J Glob Optim (2014) 60:459–481
DOI 10.1007/s10898-013-0120-8

Branch-and-Sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. Part II: Convergence analysis and numerical results

Polyxeni-M. Kleniati · Claire S. Adjiman

Received: 16 June 2013 / Accepted: 15 November 2013 / Published online: 10 January 2014

Abstract In the first part of this work, we presented a global optimization algorithm, Branch-and-Sandwich, for optimistic bilevel programming problems that satisfy a regularity condition in the inner problem (Kleniati and Adjiman in J Glob Optim, 2014). The proposed approach can be interpreted as the exploration of two solution spaces (corresponding to the inner and the outer problems) using a single branch-and-bound tree, where two pairs of lower and upper bounds are computed: one for the outer optimal objective value and the other for the inner value function. In the present paper, the theoretical properties of the proposed algorithm are investigated and finite ε-convergence to a global solution of the bilevel problem is proved. Thirty-four problems from the literature are tackled successfully.

Keywords Bilevel programming · Nonconvex inner problem · Branch and bound

1 Introduction

A bilevel programming problem is a two-person, hierarchical optimization problem having a second optimization problem as part of its constraints. In this work, the following nonlinear bilevel programming problem is considered:

$$\begin{array}{ll} \min\limits_{x,y} & F(x,y) \\ \text{s.t.} & G(x,y) \leq 0, \\ & x \in X, \\ & y \in \arg\min\limits_{y \in Y}\{f(x,y) \text{ s.t. } g(x,y) \leq 0\}, \end{array} \tag{1}$$

where the n-dimensional vector x denotes the outer (leader) variables and the m-dimensional vector y denotes the inner (follower) variables. Functions $F, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, G :

P.-M. Kleniati · C. S. Adjiman (✉)
Department of Chemical Engineering, Centre for Process Systems Engineering,
Imperial College London, London SW7 2AZ, UK
e-mail: c.adjiman@imperial.ac.uk

$\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^r$ denote the outer/inner objective and outer/inner constraint functions, all assumed to be twice continuously differentiable. Finally, the host sets for the continuous variables are closed and bounded: $X = [x^{\mathrm{L}}, x^{\mathrm{U}}] \subset \mathbb{R}^n$ and $Y = [y^{\mathrm{L}}, y^{\mathrm{U}}] \subset \mathbb{R}^m$.

Notice that due to the nonconvexity of the inner problem in (1), we adopt the so-called optimistic formulation of the bilevel problem that implies some cooperation between the leader and the follower. In particular, for different globally optimal solutions of the inner problem to which the follower is indifferent, we optimize in favor of the leader's (outer) objective and constraints; hence, the outer minimization in (1) is with respect to the whole set of variables. For more details about the optimistic formulation and its alternative pessimistic formulation, the interested reader is referred to [5,9] and [17,28], respectively.

Problem (1) has been tackled extensively for specific classes of the participating functions. For instance, the linear and the nonlinear but convex bilevel programming problems have been addressed in [4,5,9,12,13,26,27]. A thorough bibliographic review can be found in [10]. On the other hand, general bilevel programming problems with a nonconvex inner problem have received much less attention due to their intrinsic difficulties and currently are only addressed in [21,23]. The proposed methods therein are for very general nonlinear bilevel problems, restricted solely by the absence of inner equality constraints.

In our work starting from Part I [16], we also address the most general bilevel programming problem for which we assume only continuity and twice differentiability of the participating functions, compactness of the host sets and the satisfaction of a constraint qualification in the inner problem for all values of x. We assume neither a special class nor convexity of the functions involved. Furthermore, we allow outer and inner equality constraints; but, we subsume those within the set of inequality constraints in both the outer and the inner problems for the sake of a simpler presentation. The use of inner equality constraints in the context of such a general formulation is, to the best of our knowledge, considered for the first time in our proposed method, Branch-and-Sandwich. Its theoretical development was introduced in the first part of this work [16]. Briefly, we remind the reader that problem (1) can be written equivalently as follows [11,16]:

$$\begin{array}{rc} \min\limits_{x,y} & F(x,y) \\ \text{s.t.} & G(x,y) \le 0, \\ & g(x,y) \le 0, \\ & f(x,y) \le w(x), \\ & x \in X,\ \ y \in Y, \end{array} \tag{2}$$

where $w(x)$ is the optimal value function of the inner (nonconvex) problem:

$$w(x) = \min_y \{ f(x,y) \text{ s.t. } g(x,y) \le 0,\, y \in Y \}. \tag{3}$$

Remark 1 Observe that by reformulation (2) and particularly by the third constraint, i.e.,

$$f(x,y) \le w(x),$$

a restriction of problem (3), i.e., an upper bound on $w(x)$, yields a relaxation of (2); consequently, a relaxation of (1). Similarly, a relaxation of (3) gives a restriction of (2) [19].

Part of the Branch-and-Sandwich algorithm is based on the property of Remark 1. For instance, we compute a constant upper bound on $w(x)$ for all the x values over the domain under consideration and then using this upper bound on $w(x)$ in formulation (2), we derive a relaxation of the overall problem. This relaxation plays the role of our proposed lower bounding problem. In order to compute a valid upper bound on $w(x)$ for all the x values, we employ a semi-infinite formulation for the proposed inner upper bounding problem which we tackle via its tractable KKT relaxation. For this reason, a regularity condition is imposed for all the x values. Then, it is possible to employ the inner KKT conditions where necessary, e.g., in the inner upper bounding problem and in the overall lower bounding problem.

In the present article, we focus on analyzing the theoretical properties of the algorithm and testing its application on suitable numerical test problems. In particular, we prove its finite ε-convergence to a global solution of the bilevel problem and we present the numerical results from its application to thirty-four test problems from the literature. We also illustrate in detail the step-by-step application of the proposed algorithm on two test problems.

The paper is organized as follows. In Sect. 2, we recapitulate all the necessary notation introduced in [16]. In Sect. 3, we present a complete analysis of the convergence properties of the Branch-and-Sandwich algorithm along with our main convergence result. An additional theoretical result regarding the exhaustiveness of our partitioning scheme follows in Sect. 4. Illustrative examples and numerical results are presented in Sect. 5. Concluding remarks are discussed in Sect. 6. Finally, the Appendix complements Sect. 5.

2 Notation

The following assumptions, definitions and nomenclature are used throughout the paper. At the end of this section, we also provide a brief statement of the proposed algorithm.

2.1 Assumptions

In addition to common assumptions, such as continuity, twice differentiability of the participating functions and compactness of the host sets, we also make the assumption below.

Assumption 1 A constraint qualification holds for the inner problem (3) for all values of x.

2.2 Definitions and nomenclature

In this work, our aim is to compute ε-optimal solutions as defined below.

Definition 1 (ε-*optimal solution* [21]) Let ε_F and ε_f be given and fixed optimality tolerances for the outer and the inner problems, respectively. Then, a pair $(x^*, y^*) \in X \times Y$ is called an ε-feasible solution of problem (1) if it satisfies the constraints of the inner and outer problems, as well as ε_f-optimality in the inner problem:

$$G(x^*, y^*) \leq 0, \tag{4}$$

$$g(x^*, y^*) \leq 0, \tag{5}$$

$$f(x^*, y^*) \leq w(x^*) + \varepsilon_f. \tag{6}$$

An ε-feasible point is called ε-optimal if it satisfies ε_F-optimality in the outer problem:

$$F(x^*, y^*) \leq F^* + \varepsilon_F, \tag{7}$$

where F^* denotes the outer optimal objective value, i.e., the optimal objective value of the bilevel problem (1) and of its equivalent single-level problem (2).

Definition 1 implies that we apply ε_f-optimality in the inner problem throughout the paper. Next, we remind the reader of all essential definitions relevant to our branching and bounding schemes, while a full exposition can be found in Part I [16].

Definition 2 (*Node*) A node k represents (sub)domain

$$X^{(k)} \times Y^{(k)} \subseteq X \times Y.$$

The root node is the node with $k = 1$ and corresponds to the whole domain $X \times Y$.

At every node k we formulate bounding problems, which we sometimes modify to obtain tractable approximations. All the bounds resulting from the (original or approximate) formulations that we consider at a node k are summarized in Table 1. The definitions of these bounds can be found in [16, Sect. 4].

Based on the bounds computed at a node k, we may decide to fully fathom or outer fathom the node (cf. Sect. 4.6 in [16]). If the node is not fully fathomed, then we need to continue exploring it further via branching. Thus, consider that we branch at node k using bisection (cf. Sect. 4.2 in [16]). This results in two child nodes, e.g., k_1 and k_2. The replacement of the old node k by the new nodes k_1 and k_2 in the tree is managed using appropriate lists of nodes. All the lists of nodes that we use in the Branch-and-Sandwich algorithm are summarized in Tables 2 and 3 and a detailed description can be found in [16, Sect. 4.1]. In Table 3, note that every independent list $\mathscr{L}^p$, $p \in P$, consists of a collection of *sublists*:

$$\mathscr{L}^p = \left\{\mathscr{L}_1^p, \ldots, \mathscr{L}_{s_p}^p\right\}, \tag{8}$$

Table 1 Summary of bounds

Description	Symbol	Problem name [16]
(Nonconvex) inner lower bound	$f^{(k),\mathrm{L}}$	(ILB)
(Convex) relaxed inner lower bound	$\underline{f}^{(k)}$	(RILB)
(Nonconvex) inner upper bound	$f^{(k),\mathrm{U}}$	(IUB)
(Nonconvex) relaxed inner upper bound	$\bar{f}^{(k)}$	(RIUB)
(Nonconvex) inner subproblem at given x	$w^{(k)}(x)$	(ISP)
(Convex) relaxed inner subproblem at given x	$\underline{w}^{(k)}(x)$	(RISP)
(Nonconvex) outer lower bound	$\underline{F}^{(k)}$	(LB)
(Nonconvex) outer upper bound	$\bar{F}^{(k)}$	(UB)

Table 2 Summary of core lists of nodes. $\mathscr{L} \cap \mathscr{L}_{\mathrm{In}} = \emptyset$

Description	Symbol	Comment
List of open nodes w.r.t. the overall problem	$\mathscr{L}$	
List of open nodes w.r.t. the inner problem only	$\mathscr{L}_{\mathrm{In}}$	Also called list of outer-fathomed nodes

