

A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs

Hatim Djelassi¹ · Alexander Mitsos¹ 

Received: 3 April 2016 / Accepted: 9 October 2016 / Published online: 1 November 2016
© Springer Science+Business Media New York 2016

Abstract A discretization-based algorithm for the global solution of semi-infinite programs (SIPs) is proposed, which is guaranteed to converge to a feasible, ε -optimal solution finitely under mild assumptions. The algorithm is based on the hybridization of two existing algorithms. The first algorithm (Mitsos in Optimization 60(10–11):1291–1308, 2011) is based on a restriction of the right-hand side of the constraints of a discretized SIP. The second algorithm (Tsoukalas and Rustem in Optim Lett 5(4):705–716, 2011) employs a discretized oracle problem and a binary search in the objective space. Hybridization of the approaches yields an algorithm, which leverages the strong convergence guarantees and the relatively tight upper bounding problem of the first approach while employing an oracle problem adapted from the second approach to generate cheap lower bounds and adaptive updates to the restriction of the first approach. These adaptive updates help in avoiding a dense population of the discretization. The hybrid algorithm is shown to be superior to its predecessors both theoretically and computationally. A proof of finite convergence is provided under weaker assumptions than the assumptions in the references. Numerical results from established SIP test cases are presented.

Keywords SIP · NLP · Nonconvex · Feasible point method · Global optimization

1 Introduction

Semi-infinite programs (SIPs) are mathematical programs with a finite number of variables and an infinite number of constraints. This is commonly expressed by a parameterized constraint that has to hold for all possible realizations of its parameter(s). Here we consider SIPs of the form

Electronic supplementary material The online version of this article (doi:[10.1007/s10898-016-0476-7](https://doi.org/10.1007/s10898-016-0476-7)) contains supplementary material, which is available to authorized users.

✉ Alexander Mitsos
amitsos@alum.mit.edu

¹ AVT Process Systems Engineering (SVT), RWTH Aachen University, Aachen, Germany

$$\begin{aligned}
 f^* &= \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \\
 &\text{s.t. } g(\mathbf{x}, \mathbf{y}) \leq 0, \quad \forall \mathbf{y} \in \mathcal{Y} \\
 \mathcal{X} &\subseteq \mathbb{R}^{n_x}, \quad \mathcal{Y} \subseteq \mathbb{R}^{n_y}, \quad |\mathcal{Y}| = \infty,
 \end{aligned}
 \tag{SIP}$$

where the restriction to $|\mathcal{Y}| = \infty$ is made only for convenience. Furthermore, additional constraints (semi-infinite or otherwise) do not pose a problem for the methods presented here. The generalizations to arbitrarily many constraints and mixed-integer problems are discussed in Sect. 5. No convexity or concavity assumptions are made on the participating functions. The properties of the sets and functions participating in (SIP) are discussed in detail in Sect. 2.2 along with other assumptions.

SIPs emerge in many problems from science and engineering including robust design [7], model reduction [3], and design centering [19]. For a review of SIP theory, methods, and applications, the reader is referred to [6, 8, 15, 16]. The difficulty in solving (SIP) lies in the definition of the feasible set by infinitely many constraints. In order for a point $\bar{\mathbf{x}} \in \mathcal{X}$ to be feasible in (SIP), the globally optimal objective of the lower-level program

$$\max_{\mathbf{y} \in \mathcal{Y}} g(\bar{\mathbf{x}}, \mathbf{y})$$

must be non-positive. For constraint functions $g(\bar{\mathbf{x}}, \cdot)$ that are concave on \mathcal{Y} and satisfy constraint qualification, the lower-level program can be replaced equivalently by its Karush–Kuhn–Tucker (KKT) conditions and the SIP can be reduced to a finite problem. However, in the nonconcave case considered here, this approach is not applicable.

There exist approaches that approximate (SIP) by a finite problem, e.g., through replacing \mathcal{Y} with a discretization [4]. Such discretization approaches represent outer approximations and generally do not provide feasible points finitely. Bhattacharjee et al. [1] proposed a method that guarantees the generation of feasible points in finite time under appropriate assumptions. This is achieved through the replacement of the semi-infinite constraint by an overestimation based on interval extensions. Convergence of the overestimation is achieved through a subdivision of the parameter set. Subsequently, Bhattacharjee et al. [2] used the above approach in conjunction with a discretization-based lower bounding procedure to solve (SIP) approximately to global optimality in finite time. Therein, the central assumption is that there exists a sequence of SIP-Slater points, i.e., points satisfying the Slater condition for the semi-infinite constraints. Alternatively, convex overestimation approaches were proposed by Floudas and Stein [5] and then by Mitsos et al. [13]. The former was later extended to arbitrary index sets by Stein and Steuermann [20]. While providing tighter overestimations than interval extensions, these approaches are typically more expensive [13]. Mitsos [11] proposed an approach that employs a discretization-based outer approximation and an approximation that is based on discretization and restriction of the right-hand side of the discretized constraints. The algorithm is guaranteed to solve (SIP) approximately to global optimality in finite time under mild assumptions. Indeed this approach only requires the existence of a single ε -optimal SIP-Slater point. This approach has subsequently been extended to generalized SIPs (GSIPs) by Mitsos and Tsoukalas [14]. Tsoukalas and Rustem [24] proposed an approach that is centered around an oracle problem that decides whether a given objective value is attainable by (SIP) or not. Through a binary search of the objective space, the algorithm solves (SIP) approximately to global optimality under slightly stronger assumptions than the approach by Mitsos [11]. The oracle approach was originally proposed for GSIPs, bilevel, and continuous coupled min-max programs by Tsoukalas et al. [25]. As SIPs and GSIPs are closely related to bilevel programs [21], it bears mentioning that discretization-based approaches to the solu-

tion of bilevel programs have been proposed by Mitsos for the nonconvex case [12] and by Mitsos et al. for the mixed-integer case [10]. Furthermore, a method with feasible iterates for GSIPs with convex lower-level problems has been proposed by Stein and Winterfeld [22].

Here, we present a hybrid of the algorithm by Mitsos [11] and the algorithm by Tsoukalas and Rustem [24]. As its predecessors, the algorithm employs existing NLP-solvers to solve the finite subproblems and thus appropriate assumptions on the solution provided by these solvers are necessary. We propose a strategy to determine the required optimality tolerance for the NLP-solvers, which allows weakening of the assumptions. Under these relaxed assumptions, the hybrid algorithm is guaranteed to solve (SIP) approximately to global optimality in finite time. Furthermore, the hybridization addresses the issue of choosing appropriate values for the tuning parameters for the algorithm by Mitsos [11] and the hybrid algorithm shows superior overall performance over its predecessors.

The remainder of the present manuscript is organized as follows. In Sect. 2, definitions and assumptions used throughout the manuscript are introduced. In Sect. 3, the algorithm by Mitsos [11] is presented along the proposed strategy for the determination of NLP-solver tolerances and a proof of finite termination. In Sect. 4, the properties of the oracle problem and its adaptation to the hybrid algorithm are discussed, followed by the statement of the hybrid algorithm and a proof of finite convergence. In Sect. 5, generalizations of (SIP) to mixed-integer problems and multiple constraints are discussed. In Sect. 6, numerical experiments performed on established SIP benchmark problems and a newly developed problem are presented, while Sect. 7 gives conclusions and potential future work.

2 Definitions and assumptions

In this section, definitions and assumptions are introduced that are used throughout the remainder of the manuscript.

2.1 Definitions

Definition 1 (*Approximate solution*) Given an optimality tolerance ε^{NLP} , the approximate solution of a feasible nonlinear program

$$\varphi^* = \min_{z \in \mathcal{Z}} \varphi(z), \quad \mathcal{Z} \subseteq \mathbb{R}^{n_z}$$

provides a feasible solution point \bar{z} and a lower bound φ^- on the exact globally optimal objective value φ^* with

$$\varphi(\bar{z}) \geq \varphi^* \geq \varphi^- \geq \varphi(\bar{z}) - \varepsilon^{NLP}.$$

In compact notation we write

$$[\varphi^-, \varphi(\bar{z})] \ni \varphi^* = \min_{z \in \mathcal{Z}} \varphi(z), \quad \mathcal{Z} \subseteq \mathbb{R}^{n_z}.$$

Similarly, for a maximization problem, with approximate solution \bar{z} , upper bound ψ^+ , and exact globally optimal objective value ψ^* we write

$$[\psi(\bar{z}), \psi^+] \ni \psi^* = \max_{z \in \mathcal{Z}} \psi(z), \quad \mathcal{Z} \subseteq \mathbb{R}^{n_z}.$$

Definition 2 (*Lower level program*) For a given point $\bar{x} \in \mathcal{X}$, the lower level program is denoted along with its optimal objective value and approximate solution according to

Definition 1

$$[g(\bar{x}, \bar{y}), g^+(\bar{x})] \ni g^*(\bar{x}) = \max_{y \in \mathcal{Y}} g(\bar{x}, y). \tag{LLP}$$

Definition 3 (*SIP-feasibility*) A given point $\bar{x} \in \mathcal{X}$ is called SIP-feasible if and only if

$$g(\bar{x}, y) \leq 0, \quad \forall y \in \mathcal{Y},$$

or equivalently

$$g^*(\bar{x}) \leq 0.$$

Definition 4 (ε -optimality) A point $\bar{x} \in \mathbb{R}^{n_x}$ is ε -optimal if and only if it is SIP-feasible and its objective value is not more than ε worse than the optimal solution f^* of (SIP), i.e.,

$$\begin{aligned} g(\bar{x}, y) &\leq 0, \quad \forall y \in \mathcal{Y} \\ f(\bar{x}) &\leq f^* + \varepsilon. \end{aligned}$$

Definition 5 (*SIP-Slater point*) A point $\bar{x} \in \mathcal{X}$ is called SIP-Slater point if and only if it satisfies the semi-infinite constraint strictly, i.e.,

$$g(\bar{x}, y) < 0, \quad \forall y \in \mathcal{Y}.$$

Remark 1 Note that \bar{x} is not restricted to the interior of \mathcal{X} . Therefore, finite constraints are not required to be satisfied strictly at an SIP-Slater point.

2.2 Assumptions

Assumption 1 (*Host sets*) The host sets $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ are compact.

Assumption 2 (*Basic properties of functions*) The functions $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ are continuous on \mathcal{X} and $\mathcal{X} \times \mathcal{Y}$ respectively.

Remark 2 Assumptions 1 and 2 are standard in deterministic global optimization.

Assumption 3 ($\tilde{\varepsilon}^f$ -optimal SIP-Slater point) There exists some $\tilde{\varepsilon}^f > 0$ such that there exists an SIP-Slater point $x^S \in \mathcal{X}$ that is $\tilde{\varepsilon}^f$ -optimal, i.e.,

$$\begin{aligned} g(x^S, y) &< 0, \quad \forall y \in \mathcal{Y} \\ f(x^S) &\leq f^* + \tilde{\varepsilon}^f. \end{aligned}$$

Assumption 3 was first used in [11]. As discussed in [14], it is slightly weaker than the assumption in [24], which is equivalent to Assumption 3 holding for arbitrarily small $\tilde{\varepsilon}^f$.

Assumption 4 (*Approximate solution of NLPs*) If a nonlinear program (NLP) is infeasible, it is found to be infeasible without any infeasibility tolerance ($\varepsilon^{INF} = 0$). Otherwise, it can be solved globally to within an arbitrary optimality tolerance $\varepsilon^{NLP} > 0$.

Assumption 4 is a relaxation of similar assumptions in [11] and [24]. In [11], it is assumed that the approximate solution of (LLP) provides either $g^+(\bar{x}) \leq 0$ or $g(\bar{x}, \bar{y}) > 0$, which may not be given, especially for $g^*(\bar{x}) = 0$. In [24], it is assumed implicitly that all NLP subproblems are solved exactly. Obviously this is the strongest of the mentioned assumptions and can in general not be provided by numerical solution methods.

3 Restriction of the right-hand-side

The algorithm proposed by Mitsos [11] solves (SIP) by employing three finite subproblems. Through the solution of these problems by existing NLP-solvers, bounds on the solution of (SIP) are obtained. Under mild assumptions discussed later on, the algorithm terminates finitely with an ε^f -optimal point, where $\varepsilon^f > 0$ is a user-defined optimality tolerance. In Sect. 3.1, the subproblems used in the original algorithm are reviewed, followed by the statement of the algorithm with revisions in Sect. 3.2. In Sect. 3.3, a proof of finite convergence is given for the revised algorithm under weaker assumptions than in [11]. While the overall structure of the proof remains unchanged from [11], the weaker assumptions, specifically Assumption 4, require the explicit consideration of NLP-tolerances. This approach is inspired by the proofs provided in [12] and [10]. Finally, in Sect. 3.4, the choice of values for the tuning parameters of the algorithm is discussed on the basis of a newly developed benchmark case.

