J Glob Optim (2008) 42:475–513
DOI 10.1007/s10898-007-9260-z

# Global solution of bilevel programs with a nonconvex inner program

**Alexander Mitsos · Panayiotis Lemonidis · Paul I. Barton**

Received: 12 August 2007 / Accepted: 12 October 2007 / Published online: 5 December 2007


**Abstract** A bounding algorithm for the global solution of nonlinear bilevel programs involving nonconvex functions in both the inner and outer programs is presented. The algorithm is rigorous and terminates finitely to a point that satisfies $\varepsilon$-optimality in the inner and outer programs. For the lower bounding problem, a relaxed program, containing the constraints of the inner and outer programs augmented by a parametric upper bound to the parametric optimal solution function of the inner program, is solved to global optimality. The optional upper bounding problem is based on probing the solution obtained by the lower bounding procedure. For the case that the inner program satisfies a constraint qualification, an algorithmic heuristic for tighter lower bounds is presented based on the KKT necessary conditions of the inner program. The algorithm is extended to include branching, which is not required for convergence but has potential advantages. Two branching heuristics are described and analyzed. Convergence proofs are provided and numerical results for original test problems and for literature examples are presented.

**Keywords** Bilevel program · nonconvex · global optimization · branch-and-bound · MPEC

## 1 Introduction

Bilevel programs are programs where an *outer program* is constrained by an embedded *inner program*. Here, inequality constrained nonlinear bilevel programs of the form

A. Mitsos · P. Lemonidis · P. I. Barton (✉)
Department of Chemical Engineering, Massachusetts Institute of Technology, 66-464, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: pib@mit.edu

A. Mitsos
e-mail: amitsos@alum.mit.edu

P. Lemonidis
e-mail: plemonid@alum.mit.edu

$$
\begin{aligned}
f^* = \min_{\mathbf{x},\mathbf{y}} &\ f(\mathbf{x},\mathbf{y}) \\
\min \mathbf{g}(\mathbf{x},\mathbf{y}) &\leq \mathbf{0} \\
\mathbf{y} \in \arg\min_{\mathbf{z}} &\ h(\mathbf{x},\mathbf{z}) \\
\text{s.t. } &\mathbf{p}(\mathbf{x},\mathbf{z}) \leq \mathbf{0} \\
&\mathbf{q}(\mathbf{z}) \leq \mathbf{0} \\
\mathbf{x} \in X \subset \mathbb{R}^{n_x}, &\quad \mathbf{y},\mathbf{z} \in Y \subset \mathbb{R}^{n_y},
\end{aligned}
\tag{1}
$$

are considered without any convexity assumptions. The co-operative (or optimistic, weak) formulation [17] is assumed, where if for a given $\mathbf{x}$ the inner program has multiple optimal solutions $\mathbf{y}$, the outer optimizer can choose among them. In Appendix A the optimistic and pessimistic formulations are discussed in more detail. Throughout the article only real-valued solutions are considered and no methods to find all solutions are discussed. Dummy variables ($\mathbf{z}$ instead of $\mathbf{y}$) are used in the inner program since this clarifies some issues and facilitates discussion.

There are many applications of bilevel programs as well as theoretical and algorithmic contributions in the literature and the reader is directed to other publications for a review of applications and algorithms [4,10,17,18,39,45,53]. To our best knowledge, no valid algorithm has been proposed to solve bilevel programs to guaranteed global optimality when nonconvexity is present in the inner program. Bard [3] considered a simpler formulation without outer constraints and with a unique minimum for the inner problem, and proposed an algorithm based on a grid search between a lower and an upper bound of the optimal objective value, without a guarantee of convergence in the general case. Falk and Liu [21] proposed a bundle method which obtains local solutions to the inner and outer programs. Recently Tuy et al. [50] proposed an algorithm for bilevel programs satisfying a monotonicity assumption.

The following example illustrates some of the implications of nonconvexity in the inner program:

*Example 1.1* The bilevel program

$$
\begin{aligned}
&\min_{x,y} x + y \\
\min\ & y \in \arg\min_{z} x\tfrac{z^2}{2} - \tfrac{z^3}{3} \\
& x \in [-1,1], \quad y,z \in [-1,1]
\end{aligned}
$$

has the unique optimal solution $x = -1$, $y = 1$ with an objective value of 0. Stationarity of the inner objective gives $xy - y^2 = 0$ and therefore $y = 1$, $y = 0$ and $y = x$ are KKT points of the inner problem. Out of them $y = 1$ is optimal for $-1 \leq x \leq 2/3$ and $y = 0$ is optimal for $2/3 \leq x \leq 1$. Allowing for a local solution of the inner program, e.g., by replacing the inner program with its necessary but not sufficient KKT conditions leads to a relaxation of the bilevel program. The optimal solution of this relaxation is $\bar{x} = -1$, $\bar{y} = -1$ with an objective value of $-2$. This point however is not feasible in the original bilevel program because the constraint $y \in \arg\min_{z} x\frac{z^2}{2} - \frac{z^3}{3}$ is violated.

Algorithms that guarantee convergence to the global solution or stationary points have been proposed for related programs under nonconvexity, such as min-max programs [20,55], semi-infinite programs (SIP) [8,9,23], and generalized semi-infinite programs [34].

Here a bounding algorithm for the global solution of (1) is proposed allowing nonconvex functions in both the inner and outer programs as well as multiple, or even uncountably many, global minima in both the inner and outer programs. Equality constraints in the outer program would not change anything significant in the development of the algorithm and are only omitted for simplicity. The same holds for equality constraints in the inner program that do not depend on the outer variables $\mathbf{x}$. On the other hand, the presence of $\mathbf{x}$-dependent equality constraints in the inner program would require significant changes to the algorithm presented, because these constraints would violate Assumption 3 which is required for the convergence of the algorithm.

The algorithm is based on a collection of single-level optimization formulations. In the next section the assumptions necessary for finite termination of the algorithm are outlined. These assumptions also guarantee the existence of a minimum of (1). In Sect. 3 a lower bounding procedure is presented based on the solution of an optimization problem where the constraints of the inner and outer program are augmented by a parametric bound on the optimal solution value of the inner program as a function of the outer variables. In Sect. 4 an upper bounding procedure is presented, which is based on probing the solution obtained by the lower bounding procedure. In Sect. 5 the algorithmic framework is described along with a proof of finite convergence to an $\varepsilon$-optimal solution. The basic strategy of the algorithm is similar to the algorithm by Blankenship and Falk [9] for semi-infinite programs, in that the lower bounding problems become successively tighter, until the upper bounding problem is guaranteed to generate a feasible point. The novelty of the basic algorithm compared to the algorithm by Blankenship and Falk is mainly the generation of parametric upper bounds to the inner problem; this is significantly more difficult for bilevel programs than it is for semi-infinite programs. In Sect. 6 a tightening of the lower bounding problem is described based on the KKT necessary conditions of the inner problem for a class of bilevel programs satisfying additional assumptions. In Sect. 7 the algorithm is extended to allow for branching and two branching heuristics are discussed, which can accelerate convergence. In Sect. 8 a basic numerical implementation of the algorithm is described and applied to literature and original problems. Finally, the performance of the algorithm is described and improvements to the computational performance are proposed. Note also, that the algorithm proposed here exploits ideas from global optimization proposed by Floudas and coworkers [22,28] for bilevel programs with convex inner programs. The consideration of nonconvex inner programs adds significant complications, in particular that the inner program cannot be replaced by its KKT conditions. The main innovations proposed here are the convergent lower bounding problem, the analysis of requirements for the KKT based tightening of the lower bounding problem and the branching framework which is significantly different from single-level programs. Another contribution is the generation of several small but hard test-problems which can be used as a benchmark for future algorithmic proposals.

## 2 Definitions and assumptions

This section contains definitions of terms used, assumptions required by the algorithm and immediate consequences of these assumptions.

### 2.1 Definitions

**Definition 1** (*Inner Program*) For a fixed $\mathbf{x}$ we denote:

$$
\begin{aligned}
\min_{\mathbf{z}}\ & h(\mathbf{x},\mathbf{z}) \\
\text{s.t. } & \mathbf{p}(\mathbf{x},\mathbf{z}) \le \mathbf{0} \\
& \mathbf{q}(\mathbf{z}) \le \mathbf{0} \\
& \mathbf{z} \in Y,
\end{aligned} \tag{2}
$$

the *inner program*.

**Definition 2** (*Parametric Optimal Solution Function*) The parametric optimal solution value of (2) as a function of the outer variables is denoted $\bar{h}(\mathbf{x})$ and the set of optimal points $H(\mathbf{x}) \subset Y$. For infeasible inner programs ($H(\mathbf{x}) = \emptyset$) the convention $\bar{h}(\mathbf{x}) = +\infty$ is used.

Note that depending on the value of $\mathbf{x}$, the set $H(\bar{\mathbf{x}})$ can be empty, a singleton, a finite set or an infinite set.

**Definition 3** (*$\varepsilon$-Optimality*) A pair $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is called $\varepsilon$-feasible if it satisfies the constraints of the inner and outer programs and $\varepsilon_h$−optimality in the inner program, i.e.:

$$
\begin{aligned}
\mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{y}}) &\le \mathbf{0} \\
\mathbf{p}(\bar{\mathbf{x}},\bar{\mathbf{y}}) &\le \mathbf{0} \\
\mathbf{q}(\bar{\mathbf{y}}) &\le \mathbf{0} \\
h(\bar{\mathbf{x}},\bar{\mathbf{y}}) &\le \bar{h}(\bar{\mathbf{x}}) + \varepsilon_h.
\end{aligned}
$$

An $\varepsilon$-feasible point is called $\varepsilon$-optimal if it satisfies $\varepsilon_f$-optimality in the outer program, i.e.:

$$
f(\bar{\mathbf{x}},\bar{\mathbf{y}}) \le f^* + \varepsilon_f.
$$

*Remark 1* In [35] it was argued that for bilevel programs (1) with nonconvex inner programs it is only plausible to expect a finitely terminating algorithm to provide a guarantee for $\varepsilon$-optimality. Depending on the solvers used for the subproblems, the constraints may also only be satisfied within $\varepsilon$-tolerance.

**Definition 4** ($\mathbf{x}$ *Feasible in the Outer Program*) The subset of $X$ which is admissible in the outer program is denoted:

$$
X_{\text{outer}} = \{\mathbf{x} \in X : \exists\, \bar{\mathbf{y}} \in Y : \mathbf{g}(\mathbf{x},\bar{\mathbf{y}}) \le \mathbf{0}\}.
$$

**Definition 5** ($\mathbf{x}$ *Feasible in the Inner Program*) The subset of $X$ which is admissible in the inner program is denoted:

$$
X_{\text{inner}} = \{\mathbf{x} \in X : \exists\, \bar{\mathbf{y}} \in Y : \mathbf{p}(\mathbf{x},\bar{\mathbf{y}}) \le \mathbf{0},\ \mathbf{q}(\bar{\mathbf{y}}) \le \mathbf{0}\}.
$$

**Definition 6** (*Level Sets*) For a given $\bar{f} \in \mathbb{R}$ define the (potentially nonconvex) level sets

$$
Q_l(\bar{f}) = \{\mathbf{x} \in X,\ \mathbf{y} \in Y : \mathbf{g}(\mathbf{x},\mathbf{y}) \le \mathbf{0},\ \mathbf{p}(\mathbf{x},\mathbf{y}) \le \mathbf{0},\ \mathbf{q}(\mathbf{y}) \le \mathbf{0},\ f(\mathbf{x},\mathbf{y}) \le \bar{f}\}
$$

and the projection to the $X$ space

$$
X_l(\bar{f}) = \{\mathbf{x} \in X : \exists\, \mathbf{y} \in Y : (\mathbf{x},\mathbf{y}) \in Q_l(\bar{f})\}.
$$

**Definition 7** (*Partition*) A partition of a set $X^i \subset X$ is a finite collection of subsets, $X^l \subset X^i, X^{l+1} \subset X^i, \ldots, X^{l+m} \subset X^i$ such that

$$
X^i = X^l \cup X^{l+1} \cup \ldots \cup X^{l+m} \quad \text{and} \quad \text{int}\left(X^{l_1}\right) \cap \text{int}\left(X^{l_2}\right) = \emptyset, \quad \forall l_1 \ne l_2,
$$

compare also [30]. The definition for $X^i \times Y^i$ is analogous.

### 2.2 Assumptions

The algorithm presented relies on the global solution of nonconvex nonlinear programs (NLPs) and mixed-integer nonlinear programs (MINLPs) (see, e.g., [48]). In the following it is taken for granted that finite NLP and MINLP algorithms exist that can solve programs involving a finite number of inequality constraints to $\varepsilon_{NLP}$-optimality. Additional requirements imposed by such solvers, e.g., continuous second derivatives, are not discussed. On finite termination these NLP/MINLP solvers provide a lower bound to the optimal solution value and a feasible point with an objective function value that is not more than $\varepsilon_{\text{NLP}}$ larger than the lower bound [48]. All the formulated subproblems are inequality constrained with the exception of the stationarity constraint of the KKT-based lower bounds. The complementarity conditions are reformulated to inequalities involving binary variables. Since the KKT conditions are not required for convergence, an approximate solution, i.e., a relaxation, of the stationarity condition, suffices for a lower bound. Typical NLP/MINLP solvers, e.g., [48] satisfy inequalities only within a (nonzero) tolerance. To account for this, only slight modifications would be needed for the results presented here. This restriction is not made, because there is no theoretical requirement for violation of inequalities by either NLP solvers or the algorithm.

As is typical in global optimization compact host sets are required:

**Assumption 1** (*Host Sets*) The host sets $X \subset \mathbb{R}^{n_x}$, $Y \subset \mathbb{R}^{n_y}$ are Cartesian products of (compact) intervals, i.e., for all variables explicit bounds are known ($x_j \in [x_j^{LO}, x_j^{UP}]$ for $j = 1, \ldots, n_x$ and $y_j \in [y_j^{LO}, y_j^{UP}]$ for $j = 1, \ldots, n_y$).

*Remark 2* Considering arbitrary bounded polyhedra as host sets would not essentially alter the algorithm and the restriction to boxes is done for the sake of simplicity. With mild restrictions on branching, for each node $X^i \times Y^i$ we have $x_j \in [x_j^{i,LO}, x_j^{i,UP}]$ and $y_j \in [y_j^{i,LO}, y_j^{i,UP}]$.

**Assumption 2** (*Basic Properties of Functions*) The functions $f: X \times Y \to \mathbb{R}$, $\mathbf{g}: X \times Y \to \mathbb{R}^{n_g}$, $h: X \times Y \to \mathbb{R}$, and $\mathbf{p}: X \times Y \to \mathbb{R}^{n_p}$ are continuous on $X \times Y$. Similarly, $\mathbf{q}: Y \to \mathbb{R}^{n_q}$ is continuous on $Y$.

*Remark 3* By the continuity of the constraints and the compact host sets it directly follows that $X_{\text{inner}}$, $X_{\text{outer}}$ and $X_{\text{inner}} \cap X_{\text{outer}}$ are closed and therefore compact. Moreover, $Q_l(\bar{f})$ is compact for any $\bar{f}$ and therefore also $X_l(\bar{f})$ is compact. Finally, for all $\bar{\mathbf{x}} \in X_{\text{inner}}$ the minimum of the inner program exists.

**Assumption 3** (*Inner Problem*) There exists some $\tilde{\varepsilon}_f > 0$ such that for each point $\bar{\mathbf{x}} \in X_{\text{outer}} \cap X_{\text{inner}}$ at least one of the following two conditions holds:

1. For any $\varepsilon_{h1} > 0$ there exists a point $\tilde{\mathbf{z}} \in Y$ such that

$$\mathbf{p}(\bar{\mathbf{x}}, \tilde{\mathbf{z}}) < \mathbf{0}, \qquad \mathbf{q}(\tilde{\mathbf{z}}) \leq \mathbf{0}, \qquad h(\bar{\mathbf{x}}, \tilde{\mathbf{z}}) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h1}. \tag{3}$$

2. The outer objective value is $\tilde{\varepsilon}_f$ worse than the optimal objective value $f^*$

$$f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) > f^* + \tilde{\varepsilon}_f, \quad \forall \bar{\mathbf{y}} \in Y : \mathbf{p}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq \mathbf{0},\ \mathbf{q}(\bar{\mathbf{y}}) \leq \mathbf{0},\ \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq \mathbf{0} \tag{4}$$

   or equivalently $\bar{\mathbf{x}} \notin X_l(f^* + \tilde{\varepsilon}_f)$.

*Remark 4* By convention, for infeasible problems the optimal objective value is taken as infinity ($f^* = +\infty$). Therefore for infeasible problems condition (3) must hold for all $\bar{\mathbf{x}} \in X_{\text{outer}} \cap X_{\text{inner}}$.

*Remark 5* Conditions (3) and (4) can both hold for some points.