Table 3 Summary of auxiliary (pairwise disjoint) lists of nodes. $\{\mathscr{X}_p \subseteq X : p \in P\}$ is a *partition* of X

Description	Symbol	Comment
Independent list for partition set $\mathscr{X}_1$	$\mathscr{L}^1$	Contains nodes in $\mathscr{L} \cup \mathscr{L}_{\text{In}}$ corresponding to $\mathscr{X}_1 \times Y$
$\vdots$	$\vdots$	
Independent list for partition set $\mathscr{X}_p$	$\mathscr{L}^p$	Contains nodes in $\mathscr{L} \cup \mathscr{L}_{\text{In}}$ corresponding to $\mathscr{X}_p \times Y$

and has an associated best inner upper bound:

$$f^{\text{UB},p} = \max \left\{ \min_{j \in \mathscr{L}_1^p} \left\{ \bar{f}^{(j)} \right\}, \ldots, \min_{j \in \mathscr{L}_{s_p}^p} \left\{ \bar{f}^{(j)} \right\} \right\}. \tag{9}$$

Remark 2 A node that has not yet been fully fathomed belongs either to list $\mathscr{L}$ or to list $\mathscr{L}_{\text{In}}$. It must also belong to an independent list $\mathscr{L}^p$ for some $p \in P$, i.e., there must exist at least one sublist $\mathscr{L}_s^p \in \mathscr{L}^p$ such that $k \in \mathscr{L}_s^p$. For simplicity, we use the shorthand notation $k \in \mathscr{L}^p$. As a result, we have

$$\exists p \in P : k \in (\mathscr{L} \cup \mathscr{L}_{\text{In}}) \cap \mathscr{L}_p \tag{10}$$

for any node k in the branch-and-bound tree that has not yet been fully fathomed.

Finally, in order to branch at the most promising nodes among all the ones that have not yet fully fathomed, i.e., the nodes in $\mathscr{L} \cup \mathscr{L}_{\text{In}}$, we apply the following selection rule.

Definition 3 (*Selection operation* [16, Sect. 4.8]) The selection rule of the Branch-and-Sandwich algorithm is:

(i) find a node in $\mathscr{L}$ with lowest overall lower bound: $k^{\text{LB}} = \arg\min_{j \in \mathscr{L}} \{\underline{F}^{(j)}\}$;
(ii) find the corresponding $\mathscr{X}_p$ subdomain, $p \in P$, such that $k^{\text{LB}} \in \mathscr{L}^p$;
(iii) select a node $k \in \mathscr{L} \cap \mathscr{L}^p$ and a node $k_{\text{In}} \in \mathscr{L}_{\text{In}} \cap \mathscr{L}^p$, if non empty, based on

$$k := \arg\min_i \{\underline{f}^{(i)} \mid i := \arg\min_{j \in \mathscr{L}^p} \{l^{(j)}\}\}. \tag{ISR}$$

2.3 Algorithm

To end this section a brief statement of the Branch-and-Sandwich algorithm is given below.

Algorithm 1 Branch-and-Sandwich [16, Sect. 5]

Step 0: Initialize.
Step 1: Compute inner and outer bounds at the root node.
Step 2: If $\mathscr{L} = \emptyset$, stop; otherwise, select a list $\mathscr{L}^p$, a node $k \in \mathscr{L}^p \cap \mathscr{L}$ and, if relevant, a node $k_{\text{In}} \in \mathscr{L}^p \cap \mathscr{L}_{\text{In}}$.
Step 3: Branch on selected node(s) to create child nodes & amend the lists of nodes (list management).
Steps 4–5: Compute inner bounds at child nodes in $\mathscr{L} \cup \mathscr{L}_{\text{In}}$. Apply full fathoming, if needed.
Steps 6–7: Compute outer bounds at child nodes in $\mathscr{L}$. Apply outer fathoming, if needed. Goto Step 2.

3 Proof of convergence

A general theory concerning the convergence of branch-and-bounds methods can be found in [14, 15]. In this section, we apply this theory to both branch-and-bound schemes, inner and outer, of the Branch-and-Sandwich algorithm introduced in [16] in order to set the foundations for its overall convergence, proved in Theorem 6, to an ε-optimal solution of problem (1). It is assumed throughout that an exhaustive partitioning scheme is used. In Sect. 4, we show that this is true for the partitioning scheme used in the algorithm. We first recapitulate some relevant definitions used in our theoretical results.

Definition 4 (*Node diameter*) The diameter $\delta(k)$ of node k (measured by the Euclidean distance) is:

$$\begin{aligned}\delta(k) &= \{\max \|t' - t''\|_2 : t', t'' \in X^{(k)} \times Y^{(k)}\}\\ &= \sqrt{(x_1^{\mathrm{U}} - x_1^{\mathrm{L}})^2 + \cdots + (x_n^{\mathrm{U}} - x_n^{\mathrm{L}})^2 + (y_1^{\mathrm{U}} - y_1^{\mathrm{L}})^2 + \cdots + (y_m^{\mathrm{U}} - y_m^{\mathrm{L}})^2}.\end{aligned}$$

Definition 5 [14, Def. 2.4.] Let $\{k_q\}, q = 0, 1, \ldots$, be an infinite decreasing nested sequence of nodes such that any k_{q+1} is a child of k_q via a certain subdivision process. If the subdivision process is exhaustive, then

$$\lim_{q\to\infty} \delta(k_q) = 0, \tag{11}$$

or

$$\lim_{q\to\infty} k_q = \bigcap_q k_q = \{(\bar{x}, \bar{y})\},\ (\bar{x}, \bar{y}) \in X \times Y. \tag{12}$$

Definition 6 [15, Def. IV.8.] The "fathom-by-infeasibility" rule is called certain in the limit if every infinite decreasing nested sequence $\{k_q\}$ of successively refined nodes converges to a feasible singleton.

Definition 7 [15, Def. IV.4.] A bounding scheme is called consistent if at every iteration any unfathomed node can be further refined and, if any infinite decreasing sequence $\{k_q\}$ of successively refined nodes satisfies:

$$\lim_{q\to\infty} (F^{\mathrm{UB}} - \underline{F}^{(k_q)}) = 0$$

or the more practical condition below:

$$\lim_{q\to\infty} (\bar{F}^{(k_q)} - \underline{F}^{(k_q)}) = 0. \tag{13}$$

Definition 8 [15, Def. IV.6.] A selection operation is said to be bound improving if, at least each time after a finite number of iterations, at least one node where the actual lower bound is attained is selected for further partition.

Remark 3 In view of Definition 5 and Theorem 7 in the next section (cf. Theorem 1 in [16]), any infinite decreasing sequence of successively refined nodes $\{k_q\}$ in the Branch-and-Sandwich method converges to a singleton as in (12). For the sequence of associated lists $\mathscr{L}^{p_q}$ such that $k_q \in \mathscr{L}^{p_q}$, we also have that $\mathscr{X}_{p_q} \subset X$ converges to a singleton in X in view of the X exhaustive subdivision process (bisection) alone, i.e.,:

$$\lim_{q\to\infty} \mathscr{X}_{p_q} = \{\bar{x}\}, \quad \bar{x} \in X. \tag{14}$$

3.1 Inner convergence properties

Theorem 1 *Let $\{k_q\}$ be an infinite decreasing nested sequence of refined nodes. For the sequence of associated lists $\mathscr{L}^{p_q}$ such that $k_q \in \mathscr{L}^{p_q}$, we have that $\mathscr{X}_{p_q} \subset X$ converges to a singleton in X as in* (14). *Then, for the associated sequence of best inner upper bounds* (9) *we have:*

$$\lim_{q \to \infty} f^{\mathrm{UB},p_q} = w(\bar{x}).$$

Proof In the limit, i.e., as q goes to infinity, the independent list $\mathscr{L}^{p_q}$ covers a single x value, as opposed to an X subdomain. This implies that no matter how much Y subdomain it may still cover, it can include one sublist only, i.e., the sublist corresponding to that x value. Thus, $\mathscr{L}^{p_q} = \{\mathscr{L}_1^{p_q}\}$. Then, based on Eq. (9), the best inner upper bound of $\mathscr{L}^{p_q}$ is:

$$\lim_{q \to \infty} f^{\mathrm{UB},p_q} = \min_{j \in \mathscr{L}_1^{p_q}} \{\bar{f}^{(j)}\}, \tag{15}$$

where each $\bar{f}^{(j)}$ is obtained by solving a KKT relaxation (RIUB) of the inner upper bounding problem (IUB). Notice that as soon as there is only one distinct inner objective value at a stationary point over each subset $Y^{(j)}$, $j \in \mathscr{L}^{p_q}$, the relaxation (RIUB) yields an exact solution of (IUB) over $Y^{(j)}$. Thus,

$$\bar{f}^{(j)} = \max_{x \in \{\bar{x}\}} \min_{y \in Y^{(j)}} \{f(x, y) \text{ s.t. } g(x, y) \leq 0\}. \tag{16}$$

Combining (15) with (16), we have that:

$$\lim_{q \to \infty} f^{\mathrm{UB},p_q} = \min_{j \in \mathscr{L}_1^{p_q}} \{\bar{f}^{(j)}\} = \min_{j \in \mathscr{L}_1^{p_q}} \min_{y \in Y^{(j)}} \{f(\bar{x}, y) \text{ s.t. } g(\bar{x}, y) \leq 0\}.$$

Finally, by using $\mathscr{Y}_{p_q} = \bigcup_{j \in \mathscr{L}_1^{p_q}} Y^{(j)}$, we can conclude that:

$$\lim_{q \to \infty} f^{\mathrm{UB},p_q} = \min_{y \in \mathscr{Y}_{p_q}} \{f(\bar{x}, y) \text{ s.t. } g(\bar{x}, y) \leq 0\} = w(\bar{x}),$$

where the last equality holds by Lemma 2 in [16]. □

Corollary 1 *The lowest inner lower bound in $\mathscr{L}^{p_q}$ is convergent to $w(\bar{x})$ as $\mathscr{X}_{p_q}$ converges to singleton $\bar{x}$.*

Theorem 2 *The inner branch-and-bound scheme has the following properties:*

(i) a consistent bounding scheme;
(ii) a bound-improving selection operation;
(iii) it is finite ε_f-convergent.[1]

Proof (i) By Corollary 1; (ii) the inner selection rule (ISR) is bound-improving; (iii) by (i) and (ii) based on [15, Th. IV.3.]. □

[1] Given a convergence tolerance $\varepsilon > 0$, a procedure is said to be finite ε-convergent if it converges after a finite number of steps to an ε-optimal solution of the problem being solved [7, p. 291].