3.1 Review of subproblems in [11]

The first problem is the lower bounding problem (LBD) originally proposed by Blankenship and Falk [4]. A finite relaxation of (SIP) is obtained by replacing the parameter set \mathcal{Y} with a finite discretization $\mathcal{Y}^{LBD} \subsetneq \mathcal{Y}$.

$$\begin{aligned} [f^{LBD,-}, f(\bar{x}^{LBD})] \ni f^{LBD,*} &= \min_{x \in \mathcal{X}} f(x) \\ \text{s.t. } g(x, y) &\leq 0, \quad \forall y \in \mathcal{Y}^{LBD} \quad (\text{LBD}) \\ \mathcal{Y}^{LBD} &\subsetneq \mathcal{Y}, \quad |\mathcal{Y}^{LBD}| < \infty. \end{aligned}$$

Due to the fact that the discretization \mathcal{Y}^{LBD} is a subset of the parameter set \mathcal{Y} , (LBD) is a valid relaxation of (SIP). Consequently, any lower bound $f^{LBD,-}$ on its exact globally optimal objective value is a lower bound on the exact globally optimal objective value of (SIP).

The second problem is the upper bounding problem (UBD). It is obtained by first replacing the parameter set \mathcal{Y} with a finite discretization \mathcal{Y}^{UBD} and subsequently restricting the right-hand-side of the semi-infinite constraint by a restriction parameter $\varepsilon^g > 0$.

$$\begin{aligned} [f^{UBD,-}, f(\bar{x}^{UBD})] \ni f^{UBD,*} &= \min_{x \in \mathcal{X}} f(x) \\ \text{s.t. } g(x, y) &\leq -\varepsilon^g, \quad \forall y \in \mathcal{Y}^{UBD} \quad (\text{UBD}) \\ \mathcal{Y}^{UBD} &\subsetneq \mathcal{Y}, \quad |\mathcal{Y}^{UBD}| < \infty. \end{aligned}$$

In general, this problem is neither a relaxation nor a restriction of (SIP). Therefore, it is possible that a solution \bar{x}^{UBD} of (UBD) is SIP-infeasible. Consequently, an upper bound $f(\bar{x}^{UBD})$ to (UBD) is not necessarily an upper bound to (SIP). Also, due to the restriction of the right-hand-side, a lower bound $f^{UBD,-}$ to (UBD) is not necessarily a lower bound to (SIP). SIP-feasible points and therefore upper bounds to (SIP) are obtained by successively reducing the restriction parameter ε^g and populating the discretization \mathcal{Y}^{UBD} . Three cases can be distinguished in order to determine the required modification of (UBD):

1. (UBD) is infeasible. In this case, the restriction parameter ε^g is reduced in order to relax (UBD) and obtain a feasible instance of (UBD) in a following iteration.
2. (UBD) is feasible and its solution \bar{x}^{UBD} is SIP-infeasible. In this case, \mathcal{Y}^{UBD} is populated with a new discretization point in order to make \bar{x} infeasible in the following iteration of (UBD).

3. **(UBD)** is feasible and its solution \bar{x}^{UBD} is SIP-feasible. In this case, $f(\bar{x}^{UBD})$ is an upper bound to **(SIP)**. The restriction parameter ε^g is reduced in order to obtain tighter upper bounds in following iterations.

Consequently, in all three cases, **(UBD)** is refined to better approximate **(SIP)**.

The final problem is the lower-level program **(LLP)** from Definition 2.

$$[g(\bar{x}, \bar{y}), g^+(\bar{x})] \ni g^*(\bar{x}) = \max_{y \in \mathcal{Y}} g(\bar{x}, y) \tag{LLP}$$

For a given point \bar{x} , $g^*(\bar{x})$ is positive if \bar{x} is SIP-infeasible and non-positive if \bar{x} is SIP-feasible. Within the algorithm, **(LLP)** is solved for the solution points of **(LBD)** and **(UBD)** to either determine their SIP-feasibility or populate the respective discretization sets \mathcal{Y}^{LBD} and \mathcal{Y}^{UBD} with the solution point \bar{y} of **(LLP)**. To guarantee convergence, Mitsos [11] assumes that an approximate solution of **(LLP)** either provides $g^+(\bar{x}) \leq 0$ or $g(\bar{x}, \bar{y}) > 0$. Thereby, either \bar{x} is found to be SIP-feasible, or a solution $\bar{y} \in \mathcal{Y}$ is found that when added to the respective discretization makes \bar{x} infeasible in the following iteration of **(LBD)** or **(UBD)**. In the following section, the algorithm is formally stated with revisions that allow the relaxation of this assumption. This is of interest since the previous assumptions cannot be guaranteed by state-of-the-art NLP-solvers. While the case $g^+(\bar{x}) > 0 \geq g(\bar{x}, \bar{y})$ becomes increasingly unlikely for decreasing NLP-tolerances, it cannot be excluded a priori.

3.2 Algorithm statement with revisions

Here, the revised algorithm by Mitsos [11] is stated formally based on the subproblems **(LBD)**, **(UBD)**, and **(LLP)**. These problems are solved approximately according to Definition 1 with their respective tolerances $\varepsilon^{LBD}, \varepsilon^{UBD}, \varepsilon^{LBD,LLP}, \varepsilon^{UBD,LLP} > 0$. In the original statement of the algorithm, the tolerances $\varepsilon^{LBD,LLP}$ and $\varepsilon^{UBD,LLP}$ are fixed by the user, while in the revised version below they are adjusted to guarantee finite termination under the relaxed assumptions. This refinement is contained in Lines 10 and 14 of Algorithm 1.

In the upper bounding procedure, **(LLP)** is always solved with $\varepsilon^{UBD,LLP} < \varepsilon^g$ due to $\varepsilon^{UBD,LLP} = \varepsilon^g / r^{UBD,LLP}$ (Line 14) and $r^{UBD,LLP} > 1$. Consequently, if \bar{x}^{UBD} is not found to be SIP-feasible it is guaranteed to be infeasible in the next instance of **(UBD)**. This is shown in detail in Lemma 3. Note that \bar{x}^{UBD} may become feasible in a subsequent instance of **(UBD)** due to the reduction of ε^g and the fact that $g(\bar{x}^{UBD}, \bar{y}) > 0$ does not necessarily hold. Indeed, the case $g^+(\bar{x}^{UBD}) > 0 \geq g(\bar{x}^{UBD}, \bar{y})$ may occur since the contrary is not assumed as it is in [11]. This however, does not impede the finite termination of the algorithm since only a finite number of reductions of ε^g can occur before the algorithm terminates (see Lemma 4). In the lower bounding procedure, a value for $\varepsilon^{LBD,LLP}$ cannot be given a priori such that \bar{x}^{LBD} becomes infeasible in the following instance of **(LBD)**. However, in Lemma 1 it is shown that the proposed reduction strategy is sufficient to make \bar{x}^{LBD} infeasible in **(LBD)** with a finite number of reductions. With these properties, the proposed refinement strategy replaces the assumption in [11] that either $g^+(\bar{x}) \leq 0$ or $g(\bar{x}, \bar{y}) > 0$.

By Assumption 3, there exists an $\tilde{\varepsilon}^f$ -optimal SIP-Slater point and consequently **(SIP)** is feasible. Theorem 1 of the following section states that for appropriately chosen $\varepsilon^f, \varepsilon^{LBD}$, and ε^{UBD} , Algorithm 1 terminates finitely with an ε^f -optimal point x^* .

Algorithm 1: Revised approach by Mitsos [11] based on restriction of the right-hand-side. User inputs are the initial discretization sets $\mathcal{Y}^{LBD,0}, \mathcal{Y}^{UBD,0} \subsetneq \mathcal{Y}$, the initial restriction parameter $\varepsilon^{g,0} > 0$ and the corresponding reduction parameter $r^g > 1$, the initial lower-level tolerances $\varepsilon^{LBD,LLP,0}, \varepsilon^{UBD,LLP,0} > 0$ and the corresponding reduction parameter $r^{LBD,LLP}, r^{UBD,LLP} > 1$, and the tolerances $\varepsilon^{LBD}, \varepsilon^{UBD}, \varepsilon^f > 0$.

```

1 Set  $LBD \leftarrow -\infty, UBD \leftarrow \infty, \mathcal{Y}^{LBD} \leftarrow \mathcal{Y}^{LBD,0}, \mathcal{Y}^{UBD} \leftarrow \mathcal{Y}^{UBD,0}, \varepsilon^g \leftarrow \varepsilon^{g,0},$ 
    $\varepsilon^{LBD,LLP} \leftarrow \varepsilon^{LBD,LLP,0}, \varepsilon^{UBD,LLP} \leftarrow \varepsilon^{UBD,LLP,0}$ .
2 Solve (LBD) globally according to Definition 1 with  $\varepsilon^{NLP} = \varepsilon^{LBD}$  to obtain  $f^{LBD,-}$  and  $\bar{x}^{LBD}$ .
3 Set  $LBD \leftarrow f^{LBD,-}$ .
4 Solve (LLP) globally according to Definition 1 with  $\varepsilon^{NLP} = \varepsilon^{LBD,LLP}$  for  $\bar{x}^{LBD}$  to obtain
    $g^+(\bar{x}^{LBD})$  and  $\bar{y}$ .
5 if  $g^+(\bar{x}^{LBD}) \leq 0$  then
6 | Set  $UBD \leftarrow f(\bar{x}^{LBD}), x^* = \bar{x}^{LBD}$  and terminate.
7 else if  $g(\bar{x}^{LBD}, \bar{y}) > 0$  then
8 | Set  $\mathcal{Y}^{LBD} \leftarrow \mathcal{Y}^{LBD} \cup \{\bar{y}\}$ .
9 else
10 | Set  $\varepsilon^{LBD,LLP} \leftarrow (g^+(\bar{x}^{LBD}) - g(\bar{x}^{LBD}, \bar{y}))/r^{LBD,LLP}$ .
11 end
12 Solve (UBD) globally according to Definition 1 with  $\varepsilon^{NLP} = \varepsilon^{UBD}$  to obtain  $f^{UBD,-}$  and  $\bar{x}^{UBD}$ .
13 if (UBD) is feasible then
14 | if  $\varepsilon^{UBD,LLP} \geq \varepsilon^g$  then Set  $\varepsilon^{UBD,LLP} \leftarrow \varepsilon^g/r^{UBD,LLP}$ .
15 | Solve (LLP) globally according to Definition 1 with  $\varepsilon^{NLP} = \varepsilon^{UBD,LLP}$  for  $\bar{x}^{UBD}$  to obtain
    $g^+(\bar{x}^{UBD})$  and  $\bar{y}$ .
16 | if  $g^+(\bar{x}^{UBD}) \leq 0$  then
17 | | if  $f(\bar{x}^{UBD}) < UBD$  then Set  $UBD \leftarrow f(\bar{x}^{UBD}), x^* \leftarrow \bar{x}^{UBD}$ .
18 | | Set  $\varepsilon^g \leftarrow \varepsilon^g/r^g$ .
19 | else
20 | | Set  $\mathcal{Y}^{UBD} \leftarrow \mathcal{Y}^{UBD} \cup \{\bar{y}\}$ .
21 | end
22 else
23 | Set  $\varepsilon^g \leftarrow \varepsilon^g/r^g$ .
24 end
25 if  $UBD - LBD \leq \varepsilon^f$  then Terminate. else Go to Line 2.

```

Remark 3 Note that the algorithm could be changed to check for convergence after each update of the bounds, which would potentially save the last iteration of the upper bounding procedure.