*Remark 6* Condition (3) of Assumption 3 allows the construction of parametric upper bounds for the parametric optimal solution function of the inner program, thus guaranteeing convergence of the algorithm. If this assumption was required for all $\mathbf{x} \in X_{\text{outer}} \cap X_{\text{inner}}$, continuity of the constraints $\mathbf{p}$ would give $X_{\text{outer}} \cap X_{\text{inner}} = X_{\text{outer}}$, or the inner program would be feasible for all $\mathbf{x}$ that are admissible in the outer program. It has been argued that this should always be the case [4,19], but here this restriction is not made. Note that points $\mathbf{x} \notin X_{\text{inner}}$ are considered infeasible in the bilevel program. The proposed algorithm essentially checks for Condition (3) for any point visited by the lower bounding problem. If this condition is violated, the algorithm assumes that (4) is satisfied instead.

*Remark 7* Using the continuity of the inner objective function $h$, a sufficient condition for (3) is that for each $\bar{\mathbf{x}} \in X_{\text{outer}} \cap X_{\text{inner}}$ and for each solution point of the inner problem $\mathbf{z}^* \in H(\bar{\mathbf{x}})$ and for each $\varepsilon_z > 0$ there exists a point $\tilde{\mathbf{z}} \in Y$, such that

$$\mathbf{p}(\bar{\mathbf{x}}, \tilde{\mathbf{z}}) < \mathbf{0}, \quad \mathbf{q}(\tilde{\mathbf{z}}) \leq \mathbf{0}, \quad ||\tilde{\mathbf{z}} - \mathbf{z}^*|| < \varepsilon_z. \tag{5}$$

*Remark 8* In the case of differentiability of the inner problem, condition (5) and therefore also (3) can be derived from the Mangasarian–Fromowitz constraint qualification (MFCQ) [5, p. 323] for the inner program.

## 3 Lower bounding procedure

In single-level optimization, lower bounds are typically obtained by the solution of a convex relaxation. As discussed in [35], a valid relaxation of the constraint "**y** is a global minimum of the inner program" is the constraint "**y** is feasible in the inner program" [53]. It can be easily verified that the above requirement alone does not give a convergent lower bound, as Example 3.1 shows.

*Example 3.1* (Convergence of Lower Bound) Consider the linear min-max problem

$$\begin{aligned} &\min_y y \\ &\text{s.t. } y \in \arg\min_z -z \\ &\quad y, z \in [-1, 1], \end{aligned}$$

and note that the only feasible point is $y = 1$ with an optimal solution value of 1. Suppose that branching is performed and consider a node $Y^i = [y^{i,LO}, y^{i,UP}]$. Replacing the inner problem with its constraints results to a feasible problem and a lower bound of $y^{i,LO}$, which is lower than the best possible incumbent. As a consequence no node can be fathomed and the lower bound remains at $-1$.

To achieve convergence parametric upper bounds for the optimal solution function of the inner program in the lower bounding problem are included. Note that (1) is equivalent to [3]:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \ & f(\mathbf{x}, \mathbf{y}) \\ \text{s.t. } & \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{p}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{q}(\mathbf{y}) \leq \mathbf{0} \\ \mathbf{x} \in X \Rightarrow \ & h(\mathbf{x}, \mathbf{y}) \leq \bar{h}(\mathbf{x}) \\ & \mathbf{x} \in X, \quad \mathbf{y} \in Y. \end{aligned} \tag{6}$$

This reformulation has also been used by Tuy et al. [51,52] in an algorithm for linear bilevel problems. This reformulation has the advantage that while the inner program may have infinitely many optimal solution points, it always has a unique optimal objective value. Thus, by using this reformulation multiple global minima in the inner program pose no essential complication.

Let now $K$ be an index set for a finite collection of pairs $\left(V^k, \mathbf{y}^k\right)$, composed of sets $V^k \subset X$ and points $\mathbf{y}^k \in Y$, such that for each $\mathbf{y}^k$ the inner constraints are satisfied for all $\bar{\mathbf{x}} \in V^k$, i.e.,

$$\begin{aligned} \mathbf{q}(\mathbf{y}^k) &\leq \mathbf{0} \\ \mathbf{p}(\bar{\mathbf{x}}, \mathbf{y}^k) &\leq \mathbf{0}, \quad \forall \bar{\mathbf{x}} \in V^k. \end{aligned} \tag{7}$$

Then, the program

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \\ &\text{s.t. } \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ &\quad \mathbf{p}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ &\quad \mathbf{q}(\mathbf{y}) \leq \mathbf{0} \\ &\mathbf{x} \in V^k \Rightarrow h\left(\mathbf{x}, \mathbf{y}\right) \leq h\left(\mathbf{x}, \mathbf{y}^k\right), \quad \forall k \in K \\ &\quad \mathbf{x} \in X, \quad \mathbf{y} \in Y \end{aligned} \tag{8}$$

provides a relaxation of (6). Indeed, consider a point $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in X \times Y$ which is feasible in (6). It directly follows $\mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq \mathbf{0}$, $\mathbf{p}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leq \mathbf{0}$, and $\mathbf{q}(\bar{\mathbf{y}}) \leq \mathbf{0}$. Furthermore, since $\bar{\mathbf{y}}$ is a global minimum of the inner program for $\bar{\mathbf{x}}$ together with (7)

$$h\left(\bar{\mathbf{x}}, \bar{\mathbf{y}}\right) = \bar{h}\left(\bar{\mathbf{x}}\right) \leq h\left(\bar{\mathbf{x}}, \mathbf{y}^k\right), \quad \forall k \in K : \bar{\mathbf{x}} \in V^k$$

which proves that $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ is feasible in (8). Therefore, a valid lower bound can be obtained from the global solution value of (8). Note that the use of logical constraints is well established, see, e.g., [6,41,54]. In Sect. 8 a simple implementation of these constraints is described.

Note that state-of-the-art global solvers in general provide only $\varepsilon_{\text{NLP}}$-estimates to the solution of (8). To obtain a valid lower bound to (1), the final lower bound provided by the solver has to be used as the lower bound. On the other hand, the $\varepsilon_{\text{NLP}}$-optimal point furnished is used for the subsequent steps of the algorithm.

*Remark 9* Assuming that finite upper and lower bounds are available, namely $UBD$ and $LBD$ respectively, these can be augmented to problem (8) as a constraint $LBD \leq f(\mathbf{x}, \mathbf{y}) \leq UBD$ with the aim of accelerating convergence. The lower bound inherited by the parent node can be used as a lower bound $LBD$. The current incumbent can be used for $UBD$; nodes with a lower bound that does not satisfy this inequality are fathomed anyway by value dominance.

*Remark 10* An alternative to the global solution of problem (8) is to further relax it using convex relaxation methods, e.g., [48], and solve the resulting convex program with a convex solver. In this case, also a feasible point of (8) should be obtained and used in the subsequent steps of the algorithm. This can also be achieved by solving (8) with a global solver and a loose tolerance $\varepsilon_{\text{NLP}}$.

In the remainder of this section a three-step procedure is described to obtain points $\mathbf{y}^k$ and sets $V^k$ that satisfy (7). The first step is to fix the variables $\mathbf{x}$ to the values of the optimal solution $\bar{\mathbf{x}}$ obtained by the lower bounding problem (8) and to solve the inner problem globally

$$\begin{aligned} h^* = \min_{\mathbf{z}} h(\bar{\mathbf{x}}, \mathbf{z}) \\ \min \mathbf{p}(\bar{\mathbf{x}}, \mathbf{z}) \leq \mathbf{0} \\ \mathbf{q}(\mathbf{z}) \leq \mathbf{0} \\ \mathbf{z} \in Y. \end{aligned} \tag{9}$$

The results of this step are also used for the upper bounding procedure, see Sect. 4. Feasibility of (9) is guaranteed by the solution of (8). Similarly to the solution of (8), the final lower bound from the global solver needs to be used for $h^*$.

The second step is to pick $\varepsilon_{h2} > 0$ and to find a point $\mathbf{y}^k$ such that $\mathbf{p}(\bar{\mathbf{x}}, \mathbf{y}^k) < \mathbf{0}$, $\mathbf{q}(\mathbf{y}^k) \leq \mathbf{0}$ and $h(\bar{\mathbf{x}}, \mathbf{y}^k) \leq h^* + \varepsilon_{h2}$, e.g., by solution of the optimization problem

$$\begin{aligned} &\min_{\mathbf{z},u} u \\ &\text{s.t. } h(\bar{\mathbf{x}}, \mathbf{z}) \leq h^* + \varepsilon_{h2} \\ &\quad p_i(\bar{\mathbf{x}}, \mathbf{z}) \leq u, \quad i = 1, \ldots, n_p \\ &\quad \mathbf{q}(\mathbf{z}) \leq \mathbf{0} \\ &\quad \mathbf{z} \in Y, \quad u \leq 0. \end{aligned} \tag{10}$$

This problem is feasible by the solution of (9). Provided that condition (3) of Assumption 3 is satisfied, the optimal solution value of (10) is negative and $\mathbf{y}^k$ satisfies the required properties. To accelerate convergence, the solution of the inner problem (9) can be used as an initial guess. Finite convergence of the algorithm is guaranteed for sufficiently small $\varepsilon_{h2}$, see Sect. 5.1. If there are multiple solutions to (10), the path followed by the algorithm in finding an optimal solution of the bilevel program may change depending on which solution of (10) is obtained.

*Remark 11* With the further assumption of the MFCQ for the inner program, it would be possible to obtain a point $\mathbf{y}^k$ by considering the solution of (9) and taking a small step in the descent direction of the constraints $p_i$ of the inner program which are active at the optimal solution of (9), i.e., equal to zero.

The third step is to identify a set $V^k$, that satisfies (7) and contains $\bar{\mathbf{x}}$ in its interior, or its boundary coincides with the boundary of $X$. This problem has been considered by Oluwole et al. [42] in the context of kinetic model reduction and their methodology can be directly used here. Successively smaller boxes $V^k$ are guessed as shown in Subroutine 1. For a given box, (7) can in principle be checked by globally solving the nonsmooth nonconvex nonlinear optimization problem

$$u = \max_{\mathbf{x} \in V^k} \max_i p_i(\mathbf{x}, \mathbf{y}^k).$$

If $u \leq 0$, (7) is satisfied. Solving the above optimization problem is expensive and therefore here interval analysis [1,40] is employed to overestimate $u$. A consequence of this overestimation is that the largest possible $V^k$ is not obtained. For an efficient implementation the details of this procedure are important and should be tuned for the instance considered.

*Subroutine 1* (Calculating $V^k$)
Given a point $\bar{\mathbf{x}}$, a point $\mathbf{y}^k$, and the bounds of the $\mathbf{x}$ variables $\mathbf{x}^{LO}$ and $\mathbf{x}^{UP}$ valid bounds for the box $V^k = [\mathbf{v}^{k,LO}, \mathbf{v}^{k,UP}]$ are calculated. For simplicity, successively smaller boxes are guessed by scaling the box by $d \in (0, 1]$.

1. Set $d = 1$.
2. **LOOP**

   (a) **FOR** $j = 1, \ldots, n_x$ **DO**
   - **IF** $\bar{x}_j - \frac{d}{2}(x_j^{UP} - x_j^{LO}) < x_j^{LO}$ **THEN**
     – Set $v_j^{k,LO} = x_j^{LO}$.
     – Set $v_j^{k,UP} = x_j^{LO} + d\,(x_j^{UP} - x_j^{LO})$.
   - **ELSE IF** $\bar{x}_j + \frac{d}{2}(x_j^{UP} - x_j^{LO}) > x_j^{UP}$ **THEN**
     – Set $v_j^{k,LO} = x_j^{UP} - d\,(x_j^{UP} - x_j^{LO})$.
     – Set $v_j^{k,UP} = x_j^{UP}$.
   - **ELSE**
     – Set $v_j^{k,LO} = \bar{x}_j - \frac{d}{2}(x_j^{UP} - x_j^{LO})$.
     – Set $v_j^{k,UP} = \bar{x}_j + \frac{d}{2}(x_j^{UP} - x_j^{LO})$.
     **END**

   (b) Check (7) by evaluating the interval extension of $\mathbf{p}(\cdot, \mathbf{y}^k)$ on $V^k$.
   **IF** (7) is satisfied **THEN** terminate **ELSE** Reduce $d$ **END**.

   **END**

We later prove that this is a finite procedure. The computational requirement for this subroutine is typically insignificant compared to the lower bounding problems.

At this point a brief explanation of interval analysis is warranted. For a thorough analysis, the reader is referred to the literature, e.g., [1,40]. Since $V^k$ is a Cartesian product of intervals and the constraints of the inner problem $\mathbf{p}$ are continuous, the image of each real valued function $p_i(\cdot, \mathbf{y}^k) : V^k \to \mathbb{R}$ (for fixed $\mathbf{y}^k$) is an interval $[p_i^l, p_i^u]$. An interval valued function $G(V^k)$ which satisfies

$$[p_i^l, p_i^u] \subset G(V^k) = [p_i^L, p_i^U]$$

is referred to as an *inclusion function* for $p_i(\cdot, \mathbf{y}^k)$ on $V^k$. An obvious requirement on the inclusion function is convergence to the true image $[p_i^l, p_i^u]$ as $||\mathbf{v}^{k,UP} - \mathbf{v}^{k,LO}||$ is reduced. The natural interval extension is an example of such an inclusion function. It is derived by replacing each variable $x_j$ by the corresponding interval $[v_j^{k,LO}, v_j^{k,UP}]$ and evaluating the resulting expression using the rules of interval arithmetic [40]. The functions are decomposed into a finite sequence of compositions of elementary operations (e.g., multiplication, addition) and intrinsic functions, such as monomials or the exponential function. For each of the intrinsic functions and elementary operations, rules are available to construct the natural interval extension. For instance, in the addition of two intervals the lower bound is given by addition of the two lower bounds and the upper bound by addition of the two upper bounds. In general, natural interval extensions lead to an overestimation of $[p_i^l, p_i^u]$, but in special cases, such as monomials, an exact calculation is obtained. Tighter inclusion functions can be calculated using Taylor model inclusions [1]. Note also that interval analysis methods can be automated, see e.g., [49].

*Example 3.2* (Illustration of Subroutine 1) Suppose a bilevel program with $X = [0, 1]$, $Y = [-1, 1]$ and the additional (linear in $z$) constraint $zx^2 \leq 0.6$ in the inner problem. Suppose further that at some iteration of the main algorithm we have obtained $\bar{x} = 0.5$ and $y^k = 1$. Suppose finally that at each iteration of Subroutine 1 the diameter is divided by two. At the first iteration $d = 1$ is used and the interval [0, 1] is probed. The upper bound obtained by natural interval extensions of $x^2$ in [0, 1] is $1^2 = 1$ and violates the requirement $\leq 0.6$. At the second iteration $d = 0.5$ is used and the interval [0.25, 0.75] is probed. The upper bound obtained by natural interval extensions of $x^2$ in [0.25, 0.75] is $0.75^2 = 0.5625$ and satisfies the requirement $\leq 0.6$. The result of Subroutine 1 is the box $V^k = [0.25, 0.75]$.

## 4 Upper bounding procedure

As discussed in [35], currently no method exists that provides valid, convergent upper bounds for bilevel programs with nonconvex inner programs without the generation of feasible points. An upper bounding procedure is now proposed by probing the feasibility of a candidate solution $\bar{\mathbf{x}}$.

Given a candidate $\bar{\mathbf{x}}$, the first step is to solve the nonconvex inner program (9) globally and obtain an optimal solution $\bar{\mathbf{y}}$ and an optimal solution value $h^*$. For an arbitrary point $\bar{\mathbf{x}}$, this program may be infeasible, in which case no solution to the bilevel program exists for $\mathbf{x} = \bar{\mathbf{x}}$ and no upper bound can be obtained. The algorithm only considers candidates generated by the solution of the lower bounding problem (8) for which the feasibility of (9) is guaranteed. Given the solution $h^*$ the outer problem is solved for the fixed $\bar{\mathbf{x}}$

$$\begin{aligned} \min_{\mathbf{y}} \; & f(\bar{\mathbf{x}}, \mathbf{y}) \\ \text{s.t. } & \mathbf{g}(\bar{\mathbf{x}}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{p}(\bar{\mathbf{x}}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{q}(\mathbf{y}) \leq \mathbf{0} \\ & h(\bar{\mathbf{x}}, \mathbf{y}) \leq h^* + \varepsilon_h \\ & LBD \leq f(\bar{\mathbf{x}}, \mathbf{y}) \\ & \mathbf{y} \in Y, \end{aligned} \tag{11}$$

allowing an $\varepsilon_h$-violation of the inner program objective. This step is performed, because, due to potential non-uniqueness of the solutions of the inner program, a valid upper bound may be obtained even if the solution to (9) does not satisfy the outer constraints. If (11) is infeasible then no solution exists for $\mathbf{x} = \bar{\mathbf{x}}$; otherwise an upper bound is obtained. The inequality $LBD \leq f(\bar{\mathbf{x}}, \mathbf{y})$ is added to accelerate convergence of (11) and to alleviate partially the consequences of allowing $\varepsilon_h$-optimality in the inner program. Note that the solution of (11) is only an upper bound in the sense of an $\varepsilon$-feasible point.