3.2 Outer convergence properties

Recall by Definition 1 that we are concerned with computing ε-optimal solutions in the overall problem, which imply ε_f-optimality in the inner problem. For this reason, let us define the set below:

$$O_{\varepsilon_f}(x) = \{y \mid y \in Y(x),\, f(x, y) \leq w(x) + \varepsilon_f\} \tag{17}$$

as the set of ε_f-optimal solutions to the inner problem.

Theorem 3 *The fathoming rule by outer infeasibility of the Branch-and-Sandwich algorithm is certain in the limit.*

Proof We prove the theorem by contradiction. Assume that some infinite sequence $\{k_q\}$ converges to an infeasible point $(\bar{x}, \bar{y})$ in (1). There is no need to address infeasibility of $(\bar{x}, \bar{y})$ with respect to the inner and outer constraints since these constraints are included intact in the outer lower bounding problem $\underline{F}^{(k_q)}$. Thus, assume that $(\bar{x}, \bar{y})$ is infeasible with respect to ε_f-global optimality in the inner problem, i.e., $\bar{y} \notin O_{\varepsilon_f}(\bar{x})$. By feasibility of $(\bar{x}, \bar{y})$ in $\underline{F}^{(k_q)}$, we have:

$$f(\bar{x}, \bar{y}) \leq f^{\mathrm{UB}, p_q}. \tag{18}$$

However, Theorem 1 ensures that in the limit, the inner upper bound is as tight as possible:

$$\lim_{q \to \infty} f^{\mathrm{UB}, p_q} = w(\bar{x}). \tag{19}$$

In other words, there exists some finite q', for which the inner upper bound meets the specified tolerance, and the following holds:

$$|f^{\mathrm{UB}, p_q} - w(\bar{x})| \leq \varepsilon_f \;\forall\, q \geq q', \tag{20}$$

or, from (18):

$$f(\bar{x}, \bar{y}) \leq w(\bar{x}) + \varepsilon_f.$$

But, $\bar{y} \notin O_{\varepsilon_f}(\bar{x})$ implies that

$$f(\bar{x}, \bar{y}) > w(\bar{x}) + \varepsilon_f, \tag{21}$$

which is a contradiction. Hence, $\underline{F}^{(k_q)}$ is infeasible for $q \geq q'$ and a node containing $(\bar{x}, \bar{y})$ will be fathomed at iteration q'. As a result, no infinite decreasing sequence $\{k_q\}$ of refined regions can converge to an infeasible point. □

Remark 4 Notice that a decreasing sequence of successively refined nodes becomes infeasible as soon as it contains inner suboptimal KKT points only and the inner upper bound has met the specified tolerance. In particular, let

$$\Omega_{\mathrm{Global}} := \{(x, y, \mu, \lambda, \nu) \in \Omega_{\mathrm{KKT}} \mid y \in O_{\varepsilon_f}(x)\},$$

and consider a node k_q at which there only exist $(x, y, \mu, \lambda, \nu) \in \mathbb{R}^{n+3m+r}$ such that

$$(x, y) \in X^{(k_q)} \times Y^{(k_q)} \text{ and } (x, y, \mu, \lambda, \nu) \in \Omega_{\mathrm{KKT}} \setminus \Omega_{\mathrm{Global}}.$$

Then, Eq. (20) implies that

$$f(x, y) > f^{\mathrm{UB}, p_{q'}} \text{ for all } (x, y) \in X^{(k_{q'})} \times Y^{(k_{q'})},\, (x, y, \mu, \lambda, \nu) \in \Omega_{\mathrm{KKT}} \setminus \Omega_{\mathrm{Global}},$$

which makes $\underline{F}^{(k_{q'})}$ infeasible.

Finally, we can show that the Branch-and-Sandwich algorithm is convergent based on [15, Th. IV.3.]. In particular, we show that the bounding scheme is *consistent* (cf. Definition 7) and the selection operation is *bound improving* (cf. Definition 8).

Theorem 4 *The bounding scheme of the Branch-and-Sandwich algorithm is consistent.*

Proof The first requirement of Definition 7 is satisfied by construction of the algorithm. To prove that the second requirement, i.e., condition (13), is also satisfied, note first that a decreasing sequence $\{k_q\}$ of successively refined nodes converges to a feasible limit set based on Theorem 3. Then, notice that by exhaustive partitioning of domain X, in the limit where e.g., $x = \bar{x}$, the outer lower and the outer upper bounding problems are identical with the exception of the constraint on the inner objective function:

$$f(\bar{x}, y) \leq f^{\mathrm{UB},p_q}; \tag{22}$$

$$f(\bar{x}, y) \leq w(\bar{x}) + \varepsilon_f. \tag{23}$$

By Theorem 1, we know that there exists some finite q', for which:

$$|f^{\mathrm{UB},p_q} - w(\bar{x})| \leq \varepsilon_f \ \forall\, q \geq q'.$$

Thus, constraint (22) in the lower-bounding problem is written equivalently:

$$f(\bar{x}, y) \leq w(\bar{x}) + \varepsilon_f, \tag{24}$$

namely it is the same as constraint (23) in the upper-bounding problem, making both problems identical. This, along with Theorem 3 and the partitioning of the Y space, implies condition (13). □

Theorem 5 *The selection operation of the Branch-and-Sandwich algorithm (cf. Definition 3) is bound improving.*

Proof From Definition 3, recall that (i) $k^{\mathrm{LB}} \in \mathscr{L}$ is such that $k^{\mathrm{LB}} = \arg\min_{j \in \mathscr{L}}\{\underline{F}^{(j)}\}$, (ii) $\mathscr{X}_p$, $p \in P$, is such that $k^{\mathrm{LB}} \in \mathscr{L}^p$, and (iii) $k \in \mathscr{L}$ is such that $k \in \mathscr{L}^p$ satisfying (ISR). If $k = k^{\mathrm{LB}}$ then the node with the lowest overall bound is selected and our proof is complete. Let us now examine the case where $k \neq k^{\mathrm{LB}}$. Observe that the pair $(l^{(k)}, \underline{f}^{(k)})$ dominates the pair $(l^{(k^{\mathrm{LB}})}, \underline{f}^{(k^{\mathrm{LB}})})$, since otherwise node k^{LB} would have been selected. In other words, we have that either $l^{(k)} < l^{(k^{\mathrm{LB}})}$ or $\underline{f}^{(k)} < \underline{f}^{(k^{\mathrm{LB}})} \leq f^{\mathrm{UB},p}$. The branching strategy (cf. [16, Def. 6]) and the inner bounding scheme imply that the values $l^{(k)}$, $\underline{f}^{(k)}$ are non-decreasing, while Lemma 1 in [16] implies that $f^{\mathrm{UB},p}$ is non-increasing, over refined host sets. Hence, node k^{LB} will eventually be either fathomed by inner value dominance, i.e., $\underline{f}^{(k^{\mathrm{LB}})} > f^{\mathrm{UB},p}$, in which case another node will hold the lowest overall lower bound and then the same arguments apply, or selected for exploration based on $l^{(k^{\mathrm{LB}})} \leq l^{(k)}$ and/or $\underline{f}^{(k^{\mathrm{LB}})} \leq \underline{f}^{(k)}$ being satisfied. The node with the lowest overall lower bound is guaranteed to be selected after finitely many iterations due to the finite ε_f-convergence of the inner branch-and-bound scheme shown in Theorem 2. □

Our main result regarding the convergence of Branch-and-Sandwich is now stated.

Theorem 6 *The Branch-and-Sandwich algorithm is ε-convergent, such that at termination we have $F^{\mathrm{UB}} - F^{\mathrm{LB}} \leq \varepsilon_F$, where $F^{\mathrm{LB}} = \min_{k \in \mathscr{L}} \underline{F}^{(k)}$, and the incumbent solution $(x^{\mathrm{UB}}, y^{\mathrm{UB}})$ is an ε-optimal solution of problem (1).*

Proof The finite ε-convergence property of Branch-and-Sandwich follows from Theorem IV.3. in [15] and Theorems 4–5. Furthermore, termination of the algorithm implies that $F^{\text{LB}} \geq F^{\text{UB}} - \varepsilon_F$, i.e., condition (7) of outer ε_F-optimality is true. Using Theorem 3 in [16], this implies that point $(x^{\text{UB}}, y^{\text{UB}})$ is ε-optimal in (1) (cf. Definition 1). □

4 Proof of exhaustiveness

This section concerns the exhaustiveness of our subdivision process (c.f. Branch-and-Sandwich subdivision process [16, Sect. 4.2, Def. 6]) and complements Sect. 3 above. Prior to our proof of exhaustiveness, we state below a well-known preliminary lemma.

Lemma 1 (Basic Subdivision Lemma [24, Lemma 5.1]) *Let* $\{k_q\}$, $q = 0, 1, \ldots,$ *be an infinite decreasing nested sequence of nodes such that any* k_{q+1} *is a child of* k_q *via a certain subdivision process. Assume that:*

(i) for infinitely many q *the subdivision of* k_q *is a bisection;*
(ii) there exists a constant $\rho \in (0, 1)$ *such that for every* q*:*

$$\delta(k_{q+1}) \leq \rho\delta(k_q).$$

Then, the subdivision process is exhaustive.