3.3 Finite termination

The proof follows the structure of the proof provided in [11] but considers explicitly the NLP-tolerances and the revisions to the algorithm. In order to prove finite termination of Algorithm 1, it is first proven in Lemma 1 that the successive refinement of the optimality tolerance $\varepsilon^{LBD,LLP}$ guarantees that a point \bar{x}^{LBD} can only be visited a finite number of times by (LBD), unless it is SIP-feasible. Note that due to approximate solution of (LLP), an SIP-Slater point \bar{x}^{LBD} can indeed be visited by (LBD) without the algorithm terminating immediately. This however, is not a problem since at this point, the lower bound LBD is sufficiently accurate for the algorithm to terminate finitely. Indeed, given Lemma 1, Lemma 2 shows that the lower bounding procedure provides an ε^{LBD} -underestimate of f^* finitely

or in the limit. Next, it is proven in Lemma 3 that the discretization point added in the upper bounding procedure is guaranteed to make the associated point \bar{x}^{UBD} infeasible in the following iteration of (UBD). Based on this, Lemma 4 shows that the upper bounding procedure furnishes an $(\bar{\varepsilon}^f + \varepsilon^{UBD})$ -optimal point finitely. Finally, Theorem 1 proves that, given appropriate choices of the optimality tolerances ε^f , ε^{LBD} , and ε^{UBD} , Algorithm 1 terminates finitely with an ε^f -optimal point.

In the following, $\{\bar{x}^{LBD,k}\}_{k \geq 1}$ is the sequence of points produced by approximate solutions of (LBD). For each $\bar{x}^{LBD,k}$, $\{\bar{y}^{LBD,j}\}_{j \in \mathcal{J}^k}$ is the sequence of points produced by approximate solutions of (LLP) for $\bar{x} = \bar{x}^{LBD,k}$.

Lemma 1 *For some $k \geq 1$, let $\bar{x}^{LBD,k}$ be SIP-infeasible ($g^*(\bar{x}^{LBD,k}) > 0$). Assume that the optimality tolerance $\varepsilon^{LBD,LLP}$ is tightened successively according to Algorithm 1. Then, under Assumption 4, $\bar{x}^{LBD,k}$ can only be visited a finite number of times by (LBD) before one of the discretization points generated by the solution of (LLP) makes $\bar{x}^{LBD,k}$ infeasible in (LBD).*

Proof It is to show that the index set \mathcal{J}^k has finite cardinality and that there exists an index $j^k \in \mathcal{J}^k$ with $g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) > 0$.

It holds $g^*(\bar{x}^{LBD,k}) > 0$. Therefore and by approximate solution of (LLP) (Assumption 4), if at any iteration of Algorithm 1, $\bar{x}^{LBD,k}$ is evaluated and it holds $\varepsilon^{LBD,LLP} < g^*(\bar{x}^{LBD,k})$, it also holds

$$g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) \geq g^+(\bar{x}^{LBD,k}) - \varepsilon^{LBD,LLP} \geq g^*(\bar{x}^{LBD,k}) - \varepsilon^{LBD,LLP} > 0 \tag{1}$$

and $\varepsilon^{LBD,LLP}$ remains unchanged. The consequently generated discretization point \bar{y}^{LBD,j^k} renders $\bar{x}^{LBD,k}$ infeasible in (LBD) for all following iterations. If conversely $\bar{x}^{LBD,k}$ is evaluated and (1) does not hold, $\varepsilon^{LBD,LLP}$ is reduced. Therefore, it follows $|\mathcal{J}^k| = 1$ in the case that $\varepsilon^{LBD,LLP} < g^*(\bar{x}^{LBD,k})$ holds the first time $\bar{x}^{LBD,k}$ is evaluated. Otherwise, according to Algorithm 1, $\bar{x}^{LBD,k}$ is rendered infeasible in (LBD) after at most $\lceil \log_{r^{LBD,LLP}}(\varepsilon^{LBD,LLP} / g^*(\bar{x}^{LBD,k})) \rceil + 1$ reductions of $\varepsilon^{LBD,LLP}$. With $r^{LBD,LLP} > 1$ and $\varepsilon^{LBD,LLP} \geq g^*(\bar{x}^{LBD,k}) > 0$ it holds $|\mathcal{J}^k| < \infty$. \square

Remark 4 Given a finite sequence of SIP-infeasible points $\bar{x}^{LBD,k}$, Lemma 1 implies that there exists a value $\varepsilon^{LBD,LLP} > 0$ such that each point $\bar{x}^{LBD,k}$ can only be visited once by (LBD). However, the sequence of points generated by (LBD) depends on the particular instance of (SIP) and the NLP-solver used to solve the subproblems. Furthermore, it is not obvious how the existence of points $x \in \mathcal{X}$ with $g^*(x)$ positive but arbitrarily small could be excluded for nontrivial and feasible instances of (SIP) obeying Assumption 2. Consequently, it is not obvious how to determine said value for $\varepsilon^{LBD,LLP}$ a priori. Similarly, given an initial value $\varepsilon^{LBD,LLP,0}$, it is not obvious how to determine a priori whether or not a refinement of $\varepsilon^{LBD,LLP}$ will become necessary.

Lemma 2 *Let $\mathcal{Y}^{LBD,0} \subsetneq \mathcal{Y}$. Then, under Assumptions 1, 2, and 4, the lower bound $f^{LBD,-}$ produced by the lower bounding procedure of Algorithm 1 converges to an ε^{LBD} -underestimate of the globally optimal objective value f^* of (SIP) either finitely or in the limit.*

Proof It will be shown that the lower bounding procedure furnishes an SIP-feasible point either finitely or in the limit. In both cases, it will be shown that the corresponding lower bound is an ε^{LBD} -underestimate of f^* .

If for any k , (LBD) furnishes a point $\bar{x}^{LBD,k}$ with $g^*(\bar{x}^{LBD,k}) \leq 0$, then the lower bounding procedure has finitely furnished an SIP-feasible point. By definition of f^* and by SIP-feasibility of $\bar{x}^{LBD,k}$, it holds $f(\bar{x}^{LBD,k}) \geq f^*$. Furthermore, since (LBD) is a valid relaxation of (SIP) and is solved to ε^{LBD} -optimality, it holds $f^* \geq f^{LBD,-}$ and $f^{LBD,-} \geq f(\bar{x}^{LBD,k}) - \varepsilon^{LBD}$. Consequently,

$$f^* \geq f^{LBD,-} \geq f(\bar{x}^{LBD,k}) - \varepsilon^{LBD} \geq f^* - \varepsilon^{LBD}$$

is achieved finitely.

Otherwise, it will be shown that an SIP-feasible point is generated in the limit. Consider an infinite sequence of solutions to (LBD). By compactness of \mathcal{X} and Lemma 1, an infinite subsequence can be selected with the limit point $\hat{x}^{LBD} \in \mathcal{X}$, i.e., $\bar{x}^{LBD,k} \xrightarrow{k \rightarrow \infty} \hat{x}^{LBD}$. By Lemma 1 the sequences $\{\bar{y}^{LBD,j}\}_{j \in \mathcal{J}^k}$ are finite and furthermore, it holds $g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) > 0, \forall k \geq 1$. By construction of (LBD), it holds

$$g(\bar{x}^{LBD,l}, \bar{y}^{LBD,j^k}) \leq 0, \quad \forall l, k : l > k \geq 1.$$

By continuity of $g(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{Y}$ and compactness of $\mathcal{X} \times \mathcal{Y}$, $g(\cdot, \cdot)$ is uniformly continuous. Therefore, for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$g(x, \bar{y}^{LBD,j^k}) < \varepsilon, \quad \forall x \in \mathcal{X} : \|x - \bar{x}^{LBD,l}\| < \delta, \quad \forall l, k : l > k \geq 1. \tag{2}$$

Since $\bar{x}^{LBD,k} \xrightarrow{k \rightarrow \infty} \hat{x}^{LBD}$, for any $\delta > 0$ there exists K such that

$$\|\bar{x}^{LBD,l} - \bar{x}^{LBD,k}\| < \delta, \quad \forall l, k : l > k \geq K \geq 1. \tag{3}$$

Combining (2) and (3), we obtain that for any $\varepsilon > 0$, there exists K such that

$$g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) < \varepsilon, \quad \forall k \geq K \geq 1.$$

With $g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) > 0$ it follows that $g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) \xrightarrow{k \rightarrow \infty} 0$. Consequently, the limit point \hat{x}^{LBD} is SIP-feasible, i.e.,

$$g^*(\hat{x}^{LBD}) = \lim_{k \rightarrow \infty} g(\bar{x}^{LBD,k}, \bar{y}^{LBD,j^k}) = 0.$$

By definition of f^* and by SIP-feasibility of \hat{x}^{LBD} , it holds $f(\hat{x}^{LBD}) \geq f^*$. Due to $\bar{x}^{LBD,k} \rightarrow \hat{x}^{LBD}$ it follows by continuity of $f(\cdot)$ that $f(\bar{x}^{LBD,k}) \rightarrow f(\hat{x}^{LBD})$. Since (LBD) is a valid relaxation of (SIP) and by solution of (LBD) to ε^{LBD} -optimality, the lower bounds to (LBD) converge to a value in the set

$$\left[f(\hat{x}^{LBD}) - \varepsilon^{LBD}, f^* \right] \subseteq \left[f^* - \varepsilon^{LBD}, f^* \right],$$

which gives the desired property. □

In the following, $\{\bar{x}^{UBD,k}\}_{k \geq 1}$ is the sequence of points produced by approximate solutions of (UBD). Furthermore, $\{\bar{y}^{UBD,k}\}_{k \geq 1}$ is the sequence of points produced by approximate solutions of (LLP).

Lemma 3 *For any $k \geq 1$, under Assumption 4, either $\bar{x}^{UBD,k}$ is found to be SIP-feasible, or the addition of the solution $\bar{y}^{UBD,k}$ to \mathcal{Y}^{UBD} renders $\bar{x}^{UBD,k}$ infeasible in the following iteration of (UBD).*

Proof Consider the first case that $g^+(\bar{x}^{UBD,k}) \leq 0$. With $g^*(\bar{x}^{UBD,k}) \leq g^+(\bar{x}^{UBD,k})$, it follows immediately that $\bar{x}^{UBD,k}$ is found to be SIP-feasible. Otherwise it holds $g^+(\bar{x}^{UBD,k}) > 0$ and by solution of (LLP) to $\varepsilon^{UBD,LLP}$ -optimality

$$g(\bar{x}^{UBD,k}, \bar{y}^{UBD,k}) \geq g^+(\bar{x}^{UBD,k}) - \varepsilon^{UBD,LLP} > -\varepsilon^{UBD,LLP}.$$

The constraint resulting from the addition of $\bar{y}^{UBD,k}$ to \mathcal{Y}^{UBD} is

$$g(x, \bar{y}^{UBD,k}) \leq -\varepsilon^g.$$

Due to $\varepsilon^{UBD,LLP} < \varepsilon^g$, $\bar{x}^{UBD,k}$ is infeasible in the following iteration of (UBD). □

Lemma 4 *Let $\mathcal{Y}^{UBD,0} \subsetneq \mathcal{Y}$, $\varepsilon^{g,0} > 0$, and $r^g > 1$. Then, under Assumptions 1, 2, 3, and 4, the upper bounding procedure furnishes an $(\tilde{\varepsilon}^f + \varepsilon^{UBD})$ -optimal point \hat{x}^{UBD} finitely.*

Proof At each iteration of the upper bounding procedure there are three possible outcomes:

1. (UBD) is infeasible.
2. (UBD) is feasible and furnishes a point that is confirmed to be SIP-feasible by (LLP).
3. (UBD) is feasible and furnishes a point that is not confirmed to be SIP-feasible by (LLP).

In the former two cases, the restriction parameter ε^g and the lower-level tolerance $\varepsilon^{UBD,LLP}$ are reduced. In the latter case, \mathcal{Y}^{UBD} is populated. Any point $x \in \mathcal{X}$ can be characterized either as SIP-feasible or SIP-infeasible. Moreover, any point has either an objective value $f(x) \leq f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}$ or not. It will first be shown that an infinite sequence of infeasible problems (UBD) or points $\bar{x}^{UBD,k}$ with $f(\bar{x}^{UBD,k}) > f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}$ cannot occur. Subsequently, it will be shown that the case of an infinite sequence of SIP-infeasible points or points with undetermined SIP-feasibility cannot occur.

