



Global Solution of Semi-infinite Programs with Existence Constraints

Hatim Djelassi¹ · Alexander Mitsos¹ 

Received: 2 August 2019 / Accepted: 6 January 2021 / Published online: 8 February 2021
© The Author(s) 2021

Abstract

We consider what we term existence-constrained semi-infinite programs. They contain a finite number of (upper-level) variables, a regular objective, and semi-infinite existence constraints. These constraints assert that for all (medial-level) variable values from a set of infinite cardinality, there must exist (lower-level) variable values from a second set that satisfy an inequality. Existence-constrained semi-infinite programs are a generalization of regular semi-infinite programs, possess three rather than two levels, and are found in a number of applications. Building on our previous work on the global solution of semi-infinite programs (Djelassi and Mitsos in *J Glob Optim* 68(2):227–253, 2017), we propose (for the first time) an algorithm for the global solution of existence-constrained semi-infinite programs absent any convexity or concavity assumptions. The algorithm is guaranteed to terminate with a globally optimal solution with guaranteed feasibility under assumptions that are similar to the ones made in the regular semi-infinite case. In particular, it is assumed that host sets are compact, defining functions are continuous, an appropriate global nonlinear programming subsolver is used, and that there exists a Slater point with respect to the semi-infinite existence constraints. A proof of finite termination is provided. Numerical results are provided for the solution of an adjustable robust design problem from the chemical engineering literature.

Keywords Semi-infinite programming · Multi-level programming · Nonlinear programming · Nonconvex · Global optimization

Mathematics Subject Classification 90C34 · 65K05 · 90C26 · 90C33

Communicated by Johannes O. Royset.

✉ Alexander Mitsos
amitsos@alum.mit.edu

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

1 Introduction

Semi-infinite programs (SIPs) are mathematical programs that have finitely many variables and infinitely many constraints. These infinitely many constraints are commonly denoted as a parameterized inequality that must be satisfied for all parameter values from a set of infinite cardinality. In the present article, we consider a generalization of SIPs, which we term existence-constrained semi-infinite programs (ESIPs). Rather than being subject to infinitely many inequalities, ESIPs are subject to infinitely many existence constraints that in turn assert that an inequality can be satisfied by at least one solution from a set of infinite cardinality. This kind of constraint gives ESIPs a hierarchical structure with three levels rather than the two levels present in regular SIPs.

Many important applications can be cast as ESIPs. Grossmann and coworkers [1–3] consider the problem of designing flexible chemical plants based on an ESIP formulation. Grossmann et al. [2] furthermore propose a flexibility index to provide a scalar measure of the flexibility of a design considering the same problem structure. The semi-infinite existence constraints in the flexibility problems originate from a consideration of uncertainty that is analogous to the adjustable robust approach proposed two decades later by Ben-Tal et al. [4] as an extension of robust optimization. Similar to two-stage stochastic programs, adjustable robust programs comprise first-stage decisions that are fixed before realization of the uncertainty and second-stage (recourse) decisions that are fixed afterward. In contrast to two-stage stochastic programs, adjustable robust programs consider the worst case with respect to uncertainty rather than a statistical measure based on probability distributions. The consideration of a worst case together with the presence of recourse variables gives rise to semi-infinite existence constraints. More recent proposals of adjustable robust optimization models include investment problems [5], network expansion problems [6], and operational planning problems for transmission grids [7].

Published solution approaches to ESIPs stemming from the robust optimization literature usually rely on relatively strong assumptions. For example, in [1–3], a convexity assumption is made on the constraint function. Furthermore, it is assumed that the uncertainty set is polyhedral and that the recourse variables are unconstrained. Under these conditions, it can be shown that the resulting ESIP can be written equivalently as a finite program [3].

Another special case of ESIPs that has received attention in the literature is min-max-min programs. Polak and Royset [8] consider min-max-min programs absent convexity assumptions, where the inner min operator is applied over a finite set. They distinguish between finite min-max-min programs, where also the max operator is applied over a finite set and semi-infinite min-max-min programs, where the max operator is applied over a set of infinite cardinality. They employ a smoothing technique to replace the inner min operator by a smoothed approximation, essentially reducing the min-max-min program to a min-max approximation. This approximation is used and improved successively to solve finite min-max-min programs to local optimality. Similarly, semi-infinite min-max-min programs are solved through an additional discretization of the max operator. Tsoukalas et al. [9] extend this approach by also applying the smoothing technique to the max operator in finite min-max-min pro-

grams. While applying such a smoothing approach to ESIPs may seem promising, it is not directly applicable since all host sets in ESIPs may have infinite cardinality. Furthermore, it is not obvious if the smoothing technique can be extended to be applicable to sets of infinite cardinality.

To our best knowledge, there exist no published methods for solving ESIPs deterministically absent any convexity assumptions. We propose such a method based on a family of discretization-based methods that have proved particularly useful for the solution of (generalized) SIPs ((G)SIPs) absent convexity assumptions. Therefore, we provide the following review of methods for the solution of SIPs and GSIPs absent convexity assumptions. For a broader review of (G)SIP theory, applications, and methods, the reader is referred to [10–15].

The particular difficulty in solving (G)SIPs absent convexity assumptions is the fact that their lower-level programs may be nonconvex. As a consequence, the problem cannot be reduced to a finite problem by utilizing reformulations based on optimality conditions of the lower-level program. Indeed, methods capable of solving such (G)SIPs have in common that they solve, either explicitly or implicitly, the lower-level program to global optimality.

One approach of implicitly solving the lower-level program is to employ convergent overestimators of the semi-infinite constraint. Lo Bianco and Piazzzi [16] propose to use interval bounds of the constraint function to construct a genetic algorithm for the solution of SIPs. Similarly, Bhattacharjee et al. [17] propose to use interval bounds on the constraint function in a deterministic setting to provide a sequence of feasible points converging to a solution of the SIP. As an alternative to interval extensions, approaches employing convex and concave relaxations to overestimate the semi-infinite constraint in (G)SIPs are proposed by Floudas and Stein [18], Mitsos et al. [19], and Stein and Steuermann [20].

In the context of this article, the most relevant methods for solving (G)SIPs with a nonconvex lower-level program are discretization methods, i.e., methods that construct a finite approximation of the SIP by a finite discretization of the lower-level variables. Particularly algorithms building on an adaptive discretization strategy first proposed by Blankenship and Falk [21] have shown promise in solving practical problems. As such, the algorithm proposed by Bhattacharjee et al. [22] combines the previously mentioned method of generating feasible points in [17] with the algorithm in [21] as a lower bounding procedure to solve SIPs globally. The same approach is extended to the GSIP case by Lemonidis [23]. Other contributions building on [21] propose to generate feasible points by employing additional discretization-based subproblems. Mitsos [24] proposes a discretization-based upper bounding procedure used in concert with the algorithm in [21] as a lower bounding procedure. The algorithm is guaranteed to finitely solve SIPs globally with guaranteed feasibility under the assumption that there exists an ε -optimal SIP-Slater point. A GSIP algorithm following the same basic approach is proposed by Mitsos and Tsoukalas [25]. Also following a discretization approach, Tsoukalas and Rustem [26] propose a discretized oracle problem that evaluates whether or not a given target objective value can be attained by an SIP. They construct an algorithm based on a binary search of the objective space that is guaranteed to solve SIPs globally with guaranteed feasibility finitely under slightly stronger assumptions than the algorithm in [24]. Employing and adapting subproblems from

the algorithms in [24,26], we [27] propose a hybrid algorithm that inherits the stronger convergence guarantees of the algorithm in [24] while outperforming its predecessors on a standard SIP test set.