Theorem 7 (Theorem 1 in [16]) *The subdivision process of the Branch-and-Sandwich algorithm is exhaustive.*

Proof To prove our claim, we employ Lemma 1. In particular, requirement (ii) of this lemma is satisfied due to bisection (cf. [16, Def. 4]). Thus, it suffices to show requirement (i) of the lemma, i.e., that for infinitely many q the subdivision of k_q is a bisection. Let k_q express domain $X^{(k_q)} \times Y^{(k_q)}$. Without loss of generality assume that (i) k_0 is the root node; (ii) the set of variables is ordered as follows:

$$(y_1, \ldots, y_m, x_1, \ldots, x_n);$$

and (iii) all edges have equal length at k_0. This can be imposed with appropriate variable scaling if necessary. Then, for $q = 1$ to $q = m$, variables $y_1, \ldots, y_m$ are selected for branching, resulting in one independent list:

$$\mathscr{L}^{p_0} = \left\{ X^{(k_0)} \times \frac{Y^{(k_0)}}{2^m}, \ldots, X^{(k_0)} \times \frac{Y^{(k_0)}}{2^m} \right\},$$

where $p_0 \in P$. Next, for $q = m + 1$, x_1 is selected for branching at all the nodes of list $\mathscr{L}^{p_0}$, resulting in two independent lists:

$$\mathscr{L}^{p_1} = \left\{ X^{(k_1)} \times \frac{Y^{(k_0)}}{2^m}, \ldots, X^{(k_1)} \times \frac{Y^{(k_0)}}{2^m} \right\},$$

$$\mathscr{L}^{p_1'} = \left\{ X^{(k_1')} \times \frac{Y^{(k_0)}}{2^m}, \ldots, X^{(k_1')} \times \frac{Y^{(k_0)}}{2^m} \right\},$$

where $X^{(k_1)}$ and $X^{(k_1')}$ is a partition of $X^{(k_0)}$ using bisection. Consider next list $\mathscr{L}^{p_1}$ and keep branching on the x-variables. At $q = m + n$, list $\mathscr{L}^{p_1}$ has been replaced by the list $\mathscr{L}^{p_{m+n}}$ covering smaller X subdomain:

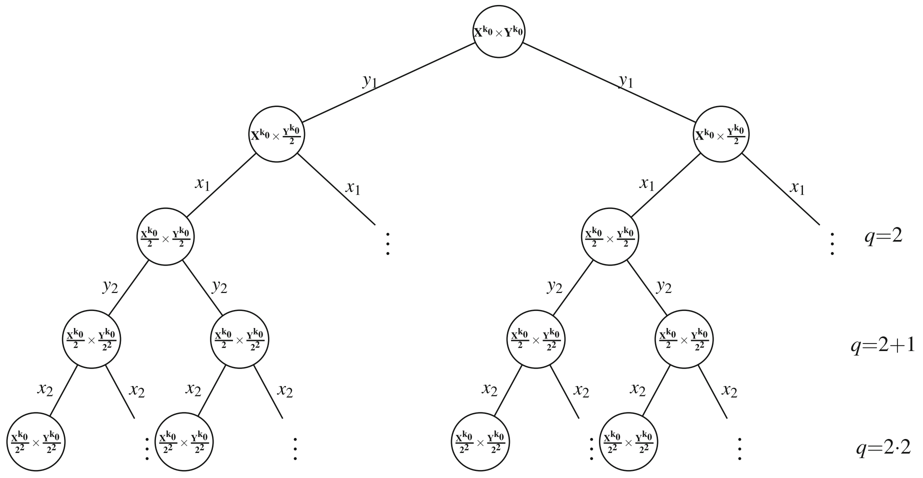


Fig. 1 Subdivision process with variable ordering $(y_1, x_1, y_2, x_2, \ldots)$

$$\mathscr{L}^{p_{m+n}} = \left\{ X^{(k_{m+n})} \times \frac{Y^{(k_0)}}{2^m}, \ldots, X^{(k_{m+n})} \times \frac{Y^{(k_0)}}{2^m} \right\},$$

where $X^{(k_{m+n})} \subset X^{(k_{m+n-1})} \ldots \subset X^{(k_1)} \subset X^{(k_0)}$ and $X^{(k_{m+n})} = X^{(k_0)}/2^n$. Variables $y_1, \ldots, y_m$ are then selected again for branching, resulting in further subdivision of the Y space, i.e.,

$$\mathscr{L}^{p_{m+n}} = \left\{ X^{(k_{m+n})} \times \frac{Y^{(k_0)}}{2^{2m}}, \ldots, X^{(k_{m+n})} \times \frac{Y^{(k_0)}}{2^{2m}} \right\}.$$

At the next step, namely after $m+n+1$ steps in total, x_1 is selected again for branching at all nodes of list $\mathscr{L}^{p_{m+n}}$ and so forth. Hence, after every $m+n$ steps, k_q is subdivided with respect to the X domain using bisection n-times and with respect to the Y domain using bisection m-times. As q goes to infinity, this results in infinitely many bisections for the sequence $\{k_q\}$. □

Remark 5 Observe that if the variables are ordered as follows:

$$\begin{array}{ll} (y_1, x_1, y_2, x_2, \ldots, y_m, x_m, x_{m+1} \ldots, x_n) & \text{if } m < n, \\ (y_1, x_1, y_2, x_2, \ldots, y_m, x_m) & \text{if } m = n, \\ (y_1, x_1, y_2, x_2, \ldots, y_n, x_n, y_{n+1}, \ldots, y_m) & \text{if } m > n, \end{array}$$

then k_q is subdivided with respect to the $X \times Y$ domain using bisection after every two steps (e.g., cf. Fig. 1).

5 Numerical results and examples

In this section, we report the application of the Branch-and-Sandwich algorithm to 34 literature problems. In the first two columns of Table 4 the problem number, the original name and the source of all the instances on which Branch-and-Sandwich was tested are shown.

Table 4 Problem instances and their statistics

No.	Example in [source]	Example in this work (Part I or Part II)	NC Inner Problem?	#Outer var. (n)	#Inner var. (m)	#Outer con. (o)	#Inner con. (r)
1	4.2 [19]		Yes	0	1	0	0
2	3.1 [20]		No	0	1	0	0
3	3.2 [20]		No	0	1	1	0
4	3.3 [20]		Yes	0	1	0	1
5	3.4 [20]		Yes	0	1	0	0
6	3.5 [20]	1 [16]	Yes	0	1	0	0
7	3.6 [20]		Yes	0	1	0	0
8	3.7 [20]		No	0	1	1	0
9	3.8 [20]		No	1	1	2	0
10	3.9 [20]		Yes	1	1	1	0
11	3.10 [20]	2	Yes	1	1	0	0
12	3.11 [20]		Yes	1	1	0	0
13	3.12 [20]		Yes	1	1	0	0
14	3.13 [20]		Yes	1	1	0	0
15		3	Yes	1	1	0	0
16	3.14 [20]		Yes	1	1	0	0
17	3.15 [20]	1	Yes	1	1	0	0
18	3.16 [20]		Yes	1	1	0	0
19	3.17 [20]		Yes	1	1	0	0
20	3.18 [20]		Yes	1	1	0	0
21	3.19 [20]		Yes	1	1	0	0
22	3.20 [20]		Yes	1	1	0	0
23	3.21 [20]	2 [16]	Yes	1	1	0	0
24	3.22 [20]		Yes	1	1	0	1
25	3.24 [20]		Yes	1	1	0	0
26	4.1 [18]		Yes	1	1	0	0
27	5.6 [8]		Yes	1	1	0	2
28	7.1.1 [5]		Yes	1	1	0	3
29	1 [25]		No	1	1	0	3
30	4.5 [20]		Yes	1	2	0	2
31	2 [6]		No	2	2	0	3
32	1 [6]		No	2	3	0	3
33	3.26 [20]		Yes	2	3	3	0
34		4	Yes	5	5	3	1

These include a few literature problems from [5,6,8,18,19,25] and all the test problems from [20] for which Assumption 1 is satisfied.[2] The two problem instances that appear to have no source in Table 4, i.e., problem No. 15 and problem No. 34, are variants of Example 3.13 and Example 3.28, respectively, from [20], and are stated in the Appendix. Both satisfy

[2] Branch-and-Sandwich was applicable to all problem instances in [20] except for Examples 3.23, 3.25, 3.27 and 3.28.

Assumption 1; the former thanks to the Abadie constraint qualification and the latter thanks to the linear/concave constraint qualification. While all problems are solved in this paper, in the third column of Table 4, we state the example numbers for the few problems selected for a more detailed discussion.

Next, from left to right, starting with the fourth column of Table 4, we report whether or not the inner problem is nonconvex (NC), the dimension of the outer variable vector, the dimension of the inner variable vector, the number of constraints in the outer problem, and the number of constraints in the inner problem.

In order to tackle problem No. 30, we first reformulated it by replacing variable $y_2 \in [0.1, 10]$ with a new variable $y_2' = 1/y_2$ with the consistent bounds. This transformation was proposed in [20, Example 4.5] to ensure the satisfiability of the Abadie constraint qualification since it yields linear constraints for the inner problem. Therefore, Assumption 1 is satisfied and our algorithm applies.

The global solution was identified successfully for all problem instances and the numerical results are summarized in Table 5. In particular, next to each problem number we report the best known optimal objective value as found in the literature (cf. Table 4). In the third column, we report the optimal objective value obtained with the Branch-and-Sandwich algorithm. This is the incumbent value F^{UB} obtained at termination; notice that the corresponding lowest outer lower bound computed at termination was within $\varepsilon_F = 10^{-3}$ absolute difference for all problem instances except for No. 19–20, where $\varepsilon_F = 10^{-1}$. The inner objective tolerance was $\varepsilon_f = 10^{-5}$ for all problem instances. The fourth column shows the number of outer upper bounding problems solved before the optimal solution was computed for the first time (N_{opt}). The fifth and sixth columns include the total number of outer-upper and outer-lower bounding problems, (#UBD and #LBD) respectively, solved for each problem instance. The seventh column contains the number of nodes required for termination for the given outer and inner objective tolerances. Finally, the last three columns report the metrics N_{opt}, #UBD and #LBD for the Mitsos et al. approach [21].

In problems No. 1–10, 12 and 18, the Branch-and-Sandwich algorithm achieved higher accuracy in the outer objective value than required to meet the convergence tolerance, e.g., $\varepsilon_F = 10^{-5}$, for the same number of nodes. There were also many instances for which convergence was achieved at the root node, such as problems No. 1–3, 5, 8–9, 12, 18, 25–29, 31–34. For instance, consider problem 12, which despite being a variation of No. 11, requires one node only for termination. For this example, the lower bounding problem computes the optimal solution of the bilevel problem at the root node; then, the convex relaxation of the inner problem for the obtained x yields the actual optimal objective value of the inner problem for this x value and, as a result, a feasible outer upper bounding problem. On the other hand, No. 19–20 were the slowest to converge because their convergence was dependent on eliminating KKT inner suboptimal points that were yielding an outer objective value lower than the actual optimal value. For instance, No. 20 terminated with $F^{\mathrm{UB}} = F^* = 0.25$, but with a lower bound of $F^{\mathrm{LB}} = 0.19$.