*Remark 12* If the solution to (9) is feasible in the outer program, an upper bound is obtained without solving (11), but (11) in general gives a better upper bound. Similarly, an upper bound can be obtained through any feasible point of (11), e.g., a local solution. For reasons of simplicity, it is assumed in the following that (11) is always solved to global optimality. Under this assumption, the algorithmic behavior is not affected by which global minimum is obtained in the solution of the inner program. If the optimal solution point of (9) satisfies the outer constraints, it can be used as an initial guess for (11).

## 5 Basic algorithm

The basic strategy of the algorithm is similar to the algorithm by Blankenship and Falk [9] for semi-infinite programs. The addition of parametric upper bounds on the optimal solution value of the inner program via the pairs $\left(V^k, \mathbf{y}^k\right)$ makes the lower bounding problems successively tighter. Finite termination is essentially achieved because either the sets $V^k$ cover $X_{\text{inner}} \cap X_{\text{outer}}$ and infeasibility is proved, or a lower bounding problem furnishes a point inside an existing set $V^k$ and close to a previously generated point and a $\varepsilon$-optimal point is obtained. The generation of parametric upper bounds is possible due to condition (3) of Assumption 3.

Input to the algorithm are the optimality tolerances $\varepsilon_f$ and $\varepsilon_h$, satisfying the assumptions of Theorem 1.

**Algorithm 1** (Basic Algorithm)

1. (**Initialization**)
   Set $LBD = -\infty$, $UBD = +\infty$, $k = 1$.
2. (**Lower Bounding**)
   Solve (8) globally.
   **IF** Feasible **THEN**
   - Set $LBD$ to the optimal objective value (final lower bound).
   - Set $\bar{\mathbf{x}}$ equal to the solution point ($\varepsilon_{\text{NLP}}$-optimal point).

   **ELSE** (**Infeasible Problem**)
   - Terminate.

   **END**
3. (**Termination**)
   **IF** $LBD \geq UBD - \varepsilon_f$ **THEN** Terminate.
4. (**Inner Problem**)
   Solve NLP (9) globally for $\mathbf{x} = \bar{\mathbf{x}}$. (Recall that feasibility of this program is guaranteed.)
   Set $h^*$ equal to the optimal objective value (final lower bound).
5. (**Populate Parametric Upper Bounds to Inner Problem**)
   Solve (10). (Recall that feasibility of this program is guaranteed.)
   - Set $\mathbf{y}^k$ equal to the solution point.
   - Obtain an appropriate set $V^k$.
   - Insert $k$ to $K$.
   - Set $k = k + 1$.
6. (**Upper Bounding**)
   Solve NLP (11) for $\mathbf{x} = \bar{\mathbf{x}}$ with $h^*$ as the upper bound for $h(\bar{\mathbf{x}}, \mathbf{y})$ and (if feasible) obtain an $\varepsilon_{\text{NLP}}$-optimal point $\bar{\mathbf{y}}$.
   **IF** Feasible and $f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) < UBD$ **THEN** set $UBD = f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and $(\mathbf{x}^*, \mathbf{y}^*) = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$.
7. (**Loop**)
   **IF** $LBD \geq UBD - \varepsilon_f$ **THEN** Terminate **ELSE** Goto step 2.

A direct consequence of the validity of the lower and upper bounding procedures is that on termination of the algorithm, if $UBD = +\infty$, the instance is infeasible. Otherwise, $UBD$ is an $\varepsilon_f$-estimate of the optimal solution value ($UBD \leq f^* + \varepsilon_f$) and $(\mathbf{x}^*, \mathbf{y}^*)$ is an $\varepsilon$-optimal point (see Definition 3) at which $UBD$ is attained. Note that depending on the problem instance it may be beneficial to swap Steps 6 and 5 and only perform the latter if $LBD < UBD - \varepsilon_f$. Note that Algorithm 1 can be applied to bilevel programs irrespectively of convexity properties.

### 5.1 Convergence proof

In this section a convergence proof for Algorithm 1 is given. Note again that no convexity or uniqueness assumptions are made for either the inner or outer programs.

**Lemma 1** (Continuity of Optimal Solution Function of Inner Problem) *The optimal objective function* $\bar{h} : X \to \mathbb{R}$ *of the inner problem is continuous for all* $\mathbf{x} \in X_{\text{inner}} \cap X_{\text{outer}}$ *satisfying* (3).

*Proof* Consider any fixed $\bar{\mathbf{x}} \in X_{\text{inner}} \cap X_{\text{outer}}$. By (3) for any $\varepsilon_{h1} > 0$, there exists $\tilde{\mathbf{y}} \in Y$ such that

$$\mathbf{p}(\bar{\mathbf{x}}, \tilde{\mathbf{y}}) < \mathbf{0}, \qquad \mathbf{q}(\tilde{\mathbf{y}}) \leq \mathbf{0} \tag{12}$$

$$h(\bar{\mathbf{x}}, \tilde{\mathbf{y}}) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h1}. \tag{13}$$

By continuity of the inner objective $h(\cdot, \tilde{\mathbf{y}})$ on $X$, for any $\varepsilon_{h3} > 0$ there exists $\delta_1 > 0$ such that

$$h(\mathbf{x}, \tilde{\mathbf{y}}) < h(\bar{\mathbf{x}}, \tilde{\mathbf{y}}) + \varepsilon_{h3}, \quad \forall \mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \delta_1. \tag{14}$$

Combining inequalities (13) and (14) it follows:

$$h(\mathbf{x}, \tilde{\mathbf{y}}) < \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h1} + \varepsilon_{h3}, \quad \forall \mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \delta_1. \tag{15}$$

By (12) and continuity of $\mathbf{p}(\cdot, \tilde{\mathbf{y}})$, there exists $\delta_2 > 0$ such that

$$\mathbf{p}(\mathbf{x}, \tilde{\mathbf{y}}) \leq \mathbf{0}, \quad \forall \mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \delta_2.$$

Together with $\mathbf{q}(\tilde{\mathbf{y}}) \leq \mathbf{0}$, $\tilde{\mathbf{y}}$ is feasible in the inner program for all $\mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \delta_2$. By the definition of $\bar{h}(\mathbf{x})$ we therefore have

$$\bar{h}(\mathbf{x}) \leq h(\mathbf{x}, \tilde{\mathbf{y}}), \quad \forall \mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \delta_2.$$

With (15) we obtain

$$\bar{h}(\mathbf{x}) < \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h1} + \varepsilon_{h3}, \quad \forall \mathbf{x} \in X : ||\bar{\mathbf{x}} - \mathbf{x}|| < \min\{\delta_1, \delta_2\}$$

which proves that $\bar{h}$ is upper semi-continuous at $\bar{\mathbf{x}}$.

By Theorem 4.2.1 in Bank et al. [2] for all $\bar{\mathbf{x}} \in X_{\text{inner}} \cap X_{\text{outer}}$ the optimal objective function $\bar{h}$ of the inner problem is lower semi-continuous. □

**Lemma 2** (Minimum of Bilevel Program Exists) *Under Assumptions* 1, 2 *and* 3, *either* (1) *is infeasible or the minimum of* (1) *exists.*

*Proof* Let for now $f^*$ denote the infimum of (1) without asserting that the minimum is attained. By Definition 6 of the level sets, the bilevel program (1) is equivalent to

$$\begin{aligned} f^* = \inf_{\mathbf{x},\mathbf{y}} \ & f(\mathbf{x}, \mathbf{y}) \\ \min\ & h(\mathbf{x}, \mathbf{y}) \leq \bar{h}(\mathbf{x}) \\ & (\mathbf{x}, \mathbf{y}) \in Q_l(f^*). \end{aligned} \tag{16}$$

Since the level set $Q_l(f^*)$ is compact (see Remark 3), so is the feasible set of (16). Noting that for all $(\mathbf{x}, \mathbf{y}) \in Q_l(f^*)$ it follows $\mathbf{x} \in X_l(f^*)$ and condition (3) of Assumption 3 is satisfied. Therefore, by Lemma 1 $\bar{h}$ is continuous on the feasible set of (16). Therefore either (16) is infeasible or its minimum is attained. As a consequence either (1) is infeasible or the minimum of (1) exists. □

**Lemma 3** (Sets $V^k$ have Nonempty Interior) *For any (arbitrary but fixed) $\varepsilon_{h2}>0$ there exists $\delta_1>0$ such that for any $\bar{f} \leq f^* + \tilde{\varepsilon}_f$ and for each point $\bar{\mathbf{x}} \in X_l(\bar{f})$ the points $\mathbf{y}^k(\bar{\mathbf{x}})$ generated in Step 5 of Algorithm 1 satisfy*

$$\mathbf{p}(\mathbf{x}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \quad \mathbf{q}(\mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \quad h(\bar{\mathbf{x}}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h2}, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_1.$$

*Note that $\delta_1$ is independent of $\bar{\mathbf{x}}$.*

*Proof* Since $\bar{f} \leq f^* + \tilde{\varepsilon}_f$, all points $\bar{\mathbf{x}} \in X_l(\bar{f})$ satisfy (3), $X_l(\bar{f})$ is compact and by Lemma 1 the optimal objective function of the inner problem $\bar{h} : X \to \mathbb{R}$ is continuous at all $\bar{\mathbf{x}} \in X_l(\bar{f})$.

Let $\bar{u}(\mathbf{x})$ denote the parametric optimal solution value of (10). By the continuity of the functions and the compactness of $X_l(\bar{f})$, $\bar{u}$ is continuous and its maximum over $\bar{\mathbf{x}} \in X_l(\bar{f})$ is attained. Since $\varepsilon_{h2} > 0$, by (3) $\bar{u}$ is strictly negative on $X_l(\bar{f})$. Therefore, there exists $\tilde{u} < 0$ such that for all $\bar{\mathbf{x}} \in X_l(\bar{f})$

$$p_i(\bar{\mathbf{x}}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \tilde{u} < 0, \ i = 1, \ldots, n_p, \quad \mathbf{q}(\mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \quad h(\bar{\mathbf{x}}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h2}.$$

Since $\mathbf{p}(\cdot, \mathbf{y})$ is continuous, and $X_l(\bar{f})$ is compact, $\mathbf{p}(\cdot, \mathbf{y})$ is uniformly continuous on $X_l(\bar{f})$. Therefore there exists $\delta_1 > 0$ (independent of $\bar{\mathbf{x}}$) such that for any $\bar{\mathbf{x}} \in X_l(\bar{f})$

$$\mathbf{p}(\mathbf{x}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \mathbf{q}(\mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \quad h(\bar{\mathbf{x}}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h2}, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_1. \qquad \square$$

*Remark 13* A direct consequence of Lemma 3 is that there exists $d_1>0$, such that all sets $V^k$ obtainable in Step 5 of Algorithm 1 satisfy

$$\min_j \left\{ v_j^{k,UP} - v_j^{k,LO} \right\} \geq d_1.$$

Interval analysis underestimates the size of these sets, but it has been shown [33] that natural interval extensions converge uniformly and therefore there exists $d_2 > 0$, such that for all $\bar{\mathbf{x}} \in X_l(\bar{f})$ the sets obtained satisfy

$$\min_j \left\{ v_j^{k,UP} - v_j^{k,LO} \right\} \geq d_2.$$

**Lemma 4** *Let $X_t \subset X$ be compact, and $\delta > 0$. Consider any infinite sequence of points $\mathbf{x}^i \in X_t$. There exists a finite index $I > 0$, such that*

$$||\mathbf{x}^I - \mathbf{x}^i|| \leq \delta, \qquad \textit{for some } i < I.$$

*Proof* Consider any infinite sequence $\mathbf{x}^i \in X_t$. Since $X_t$ is compact it is also bounded and therefore the sequence is also bounded. Therefore there exists a point $\bar{\mathbf{x}}$ and a subsequence $\mathbf{x}^{i_k}$ that converges to $\bar{\mathbf{x}}$, i.e., there exists a finite $K > 0$, such that

$$||\mathbf{x}^{i_k} - \bar{\mathbf{x}}|| \leq \delta/2, \quad \forall k \geq K.$$

Therefore

$$||\mathbf{x}^{i_{K+1}} - \mathbf{x}^{i_K}|| \leq ||\mathbf{x}^{i_{K+1}} - \bar{\mathbf{x}}|| + ||\mathbf{x}^{i_K} - \bar{\mathbf{x}}|| \leq \delta/2 + \delta/2 \leq \delta.$$

For $I = i_{K+1}$ and $i = i_K$ we have the desired result. $\square$

**Theorem 1** (Finite Termination) *If the tolerances of the optimization subproblems $\varepsilon_{\mathrm{NLP}}$ and $\varepsilon_{h2}$ in (10) satisfy*

$$0 < \varepsilon_{\mathrm{NLP}} \leq \min\{\varepsilon_f/2, \varepsilon_h, \tilde{\varepsilon}_f\}$$
$$0 < \varepsilon_{h2} < \varepsilon_h - \varepsilon_{\mathrm{NLP}},$$

*then Algorithm 1 terminates finitely.*

Since the proof of Theorem 1 is lengthy, we first present an outline of the proof. Because the lower bounding problem visits only points $\bar{\mathbf{x}} \in X_{\text{outer}} \cap X_{\text{inner}}$, it will not visit points $\tilde{\mathbf{x}} \notin X_l(f^* + \tilde{\varepsilon}_f)$. Therefore, by Assumption 3 at the points visited by the lower bounding problem it is possible to construct the parametric upper bounds to the optimal solution value of the inner program via the pairs $\left(V^k, \mathbf{y}^k\right)$. The corresponding logical constraints augmented to the lower bounding problem successively tighten the lower bounding problem to the extent that it will either become infeasible or furnish a point inside an existing $V^k$ which is also a $\varepsilon$-optimal point.

*Proof* Let $\bar{f} = f^* + \tilde{\varepsilon}_f$. Note that only points $\bar{\mathbf{x}} \in X_l(f^* + \varepsilon_{\text{NLP}}) \subset X_l(\bar{f})$ are furnished by the lower bounding problem. Points $\mathbf{x} \in X_l(\bar{f})$ satisfy condition (3) of Assumption 3 and this allows the generation of logical constraints via the pairs $\left(V^k, \mathbf{y}^k\right)$. Let $\bar{\mathbf{x}} \in X_l(\bar{f})$ be furnished by the lower bounding problem. We will show that if in a subsequent iteration the lower bounding problem furnishes a point $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ with $\hat{\mathbf{x}}$ sufficiently close to $\bar{\mathbf{x}}$, it will be $\varepsilon$-optimal. By Lemma 3 there exists $\delta_1 > 0$ such that for each point $\bar{\mathbf{x}} \in X_l(\bar{f})$ the points $\mathbf{y}^k$ generated in Step 7 of Algorithm 1 satisfy

$$\mathbf{p}(\mathbf{x}, \mathbf{y}^k) \leq \mathbf{0}, \quad \mathbf{q}(\mathbf{y}^k(\bar{\mathbf{x}})) \leq \mathbf{0}, \quad h(\bar{\mathbf{x}}, \mathbf{y}^k(\bar{\mathbf{x}})) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h2}, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_1. \quad (17)$$

Recall also

$$h(\bar{\mathbf{x}}, \mathbf{y}^k) \leq \bar{h}(\bar{\mathbf{x}}) + \varepsilon_{h2}.$$

By assumption $\varepsilon_h - \varepsilon_{NLP} - \varepsilon_{h2} > 0$. By continuity of $\bar{h}$ at $\bar{\mathbf{x}}$ there exists $\delta_2 > 0$ such that

$$\bar{h}(\bar{\mathbf{x}}) \leq \bar{h}(\mathbf{x}) + (\varepsilon_h - \varepsilon_{h2} - \varepsilon_{\text{NLP}})/2, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_2.$$

By continuity of $h(\cdot, \mathbf{y}^k)$ on $X$, there exists $\delta_3 > 0$ such that

$$h(\mathbf{x}, \mathbf{y}^k) \leq h(\bar{\mathbf{x}}, \mathbf{y}^k) + (\varepsilon_h - \varepsilon_{h2} - \varepsilon_{\text{NLP}})/2, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta_3.$$

Combining these last three inequalities gives

$$h(\mathbf{x}, \mathbf{y}^k) \leq \bar{h}(\mathbf{x}) + \varepsilon_h - \varepsilon_{\text{NLP}}, \quad \forall \mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \min\{\delta_2, \delta_3\}.$$

Therefore, together with (17), $\mathbf{y}^k$ is $\varepsilon_h$-optimal in the inner problem for all $\mathbf{x} \in X : ||\mathbf{x} - \bar{\mathbf{x}}|| < \delta$, where $\delta = \min\{\delta_1, \delta_2, \delta_3\} > 0$. Note that these $(\mathbf{x}, \mathbf{y}^k)$ are not necessarily feasible with respect to the outer constraints, and therefore termination does not occur immediately.

