BASBL: BRANCH-AND-SANDWICH BILEVEL SOLVER. II. IMPLEMENTATION AND COMPUTATIONAL STUDY WITH THE BASBLIB TEST SET*

REMIGIJUS PAULAVIČIUS†, POLYXENI-M. KLENIATI‡, AND CLAIRE S. ADJIMAN§

Abstract. We describe BASBL, our implementation of the deterministic global optimization algorithm Branch-and-Sandwich for nonconvex/nonlinear bilevel problems, within the open-source MINOTAUR framework. The solver incorporates the original Branch-and-Sandwich algorithm and modifications proposed in the first part of this work. We also introduce BASBLib, an extensive online library of bilevel benchmark problems collected from the literature and designed to enable contributions from the bilevel optimization community. We use the problems in the current release of BASBLib to analyze the performance of BASBL using different algorithmic options.

Key words. Nonconvex bilevel programming, Branch-and-Sandwich algorithm, Optimization software, BASBL solver, MINOTAUR toolkit, BASBLib

AMS subject classifications. 65K05, 90C26, 90C30, 90C57

1. Introduction.

The numerous applications of bilevel programming problems (BPP) (see e.g. [10, 28, 29, 31] and references therein) provide a strong incentive for developing efficient solvers for this large class of problems. BPP are hierarchical optimization problems in which the outer problem constraints involve the solution set of an embedded inner global optimization problem parameterized by the outer level variables $\mathbf{x} \in X$:

$$\text{(BPP)} \quad \begin{array}{lll} \min\limits_{\mathbf{x},\mathbf{y}} & F(\mathbf{x},\mathbf{y}) & \\ \text{s.t.} & \mathbf{G}(\mathbf{x},\mathbf{y}) \le \mathbf{0}, & \mathbf{H}(\mathbf{x},\mathbf{y}) = \mathbf{0}, \\ & \mathbf{x} \in X, & \mathbf{y} \in \arg\min\limits_{\mathbf{y} \in Y} \{ f(\mathbf{x},\mathbf{y}) \text{ s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \le \mathbf{0},\ \mathbf{h}(\mathbf{x},\mathbf{y}) = \mathbf{0} \}. \end{array}$$

Here the n-dimensional vector $\mathbf{x} \in X \subset \mathbb{R}^n$ denotes the outer-level (leader's) variables and the m-dimensional vector $\mathbf{y} \in Y \subset \mathbb{R}^m$ denotes the inner-level (follower's) variables. Functions $F, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ denote the outer/inner-level objective functions, $\mathbf{G} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^r$ are the vector-valued outer/inner-level inequality constraint functions and $\mathbf{H} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ and $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^s$ are vector-valued outer/inner equality constraint functions. The optimistic (co-operative) [28] formulation is assumed, where if for a given $\mathbf{x}$ the inner subproblem (ISP($\mathbf{x}$))

$$\text{(ISP}(\mathbf{x})) \quad \min_{\mathbf{y} \in Y} \{ f(\mathbf{x},\mathbf{y}) \text{ s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \le \mathbf{0},\ \mathbf{h}(\mathbf{x},\mathbf{y}) = \mathbf{0} \},$$

has multiple globally optimal solutions $\mathbf{y}$ to which the follower is indifferent, the leader can choose among them. Hence, the outer minimization in (BPP) is performed with respect to the whole set of variables.

In this work, we seek to find an ε-optimal bilevel solution to problem (BPP), which is defined as follows:

*Department of Chemical Engineering, Centre for Process Systems Engineering, Imperial College London, London SW7 2AZ, UK.

Funding: We gratefully acknowledge funding from the EPSRC through a Leadership Fellowship [EP/J003840/1].

† (remigijus.paulavicius@imperial.ac.uk).

‡ Process Systems Enterprise, 26-28 Hammersmith Grove, London W6 7HA, UK (pmkleniati@gmail.com).

§ (c.adjiman@imperial.ac.uk).

Definition 1.1 (ε-optimal bilevel solution [66]). *A pair $(\mathbf{x}^*, \mathbf{y}^*) \in X \times Y$ is an ε-optimal solution if it satisfies the inequality and equality constraints of the outer and inner problems, as well as ε_F-optimality in the outer problem and ε_f-optimality in the inner problem:*

$$\mathbf{G}(\mathbf{x}^*, \mathbf{y}^*) \le \mathbf{0}, \qquad \mathbf{g}(\mathbf{x}^*, \mathbf{y}^*) \le \mathbf{0}, \tag{1.1}$$

$$\mathbf{H}(\mathbf{x}^*, \mathbf{y}^*) = \mathbf{0}, \qquad \mathbf{h}(\mathbf{x}^*, \mathbf{y}^*) = \mathbf{0}, \tag{1.2}$$

$$F(\mathbf{x}^*, \mathbf{y}^*) \le F^* + \varepsilon_F, \quad f(\mathbf{x}^*, \mathbf{y}^*) \le w(\mathbf{x}^*) + \varepsilon_f, \tag{1.3}$$

where F^* denotes the outer optimal objective value, i.e., the optimal objective value of the bilevel problem (BPP) and $w(\mathbf{x}^*)$ is obtained from the solution of (ISP($\mathbf{x}$)).

While problem (BPP) has been studied for a long time, the few software codes that are currently available are mainly limited to special subclasses. The bilevel solver in `YALMIP` (language for advanced modeling and solution of convex and non-convex optimization problems) [54] is restricted to convex quadratic inner problems, but convexity is not a requirement on the outer problem. Similarly, `BIPA` (BIlevel Programming with Approximation methods) [21], a software based on a trust-region method for nonlinear bilevel programming problems [22], requires f to be convex in $\mathbf{y}$ for each fixed value of $\mathbf{x}$. The `EMP` (Extended Mathematical Programming) tool in `GAMS` (General Algebraic Modeling System) [23] automatically creates an MPEC (Mathematical Program with Equilibrium Constraints) [56] by expressing the lower level optimization problem via its Karush-Kuhn-Tucker (KKT) optimality conditions and finds a solution of the MPEC, not of the bilevel program. If the lower level program is nonconvex, the optimal solution of a bilevel problem may not even be a stationary point of the reduced single-level optimization problem [61]. Thus this solver is limited to bilevel problems with a convex inner problem that meets a constraint qualification. In a recent article [37], Fischetti el. al. introduced a novel publicly available solver [36] for mixed-integer bilevel linear programs. Finally, the publicly available `MibS` solver [72] requires that both the leader and the follower are purely integer problems.

To handle general bilevel problems that do not adhere to the simplifying assumptions required by these deterministic solvers, two evolutionary-based algorithms, that have been implemented in `MATLAB`, are available (http://www.bilevel.org/).

To address the lack of general deterministic bilevel solvers, we introduce the first implementation of our bilevel solver capable of handling very general (BPP) problems. General bilevel problems are very challenging and only recently have the first algorithms to tackle them been proposed: the deterministic approach of Mitsos et al. [66], the approximation method of Tsoukalas et al. [82], and the Branch-and-Sandwich (B&S) algorithm introduced in [51, 52] and extended in [69]. The solver presented here, **B**ranch-**A**nd-**S**andwich **BiL**evel algorithm (`BASBL`), is based on this latter work. It is implemented within the open-source `MINOTAUR` toolkit [57, 58, 59, 60].

The structure of this paper is as follows. In Section 2 the full implementation of `BASBL` is presented. Section 3 then describes `BASBLib`, a new library of bilevel test problems collected from the literature. In Section 4 an example of `BASBL` usage on one of the test problems is provided. In Section 5 the `BASBL` solver is applied to the problems in the library, including nonconvex bilevel benchmark problems and the impact of different algorithmic options on computational performance is investigated. Finally, in Section 6 we draw conclusions and discuss potential directions for future work.

2. Implementation of the BASBL solver. In this section, we describe the implementation of the BASBL solver. An understanding of the B&S algorithm, as described in part I of this work [69], is assumed.

2.1. Prerequisites. We start by identifying the assumptions needed to ensure the convergence of the BASBL solver to global optimality. In global optimization, it is common to assume continuity of the participating functions and compactness of the host sets, i.e., finite lower and upper bounds on all problem variables:

DEFINITION 2.2 (Prerequisites for the BASBL solver).

1. *Explicit bounds are known for all variables, i.e.* $\mathbf{x} \in X = [\mathbf{x}^{\mathrm{L}}, \mathbf{x}^{\mathrm{U}}] \subset \mathbb{R}^n$ *and* $\mathbf{y} \in Y = [\mathbf{y}^{\mathrm{L}}, \mathbf{y}^{\mathrm{U}}] \subset \mathbb{R}^m$.
2. *All functions involved in* (BPP) *must be bounded from below and/or above and continuous on* $X \times Y$.
3. *All functions involved in* (BPP) *meet the conditions imposed by the single level optimization solver used (e.g., BARON [75, 81] is applicable to factorable functions)*
4. *A constraint qualification holds for the inner problem* (ISP($\mathbf{x}$)) *for all* $\mathbf{x} \in X$ *values.*

During its execution, BASBL solver requires the global solution of nonconvex nonlinear (sub)problems (NLP), as described in [51, 69]. Note that even in the case of single-level optimization, algorithms that guarantee convergence to a global optimum in finite time exist only for special cases, e.g., linear or convex problems. For nonconvex nonlinear problems, modern solvers (see, e.g., [1, 12, 62, 81]) offer finite-time convergence to an ε-optimal solution only. That is, they provide a lower bound on the optimal objective function value and a feasible point with an objective function value that is not more than ε larger than the lower bound. Taking this into account, the BASBL solver can provide a ε-optimal bilevel solution (Definition 1.1) of (BPP) in finite time, within the tolerances used for the necessary solution of the relevant subproblems. For example, a feasibility tolerance is specified to verify the satisfaction of constraints such as $\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}$.

2.1.1. Modifications to MINOTAUR classes. The BASBL solver is built within MINOTAUR [57, 58, 60], a flexible open-source toolkit written in C++ for solving mixed-integer nonlinear problems (MINLP).

MINOTAUR provides an interface through which the user can implement a customized branch-and-bound algorithm. However, the B&S algorithm, which is the basis for the BASBL solver, differs significantly from classical branch-and-bound algorithms. In particular,

- B&S uses a list of *inner-active* nodes as well as independent list and sublists, in addition to the usual list of *active* nodes;
- every node in the Branch-and-Sandwich tree must hold lower and upper bounds for both the inner and outer problems.

We have extended the MINOTAUR search tree and nodes management facilities to enable these features.

2.1.2. NLP solvers. The nonconvex nonlinear subproblems that need to be solved during a BASBL run are shown in Table 2.1. Natively available MINOTAUR solvers (IPOPT (Interior Point OPTimizer) [84] and Filter-SQP [38]) can ensure only the local optimality of NLP problems. We therefore implemented an interface to GAMS (General Algebraic Modeling System) [23] which provides access to a range of state-of-the-art NLP solvers, including deterministic global solvers such as BARON [75, 81] and

ANTIGONE [62]. Furthermore, a native solver for the global solution of NLP problems, based on the αBB method [1, 2], has recently been implemented under the MINOTAUR framework [49]. BASBL subproblems can currently be solved to global optimality with GAMS/BARON or GAMS/ANTIGONE through system calls or using the MINOTAUR/αBB solver.

TABLE 2.1
Summary of bounding NLPs to be solved during the execution of BASBL

Problem description	Solution	Problem name†	Solver type
(Nonconvex) inner lower bound	$f^{(k),\mathrm{L}}$	(ILB(k))	Global
(Convex) relaxed inner lower bound	$\breve{f}^{(k),\mathrm{L}}$	(RILB(k))	Local
(Nonconvex) inner upper bound	$f^{(k),\mathrm{U}}$	(IUB(k))	Global
(Nonconvex) relaxed inner upper bound	$\bar{f}^{(k)}$	(RIUB(k))	Global
(Convex) relaxed inner subproblem at given $\bar{\mathbf{x}}$ over node j	$\underline{w}^{(j)}(\bar{\mathbf{x}})$	(RISP($\bar{\mathbf{x}}, j$))	Local
(Nonconvex) inner subproblem at given $\bar{\mathbf{x}}$ over node j	$w^{(j)}(\bar{\mathbf{x}})$	(ISP($\bar{\mathbf{x}}, j$))	Global
(Nonconvex) inner subproblem at given $\bar{\mathbf{x}}$ over the whole of Y	$w(\bar{\mathbf{x}})$	(ISP($\bar{\mathbf{x}}, Y$))	Global
(Nonconvex) outer lower bound	$\underline{F}^{(k)}$	(LB(k))	Global
(Nonconvex) outer upper bound at given $\bar{\mathbf{x}}$ over node k'	$\bar{F}^{(k')}(\bar{\mathbf{x}})$	(UB($\bar{\mathbf{x}}, k'$))	Global or Local
(Nonconvex) outer upper bound at given $\bar{\mathbf{x}}$ over the whole of Y	$\bar{F}(\bar{\mathbf{x}})$	(UB($\bar{\mathbf{x}}, Y$))	Global or Local

† As defined in the first part of this work [69].

2.2. Pseudo codes of the full BASBL implementation. BASBL starts with the execution of the INITIALIZE() procedure described in Algorithm 2.1. The MINOTAUR AMPL interface function READINSTANCE() is used to read a "*.nl" file. BASBL options should be provided either using command line arguments or an option file (*basbl.opt*) containing (at least) the main solver options: outer (ε_F) and inner (ε_f) optimality tolerances; inner and outer bounding schemes; node selection rule ($NodeSel$); branching variable selection strategy ($BrVarStra$); the best inner upper bound strategy ($BIUBStra$); stopping criteria, e.g., maximum no. of iterations ($MaxIter$) and maximum allowed CPU time in sec. ($MaxTime$). The complete list of input options to the BASBL solver will be presented in Subsection 4.2.

Algorithm 2.1 BASBL: initialization

Input: AMPL input file (*problem.nl*) and BASBL option file (*basbl.opt*) files
Output: **terminate** BASBL if error occurred

```
procedure INITIALIZE(problem.nl, basbl.opt)
    STARTTIMER(time)                          ▷ Initialize global timer
    READOPTIONS(basbl.opt)                    ▷ Read BASBL options from basbl.opt file
    READINSTANCE(problem.nl)                  ▷ Read problem from AMPL problem.nl file
    CREATESUBPROBLEMS()                       ▷ Create ILB,IUB,LB,UB subproblems
    L ← ∅, L_In ← ∅                           ▷ Initialize lists
    iter ← 0, p ← 1, k ← 1                    ▷ Set iteration, X-partition (p) and node (k) counters
    F^UB ← ∞, f^UB ← ∞, (x^UB, y^UB) ← ∅      ▷ Set the incumbent
end procedure
```

If the initialization procedure succeeds, BASBL then executes procedure PROCESSROOTNODE() for a root node $k = 1$, as described in Algorithm 2.2. If BASBL is not terminated at the root node, it continues to execute the main `while()` loop, described in steps 3–31 in Algorithm 2.3. On termination, if the incumbent value $F^{\mathrm{UB}} = \infty$, then the bilevel instance is infeasible. Otherwise, ε_F-optimal objective values $F^{\mathrm{UB}}, f^{\mathrm{UB}}$ and ε-optimal solution $(\mathbf{x}^{\mathrm{UB}}, \mathbf{y}^{\mathrm{UB}})$ are returned.