Here, we extend the concept of discretization methods in general and the algorithm from [27] in particular in order to solve ESIPs to global ε -optimality with guaranteed feasibility. The proposed discretization scheme prescribes to discretize what we term the medial-level variables while associating with each discretization point a vector of lower-level variables. Following this scheme, we derive the subproblems of the algorithm in [27] for the ESIP case in order to restate the algorithm for the solution of ESIPs. Convergence and finite termination of the algorithm is guaranteed under similar assumptions as in the SIP case with substantial differences arising exclusively from the fact that the revised assumptions need to take the presence of a third level into account.

The remainder of this article is organized as follows: In Sect. 2, definitions and assumptions are collected. In Sect. 3, the proposed algorithm is presented and a proof of finite termination is provided. In Sect. 4, numerical results are presented for the adjustable robust design of a reactor-cooler system proposed by Halemane and Grossmann [3]. Finally, Sect. 5 briefly concludes and gives an outlook for future work.

2 Preliminaries

This section comprises definitions and assumptions used throughout this article, as well as results that follow immediately from the assumptions made.

2.1 Definitions

Definition 2.1 (*Notation*) Throughout this article, scalars and scalar-valued functions are set in light face (e.g., φ), vectors and vector-valued functions are set in bold face (e.g., \mathbf{u}), and sets and set-valued functions are set in calligraphic font (e.g., \mathcal{U}). Furthermore, the following definitions hold.

- (i) $\mathbf{0}$ and $\mathbf{1}$ denote vectors of contextually appropriate length, containing only zeros and ones, respectively.
- (ii) Given a point $\bar{\mathbf{u}}$, $\mathcal{B}_\delta(\bar{\mathbf{u}})$ denotes an open neighborhood of $\bar{\mathbf{u}}$ with radius δ , i.e.,

$$\mathcal{B}_\delta(\bar{\mathbf{u}}) = \{\mathbf{u} : \|\mathbf{u} - \bar{\mathbf{u}}\| < \delta\}.$$

Definition 2.2 (*ESIP formulation*) We consider an ESIP of the form

$$f^* = \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \forall \mathbf{y} \in \mathcal{Y} \left[\exists \mathbf{z} \in \mathcal{Z}(\mathbf{y}) : g(\mathbf{x}, \mathbf{y}, \mathbf{z}) \leq 0 \right] \quad (\text{ESIP})$$

without any convexity or concavity assumptions and with

$$\begin{aligned} \mathcal{X} &= \{\mathbf{x} \in \mathcal{X}^0 \subsetneq \mathbb{R}^{n_x} : \mathbf{h}^u(\mathbf{x}) \leq \mathbf{0}\} \\ \mathcal{Y} &= \{\mathbf{y} \in \mathcal{Y}^0 \subsetneq \mathbb{R}^{n_y} : \mathbf{h}^m(\mathbf{y}) \leq \mathbf{0}\} \\ \mathcal{Z}(\mathbf{y}) &= \{\mathbf{z} \in \mathcal{Z}^0 \subsetneq \mathbb{R}^{n_z} : \mathbf{h}^l(\mathbf{y}, \mathbf{z}) \leq \mathbf{0}\}. \end{aligned}$$

In line with Definition 2.2, we define the notion of ESIP feasibility as follows.

Definition 2.3 (*ESIP feasibility*) A point $\bar{\mathbf{x}} \in \mathcal{X}$ is called ESIP feasible if and only if it is feasible in (ESIP), i.e.,

$$\forall \mathbf{y} \in \mathcal{Y} [\exists \mathbf{z} \in \mathcal{Z}(\mathbf{y}) : g(\bar{\mathbf{x}}, \mathbf{y}, \mathbf{z}) \leq 0].$$

Multiple generalizations can be made to (ESIP) such that the concepts discussed in the present article remain applicable. First, while (ESIP) possesses only a single scalar-valued semi-infinite existence constraint, handling multiple vector-valued constraints is straightforward with the proposed method. Similarly, the restriction to continuous variables is only made for notational convenience. Finally, while the sets \mathcal{Y} , and $\mathcal{Z}(\mathbf{y})$ are independent of \mathbf{x} , similar problems with dependencies on \mathbf{x} can be handled by the proposed method under mild assumptions. All these generalizations of (ESIP) and the conditions for their solution with the proposed method are discussed in [28, Section 5.2.4].

The proposed algorithm relies on the global solution of several nonlinear programs (NLPs) to global optimality. To this end, we define the approximate solution of an NLP as follows.

Definition 2.4 (*Approximate solution of NLPs*) The approximate global solution of a feasible NLP

$$\min_{\mathbf{u} \in \mathcal{U}} \varphi(\mathbf{u})$$

with an optimality tolerance $\varepsilon^{NLP} > 0$ yields a feasible point $\bar{\mathbf{u}}$, a lower bound φ^- , and an upper bound φ^+ on the (unknown) globally optimal objective value φ^* with

$$\varphi^- \leq \varphi^* \leq \varphi(\bar{\mathbf{u}}) = \varphi^+ \leq \varphi^- + \varepsilon^{NLP}.$$

In compact notation, we write

$$[\varphi^-, \varphi(\bar{\mathbf{u}})] \ni \varphi^* = \min_{\mathbf{u} \in \mathcal{U}} \varphi(\mathbf{u}).$$

Maximization programs are denoted analogously with the difference that it holds that

$$\varphi^- = \varphi(\bar{\mathbf{u}}) \leq \varphi^* \leq \varphi^+ \leq \varphi^- + \varepsilon^{NLP}.$$

Similarly, the proposed algorithm relies on solving a min-max program (MMP), the approximate solution of which we define as follows.

Definition 2.5 (*Approximate solution of MMPs*) The approximate global solution of a feasible MMP

$$\inf_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{v} \in \mathcal{V}} \varphi(\mathbf{u}, \mathbf{v})$$

with an optimality tolerance $\varepsilon^{MMP} > 0$ yields a feasible point $\bar{\mathbf{u}}$, a lower bound φ^- , and an upper bound φ^+ on the globally optimal objective value φ^* with

$$\varphi^- \leq \varphi^* \leq \max_{\mathbf{v} \in \mathcal{V}} \varphi(\bar{\mathbf{u}}, \mathbf{v}) \leq \varphi^+ \leq \varphi^- + \varepsilon^{MMP}.$$

In compact notation, we write

$$[\varphi^-, \varphi^+] \ni \varphi^* = \inf_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{v} \in \mathcal{V}} \varphi(\mathbf{u}, \mathbf{v})$$

Max-min programs $\sup_{\mathbf{u} \in \mathcal{U}} \min_{\mathbf{v} \in \mathcal{V}} \varphi(\mathbf{u}, \mathbf{v})$ are denoted analogously with the difference that it holds that

$$\varphi^- \leq \min_{\mathbf{v} \in \mathcal{V}} \varphi(\bar{\mathbf{u}}, \mathbf{v}) \leq \varphi^* \leq \varphi^+ \leq \varphi^- + \varepsilon^{MMP}.$$

2.2 Assumptions

We make the assumptions of compact host sets and continuous objective and constraint functions as is standard in deterministic global optimization.

Assumption 1 (*Host sets*) The host sets $\mathcal{X}^0 \subsetneq \mathbb{R}^{n_x}$, $\mathcal{Y}^0 \subsetneq \mathbb{R}^{n_y}$, $\mathcal{Z}^0 \subsetneq \mathbb{R}^{n_z}$ are compact.