At all iterations it holds $\mathcal{Y}^{UBD} \subsetneq \mathcal{Y}$. By assumption, there exists an $\tilde{\varepsilon}^f$ -optimal SIP-Slater point with

$$f(x^S) \leq f^* + \tilde{\varepsilon}^f, \quad g(x^S, y) \leq -\varepsilon^S, \quad \forall y \in \mathcal{Y}.$$

This point is feasible in (UBD) for $\varepsilon^g \leq \varepsilon^S$ irrespective of \mathcal{Y}^{UBD} . Therefore, (UBD) is feasible for $\varepsilon^g \leq \varepsilon^S$ irrespective of \mathcal{Y}^{UBD} . Moreover, the approximate global solution of (UBD) guarantees that for $\varepsilon^g \leq \varepsilon^S$, the upper bounding procedure produces points $\bar{x}^{UBD,k}$ with

$$f(\bar{x}^{UBD,k}) \leq f(x^S) + \varepsilon^{UBD} \leq f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}.$$

The restriction parameter ε^g is updated if and only if (UBD) is infeasible, or it furnishes an SIP-feasible point $\bar{x}^{UBD,k}$. As a consequence, after $\lceil \log_{r^g}(\varepsilon^{g,0}/\varepsilon^S) \rceil$ updates of ε^g the upper bounding problem generates points $\bar{x}^{UBD,k}$ with $f(\bar{x}^{UBD,k}) \leq f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}$. If one of these points is found to be SIP-feasible, the algorithm terminates. Otherwise, the restriction parameter is not updated beyond a final value $\varepsilon^{g,\min} \geq \varepsilon^S/r^g$.

It remains to be shown that it is impossible for (UBD) to furnish an infinite sequence of SIP-infeasible points or points with undetermined SIP-feasibility. Since $\varepsilon^g \geq \varepsilon^{g,\min} > 0$ holds for all iterations, this can be proven similarly to Lemma 2. Consider an infinite sequence of solutions to instances of (UBD). By compactness of \mathcal{X} and Lemma 3, we can select a convergent subsequence $\bar{x}^{UBD,k} \xrightarrow{k \rightarrow \infty} \hat{x}^{UBD}$, $\hat{x}^{UBD} \in \mathcal{X}$. Consider the corresponding solutions to (LLP) $\bar{y}^{UBD,k}$. By construction of (UBD) we have

$$g(\bar{x}^{UBD,l}, \bar{y}^{UBD,k}) \leq -\varepsilon^{g,\min}, \quad \forall l, k : l > k \geq 1.$$

By continuity of $g(\cdot, \bar{y}^{UBD,k})$ and compactness of \mathcal{X} , $g(\cdot, \bar{y}^{UBD,k})$ is uniformly continuous on \mathcal{X} . Therefore, for any $0 < \gamma < 1$, there exists $\delta > 0$ such that

$$g\left(\mathbf{x}, \bar{y}^{UBD,k}\right) < -\gamma \varepsilon^{g,\min} < 0, \quad \forall \mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \bar{\mathbf{x}}^{UBD,l}\| < \delta, \quad \forall l, k : l > k \geq 1. \tag{4}$$

Since $\bar{\mathbf{x}}^{UBD,k} \xrightarrow{k \rightarrow \infty} \hat{\mathbf{x}}^{UBD}$, for any $\delta > 0$, there exists K such that

$$\|\bar{\mathbf{x}}^{UBD,l} - \bar{\mathbf{x}}^{UBD,k}\| < \delta, \quad \forall l, k : l > k \geq K \geq 1. \tag{5}$$

Combining (4) and (5) yields that for any $0 < \gamma < 1$ there exists K such that

$$g\left(\bar{\mathbf{x}}^{UBD,k}, \bar{y}^{UBD,k}\right) < -\gamma \varepsilon^{g,\min} < 0, \quad \forall k : k \geq K \geq 1.$$

Therefore, after a finite number K of iterations, the points furnished by the upper bounding procedure are SIP-feasible. Moreover, since $\varepsilon^{UBD,LLP} < \varepsilon^{g,\min}$, these points are found by (LLP) to be SIP-feasible. □

Remark 5 In Lemma 4, it is used that ε^g is not updated beyond a final value $\varepsilon^{g,\min} \geq \varepsilon^S/r^g$. It follows by Algorithm 1 that the tolerance $\varepsilon^{UBD,LLP}$ is not updated beyond $\varepsilon^{g,\min}/r^{UBD,LLP}$. By extension, setting $\varepsilon^{UBD,LLP,0} < \varepsilon^S/r^g$ would render the tolerance update obsolete. However, determining ε^S usually requires the solution of (SIP). Consequently, finding a value for $\varepsilon^{UBD,LLP,0}$ a priori that guarantees that no tolerance updates are required is usually not possible.

Theorem 1 Let ε^f , ε^{LBD} , and ε^{UBD} be chosen such that

$$\varepsilon^f > \tilde{\varepsilon}^f + \varepsilon^{LBD} + \varepsilon^{UBD}.$$

Then, under Assumptions 1, 2, 3, and 4, Algorithm 1 terminates finitely with an ε^f -optimal point \mathbf{x}^* .

Proof By Lemma 2, the lower bound $f^{LBD,-}$ converges to an ε^{LBD} -underestimate of f^* , i.e.,

$$\lim_{k \rightarrow \infty} f^{LBD,-,k} \in \left[f^* - \varepsilon^{LBD}, f^* \right],$$

with the lower bound $f^{LBD,-,k}$ produced by instance k of (LBD). Therefore, for any $\delta > 0$ there exists K^{LBD} such that

$$f^{LBD,-,k} \geq f^* - \varepsilon^{LBD} - \delta, \quad \forall k > K^{LBD} \geq 0. \tag{6}$$

By Lemma 4, after a finite number K^{UBD} of instances of (UBD), the upper bounding procedure furnishes an $(\tilde{\varepsilon}^f + \varepsilon^{UBD})$ -optimal point $\hat{\mathbf{x}}^{UBD}$. It holds

$$f\left(\bar{\mathbf{x}}^{UBD,k}\right) \leq f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}, \quad \forall k > K^{UBD} \geq 0. \tag{7}$$

Combining (6) and (7) and choosing $\delta = \varepsilon^f - \tilde{\varepsilon}^f - \varepsilon^{LBD} - \varepsilon^{UBD} > 0$, we obtain that there exists a finite number $K = \max\{K^{LBD}, K^{UBD}\}$ of iterations of Algorithm 1 such that

$$f\left(\bar{\mathbf{x}}^{UBD,K}\right) - f^{LBD,K} \leq \varepsilon^f,$$

at which point the algorithm terminates with an ε^f -optimal point. □

Recall that ε^f , ε^{UBD} , and ε^{LBD} are tolerances set as inputs to Algorithm 1. $\tilde{\varepsilon}^f$ on the other hand is assumed to be nonzero and is, in general, unknown a priori. In case $\tilde{\varepsilon}^f$ is known, ε^{LBD} and ε^{UBD} should be chosen such that the desired ε^f can be guaranteed. Absent such information, it may be assumed that $\tilde{\varepsilon}^f$ is of similar magnitude as the chosen ε^{LBD} and ε^{UBD} . In any case, ε^{LBD} and ε^{UBD} should be substantially smaller, e.g., by one order of magnitude, than ε^f .

3.4 Choice of parameter values and benchmark problem

Although the algorithm is guaranteed to converge finitely under Assumptions 1, 2, 3, and 4, its computational performance is subject to the choice of the parameters $\varepsilon^{g,0}$ and r^g . In the following, small values for $\varepsilon^{g,0}$ and large values for r^g are termed aggressive since such a choice is aimed at achieving the termination criterion in few updates of the restriction parameter. Accordingly, large values for $\varepsilon^{g,0}$ and small values for r^g are termed conservative. In general, for aggressive parameter choices, the worst case number of reductions of ε^g that is required to obtain feasibility in (UBD) of a point x with $f(x) \leq f^* + \varepsilon^f$ is lower than for conservative choices. However, if a small value for ε^g is reached before \mathcal{Y}^{UBD} is sufficiently populated, the subsequent solutions of (UBD) may furnish many SIP-infeasible points before \mathcal{Y}^{UBD} contains the discretization points that entail an SIP-feasible iterate. This phenomenon manifests in a dense population of \mathcal{Y}^{UBD} and slow convergence of the upper bound.

For the initial restriction $\varepsilon^{g,0}$, the above discussion suggests that a conservative choice is appropriate. Starting with an aggressive value for ε^g would defeat the purpose of Algorithm 1, which is to find a small discretization that appropriately represents (SIP). While there are problems that are not particularly prone to a dense population of \mathcal{Y}^{UBD} , choosing a small value $\varepsilon^{g,0}$ for an unknown problem seems to be undesirable.

The reduction parameter r^g has already been discussed as pivotal by Mitsos [11]. Therein, it was shown for certain test problems that a wide range of values (approximately $r^g \in [2, 20]$) yields acceptable results for overall performance of the algorithm. Here we expand on that analysis by considering the impact on the upper bounding procedure specifically. If overall algorithm performance is considered, a slowly converging lower bounding procedure may mask the impact of the choice of parameters on the upper bounding procedure. In order to explore the possible adverse impact of an aggressive choice of r^g , consider the dense population problem (DP) described by

$$\begin{aligned} \mathcal{X} &= [0, 6], \quad \mathcal{Y} = [2, 6], \\ f(x) &= 10 - x, \quad g(x, y) = \frac{y^2}{1 + \exp(-40(x - y))} + x - y - 2. \end{aligned} \tag{DP}$$

Figure 1 depicts the objective function $f(\cdot)$, multiple instances $g(\cdot, y^i)$ of the semi-infinite constraint, as well as the parametric optimal objective value $g^*(\cdot)$ of (LLP). While the problem is equivalently represented by a single instance of the semi-infinite constraint ($y = 2$), it is expected that this instance is not found immediately by Algorithm 1 because it is the solution to (LLP) for values $\bar{x} \in \mathcal{X}$ with relatively large objective value $f(\bar{x})$. Rather, it is expected that the upper bounding procedure furnishes \bar{x}^{UBD} of large value. The corresponding solutions of (LLP) in turn yield large values \bar{y} . For these parameter values, $g(x, \bar{y}) \leq 0$ holds for values x that are only slightly lower than \bar{x}^{UBD} . Therefore, if ε^g is very small, many iterations are required, until an SIP-feasible point is found. Furthermore, since (LBD) is equivalent to (UBD) for $\varepsilon^g = 0$, it is also expected to converge slowly on this problem.

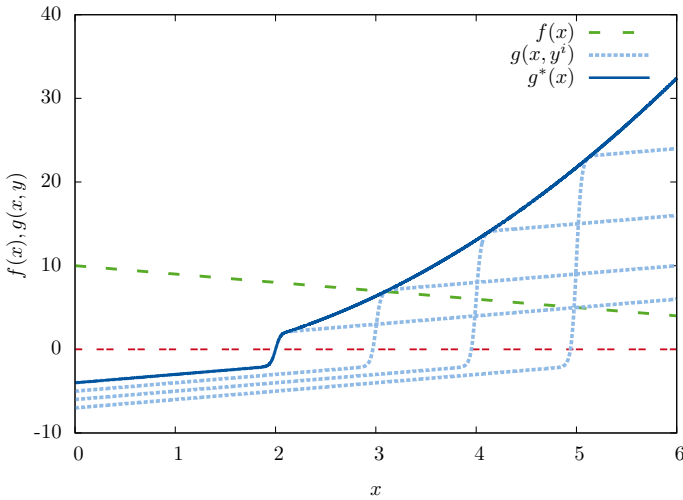


Fig. 1 Benchmark problem (DP): Objective function $f(\cdot)$, constraint function instances $g(\cdot, y^i)$ for $y^i \in \{2, 3, 4, 5\}$, and parametric optimal objective value of (LLP) $g^*(\cdot)$. Optimal solution: $x^* = 2$ with $f(x^*) = 8$

Table 1 Benchmark results for (UBD) on problem (DP)

r^g	CPU(s)		$ \mathcal{Y}^{UBD} $
	(UBD)	(LLP)	
1.2	0.94	0.81	3
2	1.26	1.32	11
4	1.64	1.75	18

The results were obtained by initializing Algorithm 1 with the exact globally optimal objective value $f(x^*) = 8$ as lower bound. Further initial settings were an empty initial discretization $\mathcal{Y}^{UBD,0} = \emptyset$, a conservative value for the initial restriction parameter $\varepsilon^{g,0} = 8$, and tolerances $\varepsilon^f = 10^{-3}$ and $\varepsilon^{UBD} = \varepsilon^{UBD,LLP} = 10^{-4}$. Multiple different values for r^g were chosen and $r^{UBD,LLP} = 10$ was chosen in all instances. Note that the choice of $r^{UBD,LLP}$ was irrelevant for the given results since no need for tolerance refinement was encountered during optimization. Details on the implementation of Algorithm 1 are provided in Sect. 6

Table 1 contains the benchmark results for the upper bounding procedure in problem (DP). It can be observed that the number discretization points that are added to \mathcal{Y}^{UBD} depends on the reduction parameter r^g . The more aggressive the choice, the more points are added due to SIP-infeasible iterates of the upper bounding procedure. Furthermore, the computational times required for the upper bound to converge are adversely impacted by aggressive choices for r^g . In the following section, a strategy is presented that aims at providing an adaptive restriction update that remains conservative on problems like (DP) and provides aggressive parameter updates when possible.