The number of nodes reported in Table 5 is based on using the proposed subdivision process of Branch-and-Sandwich (cf. [16, Def. 6]) and on solving the inner upper bounding and outer lower bounding problems, respectively (RIUB) and (LB), to global optimality, as required, using the αBB solver [1–3]. The outer upper bounding problem (UB), although nonconvex, does not need to be solved to global optimality. Hence, for all the examples tested, all (UB) subproblems were solved with the local solver MINOS [22]. Finally, the inner lower bounding problems (ILB) and the inner subproblems (ISP) were solved using the proposed convex relaxed problems (RILB) and (RISP), respectively. These problems were derived by employing the αBB convexifications techniques [2,3] and then solved with MINOS.

Table 5 Preliminary numerical results with $\varepsilon_f = 10^{-5}$ and $\varepsilon_F = 10^{-3}$ for all problems, except No. 19–20 where $\varepsilon_F = 10^{-1}$. N_{opt} is the number of outer upper bounding problems solved before the optimal solution is computed for the first time

No.	Branch-and-Sandwich method						Mitsos et al. method [21]		
	F^*	F^{UB}	N_{opt}	#UBD	#LBD	#Nodes	N_{opt}	#UBD	#LBD
1	-1	-1	1	1	1	1	–	–	–
2	1	1	1	1	1	1	1	1	3
3	∞	∞	0	0	1	1	0	1	3
4	-1	-1	2	2	3	3	1	1	3
5	1	1	1	1	1	1	1	1	3
6	0.5	0.5	6	6	11	11	1	1	3
7	-1	-1	2	2	2	3	1	1	1
8	∞	∞	0	0	1	1	0	1	3
9	0	0	1	1	1	1	2	2	2
10	-1	-1	2	2	2	3	1	1	1
11	0.5	0.5	5	5	7	11	1	1	3
12	-0.8	-0.8	1	1	1	1	1	1	1
13	0	0	3	4	11	11	14	10	17
14	-1	-1	4	7	15	27	2	2	3
15	-1	-1	4	7	15	23	–	–	–
16	0.25	0.25	5	5	10	15	1	3	7
17	0	0	2	5	8	13	1	1	3
18	-2	-2	7	7	13	19	1	1	3
19	0.1875	0.1875	4	7	35	55	2	18	37
20	0.25	0.25	13	13	25	49	3	2	3
21	-0.258	-0.259	4	4	7	11	18	14	27
22	0.3125	0.3125	5	11	26	39	3	3	5
23	0.2095	0.2094	2	2	3	3	1	2	5
24	0.2095	0.2094	2	2	3	3	8	5	11
25	-1.755	-1.755	5	5	10	11	19	14	27
26	0	0	1	1	1	1	–	–	–
27	0	0	1	1	1	1	–	–	–
28	17	17	1	1	1	1	–	–	–
29	22.5	22.5	1	1	1	1	–	–	–
30	0.193616	0.193616	2	2	2	3	1	1	1
31	1.75	1.75	1	1	1	1	–	–	–
32	29.2	29.2	1	1	1	1	–	–	–
33	-2.35	-2.35	1	1	1	1	2	2	5
34[a]	-10	-10	1	2	3	3	2[a]	2[a]	2[a]

[a] Note that for problem No. 34, our test case is a variant of Example 3.28 [20] (shown as Example 4 in "Appendix"), while the statistics in the last three columns correspond to the original Example 3.28 from [20]. The modification was introduced with the aim to satisfy the inner regularity assumption. It does not affect the optimal outer objective value or the problem size. Hence, we report the Mitsos et al. results for this test case as an indication for comparison of a medium-scale example

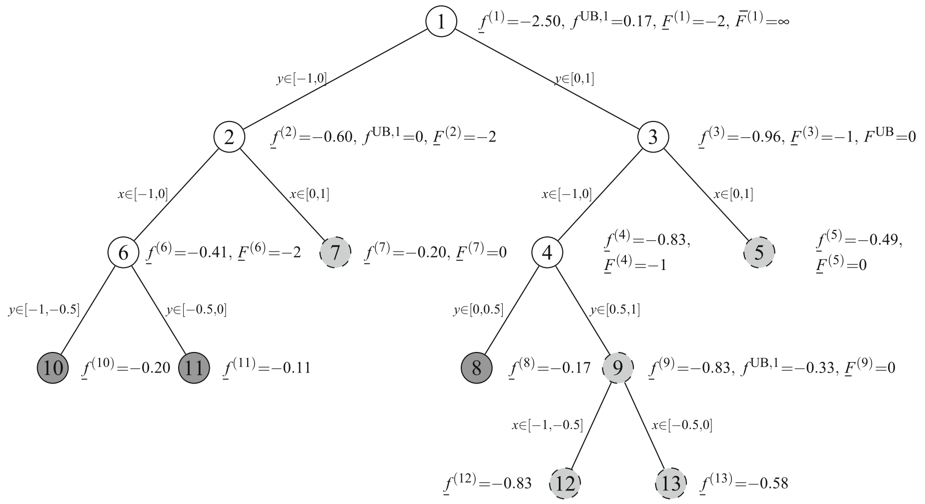
Before comparing the performance of our approach and the Mitsos et al. approach, we note that a direct comparison of performance is difficult because the formulation and size of the subproblems solved in both approaches are different. This is compounded by the fact that only some subproblems are solved to global optimality in Branch-and-Sandwich whereas all subproblems are solved to global optimality in the Mitsos et al. approach. Thus, one must be cautious in drawing conclusions from the early analysis presented here. It can be seen from Table 5 that in many cases, the numbers of subproblems in both algorithms are comparable. For example, fewer subproblems are solved with the Mistos et al. approach in problems 16, 17, 18, and fewer subproblems are solved with the Branch-and-Sandwich algorithm for problems 13, 21 and 25. These results indicate that the partitioning of the Y space can prove very beneficial. This preliminary comparison is very encouraging and motivates the further development of the Branch-and-Sandwich algorithm.

In the remainder of this section, the Branch-and-Sandwich algorithm is demonstrated in detail for two examples, problem No. 11 and problem No. 17. The computed values in the worked examples are displayed with up to two significant digits.

Example 1 (*Problem No. 17 in Table* 5, *[*20*, Example 3.15]*)

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

Example 1 has a unique optimal solution at $(x^*, y^*) = (-1, 1)$ yielding $F^* = 0$ and $f^* = -0.83$. During the course of the algorithm, 13 (RILB), 9 (RIUB), 8 (LB), 5 (RISP) and 5 (UB) problems were solved. The branch-and-bound tree for this problem is shown in Fig. 2.



1
$\underline{f}^{(1)}=-2.50,\ f^{\mathrm{UB},1}=0.17,\ \underline{F}^{(1)}=-2,\ \bar{F}^{(1)}=\infty$
$y\in[-1,0]$
$y\in[0,1]$
2
$\underline{f}^{(2)}=-0.60,\ f^{\mathrm{UB},1}=0,\ \underline{F}^{(2)}=-2$
3
$\underline{f}^{(3)}=-0.96,\ \underline{F}^{(3)}=-1,\ F^{\mathrm{UB}}=0$
$x\in[-1,0]$
$x\in[0,1]$
$x\in[-1,0]$
$x\in[0,1]$
6
$\underline{f}^{(6)}=-0.41,\ \underline{F}^{(6)}=-2$
7
$\underline{f}^{(7)}=-0.20,\ \underline{F}^{(7)}=0$
4
$\underline{f}^{(4)}=-0.83,$
$\underline{F}^{(4)}=-1$
5
$\underline{f}^{(5)}=-0.49,$
$\underline{F}^{(5)}=0$
$y\in[-1,-0.5]$
$y\in[-0.5,0]$
$y\in[0,0.5]$
$y\in[0.5,1]$
10
$\underline{f}^{(10)}=-0.20$
11
$\underline{f}^{(11)}=-0.11$
8
$\underline{f}^{(8)}=-0.17$
9
$\underline{f}^{(9)}=-0.83,\ f^{\mathrm{UB},1}=-0.33,\ \underline{F}^{(9)}=0$
$x\in[-1,-0.5]$
$x\in[-0.5,0]$
$\underline{f}^{(12)}=-0.83$
12
13
$\underline{f}^{(13)}=-0.58$

Fig. 2 Complete branch-and-bound tree for Example 1. For clarity, not all the computed values are present in the diagram. *Light-gray nodes* denote nodes that are first outer fathomed, i.e., moved to the list $\mathscr{L}_{\mathrm{In}}$, and *dark-gray nodes* denote those that are fathomed due to inner-fathoming rules. All gray nodes are eventually (fully) fathomed. Notice also that the outer-fathomed node 9 was replaced by the outer-fathomed nodes 12 and 13 within the branching framework. These two nodes were also fully fathomed in the end

Step 0: Iter $= 0$ and we set $\mathscr{L} = \mathscr{L}_{\text{In}} = \emptyset$, $F^{\text{UB}} = \infty$, $(x^{\text{UB}}, y^{\text{UB}}) = \emptyset$.

Step 1: We set $n_{\text{node}} = 1$ and compute $\underline{f}^{(1)} = -2.50$ and $\bar{f}^{(1)} = 0.17$. The latter value sets $f_X^{\text{UB}} = 0.17$. Next, we compute $\underline{F}^{(1)} = -2$ at $(x^{(1)}, y^{(1)}) = (-1, -1)$. Since the outer lower bounding is feasible, we add node 1 to the list $\mathscr{L}$ with all the computed information:

$$\mathscr{L} = \{1 : -2.50\ 0.17\ -2\ -1\ 0\ \},$$

where the last field is the level of node 1: $l^{(1)} = 0$. Also, we set $p = 1$ and

$$\mathscr{X}_1 = [-1, 1]\,, \mathscr{L}^1 = \{1\},\ f^{\text{UB},1} = f_X^{\text{UB}} = 0.17.$$

We then compute $\underline{w}(-1) = -1.50$, which yields $\bar{F}^{(1)} = \infty$.