Since $X_l(\bar{f})$ is compact and $\delta > 0$, by Lemma 4 after a finite number of iterations either the lower bounding problem becomes infeasible, in which case the algorithm terminates, or the lower bounding problem furnishes a point $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, with $\hat{\mathbf{x}}$ sufficiently close to $\bar{\mathbf{x}}$, i.e., $||\hat{\mathbf{x}} - \bar{\mathbf{x}}|| < \delta$. By construction of the lower bounding problem, this $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ satisfies the inner and outer constraints and also $h(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq h(\hat{\mathbf{x}}, \mathbf{y}^k)$ (by the logical constraint) and as a consequence

$$h(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \bar{h}(\hat{\mathbf{x}}) + \varepsilon_h - \varepsilon_{\text{NLP}}$$

or $\hat{\mathbf{y}}$ is $\varepsilon_h$-optimal in the inner problem for $\hat{\mathbf{x}}$. Note that the $\varepsilon_{\text{NLP}}$ tolerance is included here, because the global solution of the inner problem only gives a $\varepsilon_{\text{NLP}}$-estimate of $\bar{h}(\bar{\mathbf{x}})$. The lower bound $LBD$ obtained satisfies $LBD \geq f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \varepsilon_{\text{NLP}}$. The point $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is feasible in the bilevel program and therefore the upper bounding problem (Step 8) furnishes an upper bound $UBD$, satisfying $UBD \leq f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) + \varepsilon_{\text{NLP}}$. Noting now that the optimization problems are solved with tolerance $\varepsilon_{\text{NLP}} < 2\varepsilon_f$ it follows $UBD - LBD \leq 2\varepsilon_{\text{NLP}} \leq \varepsilon_f$ and the algorithm terminates. □

*Remark 14* For a finite number of iterations, an arbitrarily large $\varepsilon_{h2}$ can be used for (10). In the worse case, this will create a finite number of redundant logical constraints, but may accelerate convergence, by obtaining larger sets $V^k$ at step 7.

*Remark 15* Since it was proved that after a finite number of iterations the lower bounding problem furnishes a point that satisfies $\varepsilon$-optimality, the solution of the upper bounding problem is not required for finite convergence. On the other hand, the upper bounding procedure may accelerate convergence, and therefore it is solved at every iteration. Note also that $LBD \leq f^*$ is always guaranteed since the lower bounding problem is a valid relaxation.

### 5.2 Illustrative examples

*Example 5.1* Recall Example 1.1 and the bilevel program

$$\begin{aligned} &\min_{x,y} x + y \\ &\text{s.t. } y \in \arg\min_{z} x\frac{z^2}{2} - \frac{z^3}{3} \\ &\quad x \in [-1, 1], \quad y, z \in [-1, 1] \end{aligned}$$

with the unique optimal solution $x = -1$, $y = 1$ with an objective value of 0.

Consider the application of Algorithm 1. At the first iteration for the lower bounding problem

$$\min_{x\in[-1,1], y\in[-1,1]} x + y$$

is solved and $LBD = -2$, $\bar{x} = -1$, $\bar{y} = -1$ is obtained. Then the inner problem is solved for $\bar{x} = -1$

$$\min_{z\in[-1,1]} -1\frac{z^2}{2} - \frac{z^3}{3}$$

and $\bar{y} = 1$, $h^* = 0.888$ is obtained. Since the constraints of the inner problem do not depend on $x$, the pair $([-1, 1], 1)$ is used for the parametric upper bounds of the inner problem. The first iteration is concluded by solving the augmented upper bounding problem for $\bar{x} = 1$:

$$\begin{aligned} &\min_{y\in[-1,1]} -1 + y, \\ &\min\ -1\tfrac{y^2}{2} - \tfrac{y^3}{3} \leq 0.888 \end{aligned}$$

obtaining the unique optimal point of the bilevel problem $x^* = -1$, $y = 1$.

A second iteration is required to confirm the optimality. Now the lower bounding problem contains a parametric upper bound to the inner problem

$$\begin{aligned} &\min_{x\in[-1,1], y\in[-1,1]} x + y, \\ &\min x\tfrac{y^2}{2} - \tfrac{y^3}{3} \leq x\tfrac{(-1)^2}{2} - \tfrac{(-1)^3}{3} \end{aligned}$$

which gives $\bar{x} = -1$, $\bar{y} = 1$, $LBD = 0$. Since $LBD = UBD$ the algorithm terminates.

*Example 5.2* Consider the bilevel program

$$\begin{aligned} &\min_{x,y} x^2 - y \\ &\min\ y \in \arg\min_{z} \left((z - 1 - 0.1x)^2 - 0.5 - 0.5\,x\right)^2 \\ &\quad x \in [0, 1], \qquad y, z \in [0, 3]. \end{aligned} \tag{18}$$

For each $x \in [0, 1]$ the inner program has two global minima, $y = 1 + 0.1x \pm 0.5\sqrt{2 + 2\,x}$. Therefore (18) has the unique optimal solution $x \approx 0.2106$, $y \approx 1.799$ with an objective value of approximately $-1.755$. This example is included to demonstrate how the algorithm will proceed when the inner program has multiple global minima.

Consider the application of Algorithm 1. At the first iteration for the lower bounding problem

$$\min_{x\in[0,1],\, y\in[0,3]} x^2 - y$$

is solved and $LBD = -3$, $\bar{x} = 0$, $\bar{y} = 3$ is obtained. Then the inner problem is solved for $\bar{x} = 0$

$$\min_{z\in[0,3]} \left((z - 1)^2 - 0.5\right)^2$$

obtaining $h^* = 0$. Since the inner problem has two global minima, either $\bar{y} \approx 1.707$ or $\bar{y} \approx 0.2929$ may be obtained as the solution point. The former is more advantageous for the outer objective, and therefore suppose that the latter is obtained (worse-case analysis). Since the constraints of the inner problem do not depend on $x$, the pair ([0, 1], 0.2929) is used for the parametric upper bound of the inner problem. The first iteration is concluded by solving the augmented upper bounding problem for $\bar{x} = 0$:

$$\begin{aligned} &\min_{y\in[0,3]} 0 - y, \\ &\min\ \left((y - 1 - 0.1)^2 - 0.5 - 0.5\right)^2 \le 0 \end{aligned}$$

obtaining $\bar{y} = 1.707$ and an upper bound of $-1.707$ to the optimal objective value of the bilevel program. In subsequent iterations the parametric upper bound on the inner program leads to convergence of the lower and upper bounds on the optimal objective value. Since these parametric upper bounds are equivalent for the two global minima of the inner program, the algorithmic behavior does not depend on the choice of minimum for the inner program. Recall though that, in general, the validity range of the parametric upper bounds depends on the solution point of (10).

## 6 KKT-based tightening of lower bounding problem

In the following a tightening of the lower bounding problem is proposed based on the KKT necessary conditions, which requires additional assumptions on the inner program:

**Assumption 4** (*Assumptions for the KKT-based Lower Bound*) For all $\bar{\mathbf{x}} \in X_{\text{outer}} \cap X_{\text{inner}}$ the following three conditions hold: (i) differentiability of $h(\bar{\mathbf{x}}, \cdot)$ and $\mathbf{p}(\bar{\mathbf{x}}, \cdot)$ on some open set embedding $Y$, (ii) a constraint qualification for the inner program, and (iii) a-priori known upper bounds for the KKT multipliers.

*Remark 16* The first two parts of Assumption 4 are standard for smooth NLP solvers. The third part is not easy to verify for a general inner program, but there are many interesting applications for which it is satisfied. For instance, in feasibility and flexibility problems [27] the KKT multipliers are bounded above by one. Also, for semi-infinite programs bounds can be readily estimated [38]. For problems for which any of the three parts is violated, the KKT-based lower bounding problem is not applicable.

If Assumption 4 is satisfied, the lower bounding problem (8) can be tightened by further requiring that $\mathbf{y}$ satisfies the KKT necessary conditions for the inner program. The KKT multipliers are added to the set of variables

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y},\boldsymbol{\mu}} \; & f(\mathbf{x},\mathbf{y}) \\
\text{s.t.} \; & \mathbf{g}(\mathbf{x},\mathbf{y}) \le \mathbf{0} \\
& \mathbf{p}(\mathbf{x},\mathbf{y}) \le \mathbf{0} \\
& \mathbf{q}(\mathbf{y}) \le \mathbf{0} \\
\mathbf{x} \in V^k \Rightarrow h(\mathbf{x},\mathbf{y}) & \le h\left(\mathbf{x},\mathbf{y}^k\right), \quad \forall k \in K \\
\nabla_{\mathbf{y}} h(\mathbf{x},\mathbf{y}) + \boldsymbol{\mu}^{\mathrm{T}} \nabla_{\mathbf{y}} \tilde{\mathbf{p}}(\mathbf{x},\mathbf{y}) & = \mathbf{0} \\
\mu_j \tilde{p}_j(\mathbf{x},\mathbf{y}) & \le 0, \quad j = 1,\ldots,n_p + n_q + 2n_y \\
\mu_j & \in [0, \mu_j^{\max}], \quad j = 1,\ldots,n_p + n_q + 2n_y \\
\mathbf{x} \in X, \; & \mathbf{y} \in Y,
\end{aligned}
\tag{19}
$$

where for simplicity the constraints ($\mathbf{p}$ and $\mathbf{q}$) of the inner program have been combined and augmented (to $\tilde{\mathbf{p}}$) to include the box constraints $\mathbf{y} \in Y$, i.e.:

$$
\tilde{p}_j(\mathbf{x},\mathbf{y}) = \begin{cases}
p_j(\mathbf{x},\mathbf{y}), & j = 1,\ldots,n_p \\
q_{j-n_p}(\mathbf{y}), & j = n_p+1,\ldots,n_p+n_q \\
y_{j-n_p-n_q} - y^{UP}_{j-n_p-n_q}, & j = n_p+n_q+1,\ldots,n_p+n_q+n_y \\
-y_{j-n_p-n_q-n_y} + y^{LO}_{j-n_p-n_q-n_y}, & j = n_p+n_q+n_y+1,\ldots,n_p+n_q+2n_y.
\end{cases}
$$

At this point it is important to discuss briefly the importance of the upper bounds on the KKT multipliers. For a more thorough discussion see [35,36]. The big-M reformulation of the complementarity slackness condition needs explicit bounds for both the constraints $g_i$ and KKT multipliers $\mu_i$. Fortuny-Amat and McCarl [24] first proposed the big M-reformulation but do not specify how big M should be. Moreover, for a valid lower bound a further relaxation or a global solution of the relaxations constructed is needed, and for either of these typically all variables need to be bounded [44,48]. The regularization approach as in Stein and Still [47] does not need bounds for the regularization but if the resulting program is to be solved to global optimality bounds on the KKT multipliers are most likely needed. Recall also, that for programs with a special structure upper bounds may be known a priori or it may be easy to estimate those.

Note that as Example 6.1 shows, the addition of the KKT necessary conditions does not guarantee convergence without the parametric upper bounds for the inner program.

*Example 6.1* (Convergence of KKT-based Lower Bound) Consider the trivial bilevel program

$$\begin{aligned}
&\min_{y} y \\
&\text{s.t. } y \in \arg\min_{z} -z^2 \\
&\qquad\qquad y, z \in [-0.5, 1],
\end{aligned}$$

and note that the only feasible point is $y = 1$ with an optimal solution value of 1. The inner problem has three KKT points namely $y \in \{-0.5, 0, 1\}$. Suppose that branching is performed. At each iteration there exists a node containing $-0.5$, which gives a lower bound of $-0.5$, which is lower than the solution value of the only feasible point. As a consequence the lower bound does not converge. In order to confirm the above described behavior numerically, the solvers BARON, MINOS and KNITRO were tested through GAMS version 22.0 [11] on the MPEC described above. Irrespectively of the initial guess all three solvers converge to the global optimum of the MPEC ($y = -0.5$) which is infeasible in the bilevel program.

For bilevel programs with an inner program that is convex on $Y$ for each fixed $\bar{\mathbf{x}}$ and that satisfies Assumption 4, application of the KKT-based lower bounding problem (19) leads to convergence at the first iteration. On the other hand, if the simpler lower bounding problem (8) is used on such programs, the convexity is not exploited.

## 7 Algorithmic extension to branching

Algorithm 1 is now generalized to a branch-and-bound framework. Branching is an interesting heuristic for accelerating the convergence and has the advantage that it allows more flexibility, such as the local solution of certain subproblems. Before describing the algorithms a few points need to be clarified which are different from branching in single-level programs. While the outer variables $\mathbf{x}$ can be considered in the same manner as for single-level programs, the inner variables cannot since they participate in both the inner and outer problems. Consider a node for which $Y$ has been restricted to $Y^i$. In the lower bounding and upper bounding problems it is valid to require $\mathbf{y}$ to $Y^i$ since this is a restriction of the bilevel program and equivalent to the situation in single-level programs. On the other hand, for the solution of the inner level problem (9) the entire host set ($Y$ as opposed to $Y^i$) has to be considered, for otherwise $h^*$ would not correspond to the optimal solution of the inner problem, see also [36]. Similarly in (10) $Y$ should not be restricted to $Y^i$ in order to obtain the required point $\mathbf{y}^k$. Finally, in the KKT based lower bounding problem (19) the host set for the KKT conditions should not be restricted to $Y^i$ either, because this would generate spurious KKT points. In essence, one can branch on $\mathbf{y}$ but not on $\mathbf{z}$.

Input to the algorithm are the optimality tolerances $\varepsilon_f$ and $\varepsilon_h$, satisfying the assumptions of Theorem 2.

**Algorithm 2** (Generalized Algorithm)

1. (**Initialization**)
   Set $LBD = -\infty$, $UBD = +\infty$, $k = 1$ and $l = 1$.
   Set $K = \emptyset$ and $N = \{X \times Y\}$.
2. (**Termination Test**)
   Delete from $N$ all nodes $X^i \times Y^i$ with $LBD^i \geq UBD - \varepsilon_f$ (**Fathoming by value dominance**).
   Set $LBD = \min_{X^i \times Y^i \in N} LBD^i$.
   **IF** $N = \emptyset$ **THEN** Terminate.
3. (**Node Selection**)
   Select a node $X^i \times Y^i$ from $N$ according to some node selection heuristic.
4. (**Lower Bounding**)
   Solve (8) or (19) globally.
   **IF** Feasible **THEN**
   - Set $LBD^i$ to the optimal objective value (final lower bound).
   - Set $\bar{\mathbf{x}}$ equal to the solution point ($\varepsilon_{\mathrm{NLP}}$-optimal point).

   **ELSE** (**Fathoming by Infeasibility**)
   - Delete node $X^i \times Y^i$ from $N$ and goto step 2.

   **END**
5. (**Fathoming by Value Dominance**)
   **IF** $LBD^i \geq UBD - \varepsilon_f$ **THEN** delete node $X^i \times Y^i$ from $N$ and goto step 2.
6. (**Inner Problem**)
   Solve NLP (9) globally for $\mathbf{x} = \bar{\mathbf{x}}$. (Recall that feasibility of this program is guaranteed.)
   Set $h^*$ equal to the optimal objective value (final lower bound).
7. (**Populate Parametric Upper Bounds to Inner Problem**)
   Solve (10). (Recall that feasibility of this program is guaranteed.)
   **IF** $u^* < 0$ **THEN**
   - Set $\mathbf{y}^k$ equal to the solution point.
   - Obtain an appropriate set $V^k$.
   - Insert $k$ to $K$.
   - Set $k = k + 1$.

   **ELSE**
   - Delete node $X^i \times Y^i$ from $N$ and goto step 2.

   **END**
8. (**Upper Bounding**)
   Solve NLP (11) for $\mathbf{x} = \bar{\mathbf{x}}$ with $h^*$ as the upper bound for $h(\bar{\mathbf{x}}, \mathbf{y})$ and (if feasible) obtain an $\varepsilon_{\mathrm{NLP}}$-optimal point $\bar{\mathbf{y}}$.
   **IF** Feasible and $f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) < UBD$ **THEN** set $UBD = f(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and $(\mathbf{x}^*, \mathbf{y}^*) = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$.
9. (**Optional Branching Step**)
   Delete node $X^i \times Y^i$ from $N$.
   Partition the set $X^i \times Y^i$ into $m$ nodes $X^l \times Y^l$, $X^{l+1} \times Y^{l+1}, \ldots, X^{l+m} \times Y^{l+m}$ according to some branching heuristic.
   Set $LBD^l = LBD^{l+1} = \cdots = LBD^{l+m} = LBD^i$.
   Add the new nodes to $N$.
   Set $l = l + m$.
10. (**Loop**)
    Goto step 2.

At this point a justification of the potential fathoming at Step 7 is provided. If the lower bounding problem furnishes points $\tilde{\mathbf{x}}$ that do not satisfy (3), by Assumption 3, it follows $\tilde{\mathbf{x}} \notin X_l(f^* + \varepsilon_f)$ and therefore the lower bound (with or without the KKT heuristic) is higher than the optimal solution value: $LBD^i \geq f^* + \varepsilon_f$. The solution of the lower bounding problem is found within tolerance $\varepsilon_{\text{NLP}}$ and therefore, as long as $\varepsilon_{\text{NLP}} < \varepsilon_f$ such points can only be visited if branching is performed and only in nodes that do not contain points $\tilde{\mathbf{x}} \in X_l(f^*)$.

## 7.1 Convergence proof

In this section a convergence proof for Algorithm 2 is given. Note again that no convexity or uniqueness assumptions are made for either the inner or outer programs. For finite termination without additional assumptions on the problem instance, some restrictions on the branching and/or node selection heuristics are required. In Theorem 2, finite termination is proved for the cases of best-bound and breadth-first node selection heuristics. It is possible to show finite termination for other heuristics, but this is outside the scope of this paper.