4 Hybrid algorithm

In this section, a hybrid algorithm is derived by employing subproblems from Algorithm 1 and an oracle problem adapted from Tsoukalas and Rustem [24]. The hybridization provides an adaptive update to the restriction parameter ε^g in order to circumvent the problem of a

dense discretization \mathcal{Y}^{UBD} . The aim is for the update to reduce the restriction parameter quickly when possible while avoiding a dense population of \mathcal{Y}^{UBD} in problems similar to (DP). The original oracle problem is reviewed in Sect. 4.1 and adapted for hybridization in Sect. 4.2. In Sect. 4.3, the hybrid algorithm is formally stated and a proof of finite convergence is given in Sect. 4.4.

4.1 An oracle problem

Similarly to Algorithm 1, the algorithm proposed by Tsoukalas and Rustem [24] solves (SIP) by solving multiple finite subproblems which are obtained through discretization of the parameter set \mathcal{Y} . A so-called oracle problem is solved to determine whether or not a certain target objective function value is attainable by (SIP). Based on this problem, a binary search of the objective function space is performed, which under slightly stronger assumptions than previously, yields an ε^f -optimal point finitely. In a previous publication [25], the authors use a stochastic global optimization algorithm to solve the subproblems of a similar algorithm for generalized semi-infinite programs and related problems.

The central subproblem of the algorithm is the oracle problem (ORA). For given lower and upper bounds LBD and UBD on f^* , a target objective value is set to $f^{ORA} = \frac{1}{2}(LBD + UBD)$. With a finite discretization $\mathcal{Y}^{ORA} \subseteq \mathcal{Y}$ the oracle problem is the program

$$\min_{x \in \mathcal{X}} \max \left\{ f(x) - f^{ORA}, \max_{y \in \mathcal{Y}^{ORA}} g(x, y) \right\}.$$

To facilitate comparability with the subproblems (LBD) and (UBD), the oracle problem is here rewritten equivalently as

$$\begin{aligned} [-\zeta^+, -\bar{\zeta}] \ni -\zeta^* &= \min_{x \in \mathcal{X}, \zeta \in \mathbb{R}} -\zeta \\ \text{s.t. } f(x) - f^{ORA} &\leq -\zeta \\ g(x, y) &\leq -\zeta, \quad \forall y \in \mathcal{Y}^{ORA} \\ \mathcal{Y}^{ORA} &\subseteq \mathcal{Y}, \quad |\mathcal{Y}^{ORA}| < \infty. \end{aligned} \tag{ORA}$$

The oracle problem can be understood as a decision problem that determines whether or not the target objective value f^{ORA} is attainable by (LBD) with the discretization $\mathcal{Y}^{LBD} = \mathcal{Y}^{ORA}$. If ζ^* is negative, then there exists no point $\bar{x}^{ORA} \in \mathcal{X}$ that is feasible with respect to the discretized semi-infinite constraint and has an objective value $f(\bar{x}^{ORA}) \leq f^{ORA}$. Consequently, f^{ORA} is a lower bound to (LBD) with $\mathcal{Y}^{LBD} = \mathcal{Y}^{ORA}$ and by the properties of (LBD) also a lower bound to (SIP). If, in contrast, ζ^* is non-negative and the solution point \bar{x}^{ORA} of (ORA) is SIP-feasible, $f(\bar{x}^{ORA})$ is an upper bound to (SIP). Considering that f^{ORA} is always chosen below the current upper bound on (SIP), $f(\bar{x}^{ORA})$ is an update to the upper bound. The third possible outcome ($\zeta^* \geq 0$ and \bar{x}^{ORA} SIP-infeasible) does not yield information about bounds on the globally optimal objective value of (SIP). In this case, \mathcal{Y}^{ORA} is refined to yield a conclusive instance of (ORA) in a following iteration.

Remark 6 Due to the possibility of an instance of (ORA) being non-conclusive, the number of iterations needed to obtain a certain optimality tolerance is not known a priori.

Due to the choice $f^{ORA} = \frac{1}{2}(LBD + UBD)$, the bound update resulting from the solution of (ORA) will always be the update that closes the gap between lower and upper bound the furthest. Indeed, consider the case $f^{ORA} \geq f^*$. Obviously, f^{ORA} is an upper bound and when an SIP-feasible point \bar{x}^{ORA} is found, it holds

$$UBD - f(\bar{x}^{ORA}) \geq UBD - f^{ORA} = f^{ORA} - LBD \geq f^* - LBD,$$

where $UBD - f(\bar{x}^{ORA})$ is the reduction in the gap due to the new upper bound $f(\bar{x}^{ORA})$ and $f^* - LBD$ is the maximum possible reduction in the gap due to an exact lower bound. Similarly, if $f^{ORA} < f^*$, it holds

$$f^{ORA} - LBD = UBD - f^{ORA} > UBD - f^*,$$

where $UBD - f^*$ is the maximum possible reduction in the gap due to an exact upper bound.

The second subproblem is the lower-level program (LLP). It is solved to determine SIP-feasibility of a point \bar{x}^{ORA} or populate the discretization \mathcal{Y}^{ORA} .

The algorithm must be initialized with lower and upper bounds on f^* based on which the binary search on the objective function space is performed. Finite termination of the algorithm is contingent on Assumptions 1, 2, as well as the exact solution of all subproblems. Obviously, the latter assumption is stronger than both Assumption 4 and the original assumption by Mitsos [11] on the approximate solution of subproblems. Furthermore, it is required that the case $f^{ORA} = f^*$ does not occur and that $g^*(\cdot)$ does not have a local minimum with value 0 at the exact global solution of (SIP). This last assumption can be compared to Assumption 3. Given the structure of (SIP), the assumption by Tsoukalas and Rustem [24] gives $\tilde{\epsilon}^f$ -optimal SIP-Slater points for arbitrarily small $\tilde{\epsilon}^f$ and is therefore stronger than Assumption 3. This relation has previously been observed by Mitsos and Tsoukalas [14] for similar assumptions used in the context of generalized semi-infinite programs in [25] and [14]. A formal statement of the oracle algorithm is given in ‘‘Appendix 1’’.

4.2 Adapted oracle problem

With respect to Algorithm 1 and especially with respect to a dense population of \mathcal{Y}^{UBD} , an alternative interpretation of (ORA) is useful. In (ORA), the dummy variable $-\zeta$ is bounded from below by $f(x) - f^{ORA}$ and also by $g(x, y)$ for all $y \in \mathcal{Y}^{ORA}$. Therefore, the objective of (RES) is a compromise between minimizing $f(x)$ and maximizing the restriction of the discretized semi-infinite constraint such that the solution point remains feasible with respect to the restricted constraints. In particular, if (ORA) returns $\zeta^* \geq 0$, a point \bar{x}^{ORA} is guaranteed to exist with objective value $f(\bar{x}^{ORA}) \leq f^{ORA}$ that is feasible in (UBD) with $\mathcal{Y}^{UBD} = \mathcal{Y}^{ORA}$ and $\epsilon^g = \zeta^*$. If beyond that, \bar{x}^{ORA} is SIP-feasible, $f(\bar{x}^{ORA})$ is an upper bound to (SIP), and $\epsilon^g = \zeta^*$ is a value for the restriction parameter under which the current upper bound is attainable by (UBD) with $\mathcal{Y}^{UBD} = \mathcal{Y}^{ORA}$. Since the objective of a subsequent solution of (UBD) is to improve on the upper bound, $\epsilon^g = \zeta^*$ may be considered a sensible choice for (UBD). However, due to the involvement of term $f(x) - f^{ORA}$, ζ^* may still be an aggressive choice, which would not solve the problem of a dense population of \mathcal{Y}^{UBD} described in Sect. 3.4.

For this reason, we propose the adapted problem (RES) to provide lower and upper bounds as well as a truly conservative choice of the restriction parameter.

$$\begin{aligned} [-\eta^+, -\bar{\eta}] \ni -\eta^* &= \min_{x \in \mathcal{X}, \eta \in \mathbb{R}} -\eta \\ \text{s.t. } f(x) - f^{RES} &\leq 0 \\ g(x, y) &\leq -\eta, \quad \forall y \in \mathcal{Y}^{RES} \\ \mathcal{Y}^{RES} &\subsetneq \mathcal{Y}, \quad |\mathcal{Y}^{RES}| < \infty \end{aligned} \tag{RES}$$

Note that (RES) has the same bounding and decision properties as (ORA). If η^* is negative, f^{RES} is a lower bound to (SIP). If η^* is non-negative and the solution point \bar{x}^{RES} is SIP-feasible, $f(\bar{x}^{RES})$ is an upper bound to (SIP) and provided $f^{RES} < UBD$ an update to the upper bound. Beyond that, in the case of an SIP-feasible solution \bar{x}^{RES} , $\varepsilon^g = \eta^*$ is the largest possible restriction under which $f(\bar{x}^{RES})$ is attainable by (UBD) for $\mathcal{Y}^{UBD} = \mathcal{Y}^{RES}$. Therefore, provided that (UBD) and (RES) utilize the same discretization, any choice $\varepsilon^g > \eta^*$ will not decrease the upper bound further. Since (RES) considers the objective function only in form of a constraint, it is possible that (UBD) can improve on the upper bound for $\varepsilon^g = \eta^*$. However, it is expected that this will only yield small improvements and that a choice $\varepsilon^g = \eta^*/r^g$ with a small value for r^g will yield better results.

While these properties are desirable in the context of an adaptive update of ε^g , (RES) cannot be guaranteed to be conclusive regardless of the discretization \mathcal{Y}^{RES} under Assumptions 1, 2, 3, and 4. While very unlikely, the following proposition illustrates the point.

Proposition 1 *Let $f^{RES} = f^*$. Then, under Assumptions 1, 2, 3, and 4, (RES) cannot be guaranteed to be conclusive.*

Proof Since f^{RES} is the exact global solution to (SIP), it is both an exact lower and upper bound. We will show that based on approximate solution of (RES) and (LLP), it cannot be guaranteed to be confirmed to be either.

Due to $f^{RES} = f^*$, it holds $\eta^+ \geq 0$ irrespective of \mathcal{Y}^{RES} . Therefore, the condition for the detection of a lower bound ($\eta^+ < 0$) cannot be reached.

Furthermore, consider the following example, for which f^{RES} cannot be found to be an upper bound. Suppose that the exact globally optimal solution x^* of (SIP) is unique and that it holds $g^*(x^*) = 0$, which does not conflict with any of the assumptions. Then, x^* is the only SIP-feasible point that is also feasible in (RES). Therefore, in order to decide f^{RES} to be an upper bound to (SIP), (LLP) must be solved for x^* such that $g^+(x^*) = 0$. This, in turn, can only be guaranteed under the assumption of exact solution of (LLP). □

As mentioned, this scenario is very unlikely and numerical experiments confirm that this property has little impact on algorithm performance. However, to guarantee convergence under the given assumptions, it must be ensured that (RES) cannot lead to an infinite loop in which discretization points are added to \mathcal{Y}^{RES} . This is done by deferring to (LBD) whenever (RES) fails to be conclusive.

4.3 Algorithm statement

Here, the hybrid algorithm is stated formally based on the subproblems (LBD), (UBD), (RES), and (LLP). All subproblems are assumed to be solved approximately according to Definition 1 with their respective tolerances $\varepsilon^{LBD}, \varepsilon^{UBD}, \varepsilon^{RES}, \varepsilon^{LLP} > 0$. The subproblems (LBD), (UBD), and (RES) operate on a single discretization set $\mathcal{Y}^{LBD} = \mathcal{Y}^{UBD} = \mathcal{Y}^{RES} = \mathcal{Y}^D$ to leverage the properties discussed in Sect. 4.1. Due to the shared discretization, there is also a shared tolerance ε^{LLP} used for all problems to ensure the appropriate accuracy of the entries in the discretization.