Assumption 2 (*Defining functions*) The functions $f : \mathcal{X}^0 \rightarrow \mathbb{R}$, $\mathbf{h}^u : \mathcal{X}^0 \rightarrow \mathbb{R}^{n_h^u}$, $g : \mathcal{X}^0 \times \mathcal{Y}^0 \times \mathcal{Z}^0 \rightarrow \mathbb{R}$, $\mathbf{h}^m : \mathcal{Y}^0 \rightarrow \mathbb{R}^{n_h^m}$, and $\mathbf{h}^l : \mathcal{Y}^0 \times \mathcal{Z}^0 \rightarrow \mathbb{R}^{n_h^l}$ are continuous on their respective domains.

Proposition 2.1 *Given Assumptions 1 and 2, the sets \mathcal{X} and \mathcal{Y} are compact. Furthermore, $\mathcal{Z}(\mathbf{y})$ is compact for any $\mathbf{y} \in \mathcal{Y}^0$.*

Proof By Assumption 1, \mathcal{X}^0 and \mathcal{Y}^0 are compact. Furthermore, by Assumption 2, \mathbf{h}^u and \mathbf{h}^m are continuous on \mathcal{X}^0 and \mathcal{Y}^0 , respectively. Accordingly, the 0-sublevel sets of \mathbf{h}^u and \mathbf{h}^m are compact. Then, \mathcal{X} and \mathcal{Y} are compact by virtue of being intersections of compact sets.

Now, consider any fixed $\mathbf{y} \in \mathcal{Y}^0$. $\mathcal{Z}(\mathbf{y})$ is either empty and therefore compact or compactness of $\mathcal{Z}(\mathbf{y})$ follows as above from compactness of \mathcal{Z}^0 together with continuity of \mathbf{h}^l . \square

Furthermore, for the guaranteed provision of feasible points, the proposed algorithm relies on the existence of a point that is strictly feasible with respect to the semi-infinite existence constraints. Accordingly, we assume the existence of such an ESIP-Slater point for feasible instances of (ESIP).

Assumption 3 ($\hat{\varepsilon}^f$ -optimal ESIP-Slater point) (ESIP) is infeasible or for some $\hat{\varepsilon}^f > 0$, there exists an $\hat{\varepsilon}^f$ -optimal ESIP-Slater point in (ESIP), i.e., for some $\hat{\varepsilon}^f, \varepsilon^S > 0$ there exists a point $\mathbf{x}^S \in \mathcal{X}$ that satisfies $f(\mathbf{x}^S) \leq f^* + \hat{\varepsilon}^f$ and

$$\forall \mathbf{y} \in \mathcal{Y} \left[\exists \mathbf{z} \in \mathcal{Z}(\mathbf{y}) : g(\mathbf{x}^S, \mathbf{y}, \mathbf{z}) \leq -\varepsilon^S \right].$$

Finally, we assume that NLPs and MMPs can be solved as outlined in Definitions 2.4 and 2.5, which is usually the case given Assumptions 1 and 2.

Assumption 4 (Approximate solution) For any given $\varepsilon^{NLP}, \varepsilon^{MMP} > 0$, NLPs and MMPs can be found to be infeasible or solved according to Definitions 2.4 and 2.5, respectively.

3 Solution Algorithm

In this section, we propose an algorithm for the global solution of (ESIP), which is based on our discretization-based SIP algorithm [27]. The algorithm in [27] in turn is a hybridization of the SIP algorithms proposed in [24,26] and employs, either directly or in an adapted form, subproblems from both preceding algorithms to generate convergent lower and upper bounds on the globally optimal objective value of the SIP in question. In [27], the hybrid algorithm is shown to inherit the slightly superior convergence guarantees of [24] while providing better performance than both preceding algorithms on a standard test set.

Global solution of (ESIP) can be achieved by leaving the algorithmic structure of the hybrid SIP algorithm largely unchanged and adapting the subproblems to take the presence of a third level in (ESIP) into account. In the following, all adapted subproblems are stated and their relevant properties in relation to (ESIP) are elucidated. Finally, the algorithm is stated and a proof of finite termination is provided.

3.1 Subproblems

Before stating the subproblems that are specific to the proposed algorithm, we derive the natural subproblems of (ESIP). In SIPs, a distinction is made between the upper-level variables (variables of the SIP) and the lower-level variables (parameters of the semi-infinite constraints). Accordingly, given fixed upper-level variable values, the maximization of the semi-infinite constraint over the domain of lower-level variables is termed the lower-level program. Here, we distinguish between three levels according to their variable domains. Accordingly, we term \mathbf{x} the upper-level variables, \mathbf{y} the medial-level variables, and \mathbf{z} the lower-level variables.

Given this naming convention, we denote for given $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \mathcal{X} \times \mathcal{Y}$ the lower-level program of (ESIP)

$$g^\diamond(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \min_{\mathbf{z} \in \mathcal{Z}(\bar{\mathbf{y}})} g(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \mathbf{z}). \tag{LLP}$$

If (LLP) is feasible, the existence of the minimum asserted therein follows from Assumptions 1 and 2. Otherwise, it holds that $g^\circ(\bar{x}, \bar{y}) = \infty$ by convention.

Lemma 3.1 *Under Assumptions 1 and 2, for any $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$, (LLP) is either infeasible or attains its minimum.*

Proof For any $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$, $g(\bar{x}, \bar{y}, \cdot)$ is continuous on $\mathcal{Z}(\bar{y})$ and $\mathcal{Z}(\bar{y})$ is compact (Proposition 2.1). As a consequence and by Weierstrass’ theorem, for any $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$, (LLP) is either infeasible or attains its minimum. \square

Given Lemma 3.1 and the convention on infeasible problems, we can further write the medial-level program of (ESIP) for given $\bar{x} \in \mathcal{X}$ as

$$[g^-(\bar{x}), g^+(\bar{x})] \ni g^*(\bar{x}) = \sup_{y \in \mathcal{Y}} \min_{z \in \mathcal{Z}(y)} g(\bar{x}, y, z). \tag{MLP}$$

(MLP) is used in the proposed algorithm in order to assess feasibility in (ESIP) of iterates produced by subproblems operating on the upper-level variables. In this sense, the relation between (ESIP) and (MLP) is analogous to the relation between an SIP and its lower-level program. Indeed, a point $\bar{x} \in \mathcal{X}$ is ESIP feasible if and only if (MLP) is infeasible or its globally optimal objective is non-positive.

Lemma 3.2 *Under Assumptions 1 and 2, a point $\bar{x} \in \mathcal{X}$ is ESIP feasible if and only if (MLP) is infeasible or it holds that $g^*(\bar{x}) \leq 0$.*

The proof of Lemma 3.2 is omitted as it follows immediately by construction of (MLP) and Lemma 3.1.

All remaining subproblems are based on a discretization of the medial-level feasible set. Indeed, given a discretization index set $\mathcal{K} \subsetneq \mathbb{N}$ and discretized medial-level variable values $y^k \in \mathcal{Y}$, $k \in \mathcal{K}$, a discretized version and relaxation of (ESIP) is given by

$$\min_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad \forall k \in \mathcal{K} \left[\exists z \in \mathcal{Z}(y^k) : g(x, y^k, z) \leq 0 \right].$$

Introducing instances of the lower-level variables $z^k \in \mathcal{Z}(y^k)$, $k \in \mathcal{K}$ to the discretized program yields the lower bounding program

$$\begin{aligned} [f^{LBP,-}, f(\bar{x}^{LBP})] \ni f^{LBP,*} = & \min_{x \in \mathcal{X}, z^k \in \mathcal{Z}^0} f(x) & \tag{LBP} \\ \text{s.t.} \quad \forall k \in \mathcal{K} & \begin{bmatrix} g(x, y^k, z^k) \leq 0 \\ h^l(y^k, z^k) \leq \mathbf{0} \end{bmatrix}. \end{aligned}$$

For any valid discretization, the projection of (LBP) onto \mathcal{X} is a relaxation of (ESIP) and thereby, the solution of (LBP) provides a lower bound on the globally optimal objective value of (ESIP).