**Theorem 2** (Finite Termination) *If the tolerance of the optimization subproblems* $\varepsilon_{\text{NLP}}$ *and* $\varepsilon_{h2}$ *in* (10) *satisfy*

$$0 < \varepsilon_{\text{NLP}} \leq \min\{\varepsilon_f/2, \varepsilon_h, \tilde{\varepsilon}_f\}$$
$$0 < \varepsilon_{h2} < \varepsilon_h - \varepsilon_{\text{NLP}},$$

*the branching heuristic is exhaustive* [29] *in X and either of the following node selection heuristics is employed*

1. *breadth-first*
2. *best-bound*

*then Algorithm* 2 *terminates finitely.*

The proof of Theorem 2 is similar to Theorem 1. Since the branching heuristic is exhaustive in $X$, the nodes visited eventually become smaller than the smallest possible $V^k$ and are fathomed.

*Proof* Let $\bar{f} = f^* + \tilde{\varepsilon}_f$ and $\delta$ have the same meaning as in the proof of Theorem 1.

1. *Breadth-first node selection heuristic*
   Using the breadth-first node selection heuristic, and since the branching is exhaustive in $X$, for any $d > 0$, after a finite number of iterations for all nodes $i$ and all variables $x_j$ it follows

$$x_j^{i,UP} - x_j^{i,LO} < d, \quad \forall j = 1, \ldots, n_x, \quad \forall i \in N.$$

   Note that nodes $i$ with $X^i \cap X_{\text{inner}} \cap X_{\text{outer}} = \emptyset$ are fathomed by infeasibility. Without loss of generality such nodes are ignored in the following. At every level, one or more nodes satisfy $X_l(f^*) \cap X^i \neq \emptyset$. For all these nodes the lower bounding problem generates points $\bar{\mathbf{x}} \in X_l(f^* + \varepsilon_{NLP}) \subset X_l(\bar{f})$. Once $d$ is sufficiently small such that

$$||\mathbf{x} - \bar{\mathbf{x}}|| < \delta, \qquad \forall \mathbf{x} \in X^i$$

the lower bounding problem of all children nodes $j$ is either infeasible or furnishes $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ with $\hat{\mathbf{x}}$ sufficiently close to $\bar{\mathbf{x}}$, leading to an upper bound $UBD^j$ such that $UBD^j - LBD^j \leq 2\,\varepsilon_{\text{NLP}} \leq \varepsilon_f$. In either case all children nodes of $i$ are fathomed. If the problem is infeasible, all nodes satisfy $X_l(f^*) \cap X^i \neq \emptyset$ and are fathomed. Otherwise, at least one node contains an optimal solution and furnishes $UBD^j \leq f^* + \varepsilon_f = \bar{f}$. With this incumbent, all nodes $X^i$, for which $X_l(f^*) \cap X^i = \emptyset$ are fathomed when they are visited.

2. *Best-bound node selection heuristic*
   At each iteration, a node with the best lower bound is chosen. Therefore only nodes $i$ with an inherited lower bound $LBD^i \leq f^*$ can be chosen.
   Consider any infinite nested sequence of nodes. The nodes in this sequence satisfy $X_l(f^*) \cap X^i \neq \emptyset$, and therefore the lower bounding problem generates points $\bar{\mathbf{x}} \in X_l(f^* + \varepsilon_{\text{NLP}}) \subset X_l(\bar{f})$. Since branching is exhaustive in $X$, for any $d > 0$, after a finite number of iterations for all variables $x_j$ it follows

$$x_j^{i,UP} - x_j^{i,LO} < d, \quad \forall j = 1, \ldots, n_x.$$

   Similarly to the best-bound heuristic this leads to either fathoming by infeasibility or generation of an upper bound such that $UBD^i - LBD^i \leq \varepsilon_f$ and the node is fathomed. Since $LBD^i \leq f^*$, this upper bound is sufficient to fathom all nodes by value dominance.

□

*Remark 17* Note the requirement that the branching procedure is exhaustive allows the branching to be performed only every finite number of iterations. Moreover, the proof can be extended to the case that branching is performed only a finite number of iterations and the resulting nodes are revisited without further branching.

### 7.2 Branching heuristics

Algorithm 1 is a special case of Algorithm 2 for the extreme case of no branching. A great number of additional heuristics are conceivable, and here two of particular interest are discussed. A simple choice is to perform branch-and-bound without any distinction between $\mathbf{x}$ and $\mathbf{y}$ by bisection on the variable with the current largest range. This procedure corresponds to a common branching heuristic in global single-level optimization. Convergence is achieved by a combination of the node shrinking and the addition of parametric upper bounds to the inner problem.

*Remark 18* For the breadth-first selection heuristic and branching by bisection on the variable with the current largest range, finite convergence can be guaranteed for $\tilde{\varepsilon}_f = 0$ in condition (4).

A more elaborate and specialized branching heuristic is to branch only on the $\mathbf{x}$ variables by partitioning the node into a set of nodes in such a way that one of the children nodes corresponds to the set $V^k$. The advantage of this branching heuristic is that it avoids the use of logical constraints. Given the parent node bounds on the variables $\mathbf{x}^{i,LO}$ and $\mathbf{x}^{i,UP}$ and the box bounds $\mathbf{v}^{k,LO}$ and $\mathbf{v}^{k,UP}$ a simple procedure for this partitioning is described in Subroutine 2 and illustrated graphically in Fig. 1.

*Subroutine 2* (Branching into $2n_x + 1$ Nodes)
Two temporary vectors $\tilde{\mathbf{x}}^{LO}$ and $\tilde{\mathbf{x}}^{UP}$ and two temporary variables $t_1$ and $t_2$ are used.

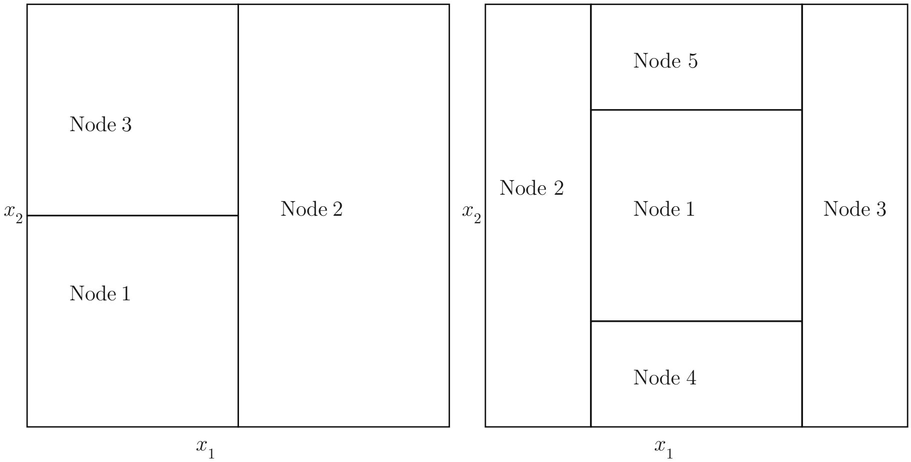


**Fig. 1** Graphical illustration of Subroutine 2 for Example 1. In the left hand side $V^k$ (box 1) is in the lower left corner of $X^i$ and therefore only three children nodes are created. In the right hand side $V^k$ (box 1) is in the interior of $X^i$ and therefore $2n_x + 1 = 5$ children nodes are created

1. (**Initialization**)
   Set $\tilde{\mathbf{x}}^{LO} = \mathbf{v}^{k,LO}$ and $\tilde{\mathbf{x}}^{UP} = \mathbf{v}^{k,UP}$.
2. (**Node corresponding to $V^k$ box**)
   Create node with $\mathbf{x} \in [\tilde{\mathbf{x}}^{LO}, \tilde{\mathbf{x}}^{UP}]$.
3. (**Up to two nodes per variable**)
   **FOR** $j = 1, \ldots, n_x$ **DO**

- Set $t_1 = \tilde{x}_j^{LO}$ and $t_2 = \tilde{x}_j^{UP}$
- **IF** $t_1 > x_j^{i,LO}$ **THEN**

– $\tilde{x}_j^{LO} = x_j^{i,LO}$ and $\tilde{x}_j^{UP} = t_1$.
– Create node with $\mathbf{x} \in [\tilde{\mathbf{x}}^{LO}, \tilde{\mathbf{x}}^{UP}]$.

- **IF** $t_2 < x_j^{i,UP}$ **THEN**

– $\tilde{x}_j^{LO} = t_2$ and $\tilde{x}_j^{UP} = x_j^{i,UP}$.
– Create node with $\mathbf{x} \in [\tilde{\mathbf{x}}^{LO}, \tilde{\mathbf{x}}^{UP}]$.

- Set $\tilde{x}_j^{LO} = x_j^{i,LO}$ and $\tilde{x}_j^{UP} = x_j^{i,UP}$.

By construction, the created nodes are a partition of the parent node and a maximum of $2n_x + 1$ children nodes are generated.

*Remark 19* When the set $V^k$ is equal to $X^i$, e.g., when the inner constraints do not depend on $\mathbf{x}$, this procedure re-creates the node $X^i$ which is equivalent to the heuristic of no branching.

*Example 7.1* (Illustration of Subroutine 2) Consider the application of Subroutine 2 for $n_x = 2$ and $X^i = [0, 1]^2$. Suppose first that $V^k = [0, 0.5]^2$. Three children nodes are created, namely $[0, 0.5]^2$, $[0.5, 1] \times [0, 1]$ and $[0, 0.5] \times [0.5, 1]$. Suppose now that $V^k = [0.25, 0.75]^2$. Five children nodes are created, namely $[0.25, 0.75]^2$, $[0, 0.25] \times [0, 1]$, $[0.75, 1] \times [0, 1]$, $[0.25, 0.75] \times [0, 0.25]$, $[0.25, 0.75] \times [0.75, 1]$. The latter case is the worse case possible since $2n_x + 1 = 5$. Both cases are illustrated in Fig. 1.

## 8 Implementation and numerical results

### 8.1 Implementation

The logical constraints in the lower bounding problem and the complementarity conditions of the KKT-based lower bounding problem were implemented using the big-M formulation [24,25]. Given a node $X^i$ and a box $V^k$ it is first checked if their intersection is empty. If it is $(X^i \cap V^k = \emptyset)$ the constraint does not need to be introduced. Also if $X^i \subset V^k$ the constraint is directly introduced

$$h(\mathbf{x}, \mathbf{y}) \leq h(\mathbf{x}, \mathbf{y}^k).$$

Otherwise up to two binary variables and constraints are introduced for each component of $\mathbf{x}$, as described in Subroutine 3. Therefore up to $2n_x + 1$ binary variables are required to formulate a logical constraint.

*Subroutine 3* (Implementation of Logical Constraints)

- Set $l = 0$
- **FOR** $j = 1, \ldots, n_x$ **DO**
  - **IF** $v_j^{k,LO} > x_j^{i,LO}$ **THEN**
    - ∗ Set $l = l + 1$ and introduce a binary variable $w_l \in \{0, 1\}$ corresponding to $x_j > v_j^{k,LO}$.
    - ∗ Introduce a constraint

$$w_l \geq \frac{x_j - v_j^{k,LO}}{x_j^{i,UP} - x_j^{i,LO}}.$$

  - **IF** $v_j^{k,UP} < x_j^{i,UP}$ **THEN**
    - ∗ Set $l = l + 1$ and introduce a binary variable $w_l \in \{0, 1\}$ corresponding to $x_j < v_j^{k,UP}$.
    - ∗ Introduce a constraint

$$w_l \geq \frac{v_j^{k,UP} - x_j}{x_j^{i,UP} - x_j^{i,LO}}.$$

  **END FOR**
- Introduce the logical constraint as

$$h(\mathbf{x}, \mathbf{y}) \leq h(\mathbf{x}, \mathbf{y}^k) + \sum_{i=1}^{l} (1 - w_i)(h^{\max} - h(\mathbf{x}, \mathbf{y}^k)),$$

  where $h^{\max} \geq h(\mathbf{x}, \mathbf{y})$ for all $(\mathbf{x}, \mathbf{y}) \in X^i \times Y^i$. $h^{\max}$ can be easily estimated using interval analysis, see, e.g., [40]. Note that unless $w_i = 1$, for all $i = 1, \ldots, l$ this constraint is redundant.

Note that the constraint $\mathbf{x} \in V^k \Rightarrow h(\mathbf{x}, \mathbf{y}) \leq h\left(\mathbf{x}, \mathbf{y}^k\right)$ is only introduced for the interior of $V^k$.

For the KKT-based lower bounds the complementarity condition $\mu_i \tilde{p}_i(\mathbf{x}, \mathbf{y})$ is reformulated with the help of a binary variable $w_l$ indicating if the constraint is active or not

$$\mu_i \leq M_i w_l$$
$$-\tilde{p}_i(\mathbf{x}, \mathbf{y}) \leq P_i(1 - w_l),$$

noting that $\tilde{p}_i(\mathbf{x}, \mathbf{y}) \leq 0$ and $\mu_i \geq 0$. By Assumption 4 an upper bound $M_i$ for the KKT multipliers is known a priori. Bounds $P_i$ on the constraints can be easily estimated by interval extension, see e.g., [40].

The algorithm was implemented in C++ and tested on a 64-bit Xeon processor 3.2GHz running Linux 2.6.13. The best-bound heuristic occurred at the node selection step, and among nodes with the same lower bound, the one that entered the set of active nodes ($N$) first was always used. As is typical with optimization codes, both an absolute and relative termination criterion was used and termination occurred if either of the criteria was satisfied.

The resulting nonconvex NLP and MINLPs were all solved globally with BARON version 7.4 [44] using GAMS version 22.0 [11] through system calls. Note that strictly speaking BARON does not satisfy the assumption about solvers, since the final lower bound furnished may not be strictly below the optimal objective function value (it can be slightly above); since tight tolerances were used for the subproblems, this consideration does not have practical implications. The occurrence of third order monomials, e.g., $x^3$ caused very slow convergence of the formulated (MI)NLPs in some of the case studies. For consistency purposes, third order monomials are therefore systematically encoded as a product of a square and a linear term, e.g., $x^2\,x$, and fourth order monomials as the product of two squares, e.g., $x^2\,x^2$.

### 8.2 Test set

To test the algorithm literature examples collected in [28,43] are used along with and a number of new test problems with either nonconvex inner problems or problems with a structure that causes convergence issues with previous proposals. The test set by Colson et al. [15,16] was not used because the inner problems are convex and would not add anything to the focus of this paper. The problem formulations with an analysis of the feasible sets and optimal solutions as well as justifications for the values used for the bounds on the KKT multipliers are given in [37].

Table 1 contains a summary of the problem properties. The first column is the label of the example. The second through sixth columns ($n_x, n_y, n_g, n_p, n_q$) contain the number of **x** variables, **y** variables, constraints in the outer problem, constraints in the inner problem that depend on the outer variables, and constraints in the inner problem (excluding box constraints) that do not depend on the outer variables. The seventh through tenth columns ($f$, $g$, $h$, $p$) contain the functional form of the outer objective, the outer constraints, the inner objective and the inner constraints: A stands for affine linear, C stands for convex nonlinear, N stands for nonconvex nonlinear, and P stands for pseudoconvex; for the outer functions the characterization is joint in **x** and **y** while for the inner functions the characterization is only for the **z**-dependence, e.g., convex means partially convex on $Y$. The eleventh column ($h$) indicates whether or not the inner objective depends on the outer variables; F stands for false (no dependence), T for true, and a dash is used for the problems without **x** variables ($n_x = 0$). Finally, the last two columns contain the optimal solution value and the set of optimal solutions as obtained by an analysis of the problems. Problems mb_2_3_01, mb_5_5_01, mb_5_5_02 and gf_3 have not yet been analyzed completely and the best available solution value, along with a (presumably) optimal solution is given. To emphasize that these problems have not been analyzed, a question mark is set next to the best available solution value.

Note that some of the problems do not contain any outer variables ($n_x = 0$). These problems can be easily solved by solving the inner problem and then solving an augmented outer program. They have been included because, despite their simplicity, they reveal problems with certain approaches.