Step 2: **Iter = 1** (since $\mathscr{L} \neq \emptyset$). We select node $1 \in \mathscr{L} \cap \mathscr{L}^1$ and remove it from $\mathscr{L}$. As a result, at this point $\mathscr{L} = \emptyset$ and $\mathscr{L}^1 = \{1\}$.

Step 3: We branch on $y = 0$, i.e., the midpoint of variable y, and create nodes:

$$\begin{aligned} 2 &:= \{(x, y) \in \mathbb{R}^2 \mid -1 \le x \le 1,\ -1 \le y \le 0\},\\ 3 &:= \{(x, y) \in \mathbb{R}^2 \mid -1 \le x \le 1,\quad 0 \le y \le 1\}. \end{aligned}$$

This results in $\mathscr{L}^1 = \{2, 3\}$ corresponding to $\mathscr{X}_1 = [-1, 1]$.

Step 4: We compute $\underline{f}^{(2)} = -0.60$ and $\underline{f}^{(3)} = -0.96$ and add both nodes to $\mathscr{L}$:

$$\begin{aligned} \mathscr{L} = \{&2 : -0.60\ 0.17\ -2\ -1\ 1\\ &3 : -0.96\ 0.17\ -2\ -1\ 1\}. \end{aligned}$$

The first and last properties, i.e., $\underline{f}^{(i)}$ and $l^{(i)}$, $i = 1, 2$, where $l^{(2)} = l^{(3)} = 1$, are set based on the new nodes 2 and 3; the other values are inherited from node 1.

Step 5: We compute $\bar{f}^{(2)} = 0$ and $\bar{f}^{(3)} = 0.17$. The former value updates $\bar{f}^{(2)}$ in $\mathscr{L}$, as well as the best inner upper bound for list $\mathscr{L}^1$: $f^{\text{UB},1} = 0$.

Step 6: We compute $\underline{F}^{(2)} = -2$ at $(-1, -1)$ and $\underline{F}^{(3)} = -1$ at $(-1, 0)$.

Step 7: For $i = 2$, we set $\bar{x} = x^{(2)} = -1$ and compute $\underline{w}^{(2)}(\bar{x}) = -0.24$ and $\underline{w}^{(3)}(\bar{x}) = -0.83$. The lowest value is given at node 3, where we compute $\bar{F}^{(3)} = 0$ at $(-1, 1)$ and, as a result, we update the incumbent value to $F^{\text{UB}} = 0$. For $i = 3$, we set $\bar{x} = x^{(3)} = -1$, but we already have computed $\underline{w}^{(2)}(\bar{x})$, $\underline{w}^{(3)}(\bar{x})$ and $\bar{F}^{(3)}$ for $\bar{x} = -1$; hence, no further computation is made. Set $n_{\text{node}} = 3$.

Step 2: **Iter = 2**. The selection rule of Definition 3 initially chooses node $2 \in \mathscr{L}$, which then points to the domain $\mathscr{X}_1$ via list $\mathscr{L}^1$, since $2 \in \mathscr{L} \cap \mathscr{L}^1$. Node $3 \in \mathscr{L} \cap \mathscr{L}^1$ is then selected due to (ISR) and removed from $\mathscr{L}$. As a result, at this point $\mathscr{L} = \{2\}$ and $\mathscr{L}^1 = \{2, 3\}$.

Step 3: We branch on $x = 0$, resulting in $\mathscr{L}^1 = \{\{2, 4\}, \{2, 5\}\}$ corresponding to $\mathscr{X}_1 = [-1, 1]$.

Step 4: We compute $\underline{f}^{(4)} = -0.83$ and $\underline{f}^{(5)} = -0.49$. Both are added to $\mathscr{L}$:

$$\begin{aligned} \mathscr{L} = \{&2 : -0.60\ 0\ \quad -2\ -1\ 1\\ &4 : -0.83\ 0.17\ -1\ -1\ 2\\ &5 : -0.49\ 0.17\ -1\ -1\ 2\}. \end{aligned}$$

Step 5: We compute $\bar{f}^{(4)} = 0$ and $\bar{f}^{(5)} = 0.17$.

Step 6: We compute $\underline{F}^{(4)} = -1$ and $\underline{F}^{(5)} = 0$, with the latter value leading to the outer fathoming of node 5, namely:

$$\begin{array}{rl} \mathscr{L} = & \{2: -0.60 \quad 0 \quad -2 \quad -1 \quad 1 \\ & \;\; 4: -0.83 \quad 0 \quad -1 \quad -1 \quad 2\}; \\ \mathscr{L}_{\text{In}} = & \{5: -0.49 \quad 0.17 \quad - \quad - \quad 2\}. \end{array}$$

Also, recall $\mathscr{L}^1 = \{\{2,4\},\{2,5\}\}$, with $f^{\text{UB},1} = 0$ and $\mathscr{X}_1 = [-1,1]$.

Step 7: For $i = 4$, we compute $\underline{w}^{(4)}(\bar{x}) = -0.83$ (we already know $\underline{w}^{(2)}(\bar{x}) = -0.24$ for $\bar{x} = -1$). The lowest value of $\underline{w}^{(j)}(\bar{x})$, $j \in \mathscr{L}^1$, being at node 4 leads to computing $\bar{F}^{(4)} = 0$ and no update of the incumbent is needed. For $i = 5$, the upper bounding procedure does not apply because 5 is no longer in $\mathscr{L}$. Set $n_{\text{node}} = 5$.

Step 2: **Iter = 3**. Node 2 is selected based on Definition 3 and removed from $\mathscr{L}$.

Step 3: At node 2, we branch on $x = 0$, resulting in $\mathscr{L}^1 = \{4,6\}$ and $\mathscr{L}^2 = \{5,7\}$ with $\mathscr{X}_1 = [-1,0]$ and $\mathscr{X}_2 = [0,1]$, respectively. The corresponding best inner upper bounds are set to $f^{\text{UB},1} = f^{\text{UB},2} = 0$.

Step 4: We compute $\underline{f}^{(6)} = -0.41$ and $\underline{f}^{(7)} = -0.20$. Both nodes are added to $\mathscr{L}$.

Step 5: We compute $\bar{f}^{(6)} = \bar{f}^{(7)} = 0$.

Step 6: We compute $\underline{F}^{(6)} = -2$ and $\underline{F}^{(7)} = 0$, with 7 being outer fathomed:

$$\begin{array}{rl} \mathscr{L} = & \{4: -0.83 \quad 0 \quad -1 \quad -1 \quad 2 \\ & \;\; 6: -0.41 \quad 0 \quad -2 \quad -1 \quad 2\}; \\ \mathscr{L}_{\text{In}} = & \{5: -0.49 \quad 0.17 \quad - \quad - \quad 2 \\ & \;\; 7: -0.20 \quad 0 \quad - \quad - \quad 2\}. \end{array}$$

Recall $\mathscr{L}^2 = \{5,7\}$, with $f^{\text{UB},2} = 0$ and $\mathscr{X}_2 = [0,1]$. This independent list can now be discarded as it no longer holds any node from $\mathscr{L}$ (cf. List-deletion fathoming rules [16, Definition 10]). Thus:

$$\begin{array}{rl} \mathscr{L} = & \{4: -0.83 \quad 0 \quad -1 \quad -1 \quad 2 \\ & \;\; 6: -0.41 \quad 0 \quad -2 \quad -1 \quad 2\}; \\ \mathscr{L}_{\text{In}} = & \emptyset. \end{array}$$

One independent list remains: $\mathscr{L}^1 = \{4,6\}$, with $f^{\text{UB},1} = 0$ and $\mathscr{X}_1 = [-1,0]$.

Step 7: For $i = 6$, we compute $\underline{w}^{(6)}(\bar{x}) = -0.24$ for $\bar{x} = -1$ and compare it with $\underline{w}^{(4)}(\bar{x}) = -0.83$; no extra computation needs to be made since the lowest value is at node 4, where $\bar{F}^{(4)}$ for $\bar{x} = -1$ has been computed. Set $n_{\text{node}} = 7$.

Step 2: **Iter = 4** and node $4 \in \mathscr{L} \cap \mathscr{L}^1$ is selected.

Step 3: At node 4, we branch on $y = 0.5$ and create nodes 8 and 9:

$$\mathscr{L}^1 = \{6,8,9\} \text{ with } f^{\text{UB},1} = 0 \text{ and } \mathscr{X}_1 = [-1,0].$$

Step 4: We compute $\underline{f}^{(8)} = -0.17$, $\underline{f}^{(9)} = -0.83$ and add both nodes to $\mathscr{L}$.

Step 5: We compute $\bar{f}^{(8)} = 0$, $\bar{f}^{(9)} = -0.33$. The latter value updates the best inner upper bound of $\mathscr{L}^1$, i.e., $f^{\text{UB},1} = -0.33$. This update causes the full fathoming of node 8 since $\underline{f}^{(8)} > f^{\text{UB},1}$ (i.e., node 8 is removed from all the lists).

Step 6: We compute $\underline{F}^{(9)} = 0$ that leads to the outer fathoming of node 9. Then the lists are:

$$\begin{array}{rl} \mathscr{L} = & \{6: -0.41 \quad 0 \quad -2 \quad -1 \quad 2\}; \\ \mathscr{L}_{\text{In}} = & \{9: -0.83 \quad -0.33 \quad - \quad - \quad 3\}, \end{array}$$

and $\mathscr{L}^1 = \{6,9\}$, with $f^{\text{UB},1} = -0.33$ and $\mathscr{X}_1 = [-1,0]$.