Algorithm 2.2 `BASBL`: root node processing

Input: Node $k = 1$
Output: Lists $\mathcal{L}, \mathcal{L}^p$

procedure ProcessRootNode(k)
  $f^{(k)} \leftarrow$ SolveILB(k) ▷ Solve (ILB(k)), see Sec. 3.1 in [69]
  if FullyFathom(k) == true **then** ▷ see Definition 2.9 in [69]
    return Infeasible problem and **terminate** `BASBL`
  end if
  $\bar{f}^{(k)} \leftarrow$ SolveIUB(k) ▷ Solve (IUB(k)), see Sec. 3.2 in [69]
  if FullyFathom(k) == true **then**
    return Infeasible problem and **terminate** `BASBL`
  else
    $f^{\mathrm{UB},p} \leftarrow \bar{f}^{(k)}$
  end if
  $[\underline{F}^{(k)}, (\bar{\mathbf{x}}^{(k)}, \bar{\mathbf{y}}^{(k)})] \leftarrow$ SolveLB(k) ▷ Solve (LB(k)), see Sec. 3.4 in [69]
  if OuterFathom(k, ε_F) == true **then** ▷ see Definition 2.10 in [69]
    return Infeasible problem and **terminate** `BASBL`
  end if
  $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}^{(k)}$
  $w(\bar{\mathbf{x}}) \leftarrow$ SolveISP($\bar{\mathbf{x}}$) ▷ Solve (ISP($\bar{\mathbf{x}}$)), see Sec. 3.5 in [69]
  $[\bar{F}(\bar{\mathbf{x}}), (\bar{\mathbf{x}}, \bar{\mathbf{y}})] \leftarrow$ SolveUB($\bar{\mathbf{x}}$) ▷ Solve (UB($\bar{\mathbf{x}}$)), see Sec. 3.5 in [69]
  if $\bar{F}(\bar{\mathbf{x}}) < \infty$ **then**
    $F^{\mathrm{UB}} \leftarrow \bar{F}(\bar{\mathbf{x}}), f^{\mathrm{UB}} \leftarrow f(\bar{\mathbf{x}}, \bar{\mathbf{y}}), (\mathbf{x}^{\mathrm{UB}}, \mathbf{y}^{\mathrm{UB}}) \leftarrow (\bar{\mathbf{x}}, \bar{\mathbf{y}})$
    if OuterFathom(k, ε_F) == true **then**
      return $F^{\mathrm{UB}}, f^{\mathrm{UB}}, (\mathbf{x}^{\mathrm{UB}}, \mathbf{y}^{\mathrm{UB}})$
      terminate `BASBL`
    end if
  end if
  $\mathcal{L} \leftarrow \{k\}, \mathcal{L}^p \leftarrow \{k\}$ ▷ Add root node to lists
end procedure

3. BASBLib - A Library of Bilevel Test Problems.

The literature on the application of bilevel programming problems is extensive and diverse (see e.g., [10, 28, 31, 78] and references therein), and there have been initial efforts to establish a systematic test library for the evaluation of the bilevel algorithms and their implementations. While there exist generators of bilevel test problems [14, 15, 16], they are limited to linear and quadratic problems. There already exist several collections of bilevel test problems, focused on special subclasses:

- Chapter 9 in [39] contains linear and quadratic problems (**19 problems in total**)
- The `GAMS EMP` Library [33] contains mainly linear and quadratic problems (**33 problems in total**)
- The test set included with `BIPA` [21] contains BPPs with a convex inner problem (**22 problems in total**)

Algorithm 2.3 Full pseudo-code of the `BASBL` implementation

Input: `AMPL` *problem.nl* file and `BASBL` option (*basbl.opt*) file. Otherwise, default `BASBL` values (see Table 4.3) are used.
Output: Best solution found $(\mathbf{x}^{\text{UB}}, \mathbf{y}^{\text{UB}})$; outer (F^{UB}) and inner (f^{UB}) objectives.

Initialize($problem.nl$) ▷ see Algorithm 2.1
ProcessRootNode(k) ▷ see Algorithm 2.2
while $\mathcal{L} \neq \emptyset$ **and** $iter \leq MaxIter$ **and** $time \leq MaxTime$ **do**
  $k \leftarrow$ SelectNode($NodeSel$) ▷ see Definition 2.8 in [69]
  $v_{\text{br}} \leftarrow$ SelectBranchingVariable($k, BrVarStra$) ▷ see Definition 2.5 in [69]
  $K \leftarrow \{k_1, k_2\} \leftarrow$ BranchNode(k, v_{br}) ▷ see Definition 2.7, Steps 1-2 in [69]
  $[\mathcal{L}, \mathcal{L}_{\text{In}}, \mathcal{L}^p] \leftarrow$ UpdateLists(k, v_{br}, K) ▷ see Definition 2.7, Step 3 in [69]
  for all $k \in K$ **do**
    $\underline{f}^{(k)} \leftarrow$ SolveILB(k)
    CheckFathoming(k) ▷ see Algorithm 2.4
    if $s^{(k)} \neq inactive$ **then** ▷ $s^{(k)}$ - state of node k: *active, inner-active, inactive*
      $\bar{f}^{(k)} \leftarrow$ SolveIUB(k)
      UpdateBIUB($BIUBStra$) ▷ see Algorithm 2.6
    end if
    if $s^{(k)} == active$ **then**
      $[\underline{F}^{(k)}, (\bar{\mathbf{x}}^{(k)}, \bar{\mathbf{y}}^{(k)})] \leftarrow$ SolveLB(k)
      CheckOuterFathoming(k, ε_F) ▷ see Algorithm 2.5
      $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}^{(k)}$
      $w(\bar{\mathbf{x}}) \leftarrow$ SolveISP($\bar{\mathbf{x}}$)
      $[\bar{F}(\bar{\mathbf{x}}), (\bar{\mathbf{x}}, \bar{\mathbf{y}})] \leftarrow$ SolveUB($\bar{\mathbf{x}}$)
      if $\bar{F}(\bar{\mathbf{x}}) < F^{\text{UB}}$ **then**
        $F^{\text{UB}} \leftarrow \bar{F}(\bar{\mathbf{x}}), f^{\text{UB}} \leftarrow f(\bar{\mathbf{x}}, \bar{\mathbf{y}}), (\mathbf{x}^{\text{UB}}, \mathbf{y}^{\text{UB}}) \leftarrow (\bar{\mathbf{x}}, \bar{\mathbf{y}})$
        for all nodes $j \in \mathcal{L}^p : k \in \mathcal{L}^p$ **do**
          CheckOuterFathoming(j, ε_F)
        end for
      end if
    end if
  end for
  $iter \leftarrow iter + 1$ ▷ Increase the iteration counter
  UpdateTimer($time$) ▷ Update global timer
end while

Algorithm 2.4 `BASBL`: check fathoming rule

Input: Node k
Output: Updated lists $\mathcal{L}, \mathcal{L}_{\text{In}}, K$

procedure CheckFathoming(k)
  if FullyFathom(k) == true **then** ▷ see Definition 2.9 in [69]
    DeleteNode(k) ▷ see Algorithm 2.7
    CheckListDeletion(p) ▷ see Definition 2.11 in [69]
    UpdateBIUB($BIUBStra$)
  end if
end procedure

Algorithm 2.5 `BASBL`: check outer fathoming rule

Input: Node k
Output: Updated lists $\mathcal{L}, \mathcal{L}_{\text{In}}, K$

procedure CheckOuterFathoming(k, ε_F)
  if OuterFathom(k, ε_F) == true **then** ▷ see Definition 2.10 in [69]
    MoveNode(k) ▷ see Algorithm 2.8
    CheckListDeletion(p)
    UpdateBIUB($BIUBStra$)
  end if
end procedure

Algorithm 2.6 `BASBL`: update the best inner upper bound

Input: $BIUBStra$
Output: Updated $f^{\text{UB},p}/f^{\text{UB},k}$

procedure UpdateBIUB($BIUBStra$)
  if $BIUBStra$ == `overList` **then** ▷ BIUB for a list $\mathcal{L}^p : k \in \mathcal{L}^p$
    Update $f^{\text{UB},p}$ ▷ see Sec. 3.3.1 in [69]
    for all nodes $j \in \mathcal{L}^p : k \in \mathcal{L}^p$ **do**
      CheckFathoming(j) ▷ see Algorithm 2.4
    end for
  else ▷ BIUB for a set of sublists $\mathcal{L}^p_s : k \in \mathcal{L}^p_s$
    Update $f^{\text{UB},k}$ ▷ see Sec. 3.3.2 in [69]
    for all nodes $j \in \mathcal{L}^p_s : k \in \mathcal{L}^p_s$ **do**
      CheckFathoming(j) ▷ see Algorithm 2.4
    end for
  end if
end procedure

Algorithm 2.7 `BASBL`: delete node

Input: Node k
Output: Updated lists $\mathcal{L}, \mathcal{L}_{\text{In}}, K$

procedure DeleteNode(k)
  if $s^{(k)} == \textit{active}$ **then** ▷ Check state of node k
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{k\}$ ▷ Delete node k from $\mathcal{L}$
  else ▷ node k is *inner-active*
    $\mathcal{L}_{\text{In}} \leftarrow \mathcal{L}_{\text{In}} \setminus \{k\}$ ▷ Delete node k from $\mathcal{L}_{\text{In}}$
  end if
  $K \leftarrow K \setminus \{k\}$ ▷ Delete node k from K
end procedure

Algorithm 2.8 `BASBL`: move node from $\mathcal{L}$ to $\mathcal{L}_{\text{In}}$

Input: Node k
Output: Updated lists $\mathcal{L}, \mathcal{L}_{\text{In}}, K$

procedure MoveNode(k)
  $\mathcal{L} \leftarrow \mathcal{L} \setminus \{k\}$ ▷ Delete node k from $\mathcal{L}$
  $\mathcal{L}_{\text{In}} \leftarrow \mathcal{L}_{\text{In}} \bigcup \{k\}$ ▷ Add node k to $\mathcal{L}_{\text{In}}$
  $K \leftarrow K \setminus \{k\}$ ▷ Delete node k from K
end procedure

- A test set for bilevel problems [64] containing either nonconvex inner problems or problems with a structure that causes convergence issues for algorithms (**36 problems in total**)
- `MIPLIB` introduced in [35] and based on `MILPLIB 3.0` [13], containing bilevel problems with only binary variables (**57 problems in total**)
- Bilevel optimization problem library, version 0.1 [73] containing binary bilevel problems (**315 problems in total**)

Thus, the goal of this section is to present an actively growing online collection of general bilevel test problems, `BASBLib` [70] gathered from various sources and devoted to bilevel programming. The library is designed as an open resource to which other researchers in the bilevel programming community can easily contribute.

3.1. Classification of BPP problems. Since bilevel programming involves two optimization problems (outer–inner), the proposed classification is based on the nature of these problems. Currently, we distinguish the following classes (types) of bilevel programming problems:

- *outer linear-inner linear* bilevel problems (LP-LP)
- *outer linear-inner quadratic* bilevel problems (LP-QP)
- *outer quadratic-inner quadratic* bilevel problems (QP-QP)
- *outer linear-inner nonlinear* bilevel problems (LP-NLP)
- *outer quadratic-inner nonlinear* bilevel problems (QP-NLP)
- *outer nonlinear-inner nonlinear* bilevel problems (NLP-NLP)

A concise summary of the problems in `BASBLib` v2.2 [71] and their properties is presented in Table 4.2. The first column denotes the problem type, the second contains the problem name and the third the original source (to the best of our knowledge) source. The fourth through ninth columns ($n, m, \#\mathbf{G}, \#\mathbf{H}, \#\mathbf{g}, \#\mathbf{h}$) contain the number of outer ($\mathbf{x}$) variables, inner ($\mathbf{y}$) variables, inequality ($\mathbf{G}$) and equality ($\mathbf{H}$) constraints in the outer problem, inequality ($\mathbf{g}$) and equality ($\mathbf{h}$) constraints in the inner problem (excluding box constraints), respectively. Finally, the last four columns contain the best known optimal solutions: the outer (F^*) and inner (f^*) objective function values and the set of optimal solutions ($\mathbf{x}^*$ and $\mathbf{y}^*$), respectively. Note that some problems do not contain any outer variables ($n = 0$), but have been included because they reveal issues with some previous approaches, as discussed in [66]. An in-depth description of `BASBLib` is provided in the online resource http://basblsolver.github.io/BASBLib/. It includes problem statements, a geometrical analysis of the problems, the best known solutions, comments on inaccuracies in the literature, sources where the problem was used, `AMPL` input files in the `BASBL` format, and finally instructions on how to use it and contribute to it.

3.2. Comments on the problems. In this section, we report some inaccuracies for problems reported in the literature. The most comprehensive and up to date source for all observed inaccuracies is http://basblsolver.github.io/BASBLib/.

In the original source [78] of problem `sib_1997_02` (shown in Eq. (3.4)) as well as in other well-known sources [10, 21], it is reported that the optimal solution occurs at $(x^*, y^*) = (4.0, 4.0)$ with $F^* = -12.0$ and $f^* = 4.0$. However, the solution obtained with `BASBL` (see Table 5.4) is $(x^*, y^*) = (2.0, 1.0)$ with $F^* = -2.0$ and $f^* = 1.0$. The solution in the literature $(x, y) = (4.0, 4.0)$ is in fact not a bilevel feasible point. When the leader (outer decision maker) selects $x = 4.0$, the follower (inner decision maker) responds with $y = 0.0$, as this is better for the follower, although it is worse for the leader. Moreover, the graphical analysis in Figure 16.1.1 in [78] between the geometrical interpretation of the problem and its formulation confirms there is an inconsistency.

$$
\begin{aligned}
\min_{x,y} \quad & x - 4y \\
\text{s.t.} \quad & y \in \arg\min_{y\in[0,10]} \quad y \\
& \qquad \text{s.t.} \qquad \begin{aligned} -x - y + 3 &\le 0 \\ -2x + y &\le 0 \\ 2x + y - 12 &\le 0 \\ -3x + 2y + 4 &\le 0 \end{aligned} \\
& x \in [0,10],\ y \in [0,10],
\end{aligned} \tag{3.4}
$$

Therefore, we introduce a variation of problem `sib_1997_02`, which we name `sib_1997_02v`. The only difference is in the fourth inner constraint, which is changed from $-3x+2y+4 \le 0$ to $3x-2y-4 \le 0$. The feasible region of problem `sib_1997_02v` coincides with the one shown in Figure 16.1.1 in [78] and the optimal solution of this problem is $(x^*, y^*) = (4.0, 4.0)$ with $F^* = -12.0$ and $f^* = 4.0$.

In the original source of problem `mb_2007_18` (`mb_1_1_11` in [64]):

$$
\begin{aligned}
\min_{x,y} \quad & -x^2 + y^2 \\
\text{s.t.} \quad & y \in \arg\min_{y\in[-1,1]} \quad xy^2 - \frac{y^4}{2} \\
& x \in [-1,1],\ y \in [-1,1]
\end{aligned} \tag{3.5}
$$

as well in [66], the reported optimal solution is achieved at $(x^*, y^*) = (0.5, 0.0)$ with $F^* = 0.25$ and $f^* = 0.0$. However, the solution obtained with `BASBL` is $(x^*, y^*) = (1.0, 0.0)$ with $F^* = -1.0$ and $f^* = 0.0$. When the leader selects $x = 1.0$, the follower responds with $y = 0.0$, and hence $(x^*, y^*) = (1.0, 0.0)$ is a bilevel feasible point.

Similarly to the situation for problem `sib_1997_02`, we introduce a variation of problem `mb_2007_18` named `mb_2007_18v`. The only difference is in the outer objective function, which is changed from $-x^2+y^2$ to x^2+y^2. The optimal solution for problem `mb_2007_18v` is $(x^*, y^*) = (0.5, 0.0)$ with $F^* = 0.25$ and $f^* = 0.0$, and this coincides with the solution of the problem `mb_1_1_11` reported in [64].