Lemma 3.3 *Let $\mathcal{K} \subsetneq \mathbb{N}$ and for all $k \in \mathcal{K}$, $y^k \in \mathcal{Y}$. Then, it holds that*

$$f^{LBP,*} \leq f^*.$$

The proof of Lemma 3.3 is omitted as it is straightforward.

An upper bounding program is obtained by again discretizing the medial-level variables as in (LBP) and then restricting the discretized constraints by a restriction parameter. Given a discretization index set $\mathcal{K} \subseteq \mathbb{N}$, discretized medial-level variable values $\mathbf{y}^k \in \mathcal{Y}$, $k \in \mathcal{K}$, instances of the lower-level variables $\mathbf{z}^k \in \mathcal{Z}(\mathbf{y}^k)$, $k \in \mathcal{K}$, and a restriction parameter $\varepsilon^g > 0$, the upper bounding program is given by

$$\begin{aligned} [f^{UBP,-}, f(\bar{\mathbf{x}}^{UBP})] \ni f^{UBP,*} = & \min_{\mathbf{x} \in \mathcal{X}, \mathbf{z}^k \in \mathcal{Z}^0} f(\mathbf{x}) & \text{(UBP)} \\ \text{s.t. } \forall k \in \mathcal{K} & \begin{bmatrix} g(\mathbf{x}, \mathbf{y}^k, \mathbf{z}^k) \leq -\varepsilon^g \\ \mathbf{h}^l(\mathbf{y}^k, \mathbf{z}^k) \leq \mathbf{0} \end{bmatrix}. \end{aligned}$$

Analogously to the SIP case discussed in [24,27], the projection of (UBP) onto \mathcal{X} is generally neither a relaxation nor a restriction of (ESIP). However, under Assumption 3 and through proper manipulation of the restriction parameter ε^g and the discretization, (UBP) can be shown to provide an ESIP feasible point and thereby an upper bound on the globally optimal objective value of (ESIP) in finitely many iterations.

Finally, the restriction program aims to maximize the restriction of the discretized constraints while attaining a target objective value. Given a discretization index set $\mathcal{K} \subseteq \mathbb{N}$, discretized medial-level variable values $\mathbf{y}^k \in \mathcal{Y}$, $k \in \mathcal{K}$, instances of the lower-level variables $\mathbf{z}^k \in \mathcal{Z}(\mathbf{y}^k)$, $k \in \mathcal{K}$, and a target objective value f^{RES} , the restriction program is given by

$$\begin{aligned} [\eta^{RES,-}, \eta^{RES,+}] \ni \eta^* = & \max_{\eta \in \mathbb{R}, \mathbf{x} \in \mathcal{X}, \mathbf{z}^k \in \mathcal{Z}^0} \eta & \text{(RES)} \\ \text{s.t. } f(\mathbf{x}) \leq & f^{RES} \\ \forall k \in \mathcal{K} & \begin{bmatrix} g(\mathbf{x}, \mathbf{y}^k, \mathbf{z}^k) \leq -\eta \\ \mathbf{h}^l(\mathbf{y}^k, \mathbf{z}^k) \leq \mathbf{0} \end{bmatrix}. \end{aligned}$$

The observations made in [27] about the relation of the restriction program to the lower and upper bounding programs also hold here. Indeed, for $\eta = 0$, exactly those points are feasible in (RES) that are members of the level set with objective function value f^{RES} of (LBP). As a consequence, if for some f^{RES} , it holds that $\eta^* < 0$, then said level set is empty, f^{RES} is a lower bound on $f^{LBP,*}$ and by extension a lower bound on f^* . If on the other hand it holds that $\eta^* \geq 0$, the associated solution $\bar{\mathbf{x}}^{RES}$ may be ESIP feasible. In case this is verified via a solution of (MLP), $f(\bar{\mathbf{x}}^{RES})$ is an upper bound on f^* and furthermore, η^* is the largest possible restriction parameter value for which the current best solution remains feasible in (UBP) for the given discretization. As argued in [27], this property motivates the use of such solutions of (RES) as updates for the restriction parameter ε^g for subsequent solutions of (UBP).

The increased difficulty of (ESIP) when compared to an SIP is reflected in all sub-problems of the proposed algorithms. Indeed, (MLP) is a max-min program, whereas its analogue in an SIP is a single-level program. As a consequence, although under Assumptions 1 and 2, (MLP) can in turn be solved according to Definition 2.5 using discretization methods such as the one proposed by Falk and Hoffman [29], the solu-

tion of a max-min program is generally more costly than the solution of a single-level program of similar size and structure. Similarly, the remaining discretization-based subproblems exhibit not only the typical increase in the number of constraints with the number of discretization points, but also an increase in the number of variables. However, the variables that are associated with discretization points appear in a block structure, which could potentially be exploited by employing a decomposition method for an efficient solution of the discretization-based subproblems. Indeed, the block structure observed here is similar to the structure observed in the subproblems of a two-stage stochastic programming algorithm, which are commonly solved using decomposition methods.

3.2 Algorithm Statement

Given the subproblems (LBP), (UBP), (RES), and (MLP), the proposed algorithm can be stated as in Algorithm 1.

Note that in contrast to the original algorithm in [27], in the case of an infeasible instance of (UBP), (LBP) is solved instead of (UBP) with a reduced restriction parameter. This change is required since Assumption 3 in contrast to its equivalent in [27] allows for (ESIP) to be infeasible. This change guarantees that an infeasible instance of (ESIP) is detected by (LBP) becoming infeasible after finitely many iterations.

3.3 Proof of Convergence

For the following discussion, note that Algorithm 1 is separated into three main blocks of instructions associated with the three subproblems (LBP), (UBP), and (RES). We term these blocks the *lower bounding procedure* (Lines 3–12), the *upper bounding procedure* (Lines 13–23), and the *restriction procedure* (Lines 26–40). In order to prove finite termination of Algorithm 1, we first establish properties of the three main procedures.

Lower bounding procedure The lower bounding procedure is required to produce a convergent sequence of lower bounds for Algorithm 1 to terminate finitely. This is achieved by successively tightening (LBP) with solution points of (MLP). As observed in [27] for the SIP case, particular attention must be paid to the optimality tolerance ε^{MMP} . In [27], a successive refinement of the optimality tolerance is performed to ensure that a sufficiently small tolerance is reached finitely. In the following lemma, we opt instead to show that a sufficiently small tolerance exists that ensures convergence of the lower bounding procedure. Practical implementations of Algorithm 1 may still employ the successive refinement strategy proposed in [27]. Indeed, in the proof of the following lemma, we let ε^{MMP} approach zero in the limit of an iteration sequence. Harwood et al. [30] show for the SIP algorithm in [24] that such a tolerance refinement is generally required to ensure convergence of the discretization-based lower bounding procedure.

