125.78	1692.35	63421	-125.78	1547.88	62218	-125.78	1579.12	62336
bmilplib-410-9	-100.77	505.02	20424	-100.77	579.73	20095	-100.77	607.44	20920	-100.77	531.88	20400	-100.77	494.06	20424
bmilplib-460-10	-102.51	1949.07	55030	-102.51	1782.80	55882	-102.51	1701.70	56265	-102.51	1857.94	55024	-102.51	1822.52	55030
bmilplib-460-1	-97.59	2961.39	86689	-97.59	3575.71	93301	-97.59	2938.02	87709	-97.59	2822.97	86632	-97.59	2915.62	86689
bmilplib-460-2	-139.00	739.75	16720	-139.00	527.76	16721	-139.00	608.40	16863	-139.00	625.95	16623	-139.00	694.03	16720
bmilplib-460-3	-86.50	1921.88	57462	-86.50	1853.40	59065	-86.50	2102.77	58229	-86.50	1995.92	57395	-86.50	1946.46	57462
bmilplib-460-4	-107.03	3321.16	95231	-107.03	3173.50	94899	-107.03	3328.91	97160	-107.03	3291.28	95035	-107.03	3301.27	95231
bmilplib-460-5	-100.50	1407.70	41312	-100.50	1366.52	40903	-100.50	1619.91	41796	-100.50	1424.54	41109	-100.50	1450.14	41312
bmilplib-460-6	-107.00	1623.54	46158	-107.00	1537.79	48966	-107.00	1483.17	46732	-107.00	1635.42	46076	-107.00	1598.68	46158

Table 5: Detailed results of Figure 2b (continued)

	whenLInt-LInt			whenLInt-LFixed			whenLFixed-LFixed			whenXYInt-LFixed			whenXYIntOr LFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-460-7	-83.75	1735.67	48973	-83.75	1606.84	49516	-83.75	1728.92	49717	-83.75	1601.34	48897	-83.75	1671.84	48973
bmilplib-460-8	-115.39	1074.21	27663	-115.39	858.20	27275	-115.39	911.51	28075	-115.39	982.30	27572	-115.39	1017.28	27663
bmilplib-460-9	-128.70	3410.18	86426	-128.70	2671.50	86656	-128.70	2715.98	87796	-128.70	2830.59	85977	-128.70	3004.49	86426
bmilplib-60-10	-186.21	14.85	5929	-186.21	6.88	9593	-186.21	6.91	6134	-186.21	6.93	5922	-186.21	9.50	5929
bmilplib-60-1	-153.20	16.56	4896	-153.20	4.84	5366	-153.20	6.50	5041	-153.20	6.71	4877	-153.20	10.92	4896
bmilplib-60-5	-116.40	15.19	11289	-116.40	7.57	11308	-116.40	8.65	13984	-116.40	8.15	11202	-116.40	11.54	11289
bmilplib-60-6	-187.31	15.38	7120	-187.31	7.08	9241	-187.31	7.34	7304	-187.31	7.62	7016	-187.31	10.44	7120
bmilplib-60-8	-232.12	8.38	3653	-232.12	2.54	4052	-232.12	3.33	3683	-232.12	3.36	3570	-232.12	5.66	3653
bmilplib-60-9	-136.50	33.85	27036	-136.50	18.90	31312	-136.50	20.22	28603	-136.50	19.61	26792	-136.50	26.84	27036

Table 6: Detailed results of Figure 3a

			linkingBranching			fractionalBranching		
Instance	r_1	r_2	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
miblp-20-15-50-0110-10-10	5	10	-206.00	1.06	423	-206.00	4.25	15741
miblp-20-15-50-0110-10-2	5	10	-398.00	9.09	2450	-398.00	947.81	2929420
miblp-20-15-50-0110-10-3	5	10	-42.00	0.62	267	-42.00	12.66	46349
miblp-20-15-50-0110-10-6	5	10	-246.00	4.56	340	-246.00	1.56	1367
miblp-20-15-50-0110-10-9	5	10	-635.00	14.05	2380	-635.00	6.52	7667
miblp-20-20-50-0110-10-10	10	10	-441.00	692.00	134585	-441.00	2867.85	7261247
miblp-20-20-50-0110-10-1	10	10	-359.00	228.67	96141	-357.00	>3600	9234364
miblp-20-20-50-0110-10-2	10	10	-659.00	3.69	3191	-659.00	3.16	5100
miblp-20-20-50-0110-10-3	10	10	-618.00	13.18	20788	-618.00	12.05	51970
miblp-20-20-50-0110-10-4	10	10	-604.00	3145.16	830808	-604.00	>3600	7988914
miblp-20-20-50-0110-10-7	10	10	-683.00	1887.37	3003967	-629.00	>3600	9709672
miblp-20-20-50-0110-10-8	10	10	-667.00	80.45	12857	-667.00	68.08	75661
miblp-20-20-50-0110-10-9	10	10	-256.00	20.99	31245	-256.00	305.78	757349
miblp-20-20-50-0110-15-1	5	15	-450.00	59.38	3506	-317.00	>3600	9813005
miblp-20-20-50-0110-15-2	5	15	-645.00	50.14	17251	-645.00	>3600	10839884
miblp-20-20-50-0110-15-3	5	15	-593.00	71.15	3081	-593.00	>3600	13042109
miblp-20-20-50-0110-15-4	5	15	-441.00	43.00	1625	-398.00	>3600	8692428
miblp-20-20-50-0110-15-5	5	15	-379.00	651.28	16715	-320.00	>3600	7284040
miblp-20-20-50-0110-15-6	5	15	-596.00	17.31	1657	-596.00	>3600	7851818
miblp-20-20-50-0110-15-7	5	15	-471.00	111.02	13405	-471.00	>3600	9675451
miblp-20-20-50-0110-15-8	5	15	-370.00	39.12	21589	-290.00	>3600	10350188
miblp-20-20-50-0110-15-9	5	15	-584.00	2.00	582	-584.00	19.58	56459
lseu-0.900000	9	80	5838.00	14.10	1023	5838.00	>3600	8743754
p0033-0.900000	4	29	4679.00	0.06	27	4679.00	5.56	28241
p0201-0.900000	21	180	15025.00	6.62	2481	15025.00	>3600	1310278
stein27-0.900000	3	24	24.00	0.02	15	24.00	419.87	702055
stein45-0.900000	5	40	40.00	0.14	63	40.00	>3600	1499583