Algorithm 2: Hybrid approach. User inputs are the initial discretization set $\mathcal{Y}^{D,0} \subsetneq \mathcal{Y}$, the initial restriction parameter $\varepsilon^{g,0} > 0$ and the corresponding reduction parameter $r^g > 1$, the initial lower-level tolerance $\varepsilon^{LLP,0} > 0$ and the corresponding reduction parameter $r^{LLP} > 1$, and the tolerances $\varepsilon^{LBD}, \varepsilon^{UBD}, \varepsilon^{RES}, \varepsilon^f > 0$.

- 1 Set $LBD \leftarrow -\infty, UBD \leftarrow \infty, f^{RES} \leftarrow \infty, \mathcal{Y}^D \leftarrow \mathcal{Y}^{D,0}, \varepsilon^g \leftarrow \varepsilon^{g,0}, \varepsilon^{LLP} \leftarrow \varepsilon^{LLP,0}, l \leftarrow 0$.
- 2 Solve (LBD) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{LBD}$ to obtain $f^{LBD,-}$ and \bar{x}^{LBD} .
- 3 Set $LBD \leftarrow f^{LBD,-}$.
- 4 Solve (LLP) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{LLP}$ for \bar{x}^{LBD} to obtain $g^+(\bar{x}^{LBD})$ and \bar{y} .
- 5 **if** $g^+(\bar{x}^{LBD}) \leq 0$ **then**
- 6 | Set $UBD \leftarrow f(\bar{x}^{LBD}), x^* = \bar{x}^{LBD}$ and terminate.
- 7 **else if** $g(\bar{x}^{LBD}, \bar{y}) > 0$ **then**
- 8 | Set $\mathcal{Y}^D \leftarrow \mathcal{Y}^D \cup \{\bar{y}\}$.
- 9 **else**
- 10 | Set $\varepsilon^{LLP} \leftarrow (g^+(\bar{x}^{LBD}) - g(\bar{x}^{LBD}, \bar{y}))/r^{LLP}$.
- 11 **end**
- 12 Solve (UBD) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{UBD}$ to obtain $f^{UBD,-}$ and \bar{x}^{UBD} .
- 13 **if** (UBD) is feasible **then**
- 14 | **if** $\varepsilon^{LLP} \geq \varepsilon^g$ **then** Set $\varepsilon^{LLP} \leftarrow \varepsilon^g/r^{LLP}$.
- 15 | Solve (LLP) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{LLP}$ for \bar{x}^{UBD} to obtain $g^+(\bar{x}^{UBD})$ and \bar{y} .
- 16 | **if** $g^+(\bar{x}^{UBD}) \leq 0$ **then**
- 17 | | **if** $f(\bar{x}^{UBD}) < UBD$ **then** Set $UBD \leftarrow f(\bar{x}^{UBD}), x^* \leftarrow \bar{x}^{UBD}$.
- 18 | | Set $\varepsilon^g \leftarrow \varepsilon^g/r^g$.
- 19 | **else**
- 20 | | Set $\mathcal{Y}^D \leftarrow \mathcal{Y}^D \cup \{\bar{y}\}$ and go to Line 12.
- 21 | **end**
- 22 **else**
- 23 | Set $\varepsilon^g \leftarrow \varepsilon^g/r^g$ and go to Line 12.
- 24 **end**
- 25 **if** $UBD - LBD \leq \varepsilon^f$ **then** Terminate.
- 26 Set $l \leftarrow 0$.
- 27 Solve (RES) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{RES}$ to obtain $\eta^+, \bar{\eta}$, and \bar{x}^{RES} .
- 28 **if** $\eta^+ < 0$ **then**
- 29 | Set $LBD \leftarrow f^{RES}, f^{RES} \leftarrow \frac{1}{2}(LBD + UBD)$ and go to Line 27.
- 30 **else if** $\bar{\eta} > 0$ **then**
- 31 | Solve (LLP) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{LLP}$ for \bar{x}^{RES} to obtain $g^+(\bar{x}^{RES})$ and \bar{y} .
- 32 | **if** $g^+(\bar{x}^{RES}) \leq 0$ **then**
- 33 | | **if** $\eta^+/r^g < \varepsilon^g$ **then** Set $\varepsilon^g \leftarrow \eta^+/r^g$.
- 34 | | Set $UBD \leftarrow f(\bar{x}^{RES}), x^* \leftarrow \bar{x}^{RES}$ and go to Line 12.
- 35 | **else if** $g(\bar{x}^{RES}, \bar{y}) > 0$ and $l \leq l_{max}$ **then**
- 36 | | Set $\mathcal{Y}^D \leftarrow \mathcal{Y}^D \cup \{\bar{y}\}, l \leftarrow l + 1$ and go to Line 27.
- 37 | **else**
- 38 | | Go to Line 2.
- 39 | **end**
- 40 **else**
- 41 | Go to Line 2.
- 42 **end**

Remark 7 Similar to Algorithm 1, the algorithm could be changed to check for convergence after each update of the bounds, which would potentially save the solution of one subproblem instance.

In Algorithm 2, the lower bounding problem (LBD) serves mainly the purpose of initializing the lower bound on (SIP). Due to the decision properties discussed in Sect. 4.1, (RES) is preferred for lower bounding. (LBD) is only used in the main loop of the algorithm if (RES) fails to be conclusive. Note that (RES) can only loop onto itself a finite number of times before it either provides a bound update or defers to one of the subproblems from Algorithm 1. Therefore, Algorithm 2 inherits the convergence guarantees from Algorithm 1. Nevertheless, for the sake of completeness, a formal proof of convergence is given in the following section.

4.4 Finite termination

The proof of finite termination for Algorithm 2 relies on Lemmata 1, 2, 3, and 4.

Theorem 2 Let ε^f , ε^{LBD} , and ε^{UBD} be chosen such that

$$\varepsilon^f > \tilde{\varepsilon}^f + \varepsilon^{LBD} + \varepsilon^{UBD}.$$

Then, under Assumptions 1, 2, 3, and 4, Algorithm 2 terminates finitely with an $\tilde{\varepsilon}^f$ -optimal point \mathbf{x}^* .

Proof By Lemma 2, the lower bounds $f^{LBD,-,k}$ to (LBD) converge to an ε^{LBD} -underestimate of f^* , i.e.,

$$\lim_{k \rightarrow \infty} f^{LBD,-,k} \in [f^* - \varepsilon^{LBD}, f^*].$$

Therefore, for any $\delta > 0$ there exists a K^{LBD} such that

$$f^{LBD,-,k} > f^* - \varepsilon^{LBD} - \delta, \quad \forall k > K^{LBD}. \tag{8}$$

By Lemma 4, after a finite number K^{UBD} of instances of (UBD), the upper bounding procedure furnishes an $(\tilde{\varepsilon}^f + \varepsilon^{UBD})$ -optimal point $\hat{\mathbf{x}}^{UBD}$. It holds

$$f(\hat{\mathbf{x}}^{UBD,l}) \leq f^* + \tilde{\varepsilon}^f + \varepsilon^{UBD}, \quad \forall l > K^{UBD}. \tag{9}$$

Combining (9) and (10), we obtain

$$f(\hat{\mathbf{x}}^{UBD,l}) - f^{LBD,-,k} < \tilde{\varepsilon}^f + \varepsilon^{UBD} + \varepsilon^{LBD} < \varepsilon^f, \quad \forall k > K^{LBD}, l > K^{UBD}$$

The first solution of (LBD) is guaranteed to provide a finite value for the lower bound $LBD > -\infty$. Similarly, after a finite number of instances of (UBD), a finite value for the upper bound $UBD < \infty$ is found. Therefore, the gap $\Delta f = UBD - LBD$ is finite when (RES) is solved for the first time.

For any $f^{RES} = \frac{1}{2}(LBD + UBD)$, (RES) is only solved a finite number of times. At this point, if a lower bound $LBD = f^{RES}$ is obtained, (RES) is solved for the updated f^{RES} . If an upper bound $UBD = f(\hat{\mathbf{x}}^{RES}) \leq f^{RES}$ is obtained, (UBD) is solved. Otherwise, (LBD) and (UBD) are solved. After at most

$$K^{RES} = \log_2 (\Delta f / \varepsilon^f) \tag{10}$$

bound updates by (RES), the optimality tolerance ε^f is reached.

(RES) either provides K^{RES} bound updates or refers K^{LBD} times to (LBD). Either way, the optimality tolerance ε^f is achieved finitely. \square

Note that the tolerance ε^{RES} is not required to fulfill any condition. This is due to the fact, that the convergence guarantee is inherited from Algorithm 1. The same is true for the solutions of (LLP) at the solution points of (RES).

5 Generalizations

In this section, the generalizations to multiply constrained SIPs and mixed-integer SIPs are discussed. For the latter case, the discussion provided by Mitsos [11] is fully applicable to Algorithm 2 and consequently, these problems pose no additional challenge as long as an MINLP-solver is available that satisfies Assumption 4.

Consider the generalization of (SIP) to multiple semi-infinite constraints according to

$$\begin{aligned}
 f^* &= \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \\
 \text{s.t. } &g_i(\mathbf{x}, \mathbf{y}) \leq 0, \quad i = 1, \dots, n_g, \quad \forall \mathbf{y} \in \mathcal{Y} \\
 &\mathcal{X} \subseteq \mathbb{R}^{n_x}, \quad \mathcal{Y} \subseteq \mathbb{R}^{n_y}, \quad |\mathcal{Y}| = \infty.
 \end{aligned}$$

Algorithm 2 is applicable with minor alterations. In particular, multiple lower-level problems must be solved to determine SIP-feasibility of a given point $\bar{\mathbf{x}}$. Furthermore, two questions emerge concerning the number of discretizations and restriction parameters to use. With respect to the number of discretizations, there is no reason to expect the use of one shared discretization is justified in the general case. This was observed previously by Mitsos [11]. With respect to the number of restriction parameters, the introduction of the restriction update changes prior observations. There are two possibilities to preserve the bounding and decision properties of (RES) when considering multiple constraints:

1. One instance of (RES) is used, in which η is bounded by all constraints. This results in a single update to a shared restriction parameter.
2. An instance of (RES) is introduced for each constraint, such that η is bounded by only one constraint, while all other constraints are considered in their original form. This results in a dedicated restriction parameter for each constraint.

While the latter option is preferable in terms of the conservatism of the updates to the restriction, it has disadvantageous solution properties. Indeed, such a problem is similar to (LBD) with respect to the constraints not bounding η . As a consequence, the finite generation of an SIP-feasible point cannot be guaranteed even under strong assumptions. Therefore, the former option seems more promising.

6 Numerical results

Algorithms 1, 2, and 3 are implemented in the general algebraic modeling system (GAMS) 24.3.1 [17] with the GAMS-F preprocessor. Note that in the implementation of Algorithm 3 in “Appendix 1”, a single instance of (LBD) is solved to initialize the lower bound with a finite value. This initialization is not part of the original algorithm by Tsoukalas and Rustem [24] and is added to remove the requirement for an initial lower bound. Furthermore, while

the authors require a finite initial value for the upper bound, this is not necessary since the first instance of (ORA) with $f^{ORA} = \infty$ is guaranteed to provide a finite upper bound. The NLP subproblems are solved with BARON 14.0.2 [18, 23]. The implementation of the algorithms is available in the supplementary material. All numerical experiments were run on a 64-bit Intel Core i3-2100 CPU with 3.1 GHz running Windows 7.

The test set consists of small-scale SIPs with up to six variables, two parameters, and four semi-infinite constraints, most of which were proposed by Watson in [26] and in some cases corrected by Mitsos et al. in [13]. The problems 2, 5, 6, 7, 8, and 9 were selected from the Watson test set, taking the corrections by [13] into account. Problems 1, 3, and 4 were excluded due to the involvement of trigonometric functions, which cannot be handled by BARON. Beyond that, the problems H, N, S, and the design centering problems D1_0_1 through D2_1_2 by Mitsos were added alongside the newly proposed dense population problem (DP). All benchmark problems are available in [9] and in the supplementary material. The labels of the problems are kept in line with the labels in the original publications while the dense population problem is labeled as DP.