Lemma 3.4 *Consider a sequence of successive iterations of the lower bounding procedure and let ε^{MMP} for each solution of (MLP) be positive but sufficiently small. Then, under Assumptions 1, 2 and 4, if (ESIP) is infeasible, (LBP) becomes infeasible after*

Algorithm 1: ESIP algorithm. Subproblems are solved globally according to Definitions 2.4 and 2.5, respectively. Inputs are parameters $\varepsilon^f, \varepsilon^{NLP}, \varepsilon^{MMP}, \varepsilon^{g,0} > 0$, and $r^g > 1$, and an initial discretization with $\mathcal{K} \subseteq \mathbb{N}$ and $\mathbf{y}^k \in \mathcal{Y}, k \in \mathcal{K}$.

```

1 set  $LBD \leftarrow -\infty, UBD \leftarrow \infty, \varepsilon^g \leftarrow \varepsilon^{g,0}$ ;
2 set  $k \leftarrow \max\{m + 1 : m \in \mathcal{K} \cup \{0\}\}$ ;
3 solve (LBP) to obtain  $f^{LBP,-}$  and  $\bar{\mathbf{x}}^{LBP}$ ;
4 if (LBP) is infeasible then
5 | terminate ((ESIP) is infeasible);
6 end
7 set  $LBD \leftarrow f^{LBP,-}$ ;
8 solve (MLP) for  $\bar{\mathbf{x}} = \bar{\mathbf{x}}^{LBP}$  to obtain  $g^+(\bar{\mathbf{x}}^{LBP})$  and  $\bar{\mathbf{y}}^{MLP}$ ;
9 if (MLP) is infeasible or  $g^+(\bar{\mathbf{x}}^{LBP}) \leq 0$  then
10 | set  $UBD \leftarrow f(\bar{\mathbf{x}}^{LBP}), \mathbf{x}^* \leftarrow \bar{\mathbf{x}}^{LBP}$  and terminate;
11 end
12 set  $\mathbf{y}^k \leftarrow \bar{\mathbf{y}}^{MLP}, \mathcal{K} \leftarrow \mathcal{K} \cup \{k\}, k \leftarrow k + 1$ ;
13 solve (UBP) to obtain  $\bar{\mathbf{x}}^{UBP}$ ;
14 if (UBP) is infeasible then
15 | set  $\varepsilon^g \leftarrow \varepsilon^g / r^g$  and go to Line 3;
16 end
17 solve (MLP) for  $\bar{\mathbf{x}} = \bar{\mathbf{x}}^{UBP}$  to obtain  $g^+(\bar{\mathbf{x}}^{UBP})$  and  $\bar{\mathbf{y}}^{MLP}$ ;
18 if (MLP) is infeasible or  $g^+(\bar{\mathbf{x}}^{UBP}) \leq 0$  then
19 | if  $f(\bar{\mathbf{x}}^{UBP}) \leq UBD$  then set  $UBD \leftarrow f(\bar{\mathbf{x}}^{UBP}), \mathbf{x}^* \leftarrow \bar{\mathbf{x}}^{UBP}$ 
20 | set  $\varepsilon^g \leftarrow \varepsilon^g / r^g$ ;
21 else
22 | set  $\mathbf{y}^k \leftarrow \bar{\mathbf{y}}^{MLP}, \mathcal{K} \leftarrow \mathcal{K} \cup \{k\}, k \leftarrow k + 1$  and go to Line 13;
23 end
24 if  $LBD \geq UBD - \varepsilon^f$  then terminate
25 set  $f^{RES} \leftarrow \frac{1}{2}(LBD + UBD), m \leftarrow 0$ ;
26 set  $m \leftarrow m + 1$ ;
27 solve (RES) to obtain  $\eta^{RES,-}, \eta^{RES,+}$ , and  $\bar{\mathbf{x}}^{RES}$ ;
28 if  $\eta^{RES,+} < 0$  then
29 | set  $LBD \leftarrow f^{RES}$  and go to Line 24;
30 else if  $\eta^{RES,-} \leq 0$  then
31 | go to Line 3;
32 end
33 solve (MLP) for  $\bar{\mathbf{x}} = \bar{\mathbf{x}}^{RES}$  to obtain  $g^+(\bar{\mathbf{x}}^{RES})$  and  $\bar{\mathbf{y}}^{MLP}$ ;
34 if (MLP) is infeasible or  $g^+(\bar{\mathbf{x}}^{RES}) \leq 0$  then
35 | if  $\eta^{RES,+} / r^g < \varepsilon^g$  then set  $\varepsilon^g \leftarrow \eta^{RES,+} / r^g$ 
36 | set  $UBD \leftarrow f(\bar{\mathbf{x}}^{RES}), \mathbf{x}^* \leftarrow \bar{\mathbf{x}}^{RES}$  and go to Line 13;
37 else if  $m < m^{max}$  then
38 | set  $\mathbf{y}^k \leftarrow \bar{\mathbf{y}}^{MLP}, \mathcal{K} \leftarrow \mathcal{K} \cup \{k\}, k \leftarrow k + 1$  and go to Line 26;
39 end
40 go to Line 3;

```

finitely many iterations of the lower bounding procedure. Otherwise, the sequence of lower bounds produced converges to an ε^{NLP} -underestimate of f^* finitely or in the limit.

Proof If (ESIP) is infeasible, each point $\bar{\mathbf{x}} \in \mathcal{X}$ furnished by (LBP) is ESIP infeasible. By compactness of \mathcal{X} , there exists $\varepsilon > 0$ independent of $\bar{\mathbf{x}}$ such that $g^*(\bar{\mathbf{x}}) > 2\varepsilon$. Therefore, by Assumption 4, ε^{MMP} can be chosen such that $0 < \varepsilon^{MMP} < \varepsilon$. It

follows that the point $\bar{y} \in \mathcal{Y}$ yielded by the solution of (MLP) satisfies

$$\min_{z \in \mathcal{Z}(\bar{y})} g(\bar{x}, \bar{y}, z) \geq g^*(\bar{x}) - \varepsilon^{MMP} > \varepsilon.$$

By continuity of g on $\mathcal{X}^0 \times \mathcal{Y}^0 \times \mathcal{Z}^0$ and compactness of $\mathcal{X}^0 \times \mathcal{Y}^0 \times \mathcal{Z}^0$, g is uniformly continuous. As a consequence, there exists $\delta > 0$ independent of \bar{x} and \bar{y} such that

$$\forall x \in \mathcal{X} \cap \mathcal{B}_\delta(\bar{x}) \forall z \in \mathcal{Z}(\bar{y}) [g(x, \bar{y}, z) > 0].$$

That is, the point added to the discretization renders an open neighborhood of \bar{x} with radius δ infeasible in (LBP). By compactness of \mathcal{X} , it follows that (LBP) is rendered infeasible in finitely many iterations of the lower bounding procedure.

If (ESIP) is feasible and (LBP) furnishes an ESIP feasible solution $\bar{x} \in \mathcal{X}$ finitely, it follows $f(\bar{x}) \geq f^*$. Furthermore, with (LBP) being a relaxation of (ESIP) ((Lemma 3.3) and by Assumption 4, it holds that

$$f(\bar{x}) - \varepsilon^{NLP} \leq f^{LBP,-} \leq f^* \leq f(\bar{x}),$$

proving that $f^{LBP,-}$ is an ε^{NLP} -underestimate of f^* .