Table 7: Detailed results of Figure 3b

			linkingBranching			fractionalBranching		
Instance	r_1	r_2	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-110-10	110	63	-177.67	153.82	75499	-177.67	116.32	55177
bmilplib-110-1	110	50	-181.67	5.35	2806	-181.67	0.93	306
bmilplib-110-2	110	45	-110.67	6.48	4174	-110.67	1.16	303
bmilplib-110-3	110	55	-215.16	5.58	3471	-215.16	0.89	360
bmilplib-110-4	110	50	-197.29	3.69	1751	-197.29	1.34	148
bmilplib-110-6	110	55	-148.25	15.06	8770	-148.25	3.84	1448
bmilplib-110-7	110	61	-160.86	4.94	2189	-160.86	0.91	205
bmilplib-110-8	110	54	-155.00	20.78	11663	-155.00	7.66	2274
bmilplib-110-9	110	58	-192.92	3.66	2151	-192.92	0.39	146
bmilplib-160-10	160	83	-189.82	33.42	9071	-189.82	7.73	728
bmilplib-160-1	160	80	-165.00	25.70	7901	-165.00	4.76	881
bmilplib-160-2	160	76	-178.24	30.59	8635	-178.24	5.95	507
bmilplib-160-3	160	86	-174.94	63.37	15002	-174.94	13.71	1102
bmilplib-160-4	160	81	-135.83	51.02	14772	-135.83	19.13	2447
bmilplib-160-5	160	83	-140.78	20.52	4948	-140.78	7.72	668
bmilplib-160-6	160	81	-111.00	24.66	8676	-111.00	5.02	627
bmilplib-160-7	160	79	-96.00	51.20	16560	-96.00	18.72	2855
bmilplib-160-8	160	85	-181.40	10.49	3444	-181.40	1.74	311
bmilplib-160-9	160	77	-207.50	15.85	4607	-207.50	2.02	268
bmilplib-210-10	210	99	-130.59	61.81	12176	-130.59	8.59	1100
bmilplib-210-1	210	101	-136.80	41.96	7429	-136.80	6.67	550
bmilplib-210-2	210	96	-117.80	113.38	22294	-117.80	19.78	2306
bmilplib-210-3	210	119	-130.80	48.19	8848	-130.80	12.93	1380
bmilplib-210-4	210	115	-162.20	26.81	4687	-162.20	3.64	309
bmilplib-210-5	210	110	-134.00	108.22	21058	-134.00	20.03	2079
bmilplib-210-6	210	115	-125.43	201.40	38443	-125.43	45.28	4875
bmilplib-210-7	210	102	-169.73	77.84	13960	-169.73	11.46	1181
bmilplib-210-8	210	116	-101.46	61.79	11105	-101.46	9.60	942
bmilplib-210-9	210	103	-184.00	879.61	142294	-184.00	240.14	9466
bmilplib-260-10	260	117	-151.73	546.55	61767	-151.73	73.10	4716
bmilplib-260-1	260	135	-139.00	85.81	10980	-139.00	10.10	887
bmilplib-260-2	260	126	-82.62	109.75	15287	-82.62	18.07	1607
bmilplib-260-3	260	120	-144.25	71.54	8769	-144.25	8.70	518
bmilplib-260-4	260	125	-117.33	259.48	32787	-117.33	66.89	4426
bmilplib-260-5	260	132	-121.00	166.03	21542	-121.00	29.19	2165
bmilplib-260-6	260	146	-124.00	197.04	25362	-124.00	40.59	2420
bmilplib-260-7	260	129	-137.80	312.12	40274	-137.80	44.45	3200
bmilplib-260-8	260	143	-119.89	87.01	10025	-119.89	10.92	1025
bmilplib-260-9	260	132	-160.00	273.66	33468	-160.00	36.36	2526
bmilplib-310-10	310	157	-141.86	121.07	9900	-141.86	5.51	397
bmilplib-310-1	310	169	-117.00	329.71	28330	-117.00	44.79	2624
bmilplib-310-2	310	154	-105.00	497.66	43399	-105.00	98.39	5372
bmilplib-310-3	310	157	-127.52	777.74	66410	-127.52	149.07	7067
bmilplib-310-4	310	152	-147.78	569.43	47711	-147.78	70.22	4767
bmilplib-310-5	310	164	-161.45	366.90	31782	-161.45	34.22	1993
bmilplib-310-6	310	148	-141.18	1191.51	101904	-141.18	169.96	8300
bmilplib-310-7	310	170	-142.00	1129.24	102000	-142.00	139.15	7263
bmilplib-310-8	310	154	-115.34	127.94	11109	-115.34	19.23	921
bmilplib-310-9	310	150	-115.65	255.27	20490	-115.65	32.31	1590
bmilplib-360-10	360	172	-108.59	234.92	13697	-108.59	25.75	1106
bmilplib-360-1	360	181	-133.00	1353.62	75421	-133.00	158.38	4923
bmilplib-360-2	360	179	-138.44	965.52	52919	-138.44	148.90	4493
bmilplib-360-3	360	195	-131.00	832.88	40487	-131.00	65.41	2654
bmilplib-360-4	360	184	-119.00	332.90	18293	-119.00	42.95	1564
bmilplib-360-5	360	194	-164.26	618.86	29947	-164.26	44.45	1713
bmilplib-360-6	360	172	-110.12	1181.02	68863	-110.12	142.81	5520
bmilplib-360-7	360	188	-105.00	634.17	30900	-105.00	89.22	3346
bmilplib-360-8	360	170	-98.25	362.89	22857	-98.25	44.85	1686
bmilplib-360-9	360	184	-127.22	736.16	40329	-127.22	50.24	2642
bmilplib-410-10	410	201	-153.37	2729.77	101447	-153.37	258.89	7428
bmilplib-410-1	410	196	-103.50	553.76	20634	-103.50	87.94	1944

Table 7: Detailed results of Figure 3b (continued)

			linkingBranching			fractionalBranching		
Instance	r_1	r_2	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-410-2	410	189	-108.59	840.56	31603	-108.59	103.57	2887
bmilplib-410-3	410	212	-96.24	871.58	32882	-96.24	147.99	3781
bmilplib-410-4	410	187	-119.50	1031.12	38489	-119.50	100.21	2995
bmilplib-410-5	410	209	-119.22	678.49	23852	-119.22	71.23	1520
bmilplib-410-6	410	206	-151.31	322.88	11130	-151.31	13.54	533
bmilplib-410-7	410	225	-123.00	560.34	22146	-123.00	35.87	1177
bmilplib-410-8	410	211	-125.78	1547.88	62218	-125.78	169.15	4480
bmilplib-410-9	410	216	-100.77	531.88	20400	-100.77	82.17	2071
bmilplib-460-10	460	217	-102.51	1857.94	55024	-102.51	228.12	4465
bmilplib-460-1	460	227	-97.59	2822.97	86632	-97.59	569.22	10803
bmilplib-460-2	460	249	-139.00	625.95	16623	-139.00	43.65	964
bmilplib-460-3	460	222	-86.50	1995.92	57395	-86.50	223.58	3882
bmilplib-460-4	460	218	-107.03	3291.28	95035	-107.03	412.76	7856
bmilplib-460-5	460	216	-100.50	1424.54	41109	-100.50	170.30	3025
bmilplib-460-6	460	222	-107.00	1635.42	46076	-107.00	236.30	4143
bmilplib-460-7	460	254	-83.75	1601.34	48897	-83.75	294.68	5252
bmilplib-460-8	460	256	-115.39	982.30	27572	-115.39	94.67	1903
bmilplib-460-9	460	224	-128.70	2830.59	85977	-128.70	327.68	6185
bmilplib-60-10	60	25	-186.21	6.93	5922	-186.21	4.29	2590
bmilplib-60-1	60	29	-153.20	6.71	4877	-153.20	2.79	1094
bmilplib-60-5	60	33	-116.40	8.15	11202	-116.40	10.08	9996
bmilplib-60-6	60	27	-187.31	7.62	7016	-187.31	3.34	1383
bmilplib-60-8	60	30	-232.12	3.36	3570	-232.12	1.15	572
bmilplib-60-9	60	26	-136.50	19.61	26792	-136.50	12.34	9888
miblp-20-20-50-0110-5-13	15	5	-519.00	2464.25	7128138	-519.00	2709.29	6515097
miblp-20-20-50-0110-5-15	15	5	-617.00	1310.67	4018058	-617.00	1130.92	4312915
miblp-20-20-50-0110-5-16	15	5	-833.00	6.60	17680	-833.00	1.80	5913
miblp-20-20-50-0110-5-17	15	5	-944.00	4.78	17679	-944.00	1.53	4038
miblp-20-20-50-0110-5-19	15	5	-431.00	26.56	77256	-431.00	25.50	116041
miblp-20-20-50-0110-5-1	15	5	-548.00	14.50	42364	-548.00	8.45	31298
miblp-20-20-50-0110-5-20	15	5	-438.00	17.30	60944	-438.00	10.82	33315
miblp-20-20-50-0110-5-6	15	5	-1061.00	63.70	222566	-1061.00	51.74	213928
lseu-0.100000	81	8	1120.00	270.78	1003976	1120.00	3.15	8603
p0033-0.500000	17	16	3095.00	10.48	20695	3095.00	0.25	1467
stein27-0.500000	14	13	19.00	5.88	12115	19.00	6.51	17648
stein45-0.100000	41	4	30.00	50.19	89035	30.00	64.27	90241
stein45-0.500000	23	22	32.00	519.59	640308	32.00	471.57	753845