Tables 2–4 contain the numerical results with the three branching heuristics presented. The branching on the **x** variables in $2n_x + 1$ nodes is only applied to the test problems for which the resulting algorithm is different than the extreme of no branching. The optimality and feasibility tolerances for BARON were set to $\varepsilon_{\text{NLP}} = 10^{-6}$ for all problems. For most

**Table 1** Summary of problem properties

| Label | $n_x$ | $n_y$ | $n_g$ | $n_p$ | $n_q$ | $f$ | $g$ | $h$ | $p$ | $h$ | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mb_0_1_01 | 0 | 1 | 0 | 0 | 0 | A | – | A | – | – | 1 | 1 |
| mb_0_1_02 | 0 | 1 | 1 | 0 | 0 | A | A | A | – | – | $\infty$ | None |
| mb_0_1_03 | 0 | 1 | 0 | 0 | 1 | A | – | C | N | – | −1 | −1 |
| mb_0_1_04 | 0 | 1 | 0 | 0 | 0 | A | – | N | – | – | 1 | 1 |
| mb_0_1_05 | 0 | 1 | 0 | 0 | 0 | A | – | N | – | – | 0.5 | 0.5 |
| mb_0_1_06 | 0 | 1 | 0 | 0 | 0 | A | – | N | – | – | −1 | −1 |
| mb_0_1_07 | 0 | 1 | 1 | 0 | 0 | A | A | A | – | – | $\infty$ | None |
| mb_1_1_01 | 1 | 1 | 2 | 0 | 0 | C | A | A | – | T | 0 | −0.567,0 |
| mb_1_1_02 | 1 | 1 | 1 | 0 | 0 | A | A | N | – | F | −1 | −1, −1 |
| mb_1_1_03 | 1 | 1 | 0 | 0 | 0 | A | – | N | – | T | 0.5 | [0.1,1],0.5 |
| mb_1_1_04 | 1 | 1 | 0 | 0 | 0 | A | – | N | – | T | −0.8 | 0,−0.8 |
| mb_1_1_05 | 1 | 1 | 0 | 0 | 0 | N | – | N | – | T | 0 | 0,0 |
| mb_1_1_06 | 1 | 1 | 0 | 0 | 0 | A | – | N | – | T | −1 | 0,1 |
| mb_1_1_07 | 1 | 1 | 0 | 0 | 0 | C | – | N | – | T | 0.25 | 0.25,0.5 |
| mb_1_1_08 | 1 | 1 | 0 | 0 | 0 | A | – | N | – | T | 0 | −1,1 |
| mb_1_1_09 | 1 | 1 | 0 | 0 | 0 | A | – | N | – | T | −2 | −1,0 and −0.5,−1 |
| mb_1_1_10 | 1 | 1 | 0 | 0 | 0 | C | – | N | – | T | 0.1875 | −0.25,±0.5 |
| mb_1_1_11 | 1 | 1 | 0 | 0 | 0 | N | – | N | – | T | 0.25 | 0.5,0 |
| mb_1_1_12 | 1 | 1 | 0 | 0 | 0 | N | – | N | – | T | −0.258 | 0.18858,−0.76759 |
| mb_1_1_13 | 1 | 1 | 0 | 0 | 0 | C | – | N | – | T | 0.3125 | 0.5,0.5 |
| mb_1_1_14 | 1 | 1 | 0 | 0 | 0 | C | – | N | – | T | 0.2095 | −0.5544,0.4554 |
| mb_1_1_15 | 1 | 1 | 0 | 1 | 0 | C | – | N | N | T | 0.2095 | −0.5544,0.4554 |
| mb_1_1_16 | 1 | 1 | 1 | 1 | 0 | C | N | C | C | F | 0.1756 | −0.4191,−1 |
| mb_1_1_17 | 1 | 1 | 0 | 0 | 0 | C | – | N | – | T | −1.755 | 0.2106,1.799 |
| mb_2_3_01 | 2 | 3 | 2 | 1 | 2 | N | N | N | N | T | −1? | −1, −1, −1,1,1 |
| mb_2_3_02 | 2 | 3 | 3 | 0 | 0 | N | N | N | – | T | −2.3535 | −1, −1, 1, 1, −0.707 |
| mb_5_5_01 | 5 | 5 | 3 | 1 | 2 | C | N | N | N | T | 2? | 0,0,0,0,0,−1, 0, −1.,0,0 |
| mb_5_5_02 | 5 | 5 | 3 | 1 | 2 | N | N | N | N | T | −10? | 1,−1, −1, −1, −1, −1, 1, −1, −1,1 |
| gf_1 | 1 | 1 | 1 | 1 | 0 | C | A | C | A | T | 2250 | 11.25,5 |
| gf_2 | 1 | 2 | 0 | 2 | 0 | N | A | A | C | F | 1 | 1,0,1 |

**Table 1** continued

| Label | $n_x$ | $n_y$ | $n_g$ | $n_p$ | $n_q$ | $f$ | $g$ | $h$ | $p$ | $h$ | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gf_3 | 2 | 3 | 0 | 2 | 1 | A | – | P | A | T | −29.2? | 0,0.9,0,0.6,0.4 |
| gf_4 | 1 | 1 | 3 | 0 | 0 | C | A | C | – | F | 9 | 3,5 |
| gf_5 | 1 | 1 | 0 | 1 | 1 | A | – | N | N | F | 0.193616 | 0.1936,9.9667,10 |
| sc_1 | 1 | 2 | 0 | 3 | 0 | A | – | A | A | F | −13 | 5,4,2 |
| sc_2 | 1 | 1 | 0 | 3 | 0 | C | – | C | A | F | 5 | 1,3 |

**Table 2** Numerical results without branching

| Label | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ | $\mu^{max}$ | $\varepsilon_h$ | $\varepsilon_{h2}^0$ | $\varepsilon_f$ | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mb_0_1_01 | 1 | 1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 1.00 | 1.00 | 1 | 1 | 2 | 0.06 | 0.29 |
| mb_0_1_02 | ∞ | – | 2 | $10^{-5}$ | NA | $10^{-4}$ | ∞ | | 0 | 1 | 2 | 0.04 | 0.29 |
| mb_0_1_03 | −1 | −1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.00 | −1.00 | 1 | 1 | 2 | 0.06 | 0.30 |
| mb_0_1_04 | 1 | −1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 1.00 | 1.00 | 1 | 1 | 2 | 0.06 | 0.30 |
| mb_0_1_05 | 0.5 | 0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.500 | 0.500 | 1 | 1 | 2 | 0.13 | 0.37 |
| mb_0_1_06 | −1 | −1 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −1 | −1 | 1 | 1 | 1 | 0.04 | 0.23 |
| mb_0_1_07 | ∞ | – | 2 | $10^{-5}$ | NA | $10^{-4}$ | ∞ | | 0 | 1 | 2 | 0.03 | 0.29 |
| mb_1_1_01 | 0 | −0.567,0 | 5 | $10^{-5}$ | NA | $10^{-4}$ | 0 | −0.567,0 | 2 | 2 | 2 | 0.08 | 0.43 |
| mb_1_1_02 | −1 | −1,−1 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −1 | −1,−1 | 1 | 1 | 1 | 0.04 | 0.22 |
| mb_1_1_03 | 0.5 | [0.1,1],0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.498 | 0.1,0.498 | 1 | 1 | 2 | 2.0 | 2.2 |
| mb_1_1_04 | −0.8 | 0,−0.8 | 100 | $10^{-5}$ | NA | $10^{-4}$ | −0.8 | 0,−0.8 | 1 | 1 | 1 | 0.04 | 0.22 |
| mb_1_1_05 | 0 | 0,0 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −0.004 | −0.004,0.00 | 9 | 9 | 9 | 4.4 | 6.0 |
| mb_1_1_06 | −1 | 0,1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.006 | −0.006,1 | 2 | 2 | 2 | 0.12 | 0.47 |
| mb_1_1_07 | 0.25 | 0.25,0.5 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.246 | 0.250,0.496 | 1 | 3 | 4 | 0.25 | 0.84 |
| mb_1_1_08 | 0 | −1,1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.00 | −1,1.00 | 1 | 1 | 2 | 0.08 | 0.31 |
| mb_1_1_09 | −2 | (−1,0),(−0.5,−1) | 2 | $10^{-5}$ | NA | $10^{-4}$ | −2.005 | −1,−0.0045 | 1 | 1 | 2 | 0.15 | 0.39 |
| mb_1_1_10 | 0.1875 | −0.25,±0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.185 | −0.250,−0.494 | 2 | 18 | 19 | 2.8 | 6.1 |
| mb_1_1_11 | 0.25 | 0.5,0 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.250 | 0.5,0.00 | 2 | 2 | 2 | 0.10 | 0.45 |
| mb_1_1_12 | −0.258 | 0.189,0.768 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −0.260 | 0.1874,0.438 | 10 | 12 | 13 | 1.7 | 3.9 |
| mb_1_1_13 | 0.3125 | 0.5,0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.310 | 0.501,0.497 | 3 | 3 | 4 | 0.26 | 0.84 |
| mb_1_1_14 | 0.2095 | −0.554,0.454 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.200 | −0.500,0.447 | 1 | 2 | 3 | 0.31 | 0.73 |
| mb_1_1_15 | 0.2095 | −0.554,0.454 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.207 | −0.567,0.454 | 5 | 5 | 5 | 1.1 | 2.3 |
| mb_1_1_16 | 0.1756 | −0.4191,−1 | NA | $10^{-5}$ | NA | $10^{-4}$ | 0.1756 | −0.4191,−1.000 | 2 | 2 | 3 | 0.08 | 0.60 |
| mb_1_1_17 | −1.755 | 0.2106,1.799 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.756 | 0.204886,1.7987 | 11 | 13 | 14 | 2.2 | 7.0 |
| mb_2_3_01 | −1? | | 20 | $10^{-5}$ | NA | $10^{-4}$ | −1.00 | 1,−1,−0.00,−1,−1 | 1 | 1 | 2 | 0.38 | 0.69 |
| mb_2_3_02 | −2.3535 | −1,−1,1,1,−0.707 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −2.35 | −1,−1,1,1.00,−0.707 | 2 | 2 | 3 | 0.17 | 0.59 |
| mb_5_5_01 | 2? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | 2 | 0,0,0,0,0,−1,0,−1,0,0 | 1 | 4 | 5 | 300 | 320 |
| mb_5_5_02 | −10? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | −10 | −1,−1,−1,1,1,−1,1,−1,−1,1 | 2 | 2 | 2 | 0.05 | 0.84 |

**Table 2** continued

| Label | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ | $\mu^{max}$ | $\varepsilon_h$ | $\varepsilon_{h2}^0$ | $\varepsilon_f$ | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gf_1 | 2250 | 11.25,5 | $10^4$ | 0.001 | 500 | 0.1 | 2250 | 11.24,5.028 | 42 | 42 | 43 | 86 | 97 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_2 | 1 | 1,0,1 | 100 | $10^{-5}$ | 0.1 | 0.15 | 1.00 | 1,−0.003,1.00 | 1 | 20 | 21 | 2.2 | 7.6 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_3 | −29.2? | 0,0.9,0,0.6,0.4 | $10^3$ | 0.1 | 1 | 0.5 | −30.3 | 0,0.38,0.515,0.768,0.042 | 15 | 15 | 16 | 20 | 24 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_4 | 9 | 3,5 | NA | $10^{-5}$ | NA | $10^{-4}$ | 9.00 | 3,5.00 | 2 | 2 | 2 | 0.10 | 0.46 |
| gf_5 | 0.1936 | 0.1936,9.9667,10 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.194 | 0.194,9.97,10 | 1 | 1 | 1 | 0.07 | 0.32 |
| sc_1 | −13 | 5,4,2 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −13 | 5,4,2 | 1 | 1 | 1 | 0.05 | 0.29 |
| sc_2 | 5 | 1,3 | 10 | $10^{-5}$ | 2 | 0.01 | 5.00 | 0.994,3.00 | 40 | 52 | 53 | 17 | 31 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |

| Label | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|
| mb_0_1_01 | 1 | 1 | 1 | 1 | 1 | 0.05 | 0.23 |
| mb_0_1_02 | ∞ | | 0 | 0 | 1 | 0.00 | 0.08 |
| mb_0_1_03 | −1.00 | −1.00 | 1 | 1 | 1 | 0.07 | 0.26 |
| mb_0_1_04 | 1.00 | 1.00 | 1 | 1 | 2 | 0.05 | 0.29 |
| mb_0_1_05 | 0.500 | 0.500 | 1 | 1 | 2 | 0.08 | 0.32 |
| mb_0_1_06 | −1 | −1 | 1 | 1 | 1 | 0.05 | 0.23 |
| mb_0_1_07 | ∞ | | 0 | 0 | 1 | 0.00 | 0.08 |
| mb_1_1_01 | 0 | −0.567,0 | 1 | 1 | 1 | 0.05 | 0.23 |
| mb_1_1_02 | −1 | −1,−1 | 1 | 1 | 1 | 0.14 | 0.33 |
| mb_1_1_03 | 0.498 | 0.100,0.498 | 1 | 1 | 2 | 0.12 | 0.36 |
| mb_1_1_04 | −0.8 | 0,−0.8 | 1 | 1 | 1 | 0.05 | 0.22 |
| mb_1_1_05 | −0.004 | 0.004,0.00 | 9 | 9 | 9 | 0.87 | 2.5 |
| mb_1_1_06 | −1.006 | −0.006,1 | 2 | 2 | 2 | 0.13 | 0.53 |
| mb_1_1_07 | 0.250 | 0.25,0.500 | 2 | 2 | 2 | 0.13 | 0.49 |
| mb_1_1_08 | 0.00 | −1,1.00 | 1 | 1 | 2 | 0.09 | 0.33 |
| mb_1_1_09 | −2.0045 | −1,−0.0045 | 1 | 1 | 2 | 0.06 | 0.30 |

**Table 2** continued

| Label | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|
| mb_1_1_10 | 0.1845 | −0.250,−0.494 | 2 | 3 | 4 | 0.27 | 0.87 |
| mb_1_1_11 | 0.2500 | 0.500,0.00 | 2 | 2 | 2 | 0.11 | 0.46 |
| mb_1_1_12 | −0.258 | 0.189,0.434 | 2 | 2 | 2 | 0.15 | 0.52 |
| mb_1_1_13 | 0.313 | 0.500,0.500 | 2 | 2 | 2 | 0.67 | 1.0 |
| mb_1_1_14 | 0.202 | −0.500,0.450 | 2 | 2 | 2 | 0.23 | 0.59 |
| mb_1_1_15 | 0.2095 | −0.554,0.455 | 1 | 1 | 1 | 0.28 | 0.53 |
| mb_1_1_16 | NA | NA | NA | NA | NA | NA | NA |
| mb_1_1_17 | −1.75472 | 0.210662,1.7991 | 1 | 1 | 1 | 0.01 | 0.024 |
| mb_2_3_01 | −1 | −1, −1, −1, 1, 1 | 1 | 1 | 1 | 0.25 | 0.51 |
| mb_2_3_02 | −2.35 | −1, −1, 1, 1.00, −0.707 | 1 | 1 | 2 | 0.16 | 0.40 |
| mb_5_5_01 | 2 | 0, 0, 0, 0, 0, −1, 0.01, −1, 0.005, 0.00797 | 7 | 7 | 7 | 360 | 400 |
| mb_5_5_02 | −10 | 1, −1, −1, −1, −1, −1, 1, −1, −1, 1 | 1 | 1 | 1 | 0.42 | 0.89 |
| gf_1 | 2250 | 11.25, 5.00 | 1 | 1 | 1 | 0.32 | 0.56 |
| | 2250 | 11.25,5.00 | 1 | 1 | 1 | 0.32 | 0.56 |
| gf_2 | 1.00 | 1,0,1.00 | 1 | 1 | 1 | 0.11 | 0.35 |
| | 1 | 1,0,1 | 1 | 1 | 1 | 0.11 | 0.35 |
| gf_3 | −29.2 | 0,0.9,0,0.6,0.4 | 1 | 1 | 1 | 0.56 | 0.80 |
| | −29.2 | 0,0.9,0,0.6,0.4 | 1 | 1 | 1 | 0.56 | 0.85 |
| gf_4 | 9 | 3,5.00 | 1 | 1 | 1 | 0.05 | 0.23 |
| gf_5 | 0.194 | 0.194,9.97,10 | 1 | 1 | 1 | 0.08 | 0.33 |
| sc_1 | −13 | 5,4,2 | 1 | 1 | 1 | 0.06 | 0.30 |
| sc_2 | 5.00 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |
| | 5.00 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |

**Table 3** Numerical results with regular branching

| Label | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ | $\mu^{max}$ | $\varepsilon_h$ | $\varepsilon_{h2}^0$ | $\varepsilon_f$ | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mb_0_1_01 | 1 | 1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 1.00 | 1.00 | 1 | 1 | 3 | 0.06 | 0.37 |
| mb_0_1_02 | ∞ | – | 2 | $10^{-5}$ | NA | $10^{-4}$ | ∞ | | 0 | 1 | 3 | 0.04 | 0.36 |
| mb_0_1_03 | −1 | −1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.00 | −1.00 | 1 | 1 | 3 | 0.08 | 0.38 |
| mb_0_1_04 | 1 | −1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 1.00 | 1.00,1.00 | 1 | 1 | 3 | 0.13 | 0.42 |
| mb_0_1_05 | 0.5 | 0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.500 | 0.500 | 1 | 1 | 3 | 0.13 | 0.43 |
| mb_0_1_06 | −1 | −1 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −1 | −1 | 1 | 1 | 1 | 0.04 | 0.22 |
| mb_0_1_07 | ∞ | – | 2 | $10^{-5}$ | NA | $10^{-4}$ | ∞ | | 0 | 1 | 3 | 0.04 | 0.36 |
| mb_1_1_01 | 0 | −0.567,0 | 5 | $10^{-5}$ | NA | $10^{-4}$ | 0 | −0.567,0 | 2 | 2 | 2 | 0.07 | 0.43 |
| mb_1_1_02 | −1 | −1,−1 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −1 | −1,−1 | 1 | 1 | 1 | 0.05 | 0.22 |
| mb_1_1_03 | 0.5 | [0.1,1],0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.498 | 0.1,0.498 | 1 | 1 | 3 | 2.1 | 2.4 |
| mb_1_1_04 | −0.8 | 0,−0.8 | 100 | $10^{-5}$ | NA | $10^{-4}$ | −0.8 | 0,−0.8 | 1 | 1 | 1 | 0.04 | 0.22 |
| mb_1_1_05 | 0 | 0,0 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −0.004 | 0.004,0.00 | 14 | 10 | 17 | 0.95 | 3.3 |
| mb_1_1_06 | −1 | 0,1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.01 | −0.01,1 | 2 | 2 | 3 | 0.13 | 0.54 |
| mb_1_1_07 | 0.25 | 0.25,0.5 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.246 | 0.250,0.4956 | 1 | 3 | 7 | 1.1 | 1.8 |
| mb_1_1_08 | 0 | −1,1 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.00 | −1,1.00 | 1 | 1 | 3 | 0.11 | 0.40 |
| mb_1_1_09 | −2 | (−1,0),(−0.5,−1) | 2 | $10^{-5}$ | NA | $10^{-4}$ | −2.00 | −1,−0.0045 | 1 | 1 | 3 | 0.16 | 0.46 |
| mb_1_1_10 | 0.1875 | −0.25,±0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.185 | −0.250,−0.494 | 2 | 18 | 37 | 2.2 | 6.5 |
| mb_1_1_11 | 0.25 | 0.5,0 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.250 | 0.500,0.00 | 3 | 2 | 3 | 0.32 | 0.73 |
| mb_1_1_12 | −0.258 | 0.189,0.768 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −0.260 | 0.187,0.438 | 18 | 14 | 27 | 1.7 | 5.0 |
| mb_1_1_13 | 0.3125 | 0.5,0.5 | 2 | $10^{-5}$ | NA | $10^{-4}$ | 0.309 | 0.500,0.496 | 3 | 3 | 5 | 0.30 | 0.95 |
| mb_1_1_14 | 0.2095 | −0.554,0.454 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.200 | −0.500,0.447 | 1 | 2 | 5 | 0.49 | 1.0 |
| mb_1_1_15 | 0.2095 | −0.554,0.454 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.207 | −0.567,0.454 | 8 | 5 | 11 | 1.3 | 2.9 |
| mb_1_1_16 | 0.1756 | −0.4191,−1 | NA | $10^{-5}$ | NA | $10^{-4}$ | −0.4191 | −1.000 | 2 | 2 | 3 | 0.08 | 0.70 |
| mb_1_1_17 | −1.755 | 0.2106,1.799 | 2 | $10^{-5}$ | NA | $10^{-4}$ | −1.756 | 0.204886,1.7987 | 19 | 14 | 27 | 2.2 | 7.0 |
| mb_2_3_01 | −1? | | 20 | $10^{-5}$ | NA | $10^{-4}$ | −1.003 | 1,−1,−0.003,−1,−1 | 1 | 1 | 3 | 0.36 | 0.72 |
| mb_2_3_02 | −2.3535 | −1,−1,1,1,−0.707 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −2.35 | −1,−1,1,1.00,−0.707 | 2 | 2 | 5 | 0.25 | 0.80 |
| mb_5_5_01 | 2? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | 2 | 0,0,0,0,0,−1,0,−1,0,0 | 1 | 7 | 15 | 460 | 540 |
| mb_5_5_02 | −10? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | −10 | −1,−1,−1,1,1,−1,1,−1,−1,1 | 2 | 2 | 2 | 0.09 | 0.82 |

**Table 3** continued

| Label | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ | $\mu^{max}$ | $\varepsilon_h$ | $\varepsilon_{h2}^0$ | $\varepsilon_f$ | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gf_1 | 2250 | 11.25,5 | $10^4$ | 0.001 | 500 | 0.1 | 2252 | 11.15,5.40 | 23 | 41 | 81 | 3.5 | 17 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_2 | 1 | 1,0,1 | 100 | $10^{-5}$ | 0.1 | 0.15 | 0.997 | 1,−0.003,1.00 | 1 | 26 | 53 | 3.1 | 11 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_3 | −29.2? | 0,0.9,0,0.6,0.4 | $10^3$ | 0.1 | 1 | 0.5 | −29.9 | 0,0.27,0.554,0.777,0 | 4 | 7 | 15 | 3.7 | 6.2 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| gf_4 | 9 | 3,5 | NA | $10^{-5}$ | NA | $10^{-4}$ | 9.00 | 3.00,5.00 | 2 | 2 | 3 | 0.12 | 0.54 |
| gf_5 | 0.1936 | 0.1936,9.9667,10 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.194 | 0.194,9.97,10 | 1 | 1 | 1 | 0.07 | 0.32 |
| sc_1 | −13 | 5,4,2 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −13 | 5,4,2 | 1 | 1 | 1 | 0.06 | 0.30 |
| sc_2 | 5 | 1,3 | 10 | $10^{-5}$ | 2 | 0.01 | 5.00 | 1,3 | 87 | 56 | 109 | 5.4 | 24 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | | | | | | |
| Label | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | | | | | | | |
| mb_0_1_01 | 1 | 1 | 1 | 1 | 1 | 0.05 | 0.22 | | | | | | |
| mb_0_1_02 | ∞ | | 0 | 0 | 1 | 0.00 | 0.08 | | | | | | |
| mb_0_1_03 | −1.00 | −1.00 | 1 | 1 | 1 | 0.08 | 0.26 | | | | | | |
| mb_0_1_04 | 1.00 | 1.00 | 1 | 1 | 3 | 0.05 | 0.35 | | | | | | |
| mb_0_1_05 | 0.500 | 0.500 | 1 | 1 | 3 | 0.09 | 0.39 | | | | | | |
| mb_0_1_06 | −1 | −1 | 1 | 1 | 1 | 0.05 | 0.23 | | | | | | |
| mb_0_1_07 | ∞ | | 0 | 0 | 1 | 0.00 | 0.08 | | | | | | |
| mb_1_1_01 | 0 | −0.567,0 | 1 | 1 | 1 | 0.04 | 0.23 | | | | | | |
| mb_1_1_02 | −1 | −1,−1 | 1 | 1 | 1 | 0.14 | 0.33 | | | | | | |
| mb_1_1_03 | 0.498 | 0.100,0.498 | 1 | 1 | 3 | 0.09 | 0.39 | | | | | | |
| mb_1_1_04 | −0.8 | 0,−0.8 | 1 | 1 | 1 | 0.04 | 0.22 | | | | | | |
| mb_1_1_05 | −0.004 | 0.004,0.00 | 14 | 10 | 17 | 0.81 | 3.0 | | | | | | |
| mb_1_1_06 | −1.01 | −0.009,1 | 2 | 2 | 3 | 0.13 | 0.55 | | | | | | |
| mb_1_1_07 | 0.250 | 0.250,0.500 | 3 | 2 | 3 | 0.16 | 0.57 | | | | | | |
| mb_1_1_08 | 0.00 | −1,1.00 | 1 | 1 | 3 | 0.10 | 0.40 | | | | | | |
| mb_1_1_09 | −2.00 | −1,−0.0045 | 1 | 1 | 3 | 0.06 | 0.36 | | | | | | |
| mb_1_1_10 | 0.1845 | −0.250,−0.494 | 2 | 3 | 7 | 0.34 | 1.1 | | | | | | |

**Table 3** continued

| Label | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|
| mb_1_1_11 | 0.250 | 0.500,0.00 | 3 | 2 | 3 | 0.12 | 0.54 |
| mb_1_1_12 | −0.258 | 0.189,0.434 | 3 | 2 | 3 | 0.15 | 0.58 |
| mb_1_1_13 | 0.3125 | 0.500,0.500 | 3 | 3 | 3 | 0.75 | 1.3 |
| mb_1_1_14 | 0.2025 | −0.500,0.4500 | 2 | 2 | 3 | 0.23 | 0.64 |
| mb_1_1_15 | 0.2095 | −0.554,0.455 | 1 | 1 | 1 | 0.29 | 0.53 |
| mb_1_1_16 | NA | NA | NA | NA | NA | NA | NA |
| mb_1_1_17 | −1.75472 | 0.210662,1.7991 | 1 | 1 | 1 | 0.01 | 0.024 |
| mb_2_3_01 | −1 | −1,−1,−1,1,1 | 1 | 1 | 1 | 0.26 | 0.51 |
| mb_2_3_02 | −2.35 | −1,−1,1,1.00,−0.707 | 1 | 1 | 3 | 0.16 | 0.47 |
| mb_5_5_01 | 2 | 0,0,0,0,0,−1,0.01,−1,0.0051,0.00797 | 4 | 4 | 4 | 95 | 106 |
| mb_5_5_02 | −10 | 1,−1,−1,−1,−1,−1,1,−1,−1,1 | 1 | 1 | 1 | 0.41 | 1.09 |
| gf_1 | 2250 | 11.25,5.00 | 1 | 1 | 1 | 0.32 | 0.57 |
| | 2250 | 11.25,5.00 | 1 | 1 | 1 | 0.33 | 0.56 |
| gf_2 | 1.00 | 1,0,1.00 | 1 | 1 | 1 | 0.11 | 0.35 |
| | 1.00 | 1,0,1.00 | 1 | 1 | 1 | 0.11 | 0.35 |
| gf_3 | −29.2 | 0, 0.9, 0, 0.6, 0.4 | 1 | 1 | 1 | 0.55 | 0.80 |
| | −29.2 | 0,0.9,0,0.6,0.4 | 1 | 1 | 1 | 0.56 | 0.84 |
| gf_4 | 9 | 3,5.00 | 1 | 1 | 1 | 0.05 | 0.23 |
| gf_5 | 0.194 | 0.194,9.97,10 | 1 | 1 | 1 | 0.09 | 0.33 |
| sc_1 | −13 | 5,4,2 | 1 | 1 | 1 | 0.06 | 0.30 |
| sc_2 | 5.00 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |
| | 5 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |

**Table 4** Numerical results with special branching on the **x** variables

| Label | $f^*$ | $\mathbf{x}^*, \mathbf{y}^*$ | $\mu^{max}$ | $\varepsilon_h$ | $\varepsilon_{h2}^0$ | $\varepsilon_f$ | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ |
|---|---|---|---|---|---|---|---|---|
| mb_1_1_15 | 0.2095 | −0.554,0.454 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.2069 | −0.567,0.454 |
| mb_1_1_16 | 0.1756 | −0.4191,−1 | NA | $10^{-5}$ | NA | $10^{-4}$ | −0.4191 | −1.000 |
| mb_2_3_01 | −1? | | 20 | $10^{-5}$ | NA | $10^{-4}$ | −1.00 | 1,−1,−0.003,−1,−1 |
| mb_5_5_01 | 2? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | 2 | 0,0,0,0,0,−1,0,−1,0,0 |
| mb_5_5_02 | −10? | | 100 | $10^{-5}$ | N/A | $10^{-4}$ | −10 | −1,−1,−1,1,1,−1,1,−1,−1,1 |
| gf_1 | 2250 | 11.25,5 | $10^4$ | 0.001 | 500 | 0.1 | 2250 | 11.18,5.30 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | |
| gf_2 | 1 | 1,0,1 | 100 | $10^{-5}$ | 0.1 | 0.15 | 0.997 | 1,−0.003,1.00 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | |
| gf_3 | −29.2? | 0,0.9,0,0.6,0.4 | $10^3$ | 0.1 | 1 | 0.5 | −29.9 | 0,0.27,0.554,0.777,0 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | |
| gf_5 | 0.1936 | 0.1936,9.9667,10 | 10 | $10^{-5}$ | NA | $10^{-4}$ | 0.194 | 0.194, 9.97,10 |
| sc_1 | −13 | 5,4,2 | 10 | $10^{-5}$ | NA | $10^{-4}$ | −13 | 5, 4, 2 |
| sc_2 | 5 | 1,3 | 10 | $10^{-5}$ | 2 | 0.01 | 5.00 | 1.00,3.00 |
| | | | | $10^{-5}$ | NA | $10^{-4}$ | | |

| Label | UBD | #UBD | #LBD | time | | $\bar{f}$ | $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ | UBD | #UBD | #LBD | time | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mb_1_1_15 | 5 | 5 | 5 | 1.1 | 2.3 | 0.2095 | −0.554 ,0.455 | 1 | 1 | 1 | 0.28 | 0.52 |
| mb_1_1_16 | 2 | 2 | 3 | 0.08 | 0.70 | NA | NA | NA | NA | NA | NA | NA |
| mb_2_3_01 | 1 | 1 | 2 | 0.38 | 0.68 | −1 | −1,−1,−1,1,1 | 1 | 1 | 1 | 0.26 | 0.50 |
| mb_5_5_01 | 1 | 4 | 5 | 308 | 338 | 2 | 0,0,0,0,0,−1,0.01,−1,0.0051,0.00797 | 7 | 7 | 7 | 361 | 402 |
| mb_5_5_02 | 2 | 2 | 2 | 0.072 | 0.835 | −10 | 1,−1,−1,−1,−1,−1,1,−1,−1,1 | 1 | 1 | 1 | 0.4 | 0.926 |
| gf_1 | 19 | 41 | 49 | 3.4 | 14 | 2250 | 11.25, 5.00 | 1 | 1 | 1 | 0.32 | 0.56 |
| | | | | | | 2250 | 11.25,5.00 | 1 | 1 | 1 | 0.32 | 0.56 |
| gf_2 | 1 | 17 | 23 | 1.2 | 5.6 | 1.00 | 1,0,1.00 | 1 | 1 | 1 | 0.11 | 0.35 |
| | | | | | | 1.00 | 1,0,1.00 | 1 | 1 | 1 | 0.10 | 0.35 |
| gf_3 | 4 | 19 | 42 | 6.9 | 15 | −29.2 | 0, 0.9, 0, 0.6, 0.4 | 1 | 1 | 1 | 0.56 | 0.81 |
| | | | | | | −29.2 | 0, 0.9, 0, 0.6, 0.4 | 1 | 1 | 1 | 0.56 | 0.85 |
| gf_5 | 1 | 1 | 1 | 0.07 | 0.32 | 0.194 | 0.194,9.97,10 | 1 | 1 | 1 | 0.09 | 0.33 |
| sc_1 | 1 | 1 | 1 | 0.05 | 0.32 | −13 | 5, 4, 2 | 1 | 1 | 1 | 0.05 | 0.30 |
| sc_2 | 32 | 65 | 125 | 4.0 | 24.7 | 5.00 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |
| | | | | | | 5.00 | 1.00,3.00 | 1 | 1 | 1 | 0.13 | 0.37 |

problems the optimality tolerance for the inner problem was set to $\varepsilon_h = 10^{-5}$ and the absolute and relative termination criteria to $\varepsilon_f = 10^{-4}$. For some of the literature problems, when the KKT heuristic is not used for the lower bound, the computational requirement is quite high, and the tolerances were set according to a tradeoff between accuracy and computational time. For these problems the solution is repeated with the KKT-based lower bounds and the default tolerances. Note that for all problems the tolerances used satisfy the assumptions in Theorems 1 and 2.

The first column (Label) has the label of the problem, while the second ($f^*$) and third ($\mathbf{x}^*, \mathbf{y}^*$) the optimal objective value and set of optimal solutions respectively, obtained by analysis. The fourth column ($\mu^{\max}$) contains the maximal values for KKT multipliers used, and the fifth ($\varepsilon_h$) the optimality tolerance for the inner program. For some problems a sequence of decreasing $\varepsilon_{h2}$ to obtain the points $y^k$ was used in step 7; the starting value is given in the sixth column ($\varepsilon_{h2}^0$). This value was decreased by a factor of 1.1 at each iteration down to $0.8\varepsilon_h$. For the rest of the problems $\varepsilon_{h2} = 0.8\varepsilon_h$ was used for all iterations. The seventh column ($\varepsilon_f$) contains the termination tolerance (absolute and relative termination criteria were set equal). To guess the boxes $V^k$, a decreasing sequence was used; each time the interval diameter was set to one and decreased by a factor of 0.9 until the interval extensions showed feasibility. Natural interval extensions were used.

The eighth through fourteenth columns contain the results obtained with the use of simple lower bounds while the fifteenth through twenty-first columns contain the results obtained with the use of the KKT heuristic for the lower bounds; $\bar{f}$ shows the optimal objective value obtained; $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ shows the optimal solution obtained; UBD shows the node at which the optimal solution was first obtained; #UBD shows the number of upper bounding calls; #LBD shows the number of lower bounding calls; the first time columns show the sum of CPU time reported by GAMS and spent in the main program, while the second time columns show the time obtained by the timing function. Note that there is a significant difference between these two times requirements, presumably due to the system calls and processing time for GAMS. Because some CPU times are very small, an average of 10 runs is presented.

### 8.3 Conclusions from numerical experiments

All the test problems were solved without significant numerical difficulties. For sufficiently small tolerances the solution furnished approaches the optimal solution. As in single-level optimization appropriate choice of tolerances is necessary for computational efficiency and accuracy of solutions. Moreover, the computational requirements are greatly dependent on the problem structure.