If (ESIP) is feasible and (LBP) does not furnish an ESIP feasible solution, let $\{\bar{x}^k\}_{k \geq 1} \subsetneq \mathcal{X}$ denote the sequence of solutions furnished by (LBP) and $\{\bar{y}^k\}_{k \geq 1}$ the sequence of associated solutions furnished by (MLP). By construction of (LBP), it holds that

$$\forall k, m : m > k \geq 1 \exists z \in \mathcal{Z}(\bar{y}^k) : [g(\bar{x}^m, \bar{y}^k, z) \leq 0].$$

By continuity of $g(\cdot, \bar{y}^k, z)$ on \mathcal{X} for all $z \in \mathcal{Z}(\bar{y}^k)$, there exists $\delta > 0$ for any $\varepsilon > 0$ such that

$$\forall k, m : m > k \geq 1 \forall x \in \mathcal{X} \cap \mathcal{B}_\delta(\bar{x}^m) \exists z \in \mathcal{Z}(\bar{y}^k) : [g(x, \bar{y}^k, z) < \varepsilon]. \tag{1}$$

By compactness of \mathcal{X} , it holds that $\bar{x}^k \xrightarrow{k \rightarrow \infty} \hat{x} \in \mathcal{X}$, yielding that there exists K for any $\delta > 0$ such that

$$\forall k, m : m > k \geq K [\bar{x}^k \in \mathcal{B}_\delta(\bar{x}^m)]. \tag{2}$$

Combining (1) and (2) yields that for any $\varepsilon > 0$, there exists K such that

$$\forall k \geq K \exists z \in \mathcal{Z}(\bar{y}^k) : [g(\bar{x}^k, \bar{y}^k, z) < \varepsilon].$$

Furthermore, it follows from Assumption 4 that for all $k \geq 1$

$$g^*(\bar{x}^k) \leq \min_{z \in \mathcal{Z}(\bar{y}^k)} g(\bar{x}^k, \bar{y}^k, z) + \varepsilon^{MMP}.$$

Then, letting $\varepsilon^{MMP} \rightarrow 0$ for $k \rightarrow \infty$, yields that the accumulation point \hat{x} is ESIP feasible and it holds $f(\hat{x}) \geq f^*$. Furthermore, by continuity of f , it follows $f(\bar{x}^k) \xrightarrow{k \rightarrow \infty} f(\hat{x})$. Together with Assumption 4 and (LBP) being a relaxation of (ESIP) (Lemma 3.3), it follows that the sequence of lower bounds produced by the lower bounding procedure converges to a value in the interval

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

proving that the sequence of lower bounds converges to an ε^{NLP} -underestimate of f^* . □

Upper bounding procedure As mentioned previously, (UBP) is not generally a restriction of (ESIP) and therefore does not necessarily yield upper bounds for (ESIP). However, as will be shown here, the upper bounding procedure as a whole does provide upper bounds reliably. To this end, it is first established that the upper bounding procedure cannot enter an infinite loop. This is of particular importance if (ESIP) is infeasible since only the lower bounding procedure can prove infeasibility of (ESIP).

Lemma 3.5 *Consider a sequence of iterations of the upper bounding procedure. Let that sequence be contiguous, i.e., let the sequence be produced by Algorithm 1 looping on Lines 13–22. Let furthermore $0 < \varepsilon^{MMP} < \varepsilon^g$ for each solution of (MLP). Then, under Assumptions 1, 2 and 4, the considered sequence of iterations of the upper bounding procedure is finite.*

Proof According to Algorithm 1, a contiguous sequence of iterations of the upper bounding procedure is terminated if in one execution, (UBP) either becomes infeasible or yields a point $\bar{x} \in \mathcal{X}$ that is found to be ESIP feasible. Furthermore, within a contiguous sequence of iterations of the upper bounding procedure, the restriction parameter ε^g remains unchanged.

Let $\bar{x} \in \mathcal{X}$ be furnished by a feasible instance of (UBP) and let \bar{x} not be found ESIP feasible. It follows that the solution of (MLP) yields $g^+(\bar{x}) > 0$. Furthermore, by Assumption 4, the point $\bar{y} \in \mathcal{Y}$ yielded by the solution of (MLP) satisfies

$$\min_{z \in \mathcal{Z}(\bar{y})} g(\bar{x}, \bar{y}, z) \geq g^+(\bar{x}) - \varepsilon^{MMP} > -\varepsilon^{MMP}.$$

By continuity of g on $\mathcal{X}^0 \times \mathcal{Y}^0 \times \mathcal{Z}^0$ and compactness of $\mathcal{X}^0 \times \mathcal{Y}^0 \times \mathcal{Z}^0$, g is uniformly continuous. As a consequence and due to $\varepsilon^g - \varepsilon^{MMP} > 0$, there exists $\delta > 0$ independent of \bar{x} and \bar{y} such that

$$\forall x \in \mathcal{X} \cap \mathcal{B}_\delta(\bar{x}) \forall z \in \mathcal{Z}(\bar{y}) \left[g(x, \bar{y}, z) > -\varepsilon^g \right].$$

That is, the point added to the discretization renders an open neighborhood of \bar{x} with radius δ infeasible in (UBP). By compactness of \mathcal{X} , it follows that (UBP) is either rendered infeasible or yields an ESIP feasible point $\bar{x} \in \mathcal{X}$ that is confirmed as such in finitely many iterations of the upper bounding procedure. □

Given Lemma 3.5 and Assumption 3, the upper bounding property of the upper bounding procedure can be proven as follows.

Lemma 3.6 *Consider a sequence of successive iterations of the upper bounding procedure and let $\varepsilon^{MMP} < \varepsilon^g$ for each solution of (MLP). Then, under Assumptions 1–4, if (ESIP) is feasible, the upper bounding procedure furnishes an $(\hat{\varepsilon}^f + \varepsilon^{NLP})$ -optimal point after finitely many iterations.*

Proof By Lemma 3.5 and $\varepsilon^{MMP} < \varepsilon^g$, it follows that for any given ε^g , the upper bounding procedure is executed only a finite number of times before either (UBP) becomes infeasible or some point is found to be ESIP feasible. In both cases, the restriction parameter ε^g is reduced according to $\varepsilon^g \leftarrow \varepsilon^g / r^g$ and similarly the restriction procedure only ever reduces ε^g . It follows that for any $\varepsilon^S > 0$ and after finitely many iterations of the upper bounding procedure, it holds $\varepsilon^g < \varepsilon^S$. Furthermore, by Assumption 3 and feasibility of (ESIP), there exists a point $\mathbf{x}^S \in \mathcal{X}$ that satisfies $f(\mathbf{x}^S) \leq f^* + \hat{\varepsilon}^f$ and

$$\forall \mathbf{y} \in \mathcal{Y} \left[\exists \mathbf{z} \in \mathcal{Z}(\mathbf{y}) : g(\mathbf{x}^S, \mathbf{y}, \mathbf{z}) \leq -\varepsilon^S \right].$$

for some $\hat{\varepsilon}^f, \varepsilon^S > 0$. As a consequence, after finitely many iterations of the upper bounding procedure, \mathbf{x}^S is feasible in (UBP) irrespective of the discretization employed. At this point, (UBP) can no longer become infeasible and by Lemma 3.5, a point $\bar{\mathbf{x}} \in \mathcal{X}$ is found to be ESIP feasible finitely. Due to Assumption 4, this point satisfies

$$f(\bar{\mathbf{x}}) \leq f(\mathbf{x}^S) + \varepsilon^{NLP} \leq f^* + \hat{\varepsilon}^f + \varepsilon^{NLP}$$

and $\bar{\mathbf{x}}$ is $(\hat{\varepsilon}^f + \varepsilon^{NLP})$ -optimal in (ESIP). □

Restriction procedure With the properties of the lower and upper bounding procedure established, the essential properties required for solving (ESIP) are already in place. Indeed, as discussed in [27] for the SIP case, the restriction procedure is not required for finite termination of Algorithm 1 but is only added to improve practical performance. Accordingly, the following lemma establishes that the restriction procedure does not impede the guarantee for finite termination by entering an infinite loop.