Table 8: Detailed results of Figure 4a

	withoutPoolWhen XYInt-LFixed			withPoolWhen XYInt-LFixed			withoutPoolWhen XYIntOrLFixed-LFixed			withPoolWhen XYIntOrLFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-110-10	-177.67	355.72	55177	-177.67	117.08	55177	-177.67	357.09	55177	-177.67	116.32	55177
bmilplib-110-1	-181.67	1.08	306	-181.67	0.94	306	-181.67	1.08	306	-181.67	0.93	306
bmilplib-110-2	-110.67	1.56	303	-110.67	1.17	303	-110.67	1.56	303	-110.67	1.16	303
bmilplib-110-3	-215.16	1.18	360	-215.16	0.91	360	-215.16	1.14	360	-215.16	0.89	360
bmilplib-110-4	-197.29	2.38	148	-197.29	1.34	148	-197.29	2.38	148	-197.29	1.34	148
bmilplib-110-6	-148.25	8.77	1448	-148.25	3.83	1448	-148.25	8.78	1448	-148.25	3.84	1448
bmilplib-110-7	-160.86	1.16	205	-160.86	0.93	205	-160.86	1.17	205	-160.86	0.91	205
bmilplib-110-8	-155.00	12.40	2274	-155.00	7.74	2274	-155.00	12.29	2274	-155.00	7.66	2274
bmilplib-110-9	-192.92	0.39	146	-192.92	0.38	146	-192.92	0.38	146	-192.92	0.39	146
bmilplib-160-10	-189.82	17.00	728	-189.82	7.70	728	-189.82	16.94	728	-189.82	7.73	728
bmilplib-160-1	-165.00	6.52	881	-165.00	4.72	881	-165.00	6.48	881	-165.00	4.76	881
bmilplib-160-2	-178.24	9.39	507	-178.24	5.98	507	-178.24	9.42	507	-178.24	5.95	507
bmilplib-160-3	-174.94	32.47	1102	-174.94	13.76	1102	-174.94	32.60	1102	-174.94	13.71	1102
bmilplib-160-4	-135.83	35.80	2447	-135.83	19.06	2447	-135.83	36.00	2447	-135.83	19.13	2447
bmilplib-160-5	-140.78	29.26	668	-140.78	7.73	668	-140.78	29.33	668	-140.78	7.72	668
bmilplib-160-6	-111.00	7.26	627	-111.00	5.01	627	-111.00	7.30	627	-111.00	5.02	627
bmilplib-160-7	-96.00	35.18	2855	-96.00	18.04	2855	-96.00	35.13	2855	-96.00	18.72	2855
bmilplib-160-8	-181.40	2.30	311	-181.40	1.67	311	-181.40	2.28	311	-181.40	1.74	311
bmilplib-160-9	-207.50	2.02	268	-207.50	2.01	268	-207.50	2.00	268	-207.50	2.02	268
bmilplib-210-10	-130.59	13.92	1100	-130.59	8.78	1100	-130.59	14.28	1100	-130.59	8.59	1100
bmilplib-210-1	-136.80	6.72	550	-136.80	6.71	550	-136.80	6.74	550	-136.80	6.67	550
bmilplib-210-2	-117.80	40.52	2306	-117.80	20.60	2306	-117.80	41.96	2306	-117.80	19.78	2306
bmilplib-210-3	-130.80	24.72	1380	-130.80	13.09	1380	-130.80	24.48	1380	-130.80	12.93	1380
bmilplib-210-4	-162.20	4.82	309	-162.20	3.65	309	-162.20	4.80	309	-162.20	3.64	309
bmilplib-210-5	-134.00	29.57	2079	-134.00	20.08	2079	-134.00	30.96	2079	-134.00	20.03	2079
bmilplib-210-6	-125.43	105.93	4875	-125.43	46.30	4875	-125.43	107.54	4875	-125.43	45.28	4875
bmilplib-210-7	-169.73	18.21	1181	-169.73	11.59	1181	-169.73	18.14	1181	-169.73	11.46	1181
bmilplib-210-8	-101.46	17.89	942	-101.46	9.69	942	-101.46	17.97	942	-101.46	9.60	942
bmilplib-210-9	-184.00	334.58	9466	-184.00	241.89	9466	-184.00	336.35	9466	-184.00	240.14	9466
bmilplib-260-10	-151.73	106.84	4716	-151.73	75.99	4716	-151.73	106.74	4716	-151.73	73.10	4716
bmilplib-260-1	-139.00	10.55	887	-139.00	9.72	887	-139.00	10.17	887	-139.00	10.10	887
bmilplib-260-2	-82.62	25.82	1607	-82.62	17.95	1607	-82.62	24.39	1607	-82.62	18.07	1607
bmilplib-260-3	-144.25	12.22	518	-144.25	8.70	518	-144.25	12.20	518	-144.25	8.70	518
bmilplib-260-4	-117.33	82.73	4426	-117.33	62.90	4426	-117.33	84.74	4426	-117.33	66.89	4426
bmilplib-260-5	-121.00	36.18	2165	-121.00	29.80	2165	-121.00	37.57	2165	-121.00	29.19	2165
bmilplib-260-6	-124.00	67.59	2420	-124.00	39.43	2420	-124.00	69.01	2420	-124.00	40.59	2420
bmilplib-260-7	-137.80	76.55	3200	-137.80	46.85	3200	-137.80	79.79	3200	-137.80	44.45	3200
bmilplib-260-8	-119.89	16.52	1025	-119.89	10.90	1025	-119.89	17.44	1025	-119.89	10.92	1025
bmilplib-260-9	-160.00	61.66	2526	-160.00	36.58	2526	-160.00	64.72	2526	-160.00	36.36	2526
bmilplib-310-10	-141.86	5.42	397	-141.86	5.95	397	-141.86	5.88	397	-141.86	5.51	397
bmilplib-310-1	-117.00	58.20	2624	-117.00	43.64	2624	-117.00	62.56	2624	-117.00	44.79	2624
bmilplib-310-2	-105.00	109.26	5372	-105.00	96.63	5372	-105.00	117.75	5372	-105.00	98.39	5372
bmilplib-310-3	-127.52	171.44	7067	-127.52	148.52	7067	-127.52	174.17	7067	-127.52	149.07	7067
bmilplib-310-4	-147.78	90.78	4767	-147.78	77.28	4767	-147.78	95.75	4767	-147.78	70.22	4767
bmilplib-310-5	-161.45	48.58	1993	-161.45	34.30	1993	-161.45	45.88	1993	-161.45	34.22	1993
bmilplib-310-6	-141.18	197.54	8300	-141.18	179.02	8300	-141.18	198.16	8300	-141.18	169.96	8300
bmilplib-310-7	-142.00	179.47	7263	-142.00	144.99	7263	-142.00	169.98	7263	-142.00	139.15	7263
bmilplib-310-8	-115.34	25.48	921	-115.34	20.24	921	-115.34	24.68	921	-115.34	19.23	921
bmilplib-310-9	-115.65	43.85	1590	-115.65	37.72	1590	-115.65	45.18	1590	-115.65	32.31	1590
bmilplib-360-10	-108.59	27.82	1106	-108.59	25.17	1106	-108.59	31.69	1106	-108.59	25.75	1106
bmilplib-360-1	-133.00	207.62	4923	-133.00	146.36	4923	-133.00	194.58	4923	-133.00	158.38	4923
bmilplib-360-2	-138.44	230.49	4493	-138.44	135.61	4493	-138.44	225.09	4493	-138.44	148.90	4493
bmilplib-360-3	-131.00	68.30	2654	-131.00	63.40	2654	-131.00	75.82	2654	-131.00	65.41	2654
bmilplib-360-4	-119.00	54.68	1564	-119.00	47.06	1564	-119.00	51.33	1564	-119.00	42.95	1564
bmilplib-360-5	-164.26	58.30	1713	-164.26	45.64	1713	-164.26	56.12	1713	-164.26	44.45	1713
bmilplib-360-6	-110.12	155.42	5520	-110.12	132.79	5520	-110.12	146.64	5520	-110.12	142.81	5520
bmilplib-360-7	-105.00	135.16	3346	-105.00	93.50	3346	-105.00	125.91	3346	-105.00	89.22	3346
bmilplib-360-8	-98.25	66.38	1686	-98.25	50.86	1686	-98.25	66.34	1686	-98.25	44.85	1686
bmilplib-360-9	-127.22	67.05	2642	-127.22	58.36	2642	-127.22	68.26	2642	-127.22	50.24	2642
bmilplib-410-10	-153.37	332.48	7428	-153.37	265.98	7428	-153.37	333.37	7428	-153.37	258.89	7428
bmilplib-410-1	-103.50	103.54	1944	-103.50	85.04	1944	-103.50	109.43	1944	-103.50	87.94	1944

Table 8: Detailed results of Figure 4a (continued)