In the original source of problem `mb_2007_19` (`mb_1_1_12` in [64]), the reported optimal solution is "$(x^*, y^*) = (\frac{7-\sqrt{13}}{18}, -\sqrt{\frac{7-\sqrt{13}}{18}}) \approx (0.189, -0.768)$" with $F^* = -0.258$. However, the solution obtained with `BASBL` is $(x^*, y^*) \approx (0.189, 0.434)$ with $F^* = -0.258$ and $f^* = -0.018$. First, note, that $-\sqrt{\frac{7-\sqrt{13}}{18}} \approx -0.434$. Next, in Mitsos et al. [66],Table 2, the reported solution is $(\bar{x}, \bar{y}) \approx (0.1874, 0.438)$, which is very close to the solution found using `BASBL`, as well to the solution of the same problem tackled in Example 3.2 in [68]. Therefore, we adopt $(0.1874, 0.438)$ as the approximate global solution.

Similarly, we introduce problem `mb_2007_22v` - variation of problem `mb_2007_22` (`mb_1_1_15` in [64]). The only difference is in the outer objective function, which is changed from $(x-0.6)^2 + y^2$ to $(x+0.6)^2 + y^2$. After this modification, the optimal solution of problem `mb_2007_22v` is $(x^*, y^*) \approx (-0.555, 0.452)$ with objective values $F^* = 0.207$ and $f^* = -0.066$. These values are very close to the values reported for problem `mb_1_1_15` in [64].

In problem `mb_2007_24` (`mb_1_1_12` in [64]), we have changed the third constraint to $-2.5 + y_1^2 + y_2^2 + y_3^2 \le 0$ (it was $2.5 + y_1^2 + y_2^2 + y_3^2 \le 0$ in the original source).

The same correction was made in [68]. Furthermore, we note there are two global solutions, $(x^*, y^*) = (-1, -1, 1, \pm 1, -\sqrt{0.5})$ whereas only one was reported in [64].

4. Solving problems using BASBL.

4.1. Problem input via the AMPL interface.

The BASBL solver reads the bilevel problem as described using the AMPL (A Mathematical Programming Language) [41] file format (.mod). Any information on the bilevel problem needed for BASBL execution is obtained through AMPL's Solver interface Library (ASL) [42] from a generated file (*.nl). AMPL/ASL does not support the modeling of programs with a multi-level structure, therefore we have developed a naming convention to define the bilevel structure of the problem. We use the prefixes "outer_" and "inner_" to distinguish information corresponding to the outer/inner problem.

We provide an example of use of the BASBL solver to solve problem sib_1997_01 (originally introduced as Example 15.3.1 in [78]), the formulation of which is presented in Example 4.1:

Example 4.1 (bilevel problem sib_1997_01).

$$
\begin{aligned}
\min_{x,y} \quad & 16x^2 + 9y^2 \\
\text{s.t.} \quad & -4x + y \leq 0 \\
& y \in \underset{y \in [0,50]}{\arg\min} \quad (x + y - 20)^4 \\
& \qquad\qquad \text{s.t.} \qquad 4x + y - 50 \leq 0 \\
& x \in [0, 12.5], \; y \in [0, 50]
\end{aligned}
$$

Listing 1 provides an AMPL modeling example for the bilevel instance described in Example 4.1. We provide the AMPL file in [71], which is directly connected to the online resource http://basblsolver.github.io/BASBLib/.

Listing 1

Description of bilevel problem sib_1997_01 in AMPL language, with the specific format required for its solution with the BASBL solver

```
var x >= 0, <= 12.5;        # Outer variable bounds
var y >= 0, <= 50;          # Inner variable bounds
var l{1..3} >= 0, <= 200;   # KKT multipliers bounds

# Outer objective
minimize outer_obj: 16*x^2 + 9*y^2;

subject to
# Outer constraints:
    outer_con_1:        -4*x + y <= 0;
# Inner objective:
    inner_obj:          (x + y -20)^4 = 0;
# Inner constraints:
    inner_con_1:        4*x + y -50 <= 0;
# KKT conditions:
    stationarity_1:     3*(x + y -20)^3 + l[1] - l[2] + l[3] = 0;
    complementarity_1: l[1]*(4*x + y -50) = 0;
    complementarity_2: l[2]*y = 0;
    complementarity_3: l[3]*(y - 50) = 0;
```

Note that although the inner objective function (inner_obj) could be construed as an outer constraint, the prefix "outer_con_" is not required.

Table 4.2: Description of the bilevel test problems in BASBLib v.2.2: data and references. [*] in the "Source" refers to the current work. **#G** (**#g**) denotes the number of outer (inner) inequality constraints, **#H** (**#h**) denotes the number of outer (inner) equality constraints. Other symbols are as defined in the text.

Bilevel problem			**Problem properties**						**Best known global solution(s)**			
Type	Name	Source	n	m	#**G**	#**H**	#**g**	#**h**	F^*	f^*	$\mathbf{x}^*$	$\mathbf{y}^*$
LP-LP	mb_2007_01	[64]	0	1	0	0	0	0	1.000	−1.000	–	1.000
	mb_2007_02	[64]	0	1	1	0	0	0	Infeas.	Infeas.	–	–
	as_2013_01	[5]	1	1	0	0	2	0	0.000	0.000	0.000	0.000
	cw_1988_01	[19]	1	1	0	0	3	0	−37.000	14.000	19.000	14.000
	lh_1994_01	[53]	1	1	0	0	3	0	−16.000	4.000	4.000	4.000
	sib_1997_02	[78]	1	1	0	0	4	0	−2.000	1.000	2.000	1.000
	sib_1997_02v	[*]	1	1	0	0	4	0	−12.000	4.000	4.000	4.000
	b_1984_01	[7]	1	1	0	0	4	0	3.111	−6.667	0.889	2.222
	aw_1990_01	[6]	1	1	0	0	5	0	−49.000	17.000	16.000	11.000
	b_1991_01	[9]	1	2	0	0	3	0	−1.000	0.000	1.000	(0.000, 0.000)
	b_1991_01v	[*]	1	2	0	0	3	0	−2.000	−1.000	0.000	(0.000, 1.000)
	cw_1990_01	[20]	1	2	0	0	3	0	−13.000	−4.000	5.000	(4.000, 2.000)
	bf_1982_02	[11]	2	2	0	0	3	0	−3.250	−4.000	(2.000, 0.000)	(1.500, 0.000)
	bf_1982_01	[11]	2	3	0	0	3	0	−26.000	3.200	(0.000, 0.900)	(0.000, 0.600, 0.400)
	s_1989_01	[76]	2	3	1	0	3	0	−14.600	0.300	(0.000, 0.650)	(0.000, 0.300, 0.000)
	ct_1982_01	[18]	2	6	0	0	0	3	−29.200	3.200	(0.000, 0.900)	(0.0, 0.6, 0.4, 0.0, 0.0, 0.0)
LP-QP	mb_2006_01	[63]	0	1	0	0	0	0	−1.000	−1.000	–	−1.000
	mb_2007_04	[64]	0	1	0	0	0	0	1.000	−1.000	–	1.000
	mb_2007_03	[64]	0	1	0	0	1	0	−1.000	1.000	–	−1.000
	b_1991_02	[9]	1	2	0	0	1	0	2.000	12.000	(2.000)	(6.000, 0.000)
	as_1984_01	[4]	2	2	1	0	2	0	0.000	200.000	(0.000, 0.000)	(−10.000, −10.000)
									0.000	100.000	(0.000, 30.000)	(−10.000, 10.000)
QP-QP	b_1998_04	[10]	1	1	0	0	0	0	81.330	−0.336	10.020	0.820
	b_1998_05	[10]	1	1	0	0	0	0	1.000	0.000	1.000	0.000
	lmp_1987_01	[55]	1	1	0	0	0	0	0.000	0.000	1.000	0.000
	y_1996_02	[86]	1	1	0	0	0	0	1.500	−2.500	0.250	0.000
	d_1992_01	[26]	1	1	0	0	1	0	31.25	4.000	1.000	1.000
	d_2000_01	[27]	1	1	0	0	2	0	0.000	−0.250	0.500	−0.500
	cw_1990_02	[20]	1	1	0	0	3	0	5.000	4.000	1.000	3.000
	tmh_2007_01	[83]	1	1	0	0	3	0	22.500	−4.500	1.500	4.500
	b_1988_01	[8]	1	1	0	0	3	0	17.000	1.000	1.000	0.000
	sa_1981_01	[77]	1	1	1	0	1	0	100.000	0.000	10.000	10.000
	sc_1998_01	[74]	1	1	3	0	0	0	9.000	0.000	3.000	5.000
	b_1998_02	[10]	2	1	0	0	0	0	0.000	−0.900	(0.800, 0.200)	1.000
	b_1998_03	[10]	2	1	0	0	0	0	0.000	−0.320	(1.000, 0.400)	0.800
	b_1998_07	[10]	1	2	0	0	4	0	−1.410	7.620	1.890	(0.890, 0.000)
	d_1978_01	[25]	2	2	0	0	0	0	−1.000	0.000	(0.500, 0.500)	(0.500, 0.500)
	fl_1995_01	[34]	2	2	0	0	0	0	−2.250	0.000	(0.750, 0.750)	(0.750, 0.750)
	sa_1981_02	[77]	2	2	2	0	0	0	225.000	100.000	(20.000, 5.000)	(10.000, 5.000)
	b_1984_02	[7]	2	2	1	0	2	0	−12.687	−6.473	(0.0, 2.0)	(1.875, 0.9063)
	dd_2012_02	[30]	2	2	0	1	2	0	−1.000	4.000	(0.707, 0.707)	(0.000, 1.000)
	as_1981_01	[3]	4	4	1	0	4	0	−6600.000	54.000	(7.0, 3.0, 12.0, 18.0)	(0.0, 10.0, 30.0, 0.0)
LP-NLP	mb_2007_05	[64]	0	1	0	0	0	0	0.500	−1.000	–	0.500
	mb_2007_06	[64]	0	1	0	0	0	0	−1.000	−1.000	–	−1.000
	mb_2007_10	[64]	1	1	0	0	0	0	0.500	−0.100	[0.100, 1.000]	0.500
	mb_2007_11	[64]	1	1	0	0	0	0	−0.800	0.000	0.000	−0.800
	mb_2007_13	[64]	1	1	0	0	0	0	−1.000	0.000	0.000	1.000
	mb_2007_13v	[*]	1	1	0	0	0	0	−2.000	−1.500	−1.000	−1.000
	mb_2007_15	[64]	1	1	0	0	0	0	0.000	−0.830	−1.000	1.000
	mb_2007_16	[64]	1	1	0	0	0	0	−2.000	0.000	−1.000	0.000

Continued on next page

Table 4.2 Continued from previous page

Bilevel problem			Problem properties						Best known global solution(s)			
Type	Name	Source	n	m	$\#\mathbf{G}$	$\#\mathbf{H}$	$\#\mathbf{g}$	$\#\mathbf{h}$	F^*	f^*	$\mathbf{x}^*$	$\mathbf{y}^*$
									−2.000	0.000	−0.500	−1.000
	ka_2014_01	[52]	1	1	0	0	0	0	−1.000	0.000	0.000	1.000
	mb_2007_09	[64]	1	1	1	0	0	0	−1.000	−1.000	−1.000	−1.000
	nwj_2016_01	[68]	1	2	0	0	1	0	2.000	0.000	(2.000)	(0.000, 0.000)
	gf_2001_01	[44]	1	2	0	0	2	0	0.190	−7.250	0.194	(9.970, 10.000)
	cg_1999_01	[17]	2	3	0	0	3	0	−29.200	0.310	(0.000, 0.900)	(0.000, 0.600, 0.400)
QP-NLP	mb_2007_12	[64]	1	1	0	0	0	0	0.000	0.000	0.000	0.000
	mb_2007_14	[64]	1	1	0	0	0	0	0.250	−0.083	0.250	0.500
	mb_2007_17	[64]	1	1	0	0	0	0	0.188	−0.016	−0.250	0.500
									0.188	−0.016	−0.250	−0.500
	mb_2007_18	[64]	1	1	0	0	0	0	−1.000	0.000	1.000	0.000
	mb_2007_18v	[*]	1	1	0	0	0	0	0.250	0.000	0.500	0.000
	mb_2007_19	[64]	1	1	0	0	0	0	−0.258	−0.018	0.189	0.434
	mb_2007_20	[64]	1	1	0	0	0	0	0.313	−0.083	0.500	0.500
	mb_2007_21	[64]	1	1	0	0	0	0	0.210	−0.069	−0.555	0.455
	mb_2007_23	[64]	1	1	0	0	0	0	−1.755	0.009	0.211	1.799
	mb_2007_22	[64]	1	1	0	0	1	0	0.189	−0.042	0.635	−0.433
	mb_2007_22v	[*]	1	1	0	0	1	0	0.210	−0.069	−0.555	0.455
	dd_2012_01	[30]	1	1	0	0	1	0	1.000	0.000	0.000	0.000
	sib_1997_01	[78]	1	1	1	0	1	0	2250.000	194.800	11.250	5.000
	yz_2010_01	[85]	1	1	0	0	1	0	1.000	−2.000	1.000	1.000
	mb_2007_08	[64]	1	1	2	0	0	0	0.000	0.000	−0.567	0.000
	c_2002_02	[21]	1	1	0	0	3	0	17.000	2.000	1.000	0.000
	c_2002_04	[21]	1	1	1	0	1	0	88.754	−0.077	0.000	0.579
NLP-NLP	c_2002_01	[21]	1	1	1	0	1	0	227.691	0.000	6.082	4.487
	c_2002_03	[21]	1	1	1	0	1	0	2.000	24.018	4.000	0.000
	c_2002_05	[21]	1	2	0	0	2	0	2.750	0.548	1.941	(0.000, 1.211)
	fz_1998_01	[40]	1	2	0	0	2	0	1.000	−1.000	1.000	(0.000, 1.000)
	mb_2007_24	[64]	2	3	3	0	0	0	−2.350	−2.000	(−1.000, −1.000)	(1.000, −0.707)
									−2.350	−2.000	(−1.000, −1.000)	(−1.000, −0.707)
	nwj_2016_02	[68]	2	3	1	0	2	0	−1.710	−2.230	(−1.000, −1.000)	(1.110, 0.310, −0.820)
	nwj_2016_04	[68]	2	3	1	0	2	0	−2.000	−1.000	(1.000, 1.000)	(0.000, 0.000, 1.000)
	nwj_2016_03	[68]	4	4	3	0	2	0	−0.437	−1.190	(0.000, 0.000, −0.707, −0.707)	(0.618, 0.000, −0.558, −0.554)
	nwj_2016_05	[68]	4	4	3	0	2	0	−3.506	−0.834	(0.544, 0.468, 0.490, 0.495)	(−0.783, −0.501, −0.288, −0.184)
	ka_2014_02	[52]	5	5	3	0	1	0	−10.000	−3.100	(1.0, −1.0, −1.0, −1.0, −1.0)	(−1.0, −1.0, −1.0, −1.0, −1.0)

Concluded

4.2. Algorithmic options. In Table 4.3 we list all options implemented in the current version of BASBL. Algorithmic options should be specified either in the BASBL option (*basbl.opt*) file, which contains the default values for the main algorithmic options, or using command line arguments.