Step 7: The upper bounding procedure does not apply since none of the new nodes are in $\mathscr{L}$ anymore. Set $n_{\text{node}} = 9$ and go back to Step 2.
Step 2: **Iter = 5** and nodes $6 \in \mathscr{L} \cap \mathscr{L}^1$ and $9 \in \mathscr{L}_{\text{In}} \cap \mathscr{L}^1$ are selected.
Step 3: At node 6, we branch on $y = -0.5$ and create nodes 10 and 11. This gives $\mathscr{L}^1 = \{9, 10, 11\}$. At node 9, we branch on $x = -0.5$ and create nodes 12 and 13. This gives the final form of the remaining independent list:

$$\mathscr{L}^1 = \{\{10, 11, 12\}, \{10, 11, 13\}\} \text{ with } f^{\text{UB},1} = 0 \text{ and } \mathscr{X}_1 = [-1, 0].$$

Step 4: We compute $\underline{f}^{(10)} = -0.20$, $\underline{f}^{(11)} = -0.11$, $\underline{f}^{(12)} = -0.83$ and $\underline{f}^{(13)} = -0.58$. The first two nodes are not added to $\mathscr{L}$ since $\underline{f}^{(10)}, \underline{f}^{(11)} > f^{\text{UB},\bar{1}}$; the other two nodes are added to $\mathscr{L}_{\text{In}}$:

$$\begin{aligned} \mathscr{L} &= \emptyset; \\ \mathscr{L}_{\text{In}} &= \{12 : -0.83\ -0.33\ -\ -\ 4 \\ &\quad\ \ \{13 : -0.58\ -0.33\ -\ -\ 4\}. \end{aligned}$$

Full fathoming of nodes 10 and 11 invokes the list deletion procedure, which deletes list $\mathscr{L}^1 = \{\{12\}, \{13\}\}$ and results in $\mathscr{L}_{\text{In}} = \emptyset$.
Step 5: There is no new node in $\mathscr{L}$ or $\mathscr{L}_{\text{In}}$; thus, we return to Step 2.
Step 2: $\mathscr{L} = \emptyset$; terminate with $F^{\text{UB}} = 0$ at $(-1, 1)$.

Remark 6 In the example above, branching on outer-fathomed nodes (here, node 9) is not strictly necessary. However, in the majority of the examples considered it is essential to ensure convergence of the bounds on the inner problem objective function.

Flexibility arises with respect to how the inner lower bounding problems (ILB) and the inner subproblems (ISP) for fixed x values are tackled. For instance, in the following example, we solve these two problems to global optimality, rather than using their relaxations, problems (RILB) and (RISP), respectively. To explore this option, we consider the application of the Branch-and-Sandwich algorithm to problem No. 11, where problems (ILB) and (ISP) are solved with αBB. Notice that the use of problem (ISP)—instead of problem (RISP)—implies that in (UB), $w^{(k')}(\bar{x})$ is used rather than $\underline{w}^{(k')}(\bar{x})$, and k' is selected based on:

$$k' := \arg\min_{j \in \mathscr{L}^p} \{w^{(j)}(\bar{x})\}.$$

Example 2 (*Problem No. 11 in Table* 5, *Example 3.10 in* [20])

$$\begin{aligned} \min_{x,y}\ & y \\ \text{s.t.}\ & y \in \arg\min_{y \in [-1,1]} x(16y^4 + 2y^3 - 8y^2 - 1.5y + 0.5) \\ & x \in [0.1, 1]\,,\ y \in [-1, 1]\,. \end{aligned}$$

This problem has an infinite number of optimal solutions at $x^* \in [0.1, 1]$ and $y^* = 0.5$, yielding $F^* = 0.5$ and $f^* \in [-1, -0.10]$ [20]. Overall, 3 (ILB), 3 (RIUB), 2 (LB), 1 (ISP) and 1 (UB) problems were solved. The branch-and-bound tree is in Fig. 3.

Step 0: Iter = 0 and we set $\mathscr{L} = \mathscr{L}_{\text{In}} = \emptyset$, $F^{\text{UB}} = \infty$.
Step 1: $n_{\text{node}} = 1$ and we compute $\underline{f}^{(1)} = -1$, $\bar{f}^{(1)} = 0.57$ and $\underline{F}^{(1)} = -0.50$ at $(x^{(1)}, y^{(1)}) = (0.71, -0.50)$. We add node 1 to the list $\mathscr{L}$ with all the computed information:

$$\mathscr{L} = \{1 : -1\ 0.57\ -0.50\ 0.71\ 0\},$$

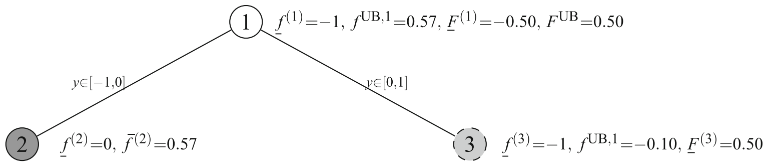

$\underline{f}^{(1)}=-1,\ f^{\mathrm{UB},1}=0.57,\ \underline{F}^{(1)}=-0.50,\ F^{\mathrm{UB}}=0.50$
$y\in[-1,0]$
$y\in[0,1]$
$\underline{f}^{(2)}=0,\ \bar{f}^{(2)}=0.57$
$\underline{f}^{(3)}=-1,\ f^{\mathrm{UB},1}=-0.10,\ \underline{F}^{(3)}=0.50$

Fig. 3 Complete branch-and-bound tree for Example 2. Again, all gray nodes are eventually (fully) fathomed. *Light-gray nodes* denote nodes that are first outer fathomed, i.e., moved to the list $\mathscr{L}_{\mathrm{In}}$, and dark-gray nodes denote those that are fully fathomed due to inner fathoming

and set:

$$p=1,\ \mathscr{X}_1=[0.1,1],\ \mathscr{L}^1=\{1\},\ f^{\mathrm{UB},1}=0.57.$$

We then compute $w(0.71)=-0.71$, which yields $\bar{F}^{(1)}=0.50$ at $(0.71,0.50)$. This updates the incumbent value: $F^{\mathrm{UB}}=0.50$.

Step 2: **Iter** $=1$ (no termination was achieved). We select node $1\in\mathscr{L}\cap\mathscr{L}^1$ and remove it from $\mathscr{L}$.

Step 3: We branch on $y=0$, resulting in $\mathscr{L}^1=\{2,3\}$, where $\mathscr{X}_1=[0.1,1]$.

Step 4: We compute $\underline{f}^{(2)}=0$ and $\underline{f}^{(3)}=-1$ and add both nodes to $\mathscr{L}$:

$$\begin{aligned}\mathscr{L}=\{2:0\ \ &0.57\ -0.50\ 0.71\ 1\\ 3:-1\ &0.57\ -0.50\ 0.71\ 1\}.\end{aligned}$$

Step 5: We compute $\bar{f}^{(2)}=0.57$ and $\bar{f}^{(3)}=-0.10$ and update the best inner upper bound for list $\mathscr{L}^1$: $f^{\mathrm{UB},1}=-0.10$. This results in node 2 being fully fathomed.

Step 6: We compute $\underline{F}^{(3)}=0.50$ at $(0.65,0.50)$. At this point list $\mathscr{L}$ is:

$$\mathscr{L}=\ \{3:-1\ -0.10\ 0.50\ 0.65\ 1\}.$$

But node 3 is outer fathomed due to outer value dominance (cf. Outer fathoming rules [16, Definition 9]) yielding:

$$\begin{aligned}\mathscr{L}\ &=\emptyset;\\ \mathscr{L}_{\mathrm{In}}&=\{3:-1\ -0.10\ -\ -\ 1\ \},\end{aligned}$$

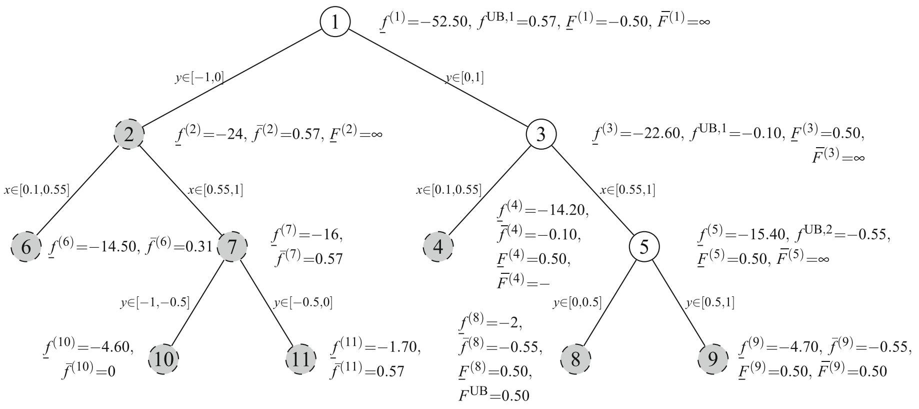
and $\mathscr{L}^1=\{3\}$ with $f^{\mathrm{UB},1}=-0.10$ and $\mathscr{X}_1=[0.1,1]$. It is clear that list $\mathscr{L}^1$ no longer holds nodes from $\mathscr{L}$ and can be discarded (cf. List-deletion fathoming rules [16, Definition 10]).

Step 7: $n_{\mathrm{node}}=3$. There is no open node left, leading back to Step 2.

Step 2: $\mathscr{L}=\emptyset$; terminate with $F^{\mathrm{UB}}=0.50$ at $(0.65,0.50)$.

Observe that in this example, 3 nodes were required for termination, and compare with the same problem instance, No. 11, in Table 5, where 11 nodes are reported. This happens because in the latter case the convex relaxed problems (RILB) and (RISP) were solved. This approach is illustrated in Fig. 4, where 11 (RILB), 11 (RIUB), 7 (LB), 11 (RISP) and 5 (UB) problems were solved overall. A few interesting points about the algorithm can be highlighted here. With (RILB) and (RISP), we cannot (fully) fathom node 2 as inner-value dominance no longer applies (the inner lower bound is too loose: $\underline{f}^{(2)}=-24$ and recall that $f^{\mathrm{UB},1}=-0.1$, where $\mathscr{X}_1=[0.1,1]$).