	withoutPoolWhen XYInt-LFixed			withPoolWhen XYInt-LFixed			withoutPoolWhen XYIntOrLFixed-LFixed			withPoolWhen XYIntOrLFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-410-2	-108.59	128.11	2887	-108.59	107.05	2887	-108.59	125.37	2887	-108.59	103.57	2887
bmilplib-410-3	-96.24	176.89	3781	-96.24	162.97	3781	-96.24	167.58	3781	-96.24	147.99	3781
bmilplib-410-4	-119.50	111.28	2995	-119.50	96.04	2995	-119.50	115.22	2995	-119.50	100.21	2995
bmilplib-410-5	-119.22	72.48	1520	-119.22	67.90	1520	-119.22	73.09	1520	-119.22	71.23	1520
bmilplib-410-6	-151.31	14.12	533	-151.31	14.66	533	-151.31	13.95	533	-151.31	13.54	533
bmilplib-410-7	-123.00	43.96	1177	-123.00	42.46	1177	-123.00	42.43	1177	-123.00	35.87	1177
bmilplib-410-8	-125.78	292.85	4480	-125.78	168.87	4480	-125.78	284.91	4480	-125.78	169.15	4480
bmilplib-410-9	-100.77	95.70	2071	-100.77	81.09	2071	-100.77	89.94	2071	-100.77	82.17	2071
bmilplib-460-10	-102.51	284.10	4465	-102.51	232.16	4465	-102.51	272.39	4465	-102.51	228.12	4465
bmilplib-460-1	-97.59	638.06	10803	-97.59	574.36	10803	-97.59	636.80	10803	-97.59	569.22	10803
bmilplib-460-2	-139.00	65.03	964	-139.00	51.23	964	-139.00	68.32	964	-139.00	43.65	964
bmilplib-460-3	-86.50	226.69	3882	-86.50	223.11	3882	-86.50	229.43	3882	-86.50	223.58	3882
bmilplib-460-4	-107.03	640.32	7856	-107.03	413.15	7856	-107.03	607.29	7856	-107.03	412.76	7856
bmilplib-460-5	-100.50	229.94	3025	-100.50	171.50	3025	-100.50	196.94	3025	-100.50	170.30	3025
bmilplib-460-6	-107.00	251.71	4143	-107.00	219.39	4143	-107.00	221.43	4143	-107.00	236.30	4143
bmilplib-460-7	-83.75	334.13	5252	-83.75	295.91	5252	-83.75	331.02	5252	-83.75	294.68	5252
bmilplib-460-8	-115.39	117.85	1903	-115.39	102.20	1903	-115.39	134.03	1903	-115.39	94.67	1903
bmilplib-460-9	-128.70	409.66	6185	-128.70	346.00	6185	-128.70	388.74	6185	-128.70	327.68	6185
bmilplib-60-10	-186.21	4.82	2590	-186.21	4.12	2590	-186.21	4.90	2590	-186.21	4.29	2590
bmilplib-60-1	-153.20	3.59	1094	-153.20	2.83	1094	-153.20	3.82	1094	-153.20	2.79	1094
bmilplib-60-5	-116.40	16.18	9996	-116.40	10.33	9996	-116.40	15.85	9996	-116.40	10.08	9996
bmilplib-60-6	-187.31	3.58	1383	-187.31	3.43	1383	-187.31	3.65	1383	-187.31	3.34	1383
bmilplib-60-8	-232.12	1.79	572	-232.12	1.16	572	-232.12	1.78	572	-232.12	1.15	572
bmilplib-60-9	-136.50	14.85	9888	-136.50	12.36	9888	-136.50	15.14	9888	-136.50	12.34	9888
miblp-20-15-50-0110-10-10	-206.00	36.93	21203	-206.00	4.22	15741	-206.00	39.37	15741	-206.00	4.25	15741
miblp-20-15-50-0110-10-2	-354.00	>3600	642894	-398.00	970.06	2929420	-354.00	>3600	596423	-398.00	947.81	2929420
miblp-20-15-50-0110-10-3	-42.00	188.57	48461	-42.00	12.65	46349	-42.00	203.74	46349	-42.00	12.66	46349
miblp-20-15-50-0110-10-6	-246.00	12.02	1385	-246.00	1.53	1367	-246.00	13.22	1367	-246.00	1.56	1367
miblp-20-15-50-0110-10-9	-635.00	25.65	7667	-635.00	6.62	7667	-635.00	26.03	7667	-635.00	6.52	7667
miblp-20-20-50-0110-10-10	-405.00	>3600	304054	-441.00	2908.94	7261247	-405.00	>3600	297842	-441.00	2867.85	7261247
miblp-20-20-50-0110-10-1	-315.00	>3600	281846	-357.00	>3600	9176214	-315.00	>3600	268812	-357.00	>3600	9234364
miblp-20-20-50-0110-10-2	-659.00	14.46	5106	-659.00	3.21	5100	-659.00	13.90	5100	-659.00	3.16	5100
miblp-20-20-50-0110-10-3	-618.00	16.74	52259	-618.00	12.21	51970	-618.00	17.08	51970	-618.00	12.05	51970
miblp-20-20-50-0110-10-4	-597.00	>3600	509252	-604.00	>3600	8018349	-597.00	>3600	502089	-604.00	>3600	7988914
miblp-20-20-50-0110-10-7	-627.00	>3600	4935849	-629.00	>3600	9735490	-627.00	>3600	4698965	-629.00	>3600	9709672
miblp-20-20-50-0110-10-8	-667.00	998.04	80603	-667.00	68.15	75661	-667.00	1085.58	75661	-667.00	68.08	75661
miblp-20-20-50-0110-10-9	-256.00	2292.26	852328	-256.00	306.60	757349	-256.00	2483.24	757349	-256.00	305.78	757349
miblp-20-20-50-0110-15-1	-246.00	>3600	250531	-317.00	>3600	9807301	-246.00	>3600	238329	-317.00	>3600	9813005
miblp-20-20-50-0110-15-2	-645.00	>3600	1416126	-645.00	>3600	10740865	-645.00	>3600	1395442	-645.00	>3600	10839884
miblp-20-20-50-0110-15-3	-476.00	>3600	189624	-593.00	>3600	13332780	-476.00	>3600	183565	-593.00	>3600	13042109
miblp-20-20-50-0110-15-4	-310.00	>3600	125392	-398.00	>3600	8670375	-310.00	>3600	125871	-398.00	>3600	8692428
miblp-20-20-50-0110-15-5	-60.00	>3600	48876	-320.00	>3600	7275287	-60.00	>3600	47748	-320.00	>3600	7284040
miblp-20-20-50-0110-15-6	-596.00	>3600	227767	-596.00	>3600	7842912	-596.00	>3600	197263	-596.00	>3600	7851818
miblp-20-20-50-0110-15-7	-471.00	>3600	393906	-471.00	>3600	9577404	-471.00	>3600	362335	-471.00	>3600	9675451
miblp-20-20-50-0110-15-8	-290.00	>3600	1772115	-290.00	>3600	10439854	-290.00	>3600	1614730	-290.00	>3600	10350188
miblp-20-20-50-0110-15-9	-584.00	269.87	56931	-584.00	19.38	56459	-584.00	271.42	56459	-584.00	19.58	56459
miblp-20-20-50-0110-5-13	-507.00	>3600	6006058	-519.00	2723.02	6515097	-507.00	>3600	6189742	-519.00	2709.29	6515097
miblp-20-20-50-0110-5-15	-617.00	1754.55	4312915	-617.00	1113.82	4312915	-617.00	1844.46	4312915	-617.00	1130.92	4312915
miblp-20-20-50-0110-5-16	-833.00	2.06	5913	-833.00	1.76	5913	-833.00	2.06	5913	-833.00	1.80	5913
miblp-20-20-50-0110-5-17	-944.00	1.61	4050	-944.00	1.50	4038	-944.00	1.66	4038	-944.00	1.53	4038
miblp-20-20-50-0110-5-19	-431.00	38.74	116041	-431.00	25.54	116041	-431.00	38.55	116041	-431.00	25.50	116041
miblp-20-20-50-0110-5-1	-548.00	13.07	31432	-548.00	8.63	31298	-548.00	13.53	31298	-548.00	8.45	31298
miblp-20-20-50-0110-5-20	-438.00	13.53	33315	-438.00	10.80	33315	-438.00	13.51	33315	-438.00	10.82	33315
miblp-20-20-50-0110-5-6	-1061.00	66.75	214009	-1061.00	51.65	213928	-1061.00	67.85	213928	-1061.00	51.74	213928
lseu-0.100000	1120.00	3.17	8603	1120.00	3.12	8603	1120.00	3.16	8603	1120.00	3.15	8603
lseu-0.900000	5838.00	>3600	2463002	5838.00	>3600	8762978	5838.00	>3600	2460978	5838.00	>3600	8743754
p0033-0.500000	3095.00	0.28	1467	3095.00	0.27	1467	3095.00	0.26	1467	3095.00	0.25	1467
p0033-0.900000	4679.00	37.10	33451	4679.00	5.58	28241	4679.00	31.19	28241	4679.00	5.56	28241
p0201-0.900000	15025.00	>3600	690347	15025.00	>3600	1309676	15025.00	>3600	689225	15025.00	>3600	1310278
stein27-0.500000	19.00	6.60	17648	19.00	6.43	17648	19.00	6.84	17648	19.00	6.51	17648
stein27-0.900000	24.00	644.53	713061	24.00	413.77	702055	24.00	638.54	702055	24.00	419.87	702055
stein45-0.100000	30.00	64.50	90241	30.00	63.84	90241	30.00	64.46	90241	30.00	64.27	90241
stein45-0.500000	32.00	465.23	753845	32.00	468.17	753845	32.00	472.32	753845	32.00	471.57	753845
stein45-0.900000	40.00	>3600	1379438	40.00	>3600	1504193	40.00	>3600	1376878	40.00	>3600	1499583