4.3. Command line interface. Interaction with the BASBL solver currently takes place using a command line interface (CLI). Upon execution, BASBL produced output (using the default log output value LogLevel = 2) is shown in Figure 4.1 (for Example 4.1 with default solver options). For each iteration, a one-line summary is printed, containing: the current iteration number (Iter), the incumbent solution found so far (Fopt, fopt), the outer optimality gap (Out.-Gap) which is equal to the difference between Fopt and the lowest outer lower bound in the current BASBL tree, the total number of solved subproblems including those solved at the current iteration, and the number of *active* (L) and *inner-active* (Lin) nodes in the BASBL tree. For all choices of Loglevel the BASBL status after termination is reported, together with the final objective values found (F* and f*) and the solution point (x*, y*). Finally, the time (wall-clock) spent on each type of BASBL subproblem and the total time are reported. BASBL also generates a file named BASBL-problem_name.dat, which contains a summary of the optimization results.

```
Command-line interface
2017-10-10            BASBL (BiLevel solver) v1.0            14:09:39
-----------------------------------------------------------------------
Outer separation gap & obj. values | # of solved subproblems | L sizes
-----------------------------------------------------------------------
Iter  Out.-Gap   Fopt     fopt      ILB  IUB   LB  ISP  UB    L  Lin
-----------------------------------------------------------------------
 0      0.00     2250.00  197.75    1    1     1   1    1     0   1
-----------------------------------------------------------------------
*** Problem name  : sib_1997_01
*** Solver status : Successful termination!
*** Solution      :
      Objectives  : F* = 2250.00
                    f* = 197.75
      Point       : x* = 11.250
                    y* = 5.000
*** Solution time (s) :
      ILB         : 0.05
      IUB         : 0.05
      LB          : 0.08
      ISP         : 0.03
      UB          : 0.03
                   -----
      Total       : 0.24
```

FIG. 4.1. *BASBL output screen using the default LogLevel = 2 value for problem sib_1997_01*

5. Numerical results. Test problems from the BASBLib library (listed in Table 4.2) are used to evaluate the performance of the BASBL solver. Where relevant, BASBL subproblems are solved to global optimality with BARON version 17.4.1 [75, 81] accessed using GAMS version 24.8.4 [23] through system calls. Absolute (GAMS option

Table 4.3
Summary of available options in the BASBL v1.0

Option Flag	**Description**	**Default value**
	Termination and output options	
`EpsF`	*Outer optimality tolerance* (ε_F)	10^{-3}
`Epsf`	*Inner optimality tolerance* (ε_f)	10^{-5}
`MaxIter`	*Maximum number of iterations*	1000
`MaxTime`	*Maximum allowed CPU time (sec.)*	10000
`LogLevel`	*Options to control log output* 0 : all log output is suppressed 1 : print only error messages and warnings, if any 2 : print main info after each iteration 3 : print all log output	`2`
	Tree management options	
`BrVarStra`	*Branching variable* (v_{br}) *selection rule:* `XY` : priority given to X variables `YX` : priority given to Y variables	`YX`
`NodeSel`	*The node selection rule:* `Fl-lf` : **Option** ($\underline{F}l$) - **Option** ($l\underline{f}$) `Fl-lff` : **Option** ($\underline{F}l$) - **Option** ($l\bar{f}$) `lF-lf` : **Option** ($l\underline{F}$) - **Option** ($l\underline{f}$) `lF-lff` : **Option** ($l\underline{F}$) - **Option** ($l\bar{f}$)	`Fl-lf`
	Bounding schemes options	
`ILB`	*Inner Lower Bounding scheme:* `strict` : $f^{(k),\mathrm{L}}$ (ILB(k)) `relaxed` : $\breve{f}^{(k),\mathrm{L}}$ (RILB(k))	`strict`
`IUB`	*Inner Upper Bounding scheme:* `relaxed` : $\bar{f}^{(k)}$ (RIUB(k))	`relaxed`
`BIUBStra`	*Best Inner Upper Bounding scheme:* `overList` : $f^{\mathrm{UB},p}$ (BIUB(p)) `overSublists` : $f^{\mathrm{UB},k}$ (BIUB(k))	`overSublists`
`ISP`	*Inner Subproblem schemes:* `relaxed` : $\underline{w}^{(j)}(\bar{\mathbf{x}})$ (RISP($\bar{\mathbf{x}}, j$)) `strict` : $w^{(j)}(\bar{\mathbf{x}})$ (ISP($\bar{\mathbf{x}}, j$)) `strictOverList` : $w(\bar{\mathbf{x}})$ (ISP($\bar{\mathbf{x}}, Y$))	`strictOverList`
`LB`	*Outer Lower Bounding scheme:* `strict` : $\underline{F}^{(k)}$ (LB(k))	`strict`
`UB`	*Outer Upper Bounding scheme:* `strict` : $\bar{F}^{(k')}(\bar{\mathbf{x}})$ (UB($\bar{\mathbf{x}}, k'$)) `strictOverList` : $\bar{F}(\bar{\mathbf{x}})$ (UB($\bar{\mathbf{x}}, Y$))	`strictOverList`
	Subsolver options	

`LocSolver`	*Specifies the local solver to be used:*
	`MINOS`
	`FilterSQP`
	`IPOPT`
		`MINOS`

`GlobSolver`	*Specifies the global solver to be used:*
	`BARON`
	`ANTIGONE`
	`alphaBB`
		`BARON`

OptCA) and relative (GAMS option OptCR) termination tolerances for BARON are set to 10^{-5}. The BASBL source code is compiled using the Clang compiler (version 3.8.0) with optimization level O3. All experiments are carried out on a computer with a 64-bit Intel(R) Xeon(R) CPU X5675 @3.07GHz processor running Red Hat Linux.

In Table 5.4 we present results using the BASBL solver with default options (see Table 4.3). In the second column (Name) we report the name of the problem, while in the third (F^{UB}) and fourth (f^{UB}) columns we report the outer and inner optimal objective values obtained with the BASBL solver. In the fifth column (Iter.) we show the total number of BASBL iterations at termination, while in the sixth column the number of outer upper bounding problems solved before the optimal solution was computed for the first time (N_{opt}) is reported. In the seventh column we report the total time (t^{B}) (in seconds of wall-clock time) spent in BASBL execution. To benchmark BASBL, we use the precision-neutral the C++11 standard library facilities defined in the *Chrono library* [47]. In our implementation we use steady_clock which guarantees that the time between clock ticks is constant. Thus, it is very well suited to measuring time intervals [24]. The implementation details of a particular clock depend on the compiler and the operating system. In our case steady_clock, compiled with Clang 3.8, has nanosecond precision. The eighth column shows the sum of times (in seconds of wall-clock time) reported by GAMS for all subproblems (t^{G}). These time measurements are based on the GAMS option etSolve, which combines the time to read and write files, the time to create the solution report, and the time taken by the actual solve. Next, in each pair of consecutive columns (starting with the ninth column and ending with the last one) the total number of subproblems solved (#) and the time spent in GAMS to solve BASBL subproblems (t^{G}) are reported for the types of subproblems indicated. Finally, for each class of bilevel test problems from the BASBLib library, we emphasize the average and median results in bold.

BASBL is able to find the optimal solution for all BASBLib problems. Since BASBL begins by solving subproblems of type (ILB) and outer/full fathoming is applied when appropriate, #ILB $\geq$ #IUB $\geq$ #LB $\geq$ #UB. The solution of the inner upper bounding problems (IUB) is the most expensive out of all BASBL subproblem types. Note that for all problems (including problems requiring a larger number of iterations) the optimal solution is found early (N_{opt} is typically small) and subsequent iterations are used to prove global optimality.

All bilevel problems from the first three classes (LP-LP, LP-QP and QP-QP) are solved at the root node. In each case, BASBL thus achieves convergence after solving at most five subproblems at the root node. BASBL convergence is based on the outer-fathoming of the root node, i.e., the root node is outer-fathomed as a consequence of the test: $\underline{F}^{(0)} \geq F^{\mathrm{UB}} - \varepsilon_F$ (see Definition 2.10 [69]). BASBL is able to solve problem mb_2007_02 without having to perform the upper bounding procedure, as a result of detecting an infeasible outer lower bounding problem $\underline{F}^{(k)} = \infty$.

The next three classes of problems (LP-NLP, QP-NLP and NLP-NLP) contain nonlinear inner problems. With the exception of three problems (mb_2007_13, mb_2007_18v, and ka_2014_02) BASBL encounters no significant numerical difficulties, but, on average, these problems require more BASBL iterations and longer execution times compared to the first three classes.

Note that there is a significant difference between the total time spent in BASBL ($t^{\mathrm{B}}(s)$) and the sum of times reported by GAMS ($t^{\mathrm{G}}(s)$). This is most likely due to following reasons. First, to run BARON, each BASBL subproblem is transformed into the GAMS format by creating an external *.gms file. Importantly, given the small size of many problems in BASBLib and the fact that the corresponding subproblems are fast to solve, much of total time in small problems is dominated by the overhead of setting up the problem. As a result, timings for such problems must be treated with caution. Second, running `GAMS` through system calls and reading results from the output (*.lst) file are slow operations. Therefore, the use of a native `MINOTAUR` global solver has great potential to reduce the observed bottlenecks, although the investigation of this aspect is beyond the scope of this work.

5.1. Investigation of the relaxed scheme. In the original B&S paper [51] computationally less expensive convex alternatives for the inner lower bounding and inner subproblems were introduced. While this has great potential to significantly reduce the complexity of these `BASBL` subproblems, algorithmic analysis in Sec. 5 of [69] revealed that this can increase the total number of `BASBL` iterations. Here we provide a further experimental investigation of the relaxed scheme and its influence on `BASBL` performance.

The bounding schemes in `BASBL` are controlled using the options presented in Table 4.3. When the **`relaxed`** option is used for the inner lower bounding problems (ILB) and the inner subproblem (ISP), appropriate options for the global solver are set automatically. For the `BARON` solver these options are shown in Figure 5.2. First, `BARON` is forced to terminate after preprocessing by setting the number of iterations (`MaxIter`) to 0. Next, the number of local searches by `BARON`'s preprocessor is limited to 0 by setting the `NumLoc` option to 0. Finally, to sample the search space for local minima without range reduction, we set the range reduction options (`PDo`, `TDo`, `MDo`, `LBTTDo`, and `OBTTDo`) to 0 (see [75] for more details).

baron.opt
```
* BARON solver options file
MaxIter 0
NumLoc 0
PDo 0
TDo 0
MDo 0
LBTTDo 0
OBTTDo 0
```

FIG. 5.2. *`BARON` options used for the relaxed `BASBL` bounding scheme*

Note that for all tested instances from `BASBLib`, neither of two formulations (**`strict`** or **`relaxed`**, see Table 4.3) of the ILB and ISP require a lot of time to achieve convergence, even using strict tolerances (Table 5.4). Therefore, the time differences between solving strict or relaxed subproblems are almost negligible. However, obtaining less tight bounding values quite often results in slower `BASBL` convergence overall. This situation is investigated in Example 5.1

EXAMPLE 5.1 (problem `b_1998_04`).

$$\begin{aligned} \min_{x,y} \quad & (x-1)^2 + (y-1)^2 \\ \text{s.t.} \quad & y \in \underset{y\in[-100,100]}{\arg\min} \quad 0.5y^2 + 500y - 50xy \\ & x \in [-100,100],\ y \in [-100,100] \end{aligned} \tag{5.6}$$

The two-dimensional problem `b_1998_04` belongs to the `QP-QP` class with variable bounds defining a wide range: $[-100,100]^2$. The solution of the strict inner lower

Table 5.4: BASBL performance solving test problems from BASBLib using default options. "T." indicates problem type. See text for further explanation of the headings.

		Solution found		Total				ILB		IUB		LB		ISP		UB	
T.	Name	$F^{\rm UB}$	$f^{\rm UB}$	Iter.	$N_{\rm opt}$	$t^{\rm B}(s)$	$t^{\rm G}(s)$	#	$t^{\rm G}(s)$	#	$t^{\rm G}(s)$	#	$t^{\rm G}(s)$	#	$t^{\rm G}(s)$	#	$t^{\rm G}(s)$
LP-LP	mb_2007_01	1.00	−1.00	0	1	0.19	0.11	1	0.02	1	0.02	1	0.02	1	0.02	1	0.02
	mb_2007_02	inf	inf	0	0	0.11	0.06	1	0.02	1	0.02	1	0.02	0	0.00	0	0.00
	as_2013_01	0.00	0.00	0	1	0.24	0.13	1	0.03	1	0.03	1	0.03	1	0.02	1	0.02
	cw_1988_01	−37.00	14.00	0	1	0.21	0.12	1	0.02	1	0.02	1	0.02	1	0.02	1	0.02
	lh_1994_01	−16.00	4.00	0	1	0.22	0.13	1	0.02	1	0.04	1	0.03	1	0.02	1	0.02
	sib_1997_02	−2.00	1.00	0	1	0.22	0.13	1	0.02	1	0.04	1	0.03	1	0.02	1	0.02
	sib_1997_02v	−12.00	4.00	0	1	0.23	0.14	1	0.02	1	0.04	1	0.04	1	0.02	1	0.02
	b_1984_01	3.11	−6.67	0	1	0.22	0.13	1	0.03	1	0.03	1	0.03	1	0.02	1	0.02
	aw_1990_01	−49.00	17.00	0	1	0.25	0.15	1	0.02	1	0.04	1	0.04	1	0.02	1	0.02
	b_1991_01	−1.00	0.00	0	1	0.22	0.12	1	0.02	1	0.03	1	0.02	1	0.02	1	0.02
	b_1991_01v	−2.00	−1.00	0	1	0.23	0.12	1	0.02	1	0.02	1	0.03	1	0.02	1	0.03
	cw_1990_01	−13.00	−4.00	0	1	0.22	0.12	1	0.02	1	0.03	1	0.03	1	0.02	1	0.02
	bf_1982_02	−3.25	−4.00	0	1	0.24	0.12	1	0.02	1	0.03	1	0.03	1	0.03	1	0.02
	bf_1982_01	−26.00	3.20	0	1	0.28	0.16	1	0.03	1	0.03	1	0.05	1	0.02	1	0.03
	s_1989_01	−14.60	0.30	0	1	0.29	0.17	1	0.02	1	0.05	1	0.05	1	0.02	1	0.02
	ct_1982_01	−29.20	3.20	0	1	0.32	0.17	1	0.02	1	0.05	1	0.05	1	0.02	1	0.02
			Average	**0.00**	**0.94**	**0.23**	**0.13**	**1.00**	**0.02**	**1.00**	**0.03**	**1.00**	**0.03**	**0.94**	**0.02**	**0.94**	**0.02**
			Median	**0.00**	**1.00**	**0.23**	**0.13**	**1.00**	**0.02**	**1.00**	**0.03**	**1.00**	**0.03**	**1.00**	**0.02**	**1.00**	**0.02**
LP-QP	mb_2006_01	−1.00	−1.00	0	1	0.21	0.12	1	0.03	1	0.03	1	0.02	1	0.02	1	0.02
	mb_2007_04	1.00	−1.00	1	1	0.38	0.22	3	0.07	3	0.07	2	0.04	1	0.02	1	0.02
	mb_2007_03	−1.00	1.00	0	1	0.29	0.20	1	0.03	1	0.05	1	0.06	1	0.04	1	0.02
	b_1991_02	2.00	12.00	0	1	0.35	0.21	1	0.05	1	0.06	1	0.04	1	0.03	1	0.04
	as_1984_01	−0.00	200.00	0	1	0.43	0.27	1	0.04	1	0.08	1	0.07	1	0.04	1	0.05
			Average	**0.20**	**1.00**	**0.33**	**0.20**	**1.40**	**0.04**	**1.40**	**0.06**	**1.20**	**0.05**	**1.00**	**0.03**	**1.00**	**0.03**
			Median	**0.00**	**1.00**	**0.35**	**0.21**	**1.00**	**0.04**	**1.00**	**0.06**	**1.00**	**0.04**	**1.00**	**0.03**	**1.00**	**0.02**
QP-QP	b_1998_04	81.33	−0.34	0	1	0.36	0.27	1	0.02	1	0.08	1	0.05	1	0.05	1	0.06
	b_1998_05	1.00	0.00	0	1	0.38	0.28	1	0.02	1	0.12	1	0.08	1	0.02	1	0.03
	lmp_1987_01	0.00	0.00	0	1	0.22	0.12	1	0.03	1	0.03	1	0.03	1	0.02	1	0.02
	y_1996_02	1.50	−2.50	0	1	0.21	0.12	1	0.03	1	0.02	1	0.02	1	0.02	1	0.02
	d_1992_01	31.25	4.00	0	1	0.31	0.21	1	0.03	1	0.03	1	0.10	1	0.03	1	0.03
	d_2000_01	0.00	−0.25	0	1	0.28	0.18	1	0.03	1	0.07	1	0.03	1	0.02	1	0.02
	cw_1990_02	5.00	4.00	0	1	0.25	0.16	1	0.03	1	0.03	1	0.06	1	0.03	1	0.02
	tmh_2007_01	22.50	−1.50	0	1	0.32	0.20	1	0.04	1	0.04	1	0.07	1	0.03	1	0.03
	b_1988_01	17.00	1.00	0	1	0.33	0.23	1	0.07	1	0.06	1	0.06	1	0.02	1	0.02
	sa_1981_01	100.00	0.00	0	1	0.23	0.15	1	0.02	1	0.04	1	0.03	1	0.02	1	0.02
	sc_1998_01	8.98	0.00	0	1	0.34	0.21	1	0.04	1	0.06	1	0.04	1	0.04	1	0.03
	b_1998_02	0.00	−0.90	0	1	0.54	0.43	1	0.07	1	0.25	1	0.03	1	0.07	1	0.02
	b_1998_03	0.00	−0.32	0	1	0.49	0.39	1	0.06	1	0.22	1	0.03	1	0.06	1	0.02
	b_1998_07	−1.41	7.62	0	1	0.38	0.25	1	0.06	1	0.08	1	0.06	1	0.02	1	0.03
	d_1978_01	−1.00	0.00	0	1	0.39	0.25	1	0.03	1	0.07	1	0.09	1	0.03	1	0.03
	fl_1995_01	−2.26	0.00	0	1	0.31	0.21	1	0.03	1	0.06	1	0.07	1	0.02	1	0.04
	sa_1981_02	224.94	100.00	0	1	0.32	0.20	1	0.03	1	0.06	1	0.04	1	0.03	1	0.04
	b_1984_02	−12.69	−6.47	0	1	0.48	0.35	1	0.07	1	0.08	1	0.07	1	0.07	1	0.07
	dd_2012_02	−1.99	8.95	0	1	0.68	0.55	1	0.02	1	0.40	1	0.04	1	0.06	1	0.03
	as_1981_01	−6600.00	57.73	0	1	0.58	0.38	1	0.04	1	0.08	1	0.16	1	0.07	1	0.03
			Average	**0.00**	**1.00**	**0.37**	**0.26**	**1.00**	**0.04**	**1.00**	**0.09**	**1.00**	**0.06**	**1.00**	**0.04**	**1.00**	**0.03**
			Median	**0.00**	**1.00**	**0.34**	**0.22**	**1.00**	**0.03**	**1.00**	**0.07**	**1.00**	**0.06**	**1.00**	**0.03**	**1.00**	**0.03**
	mb_2007_05	0.50	−1.00	1	1	0.61	0.44	3	0.11	3	0.15	2	0.10	1	0.04	1	0.05