All algorithms are initialized with empty initial discretizations $\mathcal{Y}^{LBD,0} = \mathcal{Y}^{UBD,0} = \mathcal{Y}^{ORA,0} = \mathcal{Y}^{D,0} = \emptyset$. The initial restriction parameter $\varepsilon^{g,0}$ is set to 1 in all cases but the problem DP, where a value of 8 is used. For all cases, the optimality tolerances are set to $\varepsilon^f = 10^{-3}$ and $\varepsilon^{LBD} = \varepsilon^{UBD} = \varepsilon^{ORA} = \varepsilon^{RES} = \varepsilon^{LLP,0} = \varepsilon^{LBD,LLP,0} = \varepsilon^{UBD,LLP,0} = \varepsilon^{ORA,LLP,0} = \varepsilon^{RES,LLP,0} = 10^{-4}$. ε^f , ε^{LBD} , ε^{UBD} , ε^{ORA} , and ε^{RES} are used as both absolute and relative tolerances. Further NLP-solver options are $\text{Ipsol} = 3$, $\text{nIpsol} = 4$, $\text{contol} = 10^{-12}$, $\text{boxtol} = 10^{-3}$, and $\text{inttol} = 10^{-9}$. The complete results are given in Tables 2, 3, 4 and 5 in “Appendix 2”.

In general terms, it can be observed that all algorithms show promise with respect to their scaling properties, which has previously been observed for Algorithm 1 in [11]. Indeed, while problems 7, 8, and 9 exhibit two-dimensional parameter sets \mathcal{Y} , they only require moderately more or similarly many NLP solutions as the problems with one-dimensional parameter sets. Note that the design centering problems D1_0_1 through D2_1_2 exhibit up to four semi-infinite constraints, each generating a lower-level problem that is solved to establish SIP-feasibility. Therefore, the relatively large number of NLPs solved is expected. Furthermore, the majority of the relatively large CPU requirement on these problems is due the relatively expensive problems (LBD), (UBD), (RES), and (ORA) with up to six dimensions in \mathcal{X} . This indicates that for the tested algorithms, the difficulty of an SIP is mainly determined by the difficulty of the NLP subproblems. Note however that this inference is drawn from the numerical results on a limited test set and does not derive from a theoretical analysis of the scaling properties of the algorithms.

Figure 2 depicts the performance plot comparing the tested algorithms on all problems in the test set. It can be observed that only the algorithm by Mitsos (Algorithm 1) and the hybrid algorithm (Algorithm 2) solve all problems. This is expected because of the stronger assumptions required for convergence of Algorithm 3. Indeed, in every case the failure of Algorithm 3 is due to either (ORA) or (LLP) failing to be conclusive. Recall that this is due to the use of Algorithm 3 with the approximate solution of subproblems, which violates the assumptions in [24]. Therefore, finite convergence is not guaranteed and the large number of failed problem instances can be explained.

Beyond the number of solved problems, the hybrid algorithm offers the best overall performance, both in terms of the number of quickest solutions and in terms of the time factor in adverse cases. Particularly the superior performance compared to Algorithm 1 with aggressive parameter choice $r^g = 2$, which in turn is superior to Algorithm 1 with $r^g = 1.2$

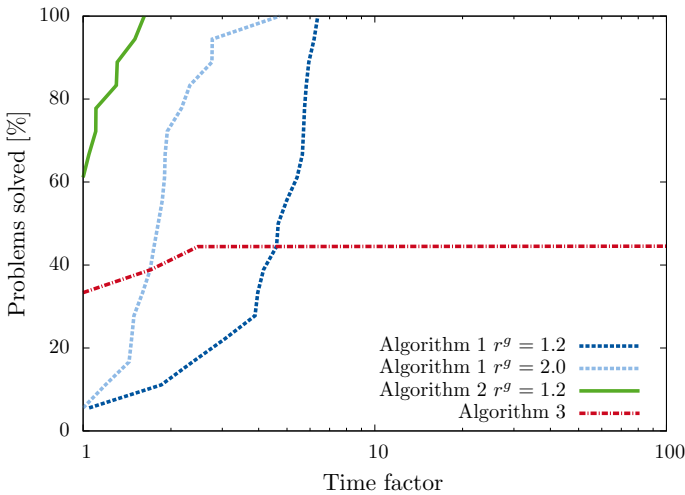


Fig. 2 Performance plot for the algorithm by Mitsos [11], Algorithm 1 ($r^g = 1.2, 2.0$), the algorithm by Tsoukalas and Rustem [24], Algorithm 3, and the hybrid algorithm, Algorithm 2 ($r^g = 1.2$)

indicates that the adaptive restriction allows the hybrid algorithm to produce competitive results despite employing a conservative reduction parameter.

7 Conclusions

A hybrid of two existing discretization algorithms for the global solution of SIPs is proposed. The algorithm employs a lower and an upper bounding problem of a bounding algorithm [11] and an adapted problem from an oracle algorithm [24]. The upper bounding problem is based on discretization and the restriction of the right-hand-side of semi-infinite constraints. Adaptive updates to the restriction parameter are introduced through the hybridization with the oracle algorithm. By virtue of the adaptive restriction, the algorithm hedges against a dense population the discretization of the parameter set and the associated poor performance. Employing a loosened assumption on the solution of the NLP subproblems, the algorithm is guaranteed to terminate finitely with an ε^f -optimal point finitely under weaker assumptions than its predecessors. The superiority of the hybrid approach is demonstrated through theoretical discussion and numerical experiments.

One possible variant of the hybrid algorithm is the solution of the added oracle problem by stochastic solvers as was done by the authors of the original oracle algorithm [25]. Possibly improving overall performance through the quicker solution of the oracle problem, this would leave the convergence guarantees intact since the solutions of the oracle problem do not need to fulfill any conditions for the algorithm to converge. Another possible task for future work is the initialization of the discretization sets to nonempty sets. While empty sets are used here to investigate the core functionality of the algorithms, nonempty sets promise performance gains, which may allow the application of the proposed methods to larger problems.

Acknowledgements We gratefully acknowledge the financial support from Deutsche Forschungsgemeinschaft (German Research Foundation) through the project “Optimization-based Control of Uncertain Systems” PAK 635/1; MA1188/34-1.

Appendix 1: Oracle algorithm

Here, the oracle algorithm is stated formally. As mentioned previously, Tsoukalas and Rustem [24] consider the exact solution of all subproblems. However, exact solutions cannot be provided for the investigated problems. Algorithm 3, is an adaptation of the original algorithm by Tsoukalas and Rustem [24] and considers the approximate solution of the subproblems according to Definition 1. Note that the convergence proof by Tsoukalas and Rustem [24] does not hold for Algorithm 3.

Algorithm 3: Oracle approach by Tsoukalas and Rustem [24]. User inputs are the initial discretization $\mathcal{Y}^{ORA,0} \subseteq \mathcal{Y}$, the initial lower and upper bounds LBD^0 and UBD^0 , the tolerances $\varepsilon^{ORA}, \varepsilon^{ORA,LLP}, \varepsilon^f > 0$

- 1 Set $LBD \leftarrow LBD^0, UBD \leftarrow UBD^0, \mathcal{Y}^{ORA} \leftarrow \mathcal{Y}^{ORA,0}$.
 - 2 Set $f^{ORA} \leftarrow \frac{1}{2}(LBD + UBD)$.
 - 3 Solve (ORA) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{ORA}$ to obtain $\zeta^+, \bar{\zeta}$, and \bar{x}^{ORA} .
 - 4 **if** $\zeta^+ < 0$ **then**
 - 5 | Set $LBD = f^{ORA}$.
 - 6 **end**
 - 7 Solve (LLP) globally according to Definition 1 with $\varepsilon^{NLP} = \varepsilon^{ORA,LLP}$ for \bar{x}^{ORA} to obtain $g^+(\bar{x}^{ORA})$ and \bar{y} .
 - 8 **if** $g^+(\bar{x}^{ORA}) \leq 0$ **then**
 - 9 | Set $UBD = f(\bar{x}^{ORA}), x^* = \bar{x}^{ORA}$.
 - 10 **else**
 - 11 | Set $\mathcal{Y}^{ORA} \leftarrow \mathcal{Y}^{ORA} \cup \{\bar{y}\}$.
 - 12 **end**
 - 13 **if** $UBD - LBD \leq \varepsilon^f$ **then** Terminate. **else** Go to Line 2.
-

Appendix: Numerical results

Table 2 Numerical results for Algorithm 1 for $r^{\delta} = 1.2$

Label	#NLPs	CPU(s)	Total			UBD	x^*
			(LBD)	(UBD)	(UBD)		
2	152	14.17	4.05 + 3.11	4.44 + 2.57	0.1943660	0.1952461	(-0.75000005, -0.61866475) ^T
5	164	13.96	3.29 + 3.63	3.28 + 3.76	4.3007537	4.3043861	(1.00947663, -0.14703292, -0.36103288) ^T
6	36	4.07	1.29 + 0.66	1.36 + 0.76	97.1491375	97.1601359	(0.71968278, -1.44971945) ^T
7	180	13.56	3.48 + 3.15	3.50 + 3.43	0.9999000	1.0007876	(-1.00039374, 0.00000000, 0.00000000) ^T
8	192	43.28	3.92 + 20.18	3.73 + 15.45	2.4356432	2.4376640	(2.51195859, -3.94019138, -4.03485959, 4.14944012, 4.46152553, 4.24410832) ^T
9	184	15.20	3.89 + 3.57	3.65 + 4.09	-12.000	-11.988	(2.99707450, 0.00000000, 0.00000000, 0.06250031, 0.00000000, -0.06250031) ^T
DI_0_1	160	16.11	3.30 + 4.06	3.57 + 5.18	-0.8367600	-0.8358239	(0.00000666, 0.83582388, 0.83582388) ^T
DI_0_2	396	470.90	332.55 + 15.18	107.97 + 15.20	-1.0289177	-1.0280937	(-0.00021631, 0.83638699, 0.83638699, -0.80106843, 0.19170675, 0.19170675) ^T
DI_1_1	172	18.03	3.78 + 4.75	3.57 + 5.93	-0.7001000	-0.6992043	(0.00000716, 0.83618436, 0.83618436) ^T
DI_1_2	306	123.52	64.75 + 10.63	36.56 + 11.58	-0.7370272	-0.7360887	(-0.00012866, 0.83631756, 0.83631756, -0.800045793, 0.19147241, 0.19147241) ^T
D2_0_1	284	56.06	16.98 + 14.94	10.11 + 14.03	-0.6608349	-0.6599785	(0.12119353, 0.98557058, 0.65997850) ^T
D2_0_2	564	279.86	130.42 + 28.18	95.34 + 25.92	-0.9258905	-0.9251709	(0.00001047, 1.17739736, 0.63529427, 0.65890546, 0.52793707, 0.28987668) ^T
D2_1_1	290	57.74	16.27 + 17.43	10.26 + 13.78	-0.4366704	-0.4359102	(0.12119353, 0.98557059, 0.66023498) ^T
D2_1_2	664	883.91	666.78 + 33.77	151.36 + 32.00	-0.4910085	-0.4902218	(0.01684796, 1.15604322, 0.64185524, 0.67566306, 0.51163274, 0.27972070) ^T
DP	107	9.71	2.55 + 3.06	2.25 + 1.85	7.9992001	8.0035402	1.99645982
H	160	15.77	5.11 + 3.30	4.42 + 2.94	-0.0003124	0.0003066	(0.43750000, 0.00030657) ^T
N	164	12.11	3.16 + 2.71	3.12 + 3.12	-0.0001000	0.0008165	(0.00000000, 0.00081645) ^T
S	240	150.21	69.29 + 4.59	71.85 + 4.48	-5.0861277	-5.0818309	(4.22101885, 3.77349621) ^T

CPU times are given for solutions of subproblems and the associated solutions of (LLP)