Table 9: Detailed results of Figure 4b

	withoutPoolWhen XYInt-LFixed			withPoolWhen XYInt-LFixed			withoutPoolWhen XYIntOrLFixed-LFixed			withPoolWhen XYIntOrLFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-110-10	-177.67	416.71	121469	-177.67	149.10	93801	-177.67	155.24	75499	-177.67	153.82	75499
bmilplib-110-1	-181.67	5.52	2892	-181.67	5.42	2878	-181.67	5.32	2806	-181.67	5.35	2806
bmilplib-110-2	-110.67	7.04	4279	-110.67	6.50	4272	-110.67	6.49	4174	-110.67	6.48	4174
bmilplib-110-3	-215.16	6.11	3599	-215.16	5.42	3556	-215.16	5.55	3471	-215.16	5.58	3471
bmilplib-110-4	-197.29	4.64	1827	-197.29	3.54	1801	-197.29	3.70	1751	-197.29	3.69	1751
bmilplib-110-6	-148.25	19.75	9578	-148.25	14.14	9392	-148.25	14.88	8770	-148.25	15.06	8770
bmilplib-110-7	-160.86	4.30	2291	-160.86	4.01	2281	-160.86	4.86	2189	-160.86	4.94	2189
bmilplib-110-8	-155.00	25.37	12659	-155.00	20.65	12498	-155.00	20.78	11663	-155.00	20.78	11663
bmilplib-110-9	-192.92	3.33	2177	-192.92	3.28	2171	-192.92	3.63	2151	-192.92	3.66	2151
bmilplib-160-10	-189.82	40.10	9425	-189.82	31.24	9344	-189.82	33.76	9071	-189.82	33.42	9071
bmilplib-160-1	-165.00	26.18	8157	-165.00	24.18	8145	-165.00	25.67	7901	-165.00	25.70	7901
bmilplib-160-2	-178.24	33.86	8719	-178.24	29.64	8710	-178.24	30.82	8635	-178.24	30.59	8635
bmilplib-160-3	-174.94	74.85	15456	-174.94	54.62	15427	-174.94	65.13	15002	-174.94	63.37	15002
bmilplib-160-4	-135.83	69.61	15513	-135.83	50.86	15354	-135.83	51.56	14772	-135.83	51.02	14772
bmilplib-160-5	-140.78	45.58	5362	-140.78	19.35	5178	-140.78	20.58	4948	-140.78	20.52	4948
bmilplib-160-6	-111.00	27.27	8878	-111.00	25.08	8864	-111.00	24.82	8676	-111.00	24.66	8676
bmilplib-160-7	-96.00	67.91	17668	-96.00	50.97	17430	-96.00	52.58	16560	-96.00	51.20	16560
bmilplib-160-8	-181.40	9.50	3564	-181.40	9.32	3540	-181.40	10.45	3444	-181.40	10.49	3444
bmilplib-160-9	-207.50	15.01	4754	-207.50	14.90	4728	-207.50	15.87	4607	-207.50	15.85	4607
bmilplib-210-10	-130.59	66.32	12596	-130.59	61.00	12497	-130.59	61.65	12176	-130.59	61.81	12176
bmilplib-210-1	-136.80	43.30	7608	-136.80	40.04	7598	-136.80	41.01	7429	-136.80	41.96	7429
bmilplib-210-2	-117.80	132.27	23482	-117.80	109.56	23096	-117.80	118.80	22294	-117.80	113.38	22294
bmilplib-210-3	-130.80	60.91	9404	-130.80	47.44	9165	-130.80	48.09	8848	-130.80	48.19	8848
bmilplib-210-4	-162.20	27.29	4812	-162.20	25.92	4806	-162.20	26.63	4687	-162.20	26.81	4687
bmilplib-210-5	-134.00	117.77	21646	-134.00	105.03	21552	-134.00	106.92	21058	-134.00	108.22	21058
bmilplib-210-6	-125.43	261.84	40736	-125.43	197.88	39941	-125.43	213.31	38443	-125.43	201.40	38443
bmilplib-210-7	-169.73	82.57	14351	-169.73	77.62	14305	-169.73	77.43	13960	-169.73	77.84	13960
bmilplib-210-8	-101.46	67.82	11406	-101.46	58.14	11347	-101.46	56.85	11105	-101.46	61.79	11105
bmilplib-210-9	-184.00	912.62	143788	-184.00	859.46	143665	-184.00	849.38	142294	-184.00	879.61	142294
bmilplib-260-10	-151.73	559.37	63243	-151.73	502.94	63026	-151.73	514.35	61767	-151.73	546.55	61767
bmilplib-260-1	-139.00	82.50	11269	-139.00	81.74	11242	-139.00	81.82	10980	-139.00	85.81	10980
bmilplib-260-2	-82.62	125.39	15896	-82.62	109.75	15712	-82.62	107.89	15287	-82.62	109.75	15287
bmilplib-260-3	-144.25	74.50	8865	-144.25	69.26	8859	-144.25	74.50	8769	-144.25	71.54	8769
bmilplib-260-4	-117.33	284.35	34237	-117.33	260.01	33931	-117.33	272.56	32787	-117.33	259.48	32787
bmilplib-260-5	-121.00	181.75	22295	-121.00	173.32	22121	-121.00	167.03	21542	-121.00	166.03	21542
bmilplib-260-6	-124.00	240.41	26166	-124.00	209.79	26023	-124.00	195.61	25362	-124.00	197.04	25362
bmilplib-260-7	-137.80	357.63	41519	-137.80	318.76	41180	-137.80	342.97	40274	-137.80	312.12	40274
bmilplib-260-8	-119.89	87.09	10445	-119.89	76.70	10332	-119.89	75.72	10025	-119.89	87.01	10025
bmilplib-260-9	-160.00	318.76	34545	-160.00	271.18	34236	-160.00	273.16	33468	-160.00	273.66	33468
bmilplib-310-10	-141.86	119.62	10032	-141.86	114.38	10033	-141.86	110.92	9900	-141.86	121.07	9900
bmilplib-310-1	-117.00	363.13	29350	-117.00	329.80	29055	-117.00	346.52	28330	-117.00	329.71	28330
bmilplib-310-2	-105.00	519.72	45235	-105.00	457.69	44841	-105.00	519.36	43399	-105.00	497.66	43399
bmilplib-310-3	-127.52	781.32	67649	-127.52	783.89	67441	-127.52	803.04	66410	-127.52	777.74	66410
bmilplib-310-4	-147.78	584.37	49620	-147.78	572.76	48950	-147.78	564.98	47711	-147.78	569.43	47711
bmilplib-310-5	-161.45	392.22	32631	-161.45	382.13	32454	-161.45	396.10	31782	-161.45	366.90	31782
bmilplib-310-6	-141.18	1191.16	103385	-141.18	1264.60	103214	-141.18	1229.32	101904	-141.18	1191.51	101904
bmilplib-310-7	-142.00	1132.52	104378	-142.00	1199.47	104166	-142.00	1142.79	102000	-142.00	1129.24	102000
bmilplib-310-8	-115.34	129.19	11336	-115.34	114.80	11293	-115.34	137.89	11109	-115.34	127.94	11109
bmilplib-310-9	-115.65	262.22	20893	-115.65	248.24	20838	-115.65	254.18	20490	-115.65	255.27	20490
bmilplib-360-10	-108.59	254.46	14127	-108.59	242.94	14064	-108.59	261.28	13697	-108.59	234.92	13697
bmilplib-360-1	-133.00	1390.85	77371	-133.00	1297.32	77066	-133.00	1380.33	75421	-133.00	1353.62	75421
bmilplib-360-2	-138.44	980.62	54874	-138.44	999.50	54354	-138.44	953.84	52919	-138.44	965.52	52919
bmilplib-360-3	-131.00	689.69	41459	-131.00	624.78	41302	-131.00	643.52	40487	-131.00	832.88	40487
bmilplib-360-4	-119.00	334.15	18870	-119.00	338.49	18813	-119.00	331.26	18293	-119.00	332.90	18293
bmilplib-360-5	-164.26	527.22	30465	-164.26	484.66	30352	-164.26	526.38	29947	-164.26	618.86	29947
bmilplib-360-6	-110.12	1147.18	70479	-110.12	1018.82	70283	-110.12	1131.67	68863	-110.12	1181.02	68863
bmilplib-360-7	-105.00	554.89	32224	-105.00	517.46	31884	-105.00	479.93	30900	-105.00	634.17	30900
bmilplib-360-8	-98.25	438.62	23402	-98.25	416.97	23337	-98.25	423.75	22857	-98.25	362.89	22857
bmilplib-360-9	-127.22	708.99	41321	-127.22	746.66	41235	-127.22	656.26	40329	-127.22	736.16	40329
bmilplib-410-10	-153.37	2756.08	103891	-153.37	2879.03	103400	-153.37	2624.30	101447	-153.37	2729.77	101447
bmilplib-410-1	-103.50	614.03	21120	-103.50	589.62	21088	-103.50	590.37	20634	-103.50	553.76	20634
bmilplib-410-2	-108.59	846.60	32377	-108.59	748.06	32251	-108.59	746.96	31603	-108.59	840.56	31603