Continued on next page

Table 5.4 Continued from previous page

T.	Name	Solution found F^{UB}	f^{UB}	Total Iter.	N_{opt}	$t^{\mathrm{B}}(s)$	$t^{\mathrm{G}}(s)$	ILB #	$t^{\mathrm{G}}(s)$	IUB #	$t^{\mathrm{G}}(s)$	LB #	$t^{\mathrm{G}}(s)$	ISP #	$t^{\mathrm{G}}(s)$	UB #	$t^{\mathrm{G}}(s)$
LP-NLP	mb_2007_06	−1.00	−1.00	0	1	0.19	0.12	1	0.03	1	0.02	1	0.03	1	0.02	1	0.02
	mb_2007_10	0.50	−0.10	1	1	0.67	0.48	3	0.12	3	0.14	2	0.11	1	0.06	1	0.05
	mb_2007_11	−0.80	0.00	0	1	0.27	0.18	1	0.04	1	0.07	1	0.02	1	0.02	1	0.02
	mb_2007_13	−1.02	−0.01	269	71	158.18	104.38	1007	32.44	967	58.43	316	8.90	93	2.30	93	2.30
	mb_2007_13v	−2.00	−1.50	0	1	0.25	0.15	1	0.02	1	0.05	1	0.03	1	0.02	1	0.03
	mb_2007_15	−0.00	−0.83	4	1	1.48	0.93	9	0.31	9	0.34	8	0.19	2	0.04	2	0.05
	mb_2007_16	−2.00	0.00	6	1	2.68	1.63	18	0.55	16	0.63	11	0.30	3	0.07	3	0.08
	ka_2014_01	−1.00	0.00	5	3	2.98	2.08	15	0.53	15	0.66	11	0.73	3	0.08	3	0.07
	mb_2007_09	−1.00	−1.00	0	1	0.20	0.11	1	0.03	1	0.02	1	0.02	1	0.02	1	0.02
	nwj_2016_01	2.00	0.00	0	1	0.23	0.13	1	0.02	1	0.04	1	0.02	1	0.02	1	0.02
	gf_2001_01	0.19	−7.25	0	1	0.26	0.15	1	0.03	1	0.03	1	0.04	1	0.02	1	0.02
	cg_1999_01	−29.20	0.31	0	1	0.45	0.28	1	0.05	1	0.08	1	0.07	1	0.04	1	0.04
			Average	**22.00**	**6.54**	**12.96**	**8.54**	**81.69**	**2.64**	**78.46**	**4.67**	**27.46**	**0.81**	**8.46**	**0.21**	**8.46**	**0.21**
			Median	**0.00**	**1.00**	**0.45**	**0.28**	**1.00**	**0.05**	**1.00**	**0.08**	**1.00**	**0.07**	**1.00**	**0.04**	**1.00**	**0.04**
QP-NLP	mb_2007_12	0.00	0.00	5	2	2.41	1.67	11	0.43	11	0.53	11	0.34	4	0.19	4	0.17
	mb_2007_14	0.24	−0.08	6	4	2.28	1.44	13	0.36	12	0.38	12	0.45	4	0.11	4	0.13
	mb_2007_17	0.18	−0.02	5	3	2.32	1.58	11	0.41	11	0.53	11	0.34	4	0.16	4	0.15
	mb_2007_18	−1.00	0.00	2	2	0.88	0.55	5	0.13	5	0.19	5	0.13	2	0.04	2	0.06
	mb_2007_18v	0.25	0.00	52	3	23.21	12.68	184	5.38	156	4.23	83	2.14	12	0.47	12	0.46
	mb_2007_19	−0.26	−0.02	0	1	0.35	0.25	1	0.05	1	0.07	1	0.05	1	0.04	1	0.04
	mb_2007_20	0.31	−0.08	7	5	2.74	1.78	15	0.47	13	0.50	13	0.49	6	0.15	6	0.16
	mb_2007_21	0.21	−0.07	3	2	2.15	1.47	8	0.54	8	0.45	7	0.29	2	0.11	2	0.09
	mb_2007_23	−1.76	0.00	0	1	0.25	0.16	1	0.03	1	0.04	1	0.04	1	0.03	1	0.02
	mb_2007_22	0.19	−0.04	1	1	1.27	1.02	3	0.21	3	0.40	3	0.31	1	0.05	1	0.05
	mb_2007_22v	0.21	−0.07	0	1	0.49	0.39	1	0.08	1	0.12	1	0.09	1	0.06	1	0.04
	dd_2012_01	1.00	0.00	0	1	0.20	0.11	1	0.02	1	0.02	1	0.02	1	0.02	1	0.02
	sib_1997_01	2250.00	197.75	0	1	0.24	0.16	1	0.03	1	0.03	1	0.06	1	0.02	1	0.02
	yz_2010_01	1.00	−2.00	0	1	0.23	0.14	1	0.02	1	0.04	1	0.02	1	0.03	1	0.02
	mb_2007_08	0.00	0.00	0	1	0.22	0.14	1	0.03	1	0.04	1	0.02	1	0.02	1	0.02
	c_2002_02	17.00	2.00	0	1	0.27	0.18	1	0.06	1	0.04	1	0.04	1	0.02	1	0.02
	c_2002_04	88.75	−0.77	0	1	0.66	0.56	1	0.04	1	0.40	1	0.04	1	0.04	1	0.04
			Average	**4.76**	**1.82**	**2.42**	**1.43**	**15.24**	**0.49**	**13.41**	**0.47**	**9.06**	**0.29**	**2.59**	**0.09**	**2.59**	**0.09**
			Median	**0.00**	**1.00**	**0.66**	**0.55**	**1.00**	**0.08**	**1.00**	**0.19**	**1.00**	**0.09**	**1.00**	**0.04**	**1.00**	**0.04**
NLP-NLP	c_2002_01	227.69	0.00	0	1	0.31	0.20	1	0.04	1	0.06	1	0.05	1	0.03	1	0.02
	c_2002_03	2.00	24.02	0	1	0.22	0.12	1	0.02	1	0.03	1	0.03	1	0.02	1	0.02
	c_2002_05	2.75	0.55	0	1	0.28	0.18	1	0.03	1	0.03	1	0.06	1	0.03	1	0.03
	fz_1998_01	1.00	−1.00	0	1	0.32	0.21	1	0.03	1	0.06	1	0.07	1	0.03	1	0.02
	mb_2007_24	−2.35	−2.00	0	1	1.03	0.87	1	0.05	1	0.62	1	0.14	1	0.03	1	0.03
	nwj_2016_02	−1.71	−2.23	0	1	1.36	1.17	1	0.12	1	0.20	1	0.62	1	0.14	1	0.09
	nwj_2016_04	−2.00	−1.00	0	1	0.61	0.45	1	0.11	1	0.16	1	0.06	1	0.06	1	0.05
	nwj_2016_03	−0.44	−1.19	0	1	1.96	1.77	1	0.10	1	0.59	1	0.99	1	0.05	1	0.04
	nwj_2016_05	−3.51	−0.83	0	1	3.13	2.92	1	0.08	1	1.96	1	0.73	1	0.08	1	0.08
	ka_2014_02	−10.00	−5.10	0	1	227.16	226.92	1	0.03	1	226.68	1	0.14	1	0.03	1	0.03
			Average	**0.00**	**1.00**	**26.62**	**26.40**	**1.00**	**0.06**	**1.00**	**25.96**	**1.00**	**0.29**	**1.00**	**0.05**	**1.00**	**0.04**
			Median	**0.00**	**1.00**	**0.82**	**0.66**	**1.00**	**0.05**	**1.00**	**0.18**	**1.00**	**0.11**	**1.00**	**0.03**	**1.00**	**0.04**

Concluded

bounding problem, ILB = −54500, is obtained in $t^{\mathrm{G}} = 0.057$ sec. and the solution of the strict inner subproblem, ISP = −0.336, is obtained after $t^{\mathrm{G}} = 0.072$ sec. These values guarantee `BASBL` convergence at the root node. However, the relaxed inner lower bound, RILB = −550000 obtained after $t^{\mathrm{G}} = 0.056$ sec. and the solution of the relaxed inner subproblem, RISP = −81.969 obtained after $t^{\mathrm{G}} = 0.056$ sec. are significantly looser. These values result in an infeasible outer upper bounding problem and final `BASBL` convergence is achieved after 3 `BASBL` iterations and 4.31 seconds of `BASBL` execution. This is much slower compared to the 0.74 sec. needed when using strict bounds. For some other test problems this has an even more negative effect.

5.2. Investigation of different branching variable and node selection rules. To test different branching and node selection variants (for more details, see Secs. 2.2.2. and 2.3 in [69]), we used all the test problems from `BASBLib` that were not solved at the root node (Iter. > 0): `mb_2007_04`, `mb_2007_05`, `mb_2007_10`, `mb_2007_13`, `mb_2007_15`, `mb_2007_16`, `ka_2014_01`, `mb_2007_12`, `mb_2007_14`, `mb_2007_17`, `mb_2007_18`, `mb_2007_18v`, `mb_2007_20`, `mb_2007_21`, `mb_2007_22`.

The experimental results are reported in Table 5.5. In total, all problems were solved using three combinations of `BASBL` tree management options (see Table 4.3). The combination of `BASBL` options used is listed in the first column, while the meaning of the remaining columns was already described in section 5. Note, that this comparison does not include node selection strategies with `lF`, as we find that the choice of `lF` or `lf` does not affect the performance of `BASBL` for the test problems used. `YX-Fl-lf` is the combination closest to the original B&S [52], which combines the highest-index branching variable strategy (`YX`) and the $(\underline{F}l) - (l\underline{f})$ node selection strategy. The other tested combinations are based on changing one of the options at a time. In comparing the different heuristics, we focus on the number of nodes explored and the `GAMS` time, t^G, rather than the total time, t^B, because the overhead associated with setting up and post-processing the `GAMS` files has a significant impact on t^B.

Comparing the different branching variable selection rules (`YX` vs. `XY`), we can observe that branching preferentially on $\mathbf{y}$ variables (`YX` strategy) reduces the total execution time by around 10% on average. However, this result is mainly influenced by the performance of `BASBL` on the most challenging problem, `mb_2007_13`. The smaller median value is obtained when branching strategy `XY` is used. In general, the `XY` strategy leads to a higher number of independent lists but a lower number of nodes inside these lists. Thanks to this, `BASBL` is able to fathom unpromising regions faster from the overall list of nodes. Also, by using the `XY` branching strategy, `BASBL` often locates the optimal solution of the bilevel problem for the first time after fewer nodes than other strategies (i.e., N_{opt} is smaller on average). However, this does not always lead to a faster `BASBL` convergence, for example for problem `mb_2007_13`. On the other hand, the `YX` strategy gives better performance in problems for which greater partitioning of the Y space leads to a faster improvement of the best inner upper bound value and full-fathoming of the nodes can be performed sooner (for example in problems `mb_2007_17` and `mb_2007_20`). Next, observe there is only a minor difference (up to around 5%) in choosing one node selection strategy over the other ($l\underline{f}$ vs $l\bar{f}$), therefore we do not include these results for the `XY`-based branching strategy, as they are almost identical to `XY-Fl-lf`.