Table 3 Numerical results for Algorithm 1 for $r^s = 2$

Label	#NLPs	CPU(s)	(LBD)		LBD	UBD	x^*
			Total	(UBD)			
2	48	4.54	1.37 + 0.99	1.33 + 0.85	0.1943660	0.1950059	(-0.75000005, -0.61847064) ^T
5	60	4.83	1.12 + 1.28	1.08 + 1.35	4.3007537	4.3031946	(1.00904771, -0.14709878, -0.36097237) ^T
6	36	3.88	1.26 + 0.74	1.20 + 0.68	97.1491375	97.2361974	(0.71187611, -1.45084232) ^T
7	56	4.04	1.12 + 0.91	1.01 + 1.00	0.9999000	1.0004883	(-1.00024414, 0.00000000, 0.00000000) ^T
8	100	19.08	1.91 + 7.89	1.78 + 7.50	2.4356385	2.4366916	(2.51098725, -3.94019064, -4.03486389, 4.14943835, 4.46152657, 4.24411159) ^T
9	92	7.63	1.78 + 1.81	1.96 + 2.08	-12.000	-11.992	(2.99804687, 0.00000000, 0.00000000, 0.06250039, 0.00000000, -0.06250039) ^T
D1_0_1	60	6.17	1.33 + 1.61	1.31 + 1.92	-0.8367600	-0.8358492	(0.00000890, 0.83584916, 0.83584916) ^T
D1_0_2	240	210.14	126.20 + 8.87	65.88 + 9.19	-1.0289295	-1.0280121	(0.00020770, 0.89075918, 0.81196061, -0.77540705, 0.21605144, 0.21605144) ^T
D1_1_1	64	6.47	1.35 + 1.55	1.39 + 2.18	-0.7001000	-0.6993300	(0.00001016, 0.83625950, 0.83625950) ^T
D1_1_2	138	46.65	20.14 + 5.10	16.89 + 4.52	-0.7370249	-0.7363415	(-0.00012651, 0.83640029, 0.83640029, -0.80111862, 0.19177072, 0.19177072) ^T
D2_0_1	128	22.38	5.21 + 6.64	5.11 + 5.42	-0.6608348	-0.6598686	(0.14859061, 0.93430069, 0.65986861) ^T
D2_0_2	326	136.40	58.80 + 15.59	46.86 + 15.15	-0.9287820	-0.9279614	(-0.00000245, 1.17842722, 0.63602590, 0.66036008, 0.52648113, 0.29193551) ^T
D2_1_1	152	26.95	6.46 + 7.07	6.55 + 6.87	-0.4366704	-0.4358170	(0.13001363, 0.96906496, 0.66016440) ^T
D2_1_2	396	334.61	216.56 + 19.08	80.38 + 18.59	-0.4912243	-0.4907927	(0.01696468, 1.15615599, 0.64207608, 0.67572133, 0.51063658, 0.28023386) ^T
DP	109	9.82	2.62 + 2.89	2.03 + 2.28	7.9992060	8.0000060	1.99999405
H	180	16.18	4.96 + 3.50	4.46 + 3.26	-0.0002792	0.0002441	(0.95311422, 0.00024409) ^T
N	52	3.55	0.93 + 0.79	0.94 + 0.89	-0.0001000	0.0004883	(0.00000000, 0.00048828) ^T
S	100	52.23	23.67 + 1.83	24.85 + 1.88	-5.0861205	-5.0825993	(3.77335028, 4.22079986) ^T

CPU times are given for solutions of subproblems and the associated solutions of (LLP)

Table 4 Numerical results for Algorithm 2

Label	#NLPs	CPU(s)		LBD	UBD	x^*		
		Total	(RES)					
2	26	2.51	0.07 + 0.06	0.54 + 0.38	1.15 + 0.31	0.1942270	0.1950138	(-0.74999995, -0.61847700) ^T
5	43	3.36	0.16 + 0.17	0.61 + 0.62	0.95 + 0.85	4.3004637	4.3026235	(1.00730322, -0.12869981, -0.37807535) ^T
6	20	5.08	0.04 + 0.12	0.29 + 0.12	4.34 + 0.17	97.1531422	97.2391444	(0.71156820, -1.45085585) ^T
7	36	2.38	0.06 + 0.06	0.61 + 0.50	0.70 + 0.45	0.9996239	1.0005338	(-1.00026685, 0.00000000, 0.00000000) ^T
8	69	11.41	0.07 + 0.07	0.92 + 3.23	2.20 + 4.92	2.4343850	2.4364040	(2.58151125, -4.13198533, -4.08953932, 4.26987320, 4.53288640, 4.22742719) ^T
9	74	6.40	0.03 + 0.12	1.36 + 1.73	1.84 + 1.32	-12.008	-11.998	(2.99940734, 0.00000000, 0.00000000, 0.03125037, 0.00000000, -0.03125037) ^T
D1_0_1	33	3.23	0.07 + 0.07	0.47 + 0.77	0.97 + 0.88	-0.8370277	-0.8362545	(-0.00002483, 0.83625455, 0.83625455) ^T
D1_0_2	103	76.05	0.14 + 0.19	6.56 + 1.36	62.23 + 5.57	-1.0291912	-1.0282922	(0.0005528, 0.85321836, 0.82928340, -0.79328877, 0.19900882, 0.19900882) ^T
D1_1_1	34	3.68	0.07 + 0.14	0.70 + 0.72	1.08 + 0.97	-0.7002049	-0.6996025	(-0.00002279, 0.83642243, 0.83642243) ^T
D1_1_2	74	31.81	0.09 + 0.19	5.92 + 2.08	20.38 + 3.15	-0.7371145	-0.7364589	(-0.00011501, 0.83648705, 0.83648705, 0.80076770, 0.19170182, 0.19169856) ^T
D2_0_1	82	14.49	0.45 + 0.67	4.09 + 3.15	3.33 + 2.80	-0.6609721	-0.6604442	(0.13123499, 0.96690844, 0.66044418) ^T
D2_0_2	175	91.28	2.54 + 0.67	24.48 + 7.83	49.01 + 6.75	-0.9288437	-0.9278613	(0.00000000, 1.17831525, 0.63596129, 0.66035048, 0.52649947, 0.29190000) ^T
D2_1_1	88	15.75	0.07 + 0.14	4.04 + 3.58	4.16 + 3.76	-0.4369370	-0.4363751	(0.13001358, 0.96906506, 0.66058693) ^T
D2_1_2	193	153.93	3.02 + 0.84	34.29 + 5.66	99.72 + 10.40	-0.4913667	-0.4906523	(0.01947920, 1.15270835, 0.64289732, 0.67771397, 0.50854300, 0.27809237) ^T
DP	28	2.08	0.06 + 0.07	0.32 + 0.35	0.93 + 0.35	7.9972287	8.0039295	1.99607046
H	105	9.42	0.09 + 0.09	0.62 + 0.60	5.44 + 2.58	-0.0005574	0.0002401	(0.03125620, 0.00024006) ^T
N	36	2.47	0.13 + 0.15	0.33 + 0.26	1.29 + 0.31	-0.0001000	0.0008483	(0.00000000, 0.00084833) ^T
S	49	32.64	1.61 + 0.13	10.45 + 0.62	19.07 + 0.76	-5.0865023	-5.0818274	(4.21848880, 3.77104989) ^T

CPU times are given for solutions of subproblems and the associated solutions of (LLP)

Table 5 Numerical results for Algorithm 3

Label	#NLPs	CPU(s)	LBD		UBD	x^*
			Total	(ORA)		
2	–	–	–	–	–	–
5	–	–	–	–	–	–
6	60	6.63	0.10 + 0.07	4.49 + 1.97	97.1817286	(0.71756923, -1.45059235) ^T
7	–	–	–	–	–	–
8	64	10.90	0.09 + 0.07	2.51 + 8.23	2.4359721	(2.58115124, -4.05352666, -4.16762615, 4.19109318, 4.53320775, 4.30519267) ^T
9	–	–	–	–	–	–
D1_0_1	–	–	–	–	–	–
D1_0_2	–	–	–	–	–	–
D1_1_1	27	3.33	0.09 + 0.10	1.87 + 1.27	-0.6998542	(-0.00000929, 0.83657289, 0.83657289) ^T
D1_1_2	–	–	–	–	–	–
D2_0_1	60	9.63	0.06 + 0.16	4.47 + 4.94	-0.6606650	(0.13384842, 0.96196874, 0.66066502) ^T
D2_0_2	–	–	–	–	–	–
D2_1_1	60	9.72	0.09 + 0.20	4.91 + 4.52	-0.4364069	(0.13384841, 0.96196875, 0.66061103) ^T
D2_1_2	–	–	–	–	–	–
DP	57	5.15	0.07 + 0.06	2.55 + 2.47	8.0001003	1.99989965
H	90	8.51	0.12 + 0.09	5.53 + 2.77	0.0002066	(0.15625935, 0.00020656) ^T
N	27	1.90	0.06 + 0.06	0.99 + 0.79	0.0003572	(0.00000000, 0.00035725) ^T
S	–	–	–	–	–	–

CPU times are given for solutions of subproblems and the associated solutions of (LLP)

References

1. Bhattacherjee, B., Green Jr., W.H., Barton, P.I.: Interval methods for semi-infinite programs. *Comput. Optim. Appl.* **30**(1), 63–93 (2005)
2. Bhattacherjee, B., Lemonidis, P., Green Jr., W.H., Barton, P.I.: Global solution of semi-infinite programs. *Math. Program.* **103**(2), 283–307 (2005)
3. Bhattacherjee, B., Schwer, D.A., Barton, P.I., Green Jr., W.H.: Optimally-reduced kinetic models: reaction elimination in large-scale kinetic mechanisms. *Combust. Flame* **135**(3), 191–208 (2003)
4. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. *J. Optim. Theory Appl.* **19**(2), 261–281 (1976)
5. 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)
6. Hettich, R., Kortamek, K.O.: Semi-infinite programming—theory, methods, and applications. *SIAM Rev.* **35**(3), 380–429 (1993)
7. Liu, Z., Gong, Y.: Semi-infinite quadratic optimisation method for the design of robust adaptive array processors. *IEE Proc. F-Radar Signal Process.* **137**(3), 177–182 (1990)
8. Lopez, M., Still, G.: Semi-infinite programming. *Eur. J. Oper. Res.* **180**, 491–518 (2007)
9. Mitsos, A.: Test Set of Semi-infinite Programs. Tech. Rep., Massachusetts Institute of Technology (2009). <http://web.mit.edu/mitsos/www/pubs/siptestset> and <http://permalink.avt.rwth-aachen.de/?id=472053>
10. Mitsos, A.: Global solution of nonlinear mixed-integer bilevel programs. *J. Glob. Optim.* **47**(4), 557–582 (2010)
11. Mitsos, A.: Global optimization of semi-infinite programs via restriction of the right-hand side. *Optimization* **60**(10–11), 1291–1308 (2011)
12. 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)
13. Mitsos, A., Lemonidis, P., Lee, C.K., Barton, P.I.: Relaxation-based bounds for semi-infinite programs. *SIAM J. Optim.* **19**(1), 77–113 (2008)
14. Mitsos, A., Tsoukalas, A.: Global optimization of generalized semi-infinite programs via restriction of the right hand side. *J. Glob. Optim.* **61**(1), 1–17 (2015)
15. Polak, E.: On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Rev.* **29**(1), 21–89 (1987)
16. Reemtsen, R., Gröner, S.: Numerical methods for semi-infinite programming: a survey. In: Reemtsen, R., Rückmann, J.-J. (eds.) *Semi-Infinite Programming*, pp. 195–275. Kluwer, Dordrecht (1998)
17. Rosenthal, R.E.: GAMS—a user’s guide. Tech. Rep., GAMS Development Corporation, Washington, DC, USA (2016)
18. Sahinidis, N.V.: Baron: a general purpose global optimization software package. *J. Glob. Optim.* **8**(2), 201–205 (1996)
19. Stein, O.: A semi-infinite approach to design centering. In: Dempe, S., Kalashnikov, V. (eds.) *Optimization with Multivalued Mappings*, pp. 209–228. Springer, New York City (2006)
20. Stein, O., Steuermann, P.: The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets. *Math. Program.* **136**(1), 183–207 (2012)
21. Stein, O., Still, G.: On generalized semi-infinite optimization and bilevel optimization. *Eur. J. Oper. Res.* **142**, 444–462 (2002)
22. Stein, O., Winterfeld, A.: Feasible method for generalized semi-infinite programming. *J. Optim. Theory Appl.* **146**(2), 419–443 (2010)
23. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. *Math. Program.* **103**(2), 225–249 (2005)
24. Tsoukalas, A., Rustem, B.: A feasible point adaptation of the Blankenship and Falk algorithm for semi-infinite programming. *Optim. Lett.* **5**(4), 705–716 (2011)
25. Tsoukalas, A., Rustem, B., Pistikopoulos, E.N.: A global optimization algorithm for generalized semi-infinite, continuous minmax with coupled constraints and bi-level problems. *J. Glob. Optim.* **44**, 235–250 (2009)
26. Watson, G.A.: Numerical experiments with globally convergent methods for semi-infinite programming problems. In: Fiacco, A.V., Kortanek, K.O. (eds.) *Semi-Infinite Programming and Applications*. Springer, Berlin, Heidelberg, pp. 193–205 (1983)