The KKT-based heuristic for the lower bound, when applicable, greatly reduces the number of iterations needed, and moreover since it gives a tighter bound, the upper bounding procedure is less likely to produce points far away from (absolutely) feasible points. On the other hand, the cost per iteration is higher because of the complementarity conditions.

As expected, when no branching is performed, the number of iterations is smaller, but the cost per iteration is higher. For the test set considered, there is no clear advantage of one branching heuristic over the other, which indicates that the optimal branching heuristic depends on the problem. As a general guideline for problems that are solved in few iterations no branching is advantageous. For larger problems, the elaborate branching to $2n_x + 1$ nodes is expected to outperform the other heuristics because no logical constraints are needed. Finally, for any given problem, typically either the elaborate branching or no branching will outperform the regular branching.

## 9 Conclusions and future work

An algorithm for the global solution of bilevel programs involving nonconvex functions was presented. The novelty is that nonconvexity in the inner program is permitted and a guaranteed global solution is obtained. Finite termination of the algorithm to an $\varepsilon$-optimal solution is proved. An implementation is described and tested on a number of original and literature test problems.

It would be interesting to consider the automatic generation of problems with difficult properties. Calamai and Vicente [12–14] have proposed such a method for bilevel programs involving linear and quadratic functions. The extension to nonconvex functions would be of interest. More importantly, several alterations to the algorithm and implementation are conceivable. An interesting alternative to the MINLP formulation of the KKT-based lower bounds is an MPEC formulation [46,47]. Also, different global MINLP algorithms could be applied, such as outer approximation [31]. Furthermore, in the case of regular branching, the introduction of the logical constraints to the lower bounding problems could be deferred until the node size is such that the parametric upper bound to the inner problem is valid for the entire node ($X^i \subset V^k$). Preliminary experimentation shows that this is not advantageous.

There are many alternatives to the implementation of logical constraints described in Subroutine 3. For instance, the number of binary variables used can be reduced to at most $n_x$ for each logical constraint, by the use of nonconvex nonlinear constraints. For instance

$$w_j \geq 1 - \frac{\left(2x_j - v_j^{k,LO} - v_j^{k,UP}\right)^2}{\left(v_j^{k,UP} - v_j^{k,LO}\right)^2}$$

enforces $w_j = 1$ if $x_j \in [v_j^{k,LO}, v_j^{k,UP}]$. Alternatively, instead of introducing the constraint for the entire box $V^k$, it may be advantageous to introduce it for an inscribed ellipsoid using a single binary variable

$$w \geq 1 - \sum_{j=1}^{n_x} \frac{\left(2x_j - v_j^{k,LO} - v_j^{k,UP}\right)^2}{\left(v_j^{k,UP} - v_j^{k,LO}\right)^2}.$$

Finally there are alternatives to the big-M formulation such as the convex hull formulation, see e.g., [26].

The algorithm was presented in a branch-and-bound framework. A more general alternative is to embed it in a generalized branch-and-cut framework, such as the one described in [32] for nonconvex MINLPs. Also, other branching heuristics could be introduced, such as branching only on a subset of the variables (the complicating variables in some sense) or branching on the inner variables for the lower bounding problem (using $Y^i$) but not for the upper bounding problem (keeping $Y$). In single-level programs algorithms incorporating domain reduction show significant performance enhancements over pure branch-and-bound [48]. It would be interesting to also consider domain reduction for bilevel programs.

The algorithm presented here relies on the global solution of the subproblems, resulting in a nested exponential procedure, and it would be interesting to, at least partially, eliminate this nested procedure. Currently, no alternative other than global solution of the inner problem exists to obtain or confirm an upper bound, but an obvious possibility is to solve the upper bounding problem only periodically. On the other hand, the lower bounding problem could be solved locally to obtain a candidate $\bar{\mathbf{x}}$ and a convex relaxation of the lower bounding problem

(8) could be solved to obtain the lower bound. Preliminary experimentation with these ideas showed slower convergence, but for problems of large size they may be beneficial. Also, to obtain points $\mathbf{y}^k$, the solution of simpler programs than (10) is conceivable.

The ideas presented here could be extended to address some related programs such as flexibility problems [27], semi-infinite programs, and bilevel programs involving binary variables. To that extent the algorithmic ideas presented here could be combined with different approaches, such as methods based on interval-extensions [7,8,33]. Finally, extensions to equality-constrained inner programs, multiple inner problems, and the pessimistic formulation as described in Appendix A are of interest.

**Acknowledgements** This work was supported by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Army Research Office under Grant DAAD19-01-1-0566. We would like to thank Cha Kun Lee for the branch-and-bound code we derivatized and for fruitful discussions.

## Appendix A Optimistic versus pessimistic formulation

The focus of the paper is the optimistic (or weak, co-operative) formulation (1). In this appendix this formulation is compared to a more general (ambiguous) case and the other extreme of the pessimistic (or strong) formulation. The applicability and limitations of the algorithms proposed to these alternative formulations are discussed.

The basic principle of a bilevel program is that two decision makers are present, each one with their own decision variables, objective and constraints. There is a hierarchy among them: the leader (or outer program, upper-level program) decides on values for the variables **x**; then the follower (or inner program, lower-level program) decides on values for the variables **y**. The resulting general formulation [4,18] is given by

$$\begin{aligned}
&\min_{\mathbf{x}} f(\mathbf{x},\mathbf{y})\\
&\text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \le \mathbf{0}\\
&\qquad \min_{\mathbf{y}} h(\mathbf{x},\mathbf{y})\\
&\qquad\quad \text{s.t. } \mathbf{p}(\mathbf{x},\mathbf{y}) \le \mathbf{0}\\
&\qquad\qquad \mathbf{q}(\mathbf{y}) \le \mathbf{0}\\
&\mathbf{x} \in X \subset \mathbb{R}^{n_x}, \quad \mathbf{y} \in Y \subset \mathbb{R}^{n_y}.
\end{aligned} \tag{20}$$

It is well-known and easy to verify that (20) is not well-defined unless there exists at most one solution to the inner program for each $\mathbf{x} \in X$ [4,18]. The optimistic formulation (1) is one extreme to resolve the ambiguity that arises in the case of multiple solutions to the inner program. The other extreme is the pessimistic formulation:

$$\begin{aligned}
f^* = &\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y})\\
&\text{s.t. } \mathbf{y} \in \arg\max_{\mathbf{w}} f(\mathbf{x},\mathbf{w})\\
&\qquad\quad \text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{w}) \le \mathbf{0}\\
&\qquad\qquad\quad \mathbf{w} \in \arg\min_{\mathbf{z}} h(\mathbf{x},\mathbf{z})\\
&\qquad\qquad\qquad \text{s.t. } \mathbf{p}(\mathbf{x},\mathbf{z}) \le \mathbf{0}\\
&\qquad\qquad\qquad\qquad \mathbf{q}(\mathbf{z}) \le \mathbf{0}\\
&\mathbf{x} \in X \subset \mathbb{R}^{n_x}, \quad \mathbf{y},\mathbf{w},\mathbf{z} \in Y \subset \mathbb{R}^{n_y}.
\end{aligned}$$

Note that points $\bar{\mathbf{x}}$ for which no $\mathbf{w}$ satisfies the outer constraints and the optimality constraint are infeasible.

For some applications either of the two extremes is appropriate while for other applications neither is. For instance, if the leader is the head of a well-organized company and the follower the head of a division inside the company, it is reasonable to assume co-operation and thus the optimistic formulation. If, on the other hand, the follower has complete information as well as an incentive for hurting the leader, the pessimistic formulation should be used. Finally, if the follower does not have any information about the objectives and constraints of the leader neither the optimistic nor the pessimistic formulation seems adequate and the ambiguous formulation (20) is closer to the reality. Other formulations to resolve the ambiguity are conceivable.

The algorithms proposed in this article require a mathematically well-defined problem. As a consequence they are not expected to be applicable to (20). On the other hand, the extension of the algorithms to the pessimistic formulation (21) is relatively simple as described in the following. The only major change required is to replace the upper bounding problem with two subproblems. The first is to maximize the constraint violation

$$\begin{aligned} v = \max_{\mathbf{y}} \max_{i} \, & g_i(\bar{\mathbf{x}}, \mathbf{y}) \\ \text{s.t. } \mathbf{p}(\bar{\mathbf{x}}, \mathbf{y}) &\leq \mathbf{0} \\ \mathbf{q}(\mathbf{y}) &\leq \mathbf{0} \\ h(\bar{\mathbf{x}}, \mathbf{y}) &\leq h^* + \varepsilon_h \\ \mathbf{y} &\in Y. \end{aligned}$$

This program is feasible for all $\bar{\mathbf{x}}$ for which the inner program is feasible. If $v>0$ then $\bar{\mathbf{x}}$ is infeasible in (21). Otherwise $\bar{\mathbf{x}}$ is feasible in (21) and its objective value is given by the optimal objective value of

$$\begin{aligned} \max_{\mathbf{y}} \, & f(\bar{\mathbf{x}}, \mathbf{y}) \\ \text{s.t. } \mathbf{p}(\bar{\mathbf{x}}, \mathbf{y}) &\leq \mathbf{0} \\ \mathbf{q}(\mathbf{y}) &\leq \mathbf{0} \\ h(\bar{\mathbf{x}}, \mathbf{y}) &\leq h^* + \varepsilon_h \\ \mathbf{y} &\in Y. \end{aligned}$$

## References

1. Alefeld, G., Mayer, G.: Interval analysis: theory and applications. J. Comput. Appl. Math. **121**(1-2), 421–464 (2000)
2. Bank, B., Guddat, J., Klatte, D., Kummer, B., Tammer, K.: Non-Linear Parametric Optimization. Birkhäuser Verlag, Stuttgart (1983)
3. Bard, J.F.: An algorithm for solving the general bilevel programming problem. Math. Oper. Res. **8**(2), 260–272 (1983)
4. Bard, J.F.: Practical Bilevel Optimization: Algorithms and Applications. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (1998)
5. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont, Massachusetts (1999)
6. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization. Athena Scientific, Belmont, Massachusetts (1997)
7. Bhattacharjee, B., Green, W.H. Jr., Barton, P.I.: Interval methods for semi-infinite programs. Comput. Optim. Appl. **30**(1), 63–93 (2005)

8. Bhattacharjee, B., Lemonidis, P., Green, W.H. Jr., Barton, P.I.: Global solution of semi-infinite programs. Math. Program. Ser. B **103**(2), 283–307 (2005)
9. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. J. Optim. Theory Appl. **19**(2), 261–281 (1976)
10. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in constraints. Oper. Res. **21**(1), 37–44 (1973)
11. Brooke, A., Kendrick, D., Meeraus, A.: GAMS: A User's Guide. The Scientific Press, Redwood City, California (1988)
12. Calamai, P.H., Vicente, L.N.: Generating linear and linear-quadratic bilevel programming-problems. SIAM J. Sci. Comput. **14**(4), 770–782 (1993)
13. Calamai, P.H., Vicente, L.N.: Generating quadratic bilevel programming test problems. ACM Trans. Math. Software **20**(1), 103–119 (1994)
14. Calamai, P.H., Vicente, L.N., Judice, J.J.: A new technique for generating quadratic-programming test problems. Math. Program. **61**(2), 215–231 (1993)
15. Colson, B.: BIPA (Bilevel programming with approximation methods): Software guide and test problems, Technical Report CRT-2002-38. Technical Report, Centre de Recherche sur les Transports, Université de Montréal, Montréal, QC, Canada (2002)
16. Colson, B., Marcotte, P., Savard, G.: A trust-region method for nonlinear bilevel programming: Algorithm and computational experience. Comput. Optim. Appl. **30**(3), 211–227 (2005)
17. Dempe, S.: Foundations of Bilevel Programming. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht (2002)
18. Dempe, S.: Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization **52**(3), 333–359 (2003)
19. Edmunds, T.A., Bard, J.F.: Algorithms for nonlinear bilevel mathematical programs. IEEE Trans. Syst. Man Cybernet. **21**(1), 83–89 (1991)
20. Falk, J.E., Hoffman, K.: A nonconvex max-min problem. Naval Res. Logist. **24**(3), 441–450 (1997)
21. Falk, J.E., Liu, J.M.: On bilevel programming, part I: general nonlinear cases. Math.Program. **70**(1), 47–72 (1995)
22. Floudas, C.A., Gümüs, Z.H., Ierapetritou, M.G.: Global optimization in design under uncertainty: feasibility test and flexibility index problems. Indust. Eng. Chem. Res. **40**(20), 4267–4282 (2001)
23. Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM J. Optim. **18**(4), 1187–1208 (2007)
24. Fortuny-Amat, J., McCarl, B.: A representation and economic interpretation of a two-level programming problem. J. Oper. Res. Soc. **32**(9), 783–792 (1981)
25. Glover, F.: Improved linear integer programming formulations of nonlinear integer programs. Manag. Sci. **22**(4), 455–460 (1975)
26. Grossmann, I.E.: Review of nonlinear mixed-integer and disjunctive programming techniques. Optim. Eng. **3**(3), 227–252 (2002)
27. Grossmann, I.E., Halemane, K.P.: Decomposition strategy for designing flexible chemical-plants. AIChE J. **28**(4), 686–694 (1982)
28. Gümüs, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**(1), 1–31 (2001)
29. Horst, R.: Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization. J. Optim. Theo Appl. **58**(1), 11–37 (1988)
30. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer, Berlin (1996)
31. Kesavan, P., Allgor, R.J., Gatzke, E.P., Barton, P.I.: Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. Math. Program. **100**(3), 517–535 (2004)
32. Kesavan, P., Barton, P.I.: Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. Comput. Chem. Eng. **24**(2–7), 1361–1366 (2000)
33. Lemonidis, P., Barton, P.I.: Global solution of generalized semi-infinite programs (In preparation)
34. Lemonidis, P., Barton, P.I.: Interval methods for generalized semi-infinite programs. In: International Conference on Parametric Optimization and Related Topics (PARAOPT VIII), Cairo, Egypt, Nov. 27–Dec. 1 2005
35. Mitsos, A.: Man-Portable power generation devices: product design and supporting algorithms, http://yoric.mit.edu/download/Reports/MitsosThesis.pdf. PhD thesis, Massachusetts Institute of Technology (2006)
36. Mitsos, A., Barton, P.I.: Issues in the development of global optimization algorithms for bilevel programs with a nonconvex inner program. Technical Report, Massachusetts Institute of Technology (2006)

37. Mitsos, A., Barton, P.I.: A test set for bilevel programs, http://yoric.mit.edu/download/Reports/bileveltestset.pdf. Technical Report, Massachusetts Institute of Technology (2006)
38. Mitsos, A., Lemonidis, P., Lee, C.K., Barton, P.I.: Relaxation-based bounds for semi-infinite programs. SIAM J. Optim. (SIOPT), Accepted for publication (2007)
39. Moore, J.T., Bard, J.F.: The mixed integer linear bilevel programming problem. Oper. Res. **38**(5), 911–921 (1990)
40. Moore, R.: Methods and Applications of Interval Analysis. SIAM, Philadelphia, PA (1979)
41. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, New York (1999)
42. Oluwole, O.O., Bhattacharjee, B., Tolsma, J.E., Barton, P.I., Green, W.H. Jr.: Rigorous valid ranges for optimally reduced kinetic models. Combust. Flame **146**(1–2), 348–365 (2006)
43. Sahin, K.H., Ciric, A.R.: A dual temperature simulated annealing approach for solving bilevel programming problems. Comput. Chem Eng. **23**(1), 11–25 (1998)
44. Sahinidis, N., Tawarmalani, M.: BARON. http://www.gams.com/solvers/baron.pdf (2005)
45. Shimizu, K., Ishizuka, Y., Bard, J.F.: Nondifferentiable and Two-Level Mathematical Programming. Kluwer Academic Publishers, Boston (1997)
46. Stein, O., Oldenburg, J., Marquardt, W.: Continuous reformulations of discrete-continuous optimization problems. Comput. Chem. Eng. **28**(10), 1951–1966 (2004)
47. Stein, O., Still, G.: Solving semi-infinite optimization problems with interior point techniques. SIAM J. Control Optim. **42**(3), 769–788 (2003)
48. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Boston (2002)
49. Tolsma, J., Barton, P.I.: DAEPACK: an open modeling environment for legacy models. Indust. Eng. Chem. Res. **39**(6), 1826–1839 (2000)
50. Tuy, H., Migdalas, A., Hoai-Phuong, N.T.: A novel approach to bilevel nonlinear programming. J. Glob. Optim. **38**(4), 527–554 (2007)
51. Tuy, H., Migdalas, A., Varbrand, P.: A global optimization approach for the linear 2-level program. J. Glob. Optim. **3**(1), 1–23 (1993)
52. Tuy, H., Migdalas, A., Varbrand, P.: A quasi-concave minimization method for solving linear 2-level programs. J. Glob. Optim. **4**(3), 243–263 (1994)
53. Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming – A bibliography review. J. Glob. Optim. **5**(3), 291–306 (1994)
54. Williams, H.P.: Model Building in Mathematical Programming, 3rd edn. John Wiley and Sons, Chichester (1990)
55. Zuhe, S., Neumaier, A., Eiermann, M.C.: Solving minimax problems by interval-methods. BIT **30**(4), 742–751 (1990)