Table 5.5: `BASBL` performance based on different variants of the branching variable and node selection rules on selected problems. See Table 4.3 for an explanation of the algorithmic options indicated in column "Opt"

		Total				ILB		IUB		LB		ISP		UB	
Opt.	Name	Iter.	N_{opt}	$t^{\text{B}}(s)$	$t^{\text{G}}(s)$	#	$t^{\text{G}}(s)$	#	$t^{\text{G}}(s)$	#	$t^{\text{G}}(s)$	#	$t^{\text{G}}(s)$	#	$t^{\text{G}}(s)$
`YX-F1-lf`	mb_2007_04	1	1	0.38	0.22	3	0.07	3	0.07	2	0.05	1	0.02	1	0.02
	mb_2007_05	1	1	0.61	0.44	3	0.11	3	0.15	2	0.10	1	0.04	1	0.05
	mb_2007_10	1	1	0.67	0.48	3	0.12	3	0.14	2	0.11	1	0.06	1	0.05
	mb_2007_13	269	56	158.18	104.38	1007	32.44	967	58.43	316	8.90	93	2.30	93	2.30
	mb_2007_15	4	1	1.48	0.93	9	0.31	9	0.34	8	0.19	2	0.04	2	0.05
	mb_2007_16	6	1	2.68	1.63	18	0.55	16	0.63	11	0.30	3	0.07	3	0.08
	ka_2014_01	5	3	2.98	2.08	15	0.53	15	0.66	11	0.73	3	0.08	3	0.07
	mb_2007_12	5	2	2.41	1.67	11	0.43	11	0.53	11	0.34	4	0.19	4	0.17
	mb_2007_14	6	4	2.28	1.44	13	0.36	12	0.38	12	0.45	4	0.11	4	0.13
	mb_2007_17	5	3	2.32	1.58	11	0.41	11	0.53	11	0.34	4	0.16	4	0.15
	mb_2007_18	2	2	0.88	0.55	5	0.13	5	0.19	5	0.13	2	0.04	2	0.06
	mb_2007_18v	52	3	23.21	12.68	184	5.38	156	4.23	83	2.14	12	0.47	12	0.46
	mb_2007_20	7	5	2.74	1.78	15	0.47	13	0.50	13	0.49	6	0.15	6	0.16
	mb_2007_21	3	2	2.15	1.47	8	0.54	8	0.45	7	0.29	2	0.11	2	0.09
	mb_2007_22	1	1	1.27	1.02	3	0.21	3	0.40	3	0.31	1	0.05	1	0.05
	Average	**24.53**	**5.73**	**13.62**	**8.82**	**87.20**	**2.80**	**82.33**	**4.51**	**33.13**	**0.99**	**9.27**	**0.26**	**9.27**	**0.26**
	Median	**5.00**	**2.00**	**2.28**	**1.47**	**11.00**	**0.41**	**11.00**	**0.45**	**11.00**	**0.31**	**3.00**	**0.08**	**3.00**	**0.08**
`YX-F1-lff`	mb_2007_04	1	1	0.39	0.23	3	0.07	3	0.07	2	0.04	1	0.02	1	0.03
	mb_2007_05	1	1	0.60	0.43	3	0.12	3	0.15	2	0.08	1	0.03	1	0.05
	mb_2007_10	1	1	0.69	0.49	3	0.13	3	0.14	2	0.12	1	0.05	1	0.05
	mb_2007_13	269	56	156.98	101.98	1008	31.21	968	57.62	316	8.73	93	2.21	93	2.22
	mb_2007_15	4	1	1.62	0.96	9	0.31	9	0.35	8	0.20	2	0.05	2	0.05
	mb_2007_16	5	1	2.31	1.43	15	0.45	14	0.57	10	0.27	3	0.06	3	0.07
	ka_2014_01	5	3	3.22	2.26	15	0.59	15	0.75	11	0.77	3	0.07	3	0.07
	mb_2007_12	5	2	2.52	1.71	11	0.44	11	0.58	11	0.35	4	0.16	4	0.18
	mb_2007_14	7	4	2.57	1.52	15	0.42	13	0.40	13	0.45	4	0.11	4	0.14
	mb_2007_17	5	3	2.50	1.69	11	0.45	11	0.56	11	0.37	4	0.16	4	0.14
	mb_2007_18	2	2	0.92	0.57	5	0.13	5	0.20	5	0.14	2	0.04	2	0.05
	mb_2007_18v	51	3	22.77	11.74	178	4.97	150	3.85	79	1.94	13	0.50	13	0.49
	mb_2007_20	7	5	3.04	1.89	15	0.49	13	0.52	13	0.51	6	0.17	6	0.18
	mb_2007_21	3	2	2.23	1.63	8	0.58	8	0.50	7	0.34	2	0.12	2	0.09
	mb_2007_22	1	1	1.56	1.18	3	0.25	3	0.42	3	0.39	1	0.06	1	0.05
	Average	**24.47**	**5.73**	**13.59**	**8.65**	**86.80**	**2.71**	**81.93**	**4.45**	**32.87**	**0.98**	**9.33**	**0.25**	**9.33**	**0.26**
	Median	**5.00**	**2.00**	**2.31**	**1.52**	**11.00**	**0.44**	**11.00**	**0.50**	**11.00**	**0.35**	**3.00**	**0.07**	**3.00**	**0.07**
`XY-F1-lf`	mb_2007_04	1	1	0.39	0.23	3	0.07	3	0.07	2	0.05	1	0.02	1	0.02
	mb_2007_05	1	1	0.60	0.43	3	0.12	3	0.15	2	0.08	1	0.04	1	0.05
	mb_2007_10	3	1	1.85	1.37	7	0.33	7	0.43	5	0.33	2	0.14	2	0.14
	mb_2007_13	363	41	174.72	105.78	1295	39.03	1242	50.90	402	10.57	111	2.54	111	2.73
	mb_2007_15	6	1	2.02	1.27	13	0.45	13	0.47	11	0.25	2	0.04	2	0.05
	mb_2007_16	4	1	1.48	0.90	10	0.27	10	0.37	9	0.21	1	0.03	1	0.03
	ka_2014_01	4	2	2.11	1.42	12	0.43	11	0.50	8	0.36	3	0.07	3	0.07
	mb_2007_12	8	2	3.41	2.71	17	0.69	17	0.81	17	0.79	5	0.21	5	0.21
	mb_2007_14	10	5	3.54	2.12	20	0.55	18	0.63	18	0.62	5	0.16	5	0.16
	mb_2007_17	8	2	3.36	2.22	17	0.67	17	0.76	17	0.50	3	0.15	3	0.13
	mb_2007_18	1	2	0.69	0.36	3	0.08	3	0.11	3	0.07	2	0.04	2	0.06
	mb_2007_18v	46	3	19.87	11.02	157	4.67	136	3.54	74	1.85	13	0.48	13	0.48
	mb_2007_20	11	5	4.48	2.88	23	0.79	23	0.86	23	0.85	7	0.18	7	0.20
	mb_2007_21	2	1	1.79	1.30	5	0.41	5	0.38	5	0.36	1	0.07	1	0.07
	mb_2007_22	2	1	2.10	1.68	5	0.44	4	0.57	4	0.40	2	0.14	2	0.13
	Average	**31.33**	**4.60**	**14.83**	**9.05**	**106.00**	**3.27**	**100.80**	**4.04**	**40.00**	**1.15**	**10.60**	**0.29**	**10.60**	**0.30**
	Median	**4.00**	**2.00**	**2.10**	**1.42**	**12.00**	**0.44**	**11.00**	**0.50**	**9.00**	**0.36**	**2.00**	**0.14**	**2.00**	**0.13**

5.3. Analysis of algorithmic progress for the most challenging problems. Here, we investigate the performance of `BASBL` for the two most difficult problems, `mb_2007_13` (see Example 5.2 and Figure 5.3) and `mb_2007_18v` (see Example 5.3 and Figure 5.5) problems.

EXAMPLE 5.2 (problem `mb_2007_13`).

$$
\begin{aligned}
&\min_{x,y} \quad x - y \\
&\text{s.t.} \quad y \in \operatorname*{arg\,min}_{y\in[-1,1]} \; \frac{xy^2}{2} - yx^3 \\
&\qquad\quad x \in [-1,1],\; y \in [-1,1]
\end{aligned}
\tag{5.7}
$$

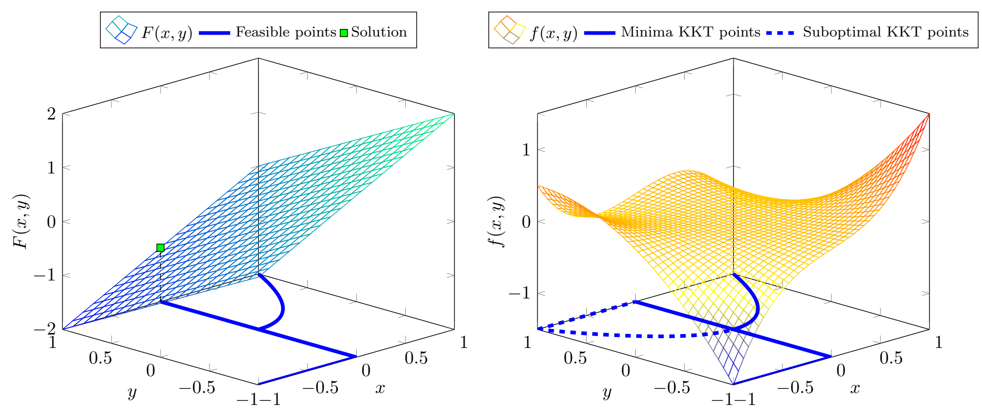


FIG. 5.3. *Graphical illustration of the outer and inner problems for problem* `mb_2007_13`*, together with their minima (solid lines) and suboptimal KKT points (dashed lines)*

The problem has the unique optimal solution $(x^*, y^*) = (0.0, 1.0)$ with the outer and inner optimal objective values $F^* = -1.0$ and $f^* = -1.0$. Stationarity of the inner objective gives $y - x^2 = 0$ and therefore for $-1 \leq x < 0$ the unique global optimum for the inner problem is $y = -1$, while for $x = 0$, all y values are trivially optimal for the inner problem. Finally, when $0 < x \leq 1$, the unique global minimum is $y = x^2$. In addition, for $-1 \leq x < 0$, suboptimal KKT points are $y = 1$ and $y = x^2$ (see Figure 5.3).

In Figure 5.4, from left to right, we illustrate the partitioned space after 2, 6 and 16 `BASBL` iterations. As before, white partitions denote (outer) active nodes, light-gray ones, nodes that are outer-fathomed, i.e., moved to $\mathcal{L}_{\text{In}}$, and dark-gray partitions those that are fully fathomed. From these partitions we can clearly see that `BASBL` convergence can be significantly influenced by suboptimal KKT points for the inner problem ($y = 1$ and $-1 \leq x < 0$ in this case) that cause slow tightening of outer lower bound values for nodes containing $y = 1$ (node $3 \in \mathcal{X}_1$, node $16 \in \mathcal{X}_3$, and nodes 52 and $59 \in \mathcal{X}_4$ in Figure 5.4) and consequently slow outer-fathoming of these nodes. Note, that this is not the case for problem `mb_2007_13v` (variant of `mb_2007_13`), where the outer-objective function changed from $x - y$ to $x + y$, keeping the rest of the problem unchanged. When solving this problem `BASBL` terminates at the root node.

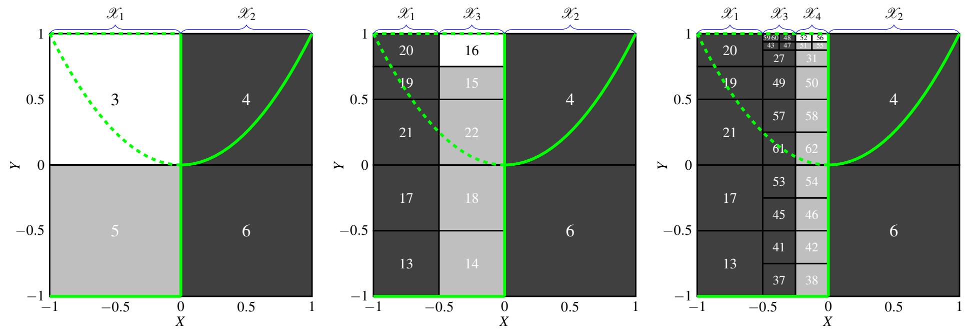


FIG. 5.4. *Illustration of the partition space after 2, 6 and 16* `BASBL` *iterations while solving* `mb_2007_13` *problem, together with their minima (solid lines) and suboptimal KKT points (dashed lines)*

Moreover, for $x = 0$, all $y \in [-1, 1]$ are optimal for the inner problem. This situation prevents any improvement of the BIUB value over the (refined) X-partitions containing $x = 0$. Therefore, the number of sublists and the number of nodes within such X-partitions increase, causing a slower shrinking of X-partitions and thus, a slow convergence of `BASBL`.

Finally, observe that the final obtained solution ($F^{\mathrm{UB}} = -1.016$) is slightly lower than the optimal solution reported in the literature $F^* = -1.0$. This is a consequence of the chosen inner tolerance $\varepsilon_f = 10^{-5}$, which leads to an approximate of the upper outer bounding problem (see Sec. 3.5 in [69]). The solution of the outer lower bounding problem over node $k = (220)$ ($(x, y) \in [-0.016, 0.000] \times [0.984, 1.000]$) is attained at the point $(\bar{x}, \bar{y}) = (-0.015625, 1.0)$. Next, the solution of the inner subproblem over the whole of Y at fixed $\bar{x} = -0.015625$ is equal to -0.007816315 and attained at point $(-0.015625, -1.0)$. Finally, the global solution of the outer upper bounding problem:

$$(5.8) \qquad \bar{F}^{(220)}(\bar{x} = -0.015625) = \begin{array}{cl} \min\limits_{y \in [-1,1]} & \bar{x} - y, \\ \text{s.t.} & \frac{\bar{x}y^2}{2} - y\bar{x}^3 \leq -0.007816315 + 0.00001, \end{array}$$

is equal to -1.015625, and is attained at $(-0.015625, 1.0)$, which is a suboptimal KKT point, but satisfies requirements of the ε-optimal bilevel solution (see Definition 1.1).

Note, that by using tighter inner tolerances (see Table 5.6) the reported solution becomes closer to -1.0. This highlights the importance played by the inner tolerance ε_f, which is responsible for the feasibility of the bilevel problem.

TABLE 5.6
`BASBL` *performance on problem* `mb_2007_13` *using different outer optimality tolerances*

Tol.	Solution found		Total								
ε_f	F^{UB}	f^{UB}	Iter.	N_{opt}	$t^{\mathrm{B}}(s)$	$t^{\mathrm{G}}(s)$	#ILB	#IUB	#LB	#ISP	#UB
10^{-5}	-1.016	-0.008	269	71	318.99	146.47	1007	967	316	93	93
10^{-6}	-1.010	-0.005	297	77	348.43	160.59	1123	1090	344	103	103
10^{-7}	-1.008	-0.004	301	79	349.84	160.79	1139	1108	347	104	104

EXAMPLE 5.3 (problem mb_2007_18v).

$$
\begin{aligned}
&\min_{x,y} \quad x^2 + y^2 \\
(5.9) \qquad &\text{s.t.} \quad y \in \operatorname*{arg\,min}_{y\in[-1,1]} \quad xy^2 - \frac{y^4}{2} \\
&\qquad\quad x \in [-1,1],\ y \in [-1,1]
\end{aligned}
$$

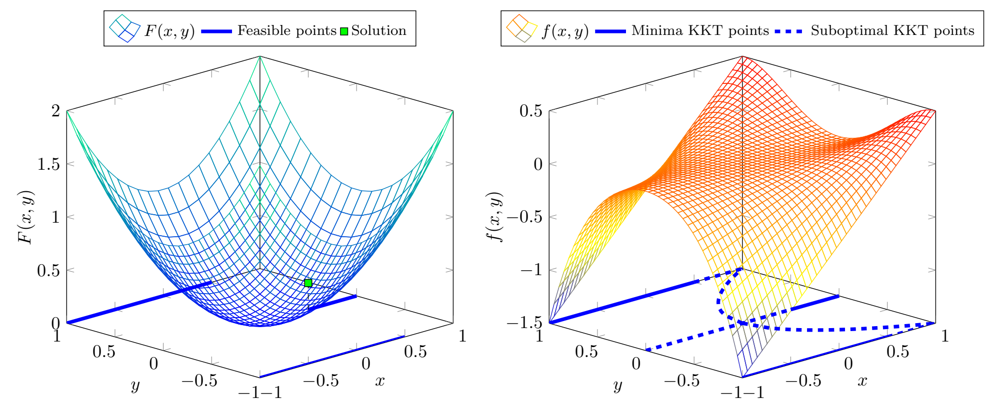


FIG. 5.5. *Graphical illustration of the outer and inner problems for instance mb_2007_18, together with their minima (solid lines) and suboptimal KKT points (dashed lines)*

The new problem has the unique optimal solution $(x^*, y^*) = (0.5, 0.0)$ with outer and inner optimal objective values $F^* = 0.25$ and $f^* = 0.0$ respectively. Stationarity of the inner objective gives $xy - y^3 = 0$ and therefore for $-1 \le x < 0.5$ the global minima for the inner problem are $y = \pm 1$. For $x = 0.5$, three global minima are $y = \pm 1$ and $y = 0$, while for $0.5 \le x \le 1.0$, the unique global minima is $y = 0$. In addition, for $-1 \le x < 0$, there are suboptimal KKT points at $y = 0$. For $0 \le x \le 0.5$, suboptimal KKT points are $y = 0$ and $y = \pm\sqrt{x}$, while for $0.5 < x \le 1.0$, suboptimal KKT points are $y = \pm\sqrt{x}$ and $y = \pm 1$ (see Figure 5.5).