Table 9: Detailed results of Figure 4b (continued)

	withoutPoolWhen XYInt-LFixed			withPoolWhen XYInt-LFixed			withoutPoolWhen XYIntOrLFixed-LFixed			withPoolWhen XYIntOrLFixed-LFixed		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-410-3	-96.24	877.76	33590	-96.24	777.04	33473	-96.24	874.84	32882	-96.24	871.58	32882
bmilplib-410-4	-119.50	957.38	39535	-119.50	1041.97	39371	-119.50	944.66	38489	-119.50	1031.12	38489
bmilplib-410-5	-119.22	677.99	24188	-119.22	681.99	24163	-119.22	583.64	23852	-119.22	678.49	23852
bmilplib-410-6	-151.31	291.60	11213	-151.31	333.36	11193	-151.31	300.03	11130	-151.31	322.88	11130
bmilplib-410-7	-123.00	514.64	22684	-123.00	601.78	22628	-123.00	588.83	22146	-123.00	560.34	22146
bmilplib-410-8	-125.78	1679.79	64013	-125.78	1692.35	63421	-125.78	1462.52	62218	-125.78	1547.88	62218
bmilplib-410-9	-100.77	561.87	20968	-100.77	607.44	20920	-100.77	539.37	20400	-100.77	531.88	20400
bmilplib-460-10	-102.51	1837.01	56744	-102.51	1701.70	56265	-102.51	1964.44	55024	-102.51	1857.94	55024
bmilplib-460-1	-97.59	2829.00	87769	-97.59	2938.02	87709	-97.59	2843.15	86632	-97.59	2822.97	86632
bmilplib-460-2	-139.00	652.85	16943	-139.00	608.40	16863	-139.00	651.97	16623	-139.00	625.95	16623
bmilplib-460-3	-86.50	1976.55	58270	-86.50	2102.77	58229	-86.50	1941.05	57395	-86.50	1995.92	57395
bmilplib-460-4	-107.03	3577.74	97665	-107.03	3328.91	97160	-107.03	3160.13	95035	-107.03	3291.28	95035
bmilplib-460-5	-100.50	1519.85	41863	-100.50	1619.91	41796	-100.50	1517.44	41109	-100.50	1424.54	41109
bmilplib-460-6	-107.00	1603.92	46833	-107.00	1483.17	46732	-107.00	1528.18	46076	-107.00	1635.42	46076
bmilplib-460-7	-83.75	1813.99	49915	-83.75	1728.92	49717	-83.75	1826.46	48897	-83.75	1601.34	48897
bmilplib-460-8	-115.39	983.08	28147	-115.39	911.51	28075	-115.39	901.08	27572	-115.39	982.30	27572
bmilplib-460-9	-128.70	3006.84	88287	-128.70	2715.98	87796	-128.70	3043.95	85977	-128.70	2830.59	85977
bmilplib-60-10	-186.21	7.00	6136	-186.21	6.91	6134	-186.21	6.70	5922	-186.21	6.93	5922
bmilplib-60-1	-153.20	7.31	5060	-153.20	6.50	5041	-153.20	6.48	4877	-153.20	6.71	4877
bmilplib-60-5	-116.40	13.21	15368	-116.40	8.65	13984	-116.40	7.98	11202	-116.40	8.15	11202
bmilplib-60-6	-187.31	7.57	7310	-187.31	7.34	7304	-187.31	7.22	7016	-187.31	7.62	7016
bmilplib-60-8	-232.12	3.66	3700	-232.12	3.33	3683	-232.12	3.16	3570	-232.12	3.36	3570
bmilplib-60-9	-136.50	22.60	28947	-136.50	20.22	28603	-136.50	19.79	26792	-136.50	19.61	26792
miblp-20-15-50-0110-10-10	-206.00	5.32	3150	-206.00	0.97	1414	-206.00	1.03	423	-206.00	1.06	423
miblp-20-15-50-0110-10-2	-398.00	427.10	229000	-398.00	11.12	27625	-398.00	9.11	2450	-398.00	9.09	2450
miblp-20-15-50-0110-10-3	-42.00	6.10	3151	-42.00	0.71	1343	-42.00	0.62	267	-42.00	0.62	267
miblp-20-15-50-0110-10-6	-246.00	5.47	1183	-246.00	2.19	853	-246.00	4.47	340	-246.00	4.56	340
miblp-20-15-50-0110-10-9	-635.00	16.24	7425	-635.00	7.86	6457	-635.00	14.46	2380	-635.00	14.05	2380
miblp-20-20-50-0110-10-10	-441.00	3568.40	1159333	-441.00	526.56	639146	-441.00	717.56	134585	-441.00	692.00	134585
miblp-20-20-50-0110-10-1	-353.00	>3600	686972	-359.00	260.31	423149	-359.00	231.63	96141	-359.00	228.67	96141
miblp-20-20-50-0110-10-2	-659.00	5.16	7729	-659.00	2.78	7547	-659.00	3.80	3191	-659.00	3.69	3191
miblp-20-20-50-0110-10-3	-618.00	14.11	42906	-618.00	10.74	38188	-618.00	13.14	20788	-618.00	13.18	20788
miblp-20-20-50-0110-10-4	-604.00	>3600	1351975	-604.00	3405.57	6452671	-604.00	3089.99	830808	-604.00	3145.16	830808
miblp-20-20-50-0110-10-7	-650.00	>3600	7509639	-683.00	3184.99	11502091	-683.00	2009.54	3003967	-683.00	1887.37	3003967
miblp-20-20-50-0110-10-8	-667.00	155.00	39744	-667.00	40.78	30873	-667.00	87.06	12857	-667.00	80.45	12857
miblp-20-20-50-0110-10-9	-256.00	77.53	127374	-256.00	23.97	76055	-256.00	21.78	31245	-256.00	20.99	31245
miblp-20-20-50-0110-15-1	-289.00	>3600	474713	-450.00	65.16	49137	-450.00	65.21	3506	-450.00	59.38	3506
miblp-20-20-50-0110-15-2	-645.00	>3600	1897076	-645.00	96.13	346065	-645.00	49.73	17251	-645.00	50.14	17251
miblp-20-20-50-0110-15-3	-593.00	>3600	231893	-593.00	65.25	42877	-593.00	72.10	3081	-593.00	71.15	3081
miblp-20-20-50-0110-15-4	-323.00	>3600	205392	-441.00	36.99	29904	-441.00	43.47	1625	-441.00	43.00	1625
miblp-20-20-50-0110-15-5	-75.00	>3600	103173	-379.00	615.16	205025	-379.00	644.67	16715	-379.00	651.28	16715
miblp-20-20-50-0110-15-6	-596.00	>3600	317849	-596.00	17.87	29923	-596.00	18.98	1657	-596.00	17.31	1657
miblp-20-20-50-0110-15-7	-471.00	>3600	1084548	-471.00	125.24	241285	-471.00	126.10	13405	-471.00	111.02	13405
miblp-20-20-50-0110-15-8	-290.00	>3600	4898557	-370.00	138.34	579309	-370.00	39.01	21589	-370.00	39.12	21589
miblp-20-20-50-0110-15-9	-584.00	20.00	12193	-584.00	1.96	4072	-584.00	2.07	582	-584.00	2.00	582
miblp-20-20-50-0110-5-13	-519.00	2582.26	10718411	-519.00	2307.90	10196998	-519.00	2516.55	7128138	-519.00	2464.25	7128138
miblp-20-20-50-0110-5-15	-617.00	2170.42	8462491	-617.00	1219.33	6630921	-617.00	1297.16	4018058	-617.00	1310.67	4018058
miblp-20-20-50-0110-5-16	-833.00	6.04	20129	-833.00	4.86	19013	-833.00	7.36	17680	-833.00	6.60	17680
miblp-20-20-50-0110-5-17	-944.00	3.71	20049	-944.00	3.99	21541	-944.00	5.21	17679	-944.00	4.78	17679
miblp-20-20-50-0110-5-19	-431.00	52.75	197157	-431.00	27.52	147268	-431.00	32.98	77256	-431.00	26.56	77256
miblp-20-20-50-0110-5-1	-548.00	16.54	69127	-548.00	12.98	64067	-548.00	17.63	42364	-548.00	14.50	42364
miblp-20-20-50-0110-5-20	-438.00	17.24	80526	-438.00	15.81	76091	-438.00	18.83	60944	-438.00	17.30	60944
miblp-20-20-50-0110-5-6	-1061.00	66.85	301416	-1061.00	58.74	284550	-1061.00	63.82	222566	-1061.00	63.70	222566
lseu-0.100000	1120.00	248.50	1071409	1120.00	248.54	1071409	1120.00	262.55	1003976	1120.00	270.78	1003976
lseu-0.900000	5838.00	>3600	1631763	5838.00	1063.02	4718749	5838.00	14.12	1023	5838.00	14.10	1023
p0033-0.500000	3095.00	6.08	33614	3095.00	6.24	33614	3095.00	10.55	20695	3095.00	10.48	20695
p0033-0.900000	4679.00	14.66	23699	4679.00	0.65	3455	4679.00	0.06	27	4679.00	0.06	27
p0201-0.900000	15025.00	>3600	1037538	15025.00	20.58	18801	15025.00	6.66	2481	15025.00	6.62	2481
stein27-0.500000	19.00	7.94	22537	19.00	7.36	21515	19.00	5.82	12115	19.00	5.88	12115
stein27-0.900000	24.00	29.12	36927	24.00	1.25	4445	24.00	0.02	15	24.00	0.02	15
stein45-0.100000	30.00	49.86	89035	30.00	50.47	89035	30.00	50.33	89035	30.00	50.19	89035
stein45-0.500000	32.00	640.82	963098	32.00	635.22	952123	32.00	516.60	640308	32.00	519.59	640308
stein45-0.900000	40.00	>3600	2651427	40.00	85.92	103661	40.00	0.16	63	40.00	0.14	63