Lemma 3.7 *Let the optimality gap UBD–LBD be finite and consider a sequence of iterations of the restriction procedure. After finitely many iterations, the restriction procedure provides a bound update, at least halving the optimality gap, or defers to the lower bounding procedure.*

Proof If the solution of (RES) yields $\eta^{RES,+} < 0$, the lower bound is updated with f^{RES} and the optimality gap is halved due to

$$LBD \leftarrow f^{RES} = \frac{1}{2}(LBD + UBD).$$

If on the other hand, the solution of (RES) yields $\eta^{RES,-} \leq 0 \leq \eta^{RES,+}$, the restriction procedure immediately defers to the lower bounding procedure in order to obtain an

update to the lower bound. Otherwise, letting \bar{x} denote the solution of (RES), if \bar{x} is found to be ESIP feasible, it provides an update to the upper bound. By construction of (RES) and $f^{RES} = \frac{1}{2}(LBD + UBD)$, the bound update satisfies

$$UBD \leftarrow f(\bar{x}) \leq \frac{1}{2}(LBD + UBD),$$

meaning the optimality gap is at least halved. Finally, if none of the above conditions are met, the restriction procedure is only executed finitely many times before deferring to the lower bounding procedure. \square

Finite termination Given Lemmas 3.4, 3.6 and 3.7 and a proper choice of tolerances, finite termination of Algorithm 1 can be proven as follows.

Theorem 3.1 *Let the tolerances in Algorithm 1 be chosen such that $\varepsilon^f > \hat{\varepsilon}^f + 2\varepsilon^{NLP}$. Let furthermore in each iteration of the lower bounding procedure $\varepsilon^{MMP} > 0$ be sufficiently small in the sense of Lemma 3.4 and in each iteration of the upper bounding procedure $0 < \varepsilon^{MMP} < \varepsilon^g$. Then, under Assumptions 1–4, Algorithm 1 terminates finitely either proving infeasibility of (ESIP) or yielding an ε^f -optimal point x^* .*

Proof If (ESIP) is infeasible, no solutions furnished by (UBP) can be ESIP feasible. Together with Lemma 3.5 and $\varepsilon^{MMP} < \varepsilon^g$ for the solution of (MLP) in the upper bounding procedure, it follows that the upper bounding procedure defers back to the lower bounding procedure finitely. With Lemma 3.4 and ε^{MMP} being chosen sufficiently small in the sense of Lemma 3.4 for the solution of (MLP) in the lower bounding procedure, it further follows that (LBP) is rendered infeasible finitely. As a consequence, Algorithm 1 terminates finitely having proven infeasibility of (ESIP).

If (ESIP) is feasible, it follows from Lemma 3.4 that the lower bounds produced by the lower bounding procedure converge to an ε^{NLP} -underestimate of f^* . Accordingly, letting $\{f^{LBP,-,k}\}_{k \geq 1}$ denote this sequence of lower bounds, there exists K^{LBP} for any $\varepsilon > 0$ such that

$$\forall k \geq K^{LBP} \left[f^{LBP,-,k} > f^* - \varepsilon^{NLP} - \varepsilon \right]. \tag{3}$$

Furthermore, it follows from Lemma 3.6 that after a finite number K^{UBP} of iterations, the upper bounding procedure furnishes an $(\hat{\varepsilon}^f + \varepsilon^{NLP})$ -optimal point. Letting $\{\bar{x}^{UBP,k}\}_{k \geq 1}$ denote the sequence of points furnished by the upper bounding procedure, it holds that

$$\forall k \geq K^{UBP} \left[f(\bar{x}^{UBP,k}) \leq f^* + \hat{\varepsilon}^f + \varepsilon^{NLP} \right]. \tag{4}$$

Combining (3) and (4) yields that for any $\varepsilon > 0$, there exists K^{LBP} such that

$$\forall k \geq K^{LBP} \forall l \geq K^{UBP} \left[f(\bar{x}^{UBP,l}) - f^{LBP,-,k} < \hat{\varepsilon}^f + 2\varepsilon^{NLP} + \varepsilon \right].$$

Due to $\varepsilon^f > \hat{\varepsilon}^f + 2\varepsilon^{NLP}$, it follows that after finitely many iterations of the lower bounding procedure and finitely many iterations of the upper bounding procedure, Algorithm 1 terminates with an ε^f -optimal point.

By construction of Algorithm 1, the restriction procedure is only executed once the optimality gap $\Delta f = UBD - LBD$ is finite. It follows by Lemma 3.7 that the restriction procedure is only executed finitely many times before providing a bound update at least halving the optimality gap or deferring to the lower bounding procedure. After at most

$$K^{RES} = \log_2 \left(\Delta f / \varepsilon^f \right)$$

bound updates by the restriction procedure, the optimality tolerance ε is reached. As a consequence, the restriction procedure either provides K^{RES} bound updates or defers sufficiently many times to the lower bounding procedure. Either way, the optimality tolerance ε^f is reached and Algorithm 1 terminates finitely. \square

4 Numerical Experiments

In this section, we provide numerical results for the solution of an ESIP example problem. To this end, we employ a C++ implementation of Algorithm 1, allowing the presence of coupling equality constraints on the medial and lower level in the sense of [31]. A detailed description of this and other generalizations is provided in [28, Section 5.2.4]. Min-max subproblems are solved using a C++ implementation of the algorithm in [29] that is similarly extended in the sense of [31] to allow coupling equality constraints on the lower level. NLP subproblems of Algorithm 1 and the min-max algorithm are solved to global optimality using MAiNGO v0.1.24 [32].

The numerical experiments are conducted on a single thread of a laptop computer with a 64-bit Intel Core i7-8550U @ 1.80 GHz (4.00 GHz boost clock speed) and 16 GB of memory, running Linux 5.1.15-arch1-1-ARCH. The CPU times presented in the following are derived from MAiNGO solution reports and represent the CPU times required for solving all NLP subproblems of Algorithm 1 and the min-max algorithm. The ESIP is solved to an absolute and relative optimality tolerance of 10^{-3} , subproblems (LBP), (UBP), and (RES) are solved to an absolute and relative optimality tolerance of 10^{-4} , and (MLP) is solved to an absolute optimality tolerance of 10^{-2} and a relative optimality tolerance of 10^{-1} .

We consider the adjustable robust design of a reactor-cooler system proposed by Halemane and Grossmann [3]. The objective of the design problem is to choose the reactor volume \hat{V} and the heat exchanger area A such that the total annualized cost (TAC) of operation is minimized subject to a semi-infinite existence constraint derived from parametric uncertainties in the model parameters and operational constraints. That is, a design is considered feasible, if for all possible uncertainty realizations, there exist operational decisions that ensure satisfaction of the operational constraints. Halemane and Grossmann [3] consider both the nominal objective function (corresponding to the nominal uncertainty realization) and the expected value of the objective function approximated by a weighted sum formulation. Here, we consider the nominal case and note that the weighted sum formulation does not pose an additional challenge apart from adding variables and constraints to the upper level of the resulting ESIP.

Table 1 Numerical results for Algorithm 1 solving [3, Example 2] with nominal objective function, compared to results reported in [3]

	\hat{V}/m^3	A/m^2	$\text{TAC}^-/(\$/\text{a})$	$\text{TAC}^+/(\$/\text{a})$
Algorithm 1	6.51	7.45	10,107	10,113
[3, Algorithm II]	6.5	8.11	–	10,160

The optimal solution is given in terms of the design variables \hat{V} and A . TAC^- and TAC^+ denote the final lower and upper bound on the total annualized cost (TAC) of operation, respectively.

The complete description of the problem instance and the ESIP formulation employed in the following are collected in [28, Section 5.3.2].

For the solution of the adjustable robust design problem, Halemane and Grossmann [3] rely on the ESIP formulation being reducible to a finite problem by considering only the vertices of the (polyhedral) medial-level feasible set. They propose an algorithm that solves a sequence of subproblems that are tightened successively through the addition of constraints associated with vertices of the medial-level feasible set. This approach is proven to be valid under the assumptions of a jointly convex constraint function g and an unconstrained lower-level feasible set [3, Theorem 2]. Both these assumptions are violated for the problem in question. Halemane and Grossmann [3] argue that although not guaranteed, the approach may still succeed in the presence of nonconvex constraint functions.