Observe that the outer tolerance (ε_F) has a huge impact on the overall BASBL performance (see Table 5.7). Similarly to problem mb_2007_13, the convergence rate is influenced by suboptimal KKT points for the inner problem ($y = 0$ and $0 \le x < 0.5$ in this case) that cause slow tightening of outer lower bound values for nodes containing $y = 0$.

TABLE 5.7
BASBL performance on problem mb_2007_18v using different outer optimality tolerances

Tol.	Solution found		Total								
ε_F	F^{UB}	f^{UB}	Iter.	N_{opt}	$t^{\text{B}}(s)$	$t^{\text{G}}(s)$	#ILB	#IUB	#LB	#ISP	#UB
10^{-3}	0.25	0.0	52	3	23.21	12.68	184	156	83	12	12
10^{-2}	0.25	0.0	40	3	17.54	9.54	139	117	65	9	9
10^{-1}	0.25	0.0	28	3	11.80	6.55	94	78	47	6	6

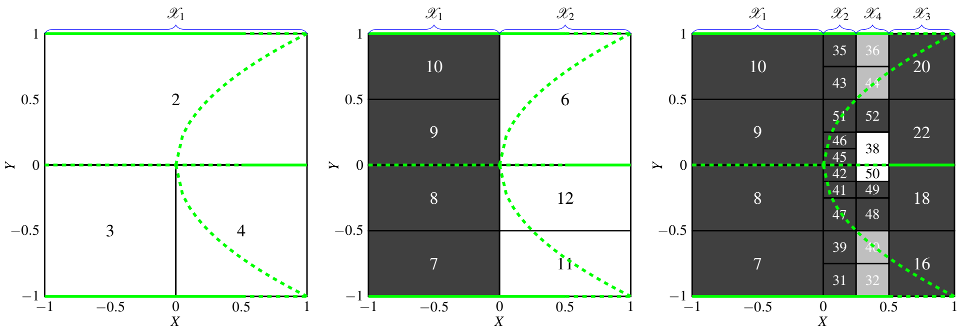


FIG. 5.6. *Illustration of the partition space after 2, 6 and 17 BASBL iterations while solving problem mb_2007_18, together with the minima (solid lines) and suboptimal KKT points (dashed lines)*

5.4. Comparison with other recent approaches. Here we comment on the performance of `BASBL` and other two recent approaches: approach of the Mitsos et al. [66] and that of Nie et al. [68]. Note that a direct comparison of performance is difficult, as the methods and their implementations are different. `BASBL` and the Mitsos et al. approaches were implemented in C++, and the subproblems were solved by `GAMS/BARON`. However, the Nie et al. approach was implemented in `MATLAB`, and the subproblems were solved by `GloptiPoly 3` [46] and `SeDuMi` [79]. Moreover, the environments and machines used are different, and a direct comparison based on execution time would be not very informative.

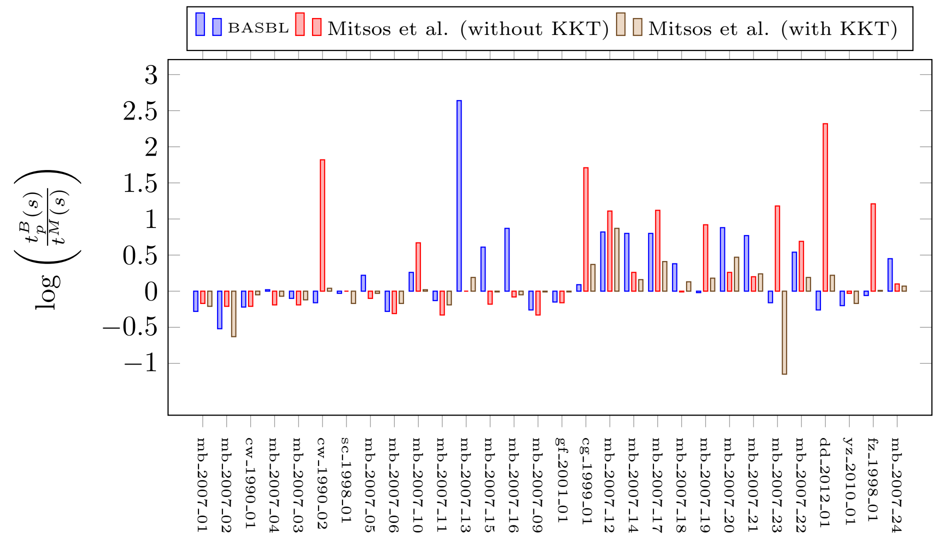
In such a situation, one way to compare algorithms is based on the number of subproblems solved. In [52] and [68], the authors compared their proposed algorithms with that of Mitsos et al. [66] based on such a criterion. In this work, for a given problem instance p, we use the following measure instead of the number of subproblems:

$$\log\left(\frac{t_p^B(s)}{t^M(s)}\right). \tag{5.10}$$

Here, t_p^B denotes the execution time, in seconds, spent solving bilevel problem p with a given implementation, and t^M is the median value of all t_p^B's for the set of bilevel problems considered using the same implementation. The median is used as it is less sensitive to outliers than an average time. This measure thus indicates the relative effort a certain algorithm requires to a given problem, relative to the median effort over all problems. Positive values of the measure from Eq. (5.10) represent "harder" problems, while negative values represent "simpler" problems for the specific algorithm.

We compare the performance of `BASBL` with two approaches from Mitsos et al. in Figure 5.7: with the KKT-based heuristic for the lower bound and without it. Note, that when the KKT heuristic is not used (shown as bars), the computational effort is much larger than when the KKT heuristic is applied (see bars). Note also that problem `mb_2007_13` requires much more computational resources when solved with `BASBL` than other problems. Finally, observe that with the exception of problems `mb_2007_13, mb_2007_15 and mb_2007_16`, the same problems appear to be challenging whether they are solved with `BASBL` or using the Mitsos et al. approach with KKT heuristic.

FIG. 5.7. *Graphical comparison of* `BASBL` *and two approaches from Mitsos et al. [66] using the measure shown in Eq.* (5.10)



Using the same approach, we compare the performance of `BASBL` with the Nie et al. approach in Figure 5.8. For the `BASBL` solver, problems `mb_2007_14`, `mb_2007_15`, `mb_2007_16` appear to be the most difficult to solve, while for the Nie et al. approach problems `mb_2007_24, nwj_2017_02, nwj_2017_03, nwj_2017_04, nwj_2017_05` are significantly harder relative to the median difficulty. Thus, the two different approaches seem to perform best on different problems.

5.5. Convergence issues with subsolvers. Choosing the best solver and options to solve subproblems is not trivial. For example, the performance of a local solver can be highly dependent on the starting point. Furthermore, a tighter tolerance is usually preferable to increase robustness, but even small changes can have a huge effect on the performance of the solver used. In this section we present two situations to illustrate these aspects.

In Example 5.4, a one-dimensional bilevel problem, `mb_2007_04`, is considered.

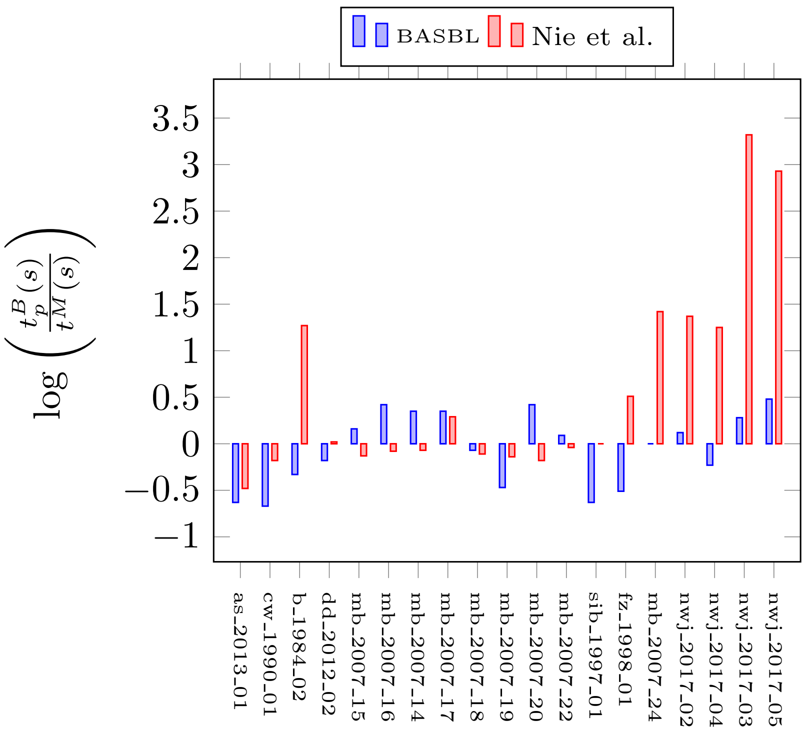
EXAMPLE 5.4 (`mb_2007_04` problem from [64]).

At the root node, the outer upper bounding problem is:

$$\bar{F}^{(1)} = \min_{y\in[-0.5,1]} \; y, \quad \text{s.t.} \quad -y^2 \le -1.0 + 0.00001, \tag{5.12}$$

where the solution of the inner subproblem $w^{(1)} = -1.0$ and an inner tolerance of $\varepsilon_f = 0.00001$ are used. Although $y = 1.0$ is a feasible point for Eq. (5.12), when solving this problem locally (without providing a starting point), `GAMS/MINOS` version 5.6 [67] reports that *"The problem is infeasible"*. Two other local solvers `GAMS/CONOPT` version 3.17C [32] and `GAMS/IPOPT` version 3.12 [84], are also unable to find a feasible point. Only when a starting point close to the minimum ($y = 1.0$) is provided, e.g., $y = 0.9$, `GAMS/MINOS` is able locating solution. The global solver `BARON`, on the other

FIG. 5.8. *Graphical comparison of* BASBL *and Nie et al. approach [68] using the measure shown in Eq.* (5.10)



$$
\begin{aligned}
&\min_{y} && y \\
&\text{s.t.} && y \in \arg\min_{y\in[-0.5,1.0]} \; -y^2 \\
& && y \in [-0.5, 1.0]
\end{aligned}
\tag{5.11}
$$

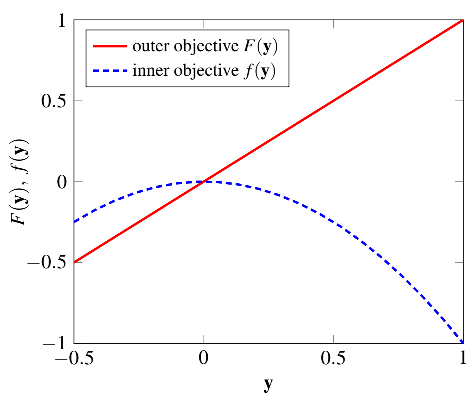


FIG. 5.9. *Graphical illustration of the inner and outer objective functions for* mb_2007_04

hand, locates this solution successfully without the need for a starting point.

As an example of the second situation, test problem mb_2007_12 is considered. The solution of the outer lower bounding problem over the node $[0, 0.5] \times [0.5, 1.0]$ is required. When the absolute termination tolerance is set to 10^{-6} in BARON, convergence is not achieved in over 1,000,000 iterations and 100 CPU seconds. However, when we set the absolute tolerance to a looser value of 10^{-5}, then BARON successfully terminates after 0.07 seconds and 0 iterations.

6. Conclusions. We presented a detailed description of the new BASBL solver for the global optimization of nonconvex/nonlinear bilevel problems. The BASBL solver is implemented in the open source MINOTAUR framework, including the use of strict or relaxed bounding problems, branching strategies and approaches to node selection. Furthermore, an online test library BASBLib, containing 81 problems, has been presented. This library has been designed to facilitate contributions from the bilevel optimization community so that it can be used to test future developments in the field.

The performance of BASBL has been investigated with detailed numerical study using the bilevel test problems in the first release of BASBLib. The results demonstrate the promising performance of BASBL. Attention has been paid to the experimental investigation of different algorithmic options. Based on this analysis, default choices for the various heuristics can be recommended as follows: the inner bounding problems should be solved using the "strict" formulations, i.e., as nonconvex problems whose global solution is sought; the YX branching strategy should be used, giving priority to branching on inner variables; the Fl-lff strategy should be used for node selection. While this combination of options gives the best average performance, as can be expected, it is not optimal for all problems. Regardless of the options used, all problems in the test library were solved successfully using BASBLib; these extensive results on small problems motivate further developments, along the possible directions discussed in the remainder of this section.

To improve performance, a quick win would be to explore ways to reduce the overheads involved in calling the global optimization solvers, through a C++ API (a beta version is available from GAMS v.24.9.1), which allows the seamless integration of GAMS into C++ applications.

It would also be interesting to test other branching heuristics. For instance, it may be beneficial to branch exclusively on the outer variables in the first few iterations of the algorithm to locate the most promising regions from the outer-level perspective faster. Once these regions are identified, priority could then be given to branching on the inner variables, keeping the number of sublists small and improving the best inner upper bound, which can accelerate full fathoming. Furthermore, it would be particularly interesting to incorporate domain reduction techniques [80] in BASBL.

Finally, the implementation of logical constraints could help to overcome the challenges of suboptimal KKT points.

To spur on further theoretical and algorithmic advances, the BASBLib resource could be expanded to include more challenging problems. In the current version, the largest problem contains only 10 variables and the most challenging problems are solved within about 175 seconds in the worst case, with most problems solved in a few seconds. The inclusion of larger problems derived from practical applications (e.g., [43, 48, 65]) would be very useful.

In order to facilitate the adoption of implementations of bilevel solvers such as BASBL, it would be beneficial to implement a universal way to conveniently model bilevel problems in AMPL. A possible way to do it is through a set of extensions to the AMPL modelling language using AMPL user functions and suffixes, as was done in [50] for modelling mixed-integer optimal control problems. Other modelling languages and/or environments such as Pyomo [45] could also be considered.

Acknowledgements. We gratefully acknowledge funding from the Leverhulme Trust through the Philip Leverhulme Prize and from the EPSRC through a Leadership Fellowship [EP/J003840/1].

Data access statement. Data underlying this article can be accessed on Zenodo at https://zenodo.org/record/897966, and used under the Creative Commons Attribution licence.

Appendix. Comment on convergence tolerance used in global NLP solvers. Note that one must be especially careful with the convergence tolerance, $\varepsilon_{\text{NLP}} > 0$, used for the global solution of the inner subproblems. Upon termination, global NLP solvers provide a lower bound (LB) on the optimal solution value and the best upper bound (BUB) at a feasible point $(\mathbf{x}^*_{\text{NLP}}, \mathbf{y}^*_{\text{NLP}})$ such that $LB \leq f(\mathbf{x}^*_{\text{NLP}}, \mathbf{y}^*_{\text{NLP}}) = BUB \leq LB + \varepsilon_{\text{NLP}}$. Since the global solution of the inner subproblems is needed for the formulation of subsequent subproblems, the quality of the approximation obtained can have a large impact on the progress of the algorithm. In particular, one can use $LB + \varepsilon_f$ or BUB in formulating subproblems. The former is more accurate in principle, but its validity depends on the relative values of ε_{NLP} and ε_f : one must choose $\varepsilon_{NLP} < \varepsilon_f$. The precision of the LB value returned by the global solver can also have an impact on convergence speed. In practice, we have found the more conservative value BUB to yield faster convergence, although this sometimes comes at a cost of reduced precision.

REFERENCES

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A global optimization method, αBB, for general twice-differentiable constrained NLPs–II. Implementation and computational results*, Computers & Chemical Engineering, 22 (1998), pp. 1159–1179, doi:10.1016/S0098-1354(98)00218-X.

[2] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, *A global optimization method, αBB, for general twice-differentiable constrained NLPs-I. Theoretical advances*, Computers & Chemical Engineering, 22 (1998), pp. 1137–1158, doi:10.1016/S0098-1354(98)00027-1.