However, node 2 can be outer-fathomed, i.e., moved to the list $\mathscr{L}_{\mathrm{In}}$, because $\bar{F}^{(2)}=\infty$. Branching on the outer-fathomed node 2 for another two levels leads to the creation of 4 descendants in total from node 2. Recall that fathoming based on the bounding information



1
$\underline{f}^{(1)}=-52.50$, $f^{\mathrm{UB},1}=0.57$, $\underline{F}^{(1)}=-0.50$, $\bar{F}^{(1)}=\infty$
$y\in[-1,0]$
$y\in[0,1]$
2
$\underline{f}^{(2)}=-24$, $\bar{f}^{(2)}=0.57$, $\underline{F}^{(2)}=\infty$
3
$\underline{f}^{(3)}=-22.60$, $f^{\mathrm{UB},1}=-0.10$, $\underline{F}^{(3)}=0.50$, $\bar{F}^{(3)}=\infty$
$x\in[0.1,0.55]$
$x\in[0.55,1]$
$x\in[0.1,0.55]$
$x\in[0.55,1]$
6
$\underline{f}^{(6)}=-14.50$, $\bar{f}^{(6)}=0.31$
7
$\underline{f}^{(7)}=-16$, $\bar{f}^{(7)}=0.57$
4
$\underline{f}^{(4)}=-14.20$, $\bar{f}^{(4)}=-0.10$, $\underline{F}^{(4)}=0.50$, $\bar{F}^{(4)}=-$
5
$\underline{f}^{(5)}=-15.40$, $f^{\mathrm{UB},2}=-0.55$, $\underline{F}^{(5)}=0.50$, $\bar{F}^{(5)}=\infty$
$y\in[-1,-0.5]$
$y\in[-0.5,0]$
$y\in[0,0.5]$
$y\in[0.5,1]$
$\underline{f}^{(10)}=-4.60$, $\bar{f}^{(10)}=0$
10
11
$\underline{f}^{(11)}=-1.70$, $\bar{f}^{(11)}=0.57$
$\underline{f}^{(8)}=-2$, $\bar{f}^{(8)}=-0.55$, $\underline{F}^{(8)}=0.50$, $F^{\mathrm{UB}}=0.50$
8
9
$\underline{f}^{(9)}=-4.70$, $\bar{f}^{(9)}=-0.55$, $\underline{F}^{(9)}=0.50$, $\bar{F}^{(9)}=0.50$

Fig. 4 Complete branch-and-bound tree for Example 2 with (RILB) and (RISP). Nodes in light gray were outer fathomed. There is no inner fathoming in this case

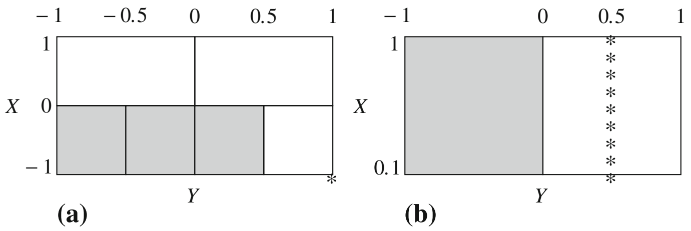


Fig. 5 Regions discarded due to inner fathoming. **a** Example 1, **b** Example 2

of the overall problem implies that the optimal solution of the bilevel problem cannot be found in the fathomed region but does not imply anything for the optimal solution of the inner problem (in the X subdomain under consideration). We have introduced the concept of outer fathoming to recognize and make use of such situations. The fathoming-by-infeasibility rule, for example, may be satisfied due to infeasible outer constraints, if present. In the case of Problem No. 11, where no outer constraints are present, the infeasibility of the lower bounding problem can only be due to inner information, which means that in this case full fathoming of node 2 could have been detected. In general, if we were able to detect/check where the infeasibility of the outer lower bounding comes from, then the Branch-and-Sandwich algorithm in its most general form, i.e., using the convex relaxed problems (RILB) and (RISP), would require 7 nodes for problem No. 11.

Finally, to highlight the effects of inner fathoming, we show the regions that are eliminated due to inner fathoming as dark boxes in Fig. 5 for the two bilevel problems presented in detail, i.e., problem No. 11 and problem No. 17. The optimal solutions of the two problems are marked with stars. Observe that significant portions of the solution space are eliminated from consideration thanks to inner fathoming and that no region where optimal solutions lie is discarded.

6 Conclusions

The present paper constitutes the second part of our work on developing a branch-and-bound scheme, the Branch-and-Sandwich algorithm, for the solution of optimistic bilevel programming problems that satisfy an appropriate regularity condition in the inner problem. In this article, we explored the convergence properties of the Branch-and-Sandwich algorithm and we proved finite convergence to an ε-optimal global solution.

We demonstrated our algorithm in detail for two test cases. With this detailed exposition, we wanted to (i) illustrate the proposed branching scheme and the use of auxiliary lists; (ii) highlight the flexibility that our method offers in how the proposed bounding problems are tackled; and also, (iii) point out that with the inner bounding scheme and the corresponding fathoming rules, it is possible to eliminate large portions of the inner space.

The Branch-and-Sandwich algorithm was tested on 34 small problems with promising numerical results. The full implementation of the algorithm and computational performance are the focus of current work. Alternative choices in the way each step of the proposed algorithm is tackled, as well as different branching strategies, need to be explored. The implementation of the proposed method will also allow us to evaluate its performance on larger problems.

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

7 Appendix: Additional test problems

Example 3 (*Problem No. 15 in Table 5, Variant of Example 3.13 in* [20])

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

Example 3 has the unique optimal solution $(x^*, y^*) = (0, 1)$ with $F^* = -1$ and $f^* = 0$. In particular, based on the inner KKT conditions: (i) $y \in \{-1, 0, \frac{1}{3}\}$ for $-1 \le x < 0$, with $y = -1$ being the unique global minimum for these x values; (ii) $y \in [-1, 1]$ for $x = 0$, with all y values being global minima; (iii) $y \in \{0, \frac{1}{3}, 1\}$ for $0 < x \le 1$, with $y = 1$ being the unique global minimum.

Example 4 (*Problem No. 34, Variant of Example 3.28* [20] *with the same optimal objective value of* -10)

$$\begin{aligned} &\min_{x,y} \sum_{j=1}^{5} -x_j^2 - y_j^2 \\ &\text{s.t.} \;\; y_1 y_2 - x_1 \le 0 \\ &\qquad x_2 y_1^2 \le 0 \\ &\qquad x_1 - \exp(x_2) + y_3 \le 0 \\ &\qquad y \in \arg\min_{y\in[-1,1]^5} \left\{ y_1^3 + y_2^2 x_1 + y_2^2 x_2 + 0.1 y_3 + (y_4^2 + y_5^2) x_3 x_4 x_5 \text{ s.t. } x_1 - 0.2 - y_3^2 \le 0 \right\} \\ &\qquad x \in [-1, 1]^5, \, y \in [-1, 1]^5. \end{aligned}$$

References

1. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of mixed-integer nonlinear problems. AIChE J. **46**(9), 1769–1797 (2000)
2. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs—I. Theoretical advances. Comput. Chem. Eng. **22**(9), 1137–1158 (1998)
3. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs-II. Implementation and computational results. Comput. Chem. Eng. **22**(9), 1159–1179 (1998)
4. Bard, J.F.: Convex two-level optimization. Math. Program. **40**(1, Ser. A), 15–27 (1988)
5. Bard, J.F.: Practical Bilevel Optimization, Nonconvex Optimization and Its Applications, vol. 30. Kluwer, Dordrecht (1998)
6. Bard, J.F., Falk, J.E.: An explicit solution to the multilevel programming problem. Comput. Oper. Res. **9**(1), 77–100 (1982). Mathematical programming with parameters and multilevel constraints
7. Bhattacharjee, B., Lemonidis, P., Green, W.H., Jr., Barton, P.I.: Global solution of semi-infinite programs. Math. Program. 103(2, Ser. B), 283–307 (2005)
8. Dempe, S.: A bundle algorithm applied to bilevel programming problems with non-unique lower level solutions. Comput. Optim. Appl. **15**(2), 145–166 (2000)
9. Dempe, S.: Foundations of Bilevel Programming, Nonconvex Optimization and Its Applications, vol. 61. Kluwer, Dordrecht (2002)
10. Dempe, S.: Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization **52**(3), 333–359 (2003)
11. Dempe, S., Zemkoho, A.B.: The generalized Mangasarian-Fromowitz constraint qualification and optimality conditions for bilevel programs. J. Optim. Theory Appl. **148**(1), 46–68 (2011)
12. Faísca, N.P., Dua, V., Rustem, B., Saraiva, P.M., Pistikopoulos, E.N.: Parametric global optimisation for bilevel programming. J. Glob. Optim. **38**(4), 609–623 (2007)
13. Gümüş, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**(1), 1–31 (2001)
14. 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. Theory Appl. **58**(1), 11–37 (1988)
15. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer, Berlin (1996)
16. Kleniati, P.M., Adjiman, C.S.: Branch-and-Sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. Part I: Theoretical development. J. Glob. Optim. (2014). doi:10.1007/s10898-013-0121-7
17. Loridan, P., Morgan, J.: Weak via strong Stackelberg problem: new results. J. Glob. Optim. **8**(3), 263–287 (1996). Hierarchical and bilevel programming
18. Lucchetti, R., Mignanego, F., Pieri, G.: Existence theorems of equilibrium points in Stackelberg games with constraints. Optimization **18**(6), 857–866 (1987)
19. 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)
20. Mitsos, A., Barton, P.I.: A test set for bilevel programs. Technical report, Massachusetts Institute of Technology (2010). http://yoric.mit.edu/sites/default/files/bileveltestset.pdf
21. Mitsos, A., Lemonidis, P., Barton, P.I.: Global solution of bilevel programs with a nonconvex inner program. J. Glob. Optim. **42**(4), 475–513 (2008)
22. Murtagh, B.A., Saunders, M.A.: Minos 5.4 users guide (revised). Technical report sol 83–20r, Department of Operations Research, Stanford University (1995)
23. Tsoukalas, A., Rustem, B., Pistikopoulos, E.N.: A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. J. Glob. Optim. **44**(2), 235–250 (2009)
24. Tuy, H.: Convex Analysis and Global Optimization, Nonconvex Optimization and Its Applications, vol. 22. Kluwer, Dordrecht (1998)
25. Tuy, H., Migdalas, A., Hoai-Phuong, N.T.: A novel approach to bilevel nonlinear programming. J. Glob. Optim. **38**(4), 527–554 (2007)
26. Tuy, H., Migdalas, A., Värbrand, P.: A global optimization approach for the linear two-level program. J. Glob. Optim. **3**(1), 1–23 (1993)
27. Visweswaran, V., Floudas, C.A., Ierapetritou, M.G., Pistikopoulos, E.N.: A decomposition-based global optimization approach for solving bilevel linear and quadratic programs. In: State of the Art in Global

Optimization. Nonconvex Optimization and Its Applications (Princeton, NJ, 1995), vol. 7, pp. 139–162. Kluwer, Dordrecht (1996)

28. Wiesemann, W., Tsoukalas A. Kleniati, P.M., Rustem, B.: Pessimistic bilevel optimization. SIAM J. Optim. **23**(1), 353–380 (2013)