Table 10: Detailed results of Figure 5

	noHeuristics			impObjectiveCut			secondLevelPriority			weightedSums		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-110-10	-177.67	116.32	55177	-177.67	117.32	55177	-177.67	146.32	55177	-177.67	1024.90	55177
bmilplib-110-1	-181.67	0.93	306	-181.67	0.96	306	-181.67	2.90	306	-181.67	67.04	306
bmilplib-110-2	-110.67	1.16	303	-110.67	1.26	303	-110.67	1.64	303	-110.67	25.89	303
bmilplib-110-3	-215.16	0.89	360	-215.16	1.50	361	-215.16	1.93	360	-215.16	30.52	360
bmilplib-110-4	-197.29	1.34	148	-197.29	1.54	148	-197.29	2.79	148	-197.29	68.13	148
bmilplib-110-6	-148.25	3.84	1448	-148.25	4.24	1448	-148.25	4.84	1446	-148.25	59.02	1446
bmilplib-110-7	-160.86	0.91	205	-160.86	2.19	205	-160.86	1.98	205	-160.86	52.88	207
bmilplib-110-8	-155.00	7.66	2274	-155.00	7.68	2274	-155.00	10.06	2274	-155.00	73.62	2274
bmilplib-110-9	-192.92	0.39	146	-192.92	1.33	146	-192.92	1.15	146	-192.92	16.48	154
bmilplib-160-10	-189.82	7.73	728	-189.82	7.89	728	-189.82	11.28	728	-189.82	65.95	728
bmilplib-160-1	-165.00	4.76	881	-165.00	4.76	881	-165.00	5.82	893	-165.00	51.48	893
bmilplib-160-2	-178.24	5.95	507	-178.24	6.64	507	-178.24	6.96	507	-178.24	53.77	507
bmilplib-160-3	-174.94	13.71	1102	-174.94	13.57	1102	-174.94	15.78	1102	-174.94	70.46	1102
bmilplib-160-4	-135.83	19.13	2447	-135.83	18.91	2463	-135.83	24.00	2447	-135.83	116.48	2422
bmilplib-160-5	-140.78	7.72	668	-140.78	8.94	668	-140.78	10.34	668	-140.78	56.74	668
bmilplib-160-6	-111.00	5.02	627	-111.00	5.03	627	-111.00	5.40	627	-111.00	19.14	627
bmilplib-160-7	-96.00	18.72	2855	-96.00	17.83	2855	-96.00	19.22	2855	-96.00	74.44	2855
bmilplib-160-8	-181.40	1.74	311	-181.40	2.12	311	-181.40	2.44	311	-181.40	28.91	311
bmilplib-160-9	-207.50	2.02	268	-207.50	2.85	268	-207.50	2.98	268	-207.50	40.33	268
bmilplib-210-10	-130.59	8.59	1100	-130.59	8.66	1100	-130.59	10.02	1100	-130.59	73.44	1100
bmilplib-210-1	-136.80	6.67	550	-136.80	6.68	550	-136.80	7.59	550	-136.80	80.52	550
bmilplib-210-2	-117.80	19.78	2306	-117.80	20.49	2306	-117.80	22.29	2298	-117.80	149.96	2308
bmilplib-210-3	-130.80	12.93	1380	-130.80	13.04	1386	-130.80	16.16	1380	-130.80	218.18	1380
bmilplib-210-4	-162.20	3.64	309	-162.20	5.41	309	-162.20	4.63	309	-162.20	41.06	309
bmilplib-210-5	-134.00	20.03	2079	-134.00	20.03	2079	-134.00	22.31	2079	-134.00	161.23	2079
bmilplib-210-6	-125.43	45.28	4875	-125.43	46.46	4875	-125.43	52.83	4878	-125.43	230.84	4875
bmilplib-210-7	-169.73	11.46	1181	-169.73	12.35	1153	-169.73	14.19	1181	-169.73	148.58	1152
bmilplib-210-8	-101.46	9.60	942	-101.46	12.36	976	-101.46	12.71	942	-101.46	255.24	942
bmilplib-210-9	-184.00	240.14	9466	-184.00	242.65	9466	-184.00	267.90	9466	-184.00	1639.04	9466
bmilplib-260-10	-151.73	73.10	4716	-151.73	75.91	4716	-151.73	119.40	4716	-151.73	783.50	4716
bmilplib-260-1	-139.00	10.10	887	-139.00	11.58	887	-139.00	11.79	887	-139.00	135.30	887
bmilplib-260-2	-82.62	18.07	1607	-82.62	18.48	1607	-82.62	29.56	1607	-82.62	195.56	1607
bmilplib-260-3	-144.25	8.70	518	-144.25	9.34	518	-144.25	9.60	518	-144.25	105.89	518
bmilplib-260-4	-117.33	66.89	4426	-117.33	62.45	4426	-117.33	80.66	4426	-117.33	671.57	4426
bmilplib-260-5	-121.00	29.19	2165	-121.00	27.30	2165	-121.00	30.22	2165	-121.00	212.90	2165
bmilplib-260-6	-124.00	40.59	2420	-124.00	39.48	2420	-124.00	42.70	2420	-124.00	426.31	2420
bmilplib-260-7	-137.80	44.45	3200	-137.80	45.80	3200	-137.80	48.73	3200	-137.80	267.10	3200
bmilplib-260-8	-119.89	10.92	1025	-119.89	10.46	1025	-119.89	11.77	1025	-119.89	102.82	1025
bmilplib-260-9	-160.00	36.36	2526	-160.00	38.08	2526	-160.00	39.54	2526	-160.00	231.78	2526
bmilplib-310-10	-141.86	5.51	397	-141.86	6.24	397	-141.86	6.86	397	-141.86	95.24	397
bmilplib-310-1	-117.00	44.79	2624	-117.00	41.07	2624	-117.00	44.15	2624	-117.00	381.97	2624
bmilplib-310-2	-105.00	98.39	5372	-105.00	94.00	5372	-105.00	97.58	5372	-105.00	385.78	5372
bmilplib-310-3	-127.52	149.07	7067	-127.52	141.25	7067	-127.52	151.24	7067	-127.52	694.04	7067
bmilplib-310-4	-147.78	70.22	4767	-147.78	73.75	4908	-147.78	74.72	4863	-147.78	510.45	4892
bmilplib-310-5	-161.45	34.22	1993	-161.45	34.42	1993	-161.45	35.32	1993	-161.45	205.57	1993
bmilplib-310-6	-141.18	169.96	8300	-141.18	172.88	8300	-141.18	214.66	8300	-141.18	>3600	4459
bmilplib-310-7	-142.00	139.15	7263	-142.00	140.50	7263	-142.00	164.46	7263	-142.00	1416.36	7263
bmilplib-310-8	-115.34	19.23	921	-115.34	19.53	952	-115.34	23.06	921	-115.34	425.10	921
bmilplib-310-9	-115.65	32.31	1590	-115.65	32.39	1590	-115.65	34.56	1590	-115.65	151.96	1590
bmilplib-360-10	-108.59	25.75	1106	-108.59	23.23	1106	-108.59	26.41	1102	-108.59	156.13	1102
bmilplib-360-1	-133.00	158.38	4923	-133.00	148.68	4920	-133.00	188.86	4923	-120.00	>3600	1611
bmilplib-360-2	-138.44	148.90	4493	-138.44	138.96	4493	-138.44	160.91	4493	-138.44	1961.79	4493
bmilplib-360-3	-131.00	65.41	2654	-131.00	56.59	2654	-131.00	66.60	2654	-131.00	609.13	2654
bmilplib-360-4	-119.00	42.95	1564	-119.00	44.65	1564	-119.00	61.20	1564	-119.00	825.58	1564
bmilplib-360-5	-164.26	44.45	1713	-164.26	45.39	1713	-164.26	41.95	1713	-164.26	221.40	1713
bmilplib-360-6	-110.12	142.81	5520	-110.12	127.35	5520	-110.12	148.17	5520	-110.12	2890.94	5556
bmilplib-360-7	-105.00	89.22	3346	-105.00	88.94	3346	-105.00	86.08	3346	-105.00	357.79	3346
bmilplib-360-8	-98.25	44.85	1686	-98.25	48.77	1686	-98.25	72.01	1686	-98.25	2774.87	1686
bmilplib-360-9	-127.22	50.24	2642	-127.22	52.33	2642	-127.22	58.90	2642	-127.22	1191.63	2642
bmilplib-410-10	-153.37	258.89	7428	-153.37	254.18	7428	-153.37	278.87	7428	-153.37	2497.78	7428
bmilplib-410-1	-103.50	87.94	1944	-103.50	76.78	1944	-103.50	88.08	1944	-103.50	2256.70	1944