[3] E. Aiyoshi and K. Shimizu, *Hierarchical decentralized systems and its new solution by a barrier method*, IEEE Transactions on Systems, Man and Cybernetics, (1981), pp. 444–449, doi:10.1109/TSMC.1981.4308712.

[4] E. Aiyoshi and K. Shimizu, *A solution method for the static constrained Stackelberg problem via penalty method*, IEEE Transactions on Automatic Control, 29 (1984), pp. 1111–1114, doi:10.1109/TAC.1984.1103455.

[5] G. B. Allende and G. Still, *Solving bilevel programs with the KKT-approach*, Mathematical Programming, 138 (2013), pp. 309–332, doi:10.1007/s10107-012-0535-x.

[6] G. Anandalingam and D. White, *A solution method for the linear static Stackelberg problem using penalty functions*, IEEE Transactions on Automatic Control, 35 (1990), pp. 1170–1173, doi:10.1109/9.58565.

[7] J. F. Bard, *Optimality conditions for the bilevel programming problem*, Naval Research Logistics Quarterly, 31 (1984), pp. 13–26, doi:10.1002/nav.3800310104.

[8] J. F. Bard, *Convex two-level optimization*, Mathematical Programming, 40 (1988), pp. 15–27, doi:10.1007/BF01580720.

[9] J. F. Bard, *Some properties of the bilevel programming problem*, Journal of Optimization Theory and Applications, 68 (1991), pp. 371–378, doi:10.1007/BF00941574.

[10] J. F. Bard, *Practical Bilevel Optimization*, vol. 30 of Nonconvex Optimization and Its Applications, Springer US, 1998, doi:10.1007/978-1-4757-2836-1.

[11] J. F. Bard and J. E. Falk, *An explicit solution to the multi-level programming problem*, Computers & Operations Research, 9 (1982), pp. 77–100, doi:10.1016/0305-0548(82)90007-7.

[12] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optimization Methods & Software, 24 (2009), pp. 597–634, doi:10.1080/10556780903087124.

[13] R. Bixby, S. Ceria, C. McZeal, and M. Savelsbergh, *An updated mixed integer programming library: Miplib 3.0*, 1996.
[14] P. H. Calamai and L. N. Vicente, *Generating linear and linear-quadratic bilevel programming problems*, SIAM Journal on Scientific Computing, 14 (1993), pp. 770–782, doi:10.1137/0914049.
[15] P. H. Calamai and L. N. Vicente, *Generating quadratic bilevel programming test problems*, ACM Transactions on Mathematical Software (TOMS), 20 (1994), pp. 103–119, doi:10.1145/174603.174411.
[16] P. H. Calamai, L. N. Vicente, and J. J. Júdice, *A new technique for generating quadratic programming test problems*, Mathematical Programming, 61 (1993), pp. 215–231, doi:10.1007/BF01582148.
[17] H. I. Calvete and C. Galé, *The bilevel linear/linear fractional programming problem*, European Journal of Operational Research, 114 (1999), pp. 188–197, doi:10.1016/S0377-2217(98)00078-2.
[18] W. Candler and R. Townsley, *A linear two-level programming problem*, Computers & Operations Research, 9 (1982), pp. 59–76, doi:10.1016/0305-0548(82)90006-5.
[19] P. Clark and A. Westerberg, *A note on the optimality conditions for the bilevel programming problem*, Naval Research Logistics (NRL), 35 (1988), pp. 413–418, doi:10.1002/1520-6750(198810)35:5¡413::AID-NAV3220350505¿3.0.CO;2-6.
[20] P. A. Clark and A. W. Westerberg, *Bilevel programming for steady-state chemical process design-I. fundamentals and algorithms*, Computers & Chemical Engineering, 14 (1990), pp. 87–97, doi:10.1016/0098-1354(90)87007-C.
[21] B. Colson, *BIPA(BIlevel Programming with Approximation Methods)(software guide and test problems)*, Cahiers du GERAD, (2002).
[22] B. Colson, P. Marcotte, and G. Savard, *A trust-region method for nonlinear bilevel programming: algorithm and computational experience*, Computational Optimization and Applications, 30 (2005), pp. 211–227, doi:10.1007/s10589-005-4612-4.
[23] G. D. Corporation, *General Algebraic Modeling System (GAMS) Release 24.8.4*. http://www.gams.com/, 2017.
[24] *C++ reference*. http://en.cppreference.com/w/cpp/chrono/steady_clock, 2017. [Online; accessed 24-February-2017].
[25] A. H. De Silva, *Sensitivity formulas for nonlinear factorable programming and their application to the solution of an implicitly defined optimization model of U. S. crude oil production*, Phd thesis, George Washington University, 1978.
[26] S. Dempe, *A necessary and a sufficient optimality condition for bilevel programming problems*, Optimization, 25 (1992), pp. 341–354, doi:10.1080/02331939208843831.
[27] S. Dempe, *A bundle algorithm applied to bilevel programming problems with non-unique lower level solutions*, Computational Optimization and Applications, 15 (2000), pp. 145–166, doi:10.1023/A:1008735010803.
[28] S. Dempe, *Foundations of Bilevel Programming*, vol. 61 of Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Boston, 2002, doi:10.1007/b101970.
[29] S. Dempe, *Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints*, Optimization, 52 (2003), pp. 333–359, doi:10.1080/0233193031000149894.
[30] S. Dempe and J. Dutta, *Is bilevel programming a special case of a mathematical program with complementarity constraints?*, Mathematical programming, 131 (2012), pp. 37–48, doi:10.1007/s10107-010-0342-1.
[31] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova, *Bilevel programming problems*, Energy Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, doi:10.1007/978-3-662-45827-3.
[32] A. S. Drud, *CONOPT–a large-scale GRG code*, ORSA Journal on Computing, 6 (1994), pp. 207–216.
[33] *EMPlib Library*. https://www.gams.com/emplib/libhtml/alfindx.htm. Accessed: 2016-11-17.
[34] J. E. Falk and J. Liu, *On bilevel programming, Part I: general nonlinear cases*, Mathematical Programming, 70 (1995), pp. 47–72, doi:10.1007/BF01585928.
[35] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl, *Intersection Cuts for Bilevel Optimization*, Springer International Publishing, 2016, pp. 77–88, doi:10.1007/978-3-319-33461-5_7.
[36] M. Fischetti, I. Ljubi, M. Monaci, and M. Sinnl, *Instances and solver software for mixed-integer bilevel linear problems*. https://msinnl.github.io/pages/bilevel.html. Accessed: 2017-09-12.
[37] M. Fischetti, I. Ljubi, M. Monaci, and M. Sinnl, *A new general-purpose algorithm for mixed-integer bilevel linear programs*, Operations Research, (2017),

doi:10.1287/opre.2017.1650.

[38] R. Fletcher, S. Leyffer, and P. L. Toint, *On the global convergence of a filter–SQP algorithm*, SIAM Journal on Optimization, 13 (2002), pp. 44–59, doi:10.1137/S105262340038081X.

[39] C. A. Floudas, P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Schweiger, *Handbook of test problems in local and global optimization*, Springer Science & Business Media, 1999, doi:10.1007/978-1-4757-3040-1.

[40] C. A. Floudas and S. Zlobec, *Optimality and duality in parametric convex lexicographic programming*, Springer US, Boston, MA, 1998, pp. 359–379, doi:10.1007/978-1-4613-0307-7_16.

[41] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, second ed., 2002, doi:10.1287/mnsc.36.5.519.

[42] D. M. Gay, *Hooking your solver to AMPL*, tech. report, Citeseer, 1997.

[43] I. E. Grossmann and L. T. Biegler, *Part II. Future perspective on optimization*, Computers & Chemical Engineering, 28 (2004), pp. 1193–1218, doi:10.1016/j.compchemeng.2003.11.006.

[44] Z. H. Gümüş and C. A. Floudas, *Global optimization of nonlinear bilevel programming problems*, Journal of Global Optimization, 20 (2001), pp. 1–31, doi:10.1023/A:1011268113791.

[45] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo–optimization modeling in python*, vol. 67, Springer Science & Business Media, 2012, doi:10.1007/978-1-4614-3226-5.

[46] D. Henrion, J.-B. Lasserre, and J. Löfberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optimization Methods and Software, 24 (2009), pp. 761–779, doi:10.1080/10556780802699201.

[47] H. E. Hinnant, W. E. Brown, J. Garland, and M. Paterno, *A Foundation to sleep on: clocks, points in time, and time durations*. http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2008/n2661.html, 2008. [Online; accessed 4-June-2016].

[48] R. S. Kamath, L. T. Biegler, and I. E. Grossmann, *Modeling multistream heat exchangers with and without phase changes for simultaneous optimization and heat integration*, AIChE Journal, 58 (2012), pp. 190–204, doi:10.1002/aic.12565.

[49] N. Kazazakis and C. S. Adjiman, *Arbitrarily tight αBB underestimators of sub-optimal domains for general non-linear functions*, Journal of Global Optimization, (2017). submitted.

[50] C. Kirches and S. Leyffer, *TACO: a toolkit for AMPL control optimization*, Mathematical Programming Computation, 5 (2013), pp. 227–265, doi:10.1007/s12532-013-0054-7.

[51] 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 (2014), pp. 425–458, doi:10.1007/s10898-013-0121-7.

[52] 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 (2014), pp. 459–481, doi:10.1007/s10898-013-0120-8.

[53] Y.-H. Liu and S. M. Hart, *Characterizing an optimal solution to the linear bilevel programming problem*, European Journal of Operational Research, 73 (1994), pp. 164–166, doi:10.1016/0377-2217(94)90155-4.

[54] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Computer Aided Control Systems Design, 2004 IEEE International Symposium on, IEEE, 2004, pp. 284–289.

[55] R. Lucchetti, F. Mignanego, and G. Pieri, *Existence theorems of equilibrium points in Stackelberg*, Optimization, 18 (1987), pp. 857–866, doi:10.1080/02331938708843300.

[56] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints*, Cambridge University Press, 1996.

[57] A. Mahajan, S. Leyffer, and C. Kirches, *Solving mixed-integer nonlinear programs by QP-diving*, Preprint ANL/MCS-2071-0312, Argonne National Laboratory, Mathematics and Computer Science Division, (2012).

[58] A. Mahajan, S. Leyffer, J. Linderoth, J. Luedtke, and T. Munson, *MINOTAUR: a toolkit for solving mixed-integer nonlinear optimization*. http://wiki.mcs.anl.gov/minotaur, 2011. [Online; accessed 4-June-2016].

[59] A. Mahajan, S. Leyffer, J. Linderoth, J. Luedtke, T. Munson, and C. S. Division, *Minotaur: A Mixed-Integer Nonlinear Optimization Toolkit*, 2017. preprint on webpage at http://www.optimization-online.org/DB_FILE/2017/10/6275.pdf.

[60] A. Mahajan and T. Munson, *Exploiting second-order cone structure for global optimization*,

Technical report ANL/MCS-P1801-1010, Argonne National Laboratory, (2010).

[61] J. A. Mirrlees, *The theory of moral hazard and unobservable behaviour: Part I*, The Review of Economic Studies, 66 (1999), pp. 3–21.

[62] R. Misener and C. A. Floudas, *ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations*, Journal of Global Optimization, 59 (2014), pp. 503–526, doi:10.1007/s10898-014-0166-2.

[63] A. Mitsos and P. I. Barton, *Issues in the development of global optimization algorithms for bilevel programs with a nonconvex inner program*, technical report, Massachusetts Institute of Technology, 2006.

[64] A. Mitsos and P. I. Barton, *A test set for bilevel programs*. http://www.researchgate.net/publication/228455291, 2007. [Last updated September 19, 2007].

[65] A. Mitsos, G. M. Bollas, and P. I. Barton, *Bilevel optimization formulation for parameter estimation in liquid-liquid phase equilibrium problems*, Chemical Engineering Science, 64 (2009), pp. 548–559, doi:10.1016/j.ces.2008.09.034.

[66] A. Mitsos, P. Lemonidis, and P. I. Barton, *Global solution of bilevel programs with a nonconvex inner program*, Journal of Global Optimization, 42 (2008), pp. 475–513, doi:10.1007/s10898-007-9260-z.

[67] B. A. Murtagh, P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *GAMS/MINOS 5.6, Large Scale Nonlinear Solver*, 2014.

[68] J. Nie, L. Wang, and J. J. Ye, *Bilevel Polynomial Programs and Semidefinite Relaxation Methods*, SIAM Journal on Optimization, 27 (2017), pp. 1728–1757, doi:10.1137/15M1052172.

[69] R. Paulavičius and C. S. Adjiman, *BASBL: Branch-And-Sandwich BiLevel solver. I. Algorithmic improvements and extensions*, SIAM Journal on Optimization, (2017). Submitted.

[70] R. Paulavičius and C. S. Adjiman, *BASBLib - a library of bilevel test problems*, 2017, doi:10.5281/zenodo.595940.

[71] R. Paulavičius and C. S. Adjiman, *BASBLib - a library of bilevel test problems, v2.2*, Sept. 2017, doi:10.5281/zenodo.897966.

[72] T. Ralphs, *Mixed Integer Bilevel Solver*. https://github.com/tkralphs/MibS. Accessed: 2017-09-12.

[73] T. Ralphs and E. Adams, *Bilevel instance library*. http://coral.ise.lehigh.edu/data-sets/bilevel-instances/, 2016. Accessed: 2017-09-12.

[74] K. H. Sahin and A. R. Ciric, *A dual temperature simulated annealing approach for solving bilevel programming problems*, Computers & Chemical Engineering, 23 (1998), pp. 11–25, doi:10.1016/S0098-1354(98)00267-1.

[75] N. V. Sahinidis, *BARON 17.4.1: global optimization of mixed-integer nonlinear programs,* User's Manual, 2017.

[76] G. Savard, *Contribution à la programmation mathématique à deux niveaux*, phd thesis, Ecole Polytechnique de Montréal, Université de Montréal, 1989.

[77] K. Shimizu and E. Aiyoshi, *A new computational method for Stackelberg and min-max problems by use of a penalty method*, IEEE Transactions on Automatic Control, 26 (1981), pp. 460–466, doi:10.1109/TAC.1981.1102607.

[78] K. Shimizu, Y. Ishizuka, and J. F. Bard, *Nondifferentiable and two-level mathematical programming*, vol. 102, Kluwer Academic Publishers, Boston, 1997, doi:10.1016/S0377-2217(97)00228-2.

[79] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization methods and software, 11 (1999), pp. 625–653, doi:10.1080/10556789908805766.

[80] M. Tawarmalani and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: theory, algorithms, software, and applications*, Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Boston, 2002, doi:10.1007/978-1-4757-3532-1.

[81] M. Tawarmalani and N. V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Mathematical Programming, 103 (2005), pp. 225–249, doi:10.1007/s10107-005-0581-8.

[82] A. Tsoukalas, B. Rustem, and E. N. Pistikopoulos, *A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems*, Journal of Global Optimization, 44 (2008), pp. 235–250, doi:10.1007/s10898-008-9321-y.

[83] H. Tuy, A. Migdalas, and N. Hoai-Phuong, *A novel approach to bilevel nonlinear programming*, Journal of Global Optimization, 38 (2007), pp. 527–554, doi:10.1007/s10898-006-9093-1.

[84] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, 106 (2006), pp. 25–57, doi:10.1007/s10107-004-0559-y.

[85] J. J. Ye and D. Zhu, *New necessary optimality conditions for bilevel programs by combining the MPEC and value function approaches*, SIAM Journal on Optimization, 20 (2010), pp. 1885–1905, doi:10.1137/080725088.

[86] A. Yezza, *First-order necessary optimality conditions for general bilevel programming problems*, Journal of Optimization Theory and Applications, 89 (1996), pp. 189–219, doi:10.1007/BF02192648.