For the solution of the problem using Algorithm 1, its algorithmic parameters are set to $\varepsilon^{g,0} = 10^{-2}$ and $r^g = 2$. As is apparent from Table 1, Algorithm 1 yields results that are consistent with the results reported in [3]. We find a slightly better solution than [3], which is likely due to a difference in the termination criteria employed. Note also that one parameter of the problem is changed from the value reported in [3] since the reported parameter value does not yield consistent results [28, Section 5.3.2]. The results confirm that the results reported in [3] are correct despite the fact that the problem violates the assumptions made therein. Indeed, the only discretization point added by Algorithm 1 is a vertex of the medial-level feasible set, which indicates that the problem can be reduced to a finite problem by considering that vertex.

Termination of Algorithm 1 is achieved after 1 iteration of the lower bounding procedure, 1 iteration of the upper bounding procedure, 7 iterations of the restriction procedure, and overall 6.15 CPU seconds.

5 Conclusions

We propose a discretization-based algorithm for the solution of existence-constrained semi-infinite programs (ESIPs) based on our previous work on the solution of semi-infinite programs (SIPs) [27]. While SIPs possess two levels, the lower of which is discretized in discretization algorithms, the presence of semi-infinite existence constraints adds a third level. The proposed algorithm performs a discretization of the medial-level variables while introducing for each discretization point a vector of lower-level variables. By this approach, the subproblems constructed by the algorithm obtain

the same bounding properties as their counterparts in the SIP algorithm [27]. Finite termination with an ESIP feasible and ε -optimal solution is proven under assumptions that are similar to the ones made for the SIP case.

The proposed algorithm is implemented, and an adjustable robust design problem from the chemical engineering literature is solved as an example problem. While the problem has been solved previously using a method that requires a convexity assumption to reduce the problem to a finite counterpart, the proposed algorithm does not require such a property. The obtained results are consistent with the ones reported in the literature.

The proposed approach of generating subproblems yields a vector of lower-level variables for each discretization point in the discretization-based subproblems. As a consequence, these subproblems grow in terms of the number of constraints and variables as the number of discretization points increases. It is therefore expected that for problem instances that require many iterations of the ESIP algorithm, standard general-purpose solvers will encounter tractability issues in solving these subproblems. However, the affected subproblems also exhibit a block structure that may, depending on the particular problem structure, enable the use of decomposition methods with a favorable scaling behavior. Indeed, the block structure mirrors a similar structure in two-stage stochastic programs, the solution of which is often achieved using appropriate decomposition methods. Nevertheless, care should be taken to minimize the number of required iterations, e.g., through tuning algorithmic parameters and improving the algorithm.

Acknowledgements We gratefully acknowledge the financial support provided by Réseau de transport d'électricité (RTE, France) through the project "Bilevel Optimization for Worst-case Analysis of Power Grids." Furthermore, we thank Oliver Stein for helpful comments that lead to an improvement in this article.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Grossmann, I.E., Halemane, K.P.: Decomposition strategy for designing flexible chemical plants. *AIChE J.* **28**(4), 686–694 (1982)
2. Grossmann, I.E., Halemane, K.P., Swaney, R.E.: Optimization strategies for flexible chemical processes. *Comput. Chem. Eng.* **7**(4), 439–462 (1983)
3. Halemane, K.P., Grossmann, I.E.: Optimal process design under uncertainty. *AIChE J.* **29**(3), 425–433 (1983)
4. Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A.: Adjustable robust solutions of uncertain linear programs. *Math. Program.* **99**(2), 351–376 (2004)
5. Takeda, A., Taguchi, S., Tütüncü, R.H.: Adjustable robust optimization models for a nonlinear two-period system. *J. Optim. Theory Appl.* **136**(2), 275–295 (2007)

6. Ordóñez, F., Zhao, J.: Robust capacity expansion of network flows. *Networks* **50**(2), 136–145 (2007)
7. Djelassi, H., Fliscounakis, S., Mitsos, A., Panciatici, P.: Hierarchical programming for worst-case analysis of power grids. In: *Power Systems Computation Conference*, Dublin, Ireland (2018)
8. Polak, E., Royset, J.O.: Algorithms for finite and semi-infinite min–max–min problems using adaptive smoothing techniques. *J. Optim. Theory Appl.* **119**(3), 421–457 (2003)
9. Tsoukalas, A., Pappas, P., Rustem, B.: A smoothing algorithm for finite min–max–min problems. *Optim. Lett.* **3**(1), 49–62 (2008)
10. Polak, E.: On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Rev.* **29**(1), 21–89 (1987)
11. Hetlich, R., Kortanek, K.O.: Semi-infinite programming: theory, methods, and applications. *SIAM Rev.* **35**(3), 380–429 (1993)
12. Reemtsen, R., Görner, S.: Numerical methods for semi-infinite programming: a survey. In: Reemtsen, R., Rüdmann, J.J. (eds.) *Semi-infinite Programming*, pp. 195–275. Springer, Berlin (1998)
13. López, M.A., Still, G.: Semi-infinite programming. *Eur. J. Oper. Res.* **180**(2), 491–518 (2007)
14. Guerra Vázquez, F., Rückmann, J.J., Stein, O., Still, G.: Generalized semi-infinite programming: a tutorial. *J. Comput. Appl. Math.* **217**(2), 394–419 (2008)
15. Stein, O.: How to solve a semi-infinite optimization problem. *Eur. J. Oper. Res.* **223**(2), 312–320 (2012)
16. Lo Bianco, C.G., Piazzì, A.: A hybrid algorithm for infinitely constrained optimization. *Int. J. Syst. Sci.* **32**(1), 91–102 (2001)
17. Bhattacharjee, B., Green, W.H., Barton, P.I.: Interval methods for semi-infinite programs. *Comput. Optim. Appl.* **30**(1), 63–93 (2005)
18. Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. *SIAM J. Optim.* **18**(4), 1187–1208 (2008)
19. 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)
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. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. *J. Optim. Theory Appl.* **19**(2), 261–281 (1976)
22. Bhattacharjee, B., Lemonidis, P., Green, W.H., Barton, P.I.: Global solution of semi-infinite programs. *Math. Program.* **103**(2), 283–307 (2005)
23. Lemonidis, P.: Global optimization algorithms for semi-infinite and generalized semi-infinite programs. Ph.D. thesis. Massachusetts Institute of Technology, Boston, MA (2008)
24. Mitsos, A.: Global optimization of semi-infinite programs via restriction of the right-hand side. *Optimization* **60**(10–11), 1291–1308 (2011)
25. 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)
26. 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)
27. Djelassi, H., Mitsos, A.: A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs. *J. Glob. Optim.* **68**(2), 227–253 (2017)
28. Djelassi, H.: Discretization-based algorithms for the global solution of hierarchical programs. Ph.D. thesis. RWTH Aachen University, Aachen, Germany (2020)
29. Falk, J.E., Hoffman, K.: A nonconvex max–min problem. *Nav. Res. Logist. Q.* **24**(3), 441–450 (1977)
30. Harwood, S.M., Papageorgiou, D.J., Trespalacios, F.: A note on semi-infinite program bounding methods (2019). [arXiv:1912.01763v1](https://arxiv.org/abs/1912.01763v1)
31. Djelassi, H., Glass, M., Mitsos, A.: Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. *J. Glob. Optim.* **75**(2), 341–392 (2019)
32. Bongartz, D., Najman, J., Sass, S., Mitsos, A.: MAiNGO—McCormick-based algorithm for mixed-integer nonlinear global optimization. Technical report. Process Systems Engineering (AVT.SVT), RWTH Aachen University (2018). http://www.avt.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaabclahw. Accessed 3 Jul 2019