Table 10: Detailed results of Figure 5 (continued)

	noHeuristics			impObjectiveCut			secondLevelPriority			weightedSums		
Instance	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes	BestSol	Time(s)	Nodes
bmilplib-410-2	-108.59	103.57	2887	-108.59	92.86	2887	-108.59	97.35	2887	-108.59	947.06	2887
bmilplib-410-3	-96.24	147.99	3781	-96.24	157.30	3781	-96.24	257.83	3781	-79.49	>3600	878
bmilplib-410-4	-119.50	100.21	2995	-119.50	100.04	2995	-119.50	100.85	2995	-119.50	1042.52	2995
bmilplib-410-5	-119.22	71.23	1520	-119.22	61.46	1527	-119.22	107.73	1520	-119.22	426.60	1527
bmilplib-410-6	-151.31	13.54	533	-151.31	13.74	533	-151.31	17.48	533	-151.31	727.04	533
bmilplib-410-7	-123.00	35.87	1177	-123.00	35.79	1175	-123.00	41.41	1177	-123.00	406.53	1177
bmilplib-410-8	-125.78	169.15	4480	-125.78	171.56	4480	-125.78	190.33	4480	-125.78	1408.43	4488
bmilplib-410-9	-100.77	82.17	2071	-100.77	79.34	2071	-100.77	85.38	2071	-100.77	478.13	2071
bmilplib-460-10	-102.51	228.12	4465	-102.51	204.04	4465	-102.51	238.28	4465	-102.51	1557.08	4465
bmilplib-460-1	-97.59	569.22	10803	-97.59	610.72	10803	-97.59	666.13	10803	-93.40	>3600	1240
bmilplib-460-2	-139.00	43.65	964	-139.00	46.56	964	-139.00	45.27	964	-139.00	557.54	964
bmilplib-460-3	-86.50	223.58	3882	-86.50	219.35	3882	-86.50	251.89	3882	-82.83	>3600	2966
bmilplib-460-4	-107.03	412.76	7856	-107.03	425.15	7856	-107.03	399.79	7794	-102.61	>3600	4784
bmilplib-460-5	-100.50	170.30	3025	-100.50	189.63	3025	-100.50	177.04	3025	-100.50	1788.18	3025
bmilplib-460-6	-107.00	236.30	4143	-107.00	205.48	4143	-107.00	266.17	4143	-107.00	2457.96	4143
bmilplib-460-7	-83.75	294.68	5252	-83.75	282.83	5252	-83.75	341.16	5252	-83.75	>3600	1538
bmilplib-460-8	-115.39	94.67	1903	-115.39	92.26	1908	-115.39	140.92	1903	-103.50	>3600	611
bmilplib-460-9	-128.70	327.68	6185	-128.70	316.77	6227	-128.70	318.04	6185	-128.70	3166.24	6185
bmilplib-60-10	-186.21	4.29	2590	-186.21	4.32	2590	-186.21	4.68	2590	-186.21	22.24	2590
bmilplib-60-1	-153.20	2.79	1094	-153.20	3.01	1094	-153.20	3.47	1094	-153.20	18.65	1094
bmilplib-60-5	-116.40	10.08	9996	-116.40	10.41	9996	-116.40	12.63	9996	-116.40	35.73	9996
bmilplib-60-6	-187.31	3.34	1383	-187.31	3.68	1383	-187.31	4.07	1383	-187.31	22.35	1383
bmilplib-60-8	-232.12	1.15	572	-232.12	1.22	572	-232.12	1.61	572	-232.12	9.46	572
bmilplib-60-9	-136.50	12.34	9888	-136.50	12.45	9888	-136.50	13.56	9888	-136.50	47.50	9888
miblp-20-15-50-0110-10-10	-206.00	1.06	423	-206.00	1.07	423	-206.00	1.01	423	-206.00	1.81	423
miblp-20-15-50-0110-10-2	-398.00	9.09	2450	-398.00	9.43	2450	-398.00	8.85	2450	-398.00	15.52	2450
miblp-20-15-50-0110-10-3	-42.00	0.62	267	-42.00	0.72	267	-42.00	0.68	267	-42.00	1.36	267
miblp-20-15-50-0110-10-6	-246.00	4.56	340	-246.00	4.43	340	-246.00	4.24	340	-246.00	6.40	340
miblp-20-15-50-0110-10-9	-635.00	14.05	2380	-635.00	14.97	2382	-635.00	15.48	2380	-635.00	19.90	2380
miblp-20-20-50-0110-10-10	-441.00	692.00	134585	-441.00	758.37	134585	-441.00	743.82	134585	-441.00	1113.34	134585
miblp-20-20-50-0110-10-1	-359.00	228.67	96141	-359.00	232.30	96141	-359.00	242.52	96141	-359.00	407.03	96141
miblp-20-20-50-0110-10-2	-659.00	3.69	3191	-659.00	4.15	3191	-659.00	3.94	3191	-659.00	9.48	3191
miblp-20-20-50-0110-10-3	-618.00	13.18	20788	-618.00	13.68	20788	-618.00	14.77	20788	-618.00	54.78	20715
miblp-20-20-50-0110-10-4	-604.00	3145.16	830808	-604.00	3286.17	830808	-604.00	3223.61	830808	-604.00	>3600	666302
miblp-20-20-50-0110-10-7	-683.00	1887.37	3003967	-683.00	1960.64	3014456	-683.00	2049.54	3004274	-683.00	>3600	2542388
miblp-20-20-50-0110-10-8	-667.00	80.45	12857	-667.00	94.47	12857	-667.00	87.72	12858	-667.00	134.18	12859
miblp-20-20-50-0110-10-9	-256.00	20.99	31245	-256.00	21.84	31245	-256.00	22.38	31245	-256.00	38.71	31245
miblp-20-20-50-0110-15-1	-450.00	59.38	3506	-450.00	59.36	3506	-450.00	62.04	3506	-450.00	85.24	3506
miblp-20-20-50-0110-15-2	-645.00	50.14	17251	-645.00	52.11	17251	-645.00	61.00	17251	-645.00	222.62	17251
miblp-20-20-50-0110-15-3	-593.00	71.15	3081	-593.00	66.66	3081	-593.00	67.67	3081	-593.00	95.58	3083
miblp-20-20-50-0110-15-4	-441.00	43.00	1625	-441.00	43.74	1625	-441.00	39.84	1625	-441.00	57.56	1625
miblp-20-20-50-0110-15-5	-379.00	651.28	16715	-379.00	634.58	16715	-379.00	639.80	16715	-379.00	750.81	16715
miblp-20-20-50-0110-15-6	-596.00	17.31	1657	-596.00	17.40	1657	-596.00	18.50	1657	-596.00	25.94	1657
miblp-20-20-50-0110-15-7	-471.00	111.02	13405	-471.00	109.14	13405	-471.00	113.83	13405	-471.00	191.29	13405
miblp-20-20-50-0110-15-8	-370.00	39.12	21589	-370.00	39.12	21589	-370.00	44.93	21589	-370.00	122.50	21589
miblp-20-20-50-0110-15-9	-584.00	2.00	582	-584.00	2.24	582	-584.00	1.93	582	-584.00	3.44	584
miblp-20-20-50-0110-5-13	-519.00	2709.29	6515097	-519.00	2756.40	6514987	-519.00	2828.20	6521367	-519.00	3465.54	6515355
miblp-20-20-50-0110-5-15	-617.00	1130.92	4312915	-617.00	1123.27	4312915	-617.00	1202.16	4312915	-617.00	1826.00	4309751
miblp-20-20-50-0110-5-16	-833.00	1.80	5913	-833.00	1.76	5913	-833.00	2.22	5925	-833.00	9.74	5925
miblp-20-20-50-0110-5-17	-944.00	1.53	4038	-944.00	1.48	4037	-944.00	1.84	4038	-944.00	8.31	4212
miblp-20-20-50-0110-5-19	-431.00	25.50	116041	-431.00	26.02	116041	-431.00	27.33	116065	-431.00	45.05	116285
miblp-20-20-50-0110-5-1	-548.00	8.45	31298	-548.00	9.08	31298	-548.00	9.13	31298	-548.00	22.96	31298
miblp-20-20-50-0110-5-20	-438.00	10.82	33315	-438.00	10.92	33315	-438.00	11.25	33315	-438.00	17.54	33130
miblp-20-20-50-0110-5-6	-1061.00	51.74	213928	-1061.00	52.57	213928	-1061.00	56.75	214449	-1061.00	235.63	234647
lseu-0.100000	1120.00	3.15	8603	1120.00	3.11	8603	1120.00	3.82	13352	1120.00	45.32	22411
lseu-0.900000	5838.00	14.10	1023	5838.00	14.11	1023	5838.00	251.51	1023	5838.00	21.57	1023
p0033-0.500000	3095.00	0.25	1467	3095.00	0.28	1467	3095.00	0.29	1471	3095.00	0.80	2239
p0033-0.900000	4679.00	0.06	27	4679.00	0.06	27	4679.00	0.07	27	4679.00	0.12	27
p0201-0.900000	15025.00	6.62	2481	15025.00	7.09	2481	15025.00	359.66	2481	15025.00	34.52	2481
stein27-0.500000	19.00	6.51	17648	19.00	6.44	17648	19.00	6.76	16969	19.00	9.57	16969
stein27-0.900000	24.00	0.02	15	24.00	0.02	15	24.00	0.02	15	24.00	1.62	15
stein45-0.100000	30.00	64.27	90241	30.00	64.12	90241	30.00	83.80	90002	30.00	256.23	58729
stein45-0.500000	32.00	471.57	753845	32.00	459.73	753845	32.00	511.05	862249	32.00	880.47	565867
stein45-0.900000	40.00	0.14	63	40.00	0.16	63	40.00	0.16	63	40.00	72.23	63