Journal of Global Optimization (2025) 93:27–61
https://doi.org/10.1007/s10898-025-01515-3

Discretization algorithms for generalized semi-infinite programs with coupling equality constraints under local solution stability

Aron Zingler[2] · Adrian W. Lipow[2] · Alexander Mitsos[1,2,3]

Received: 5 December 2024 / Accepted: 12 June 2025 / Published online: 25 June 2025

Abstract

Existing algorithms for generalized semi-infinite programs can only handle lower-level constraints containing equality constraints depending on upper-level variables (so-called coupling equality constraints) under limiting assumptions. More specifically, discretization-based algorithms require that the coupling equality constraints result in some lower-level variables being determined uniquely as implicit functions of the other lower-level and upper-level variables. We propose an adaptation of the discretization-based algorithm of Blankenship & Falk and demonstrate it can handle coupling equality constraints under the weaker assumption of stability of the solution set for these constraints in the sense of Lipschitz lower semi-continuity. The key idea is to allow a perturbation of the lower-level variable values from discretization points in connection with changes in the upper-level variables in the discretized upper-level problem. We enforce that these perturbed values satisfy the coupling equality constraints while remaining close to the discretization point, provided we can guarantee the stability of the solution in the sense that a nearby solution exists for small changes of the upper-level variables. We provide concrete realizations of the algorithm for three different situations: i) when knowledge about a certain Lipschitz constant is available, ii) when the coupling equality constraints are assumed to have full rank, and iii) when the coupling equality constraints are additionally linear in the lower-level variables. Numerical experiments on small test problems and a physically motivated problem related to power flow illustrate that the approach can be successfully applied to solve the challenging problems, but is currently limited in terms of scalability.

Keywords Semi-Infinite Programming · Generalized Semi-Infinite Programming · Nonconvex · Equality Constraints · Global Optimization

✉ Alexander Mitsos
amitsos@alum.mit.edu

1 JARA-CSD, Aachen 52056, Germany

2 Process Systems Engineering (AVT.SVT), RWTH Aachen University, Aachen 52074, Germany

3 Institute of Climate and Energy Systems: Energy Systems Engineering (ICE-1), Forschungszentrum Jülich GmbH, Jülich 52425, Germany

1 Introduction

We are concerned with the generalized semi-infinite optimization program (GSIP)

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x},\boldsymbol{y}) \leqslant 0, \quad \forall \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \end{aligned} \tag{GSIP-eq}$$

where the description of the feasible region of the lower level $\mathcal{Y}(\boldsymbol{x})$ contains coupling constraints, both equality $\boldsymbol{h}^{eq}$ and inequality $\boldsymbol{h}^{ineq}$ as

$$\mathcal{Y}(\boldsymbol{x}) := \{\boldsymbol{y} \in \bar{\mathcal{Y}} \subsetneq \mathbb{R}^{n_y} \mid \boldsymbol{h}^{ineq}(\boldsymbol{x},\boldsymbol{y}) \leqslant \boldsymbol{0},\ \boldsymbol{h}^{eq}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}\}. \tag{1}$$

We call $\mathcal{X}$ the host set of the upper-level variables $\boldsymbol{x}$ and $\bar{\mathcal{Y}}$ the host set of the lower-level variables $\boldsymbol{y}$. Throughout, we make the following standard assumption of compactness and continuity:

Assumption 1 The hosts sets $\mathcal{X}$ and $\bar{\mathcal{Y}}$ are compact and the functions f, g, $\boldsymbol{h}^{ineq}$ and $\boldsymbol{h}^{eq}$ are continuous on the hosts sets $\mathcal{X}$ and $\bar{\mathcal{Y}}$.

Solving GSIPs and the related problem classes of (standard) semi-infinite problems (SIP) and bilevel-optimization problems (BLP) has many applications such as design centering, robust optimization, Chebyshev approximation and parameter estimation in thermodynamics of mixtures (see [1, 2] and references therein). Accordingly, a multitude of approaches for handling this problem have been discussed in the literature under different assumptions on the lower-level problem:

$$\max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} g(\boldsymbol{x},\boldsymbol{y}). \tag{LLP}$$

If problem (LLP) is convex, duality or stationarity can be used to formulate (GSIP-eq) as a single-level optimization problem. For example, in [3], the KKT conditions of the lower-level problem are embedded, leading to a problem with equilibrium constraints that must be smoothed. In this case, the inclusion of coupling equality constraints is straightforward [3]. However, convexity of the lower level is a strong assumption that is often violated in practice. Therefore, methods allowing for nonconvex lower-level problems are of interest.

In this work, we specifically focus on the class of adaptive discretization-based algorithms that are applicable to the case of nonconvex lower-level problems and aim for a global solution. This requires the global solution of the subproblems, which can be done using existing numerical solvers in a black-box fashion. Other approaches for solving SIPs such as approaches based on convexification of the lower-level [4, 5], based on interval arithmetic [6, 7] or the lower-bounding approach of [8] could potentially be extended to find global optima of GSIP with nonconvex lower-level. However, discussions of the required extensions, as well as additional ones that would enable the incorporation of coupling equality constraints in these approaches, appear to be absent from the existing literature. As a result, a detailed comparison of solution approaches for GSIP is outside the scope of our work. Interested readers are instead referred to [1, 2, 9] for an overview. Outside of algorithms directly targeted for the solution of GSIPs, existing algorithms for the solution of BLPs are of interest because the problem classes of GSIP and BLP are heavily connected [10]. Therefore, an approach for BLP with coupling equality constraints could give insights into handling coupling equality constraints in the context of GSIP. However, while many different algorithms for BLP with nonconvex lower-level exist [11–13], they are not applicable for coupling equality constraints.

The solution of GSIP with coupling equalities is not straightforward in adaptive discretization methods. For example, the approaches by Mitsos and Tsoukalas [14] or Tsoukalas et al.

[15, 16] can be applied to solve GSIP globally without assuming convexity; however, they rely on a critical assumption that restricts their applicability in the context of coupling equality constraints. Specifically, one of the necessary assumptions (see Assumption 2) requires, roughly speaking, that it is safe to ignore lower-level points on the boundary of the set defined by the coupling lower-level constraints. Notably, this means no coupling equality constraints, even when linear, are allowed, which can be limiting.

To address this limitation, two discretization-based methods that allow coupling equality constraints under certain circumstances have been presented in the open literature. In the approach of Djelassi et al. [17], we assumed that the lower-level variables are a priori divided into dependent and independent variables in such a way that the dependent variables are uniquely defined by the values of the upper- and independent lower-level variables and possibly by so-called selection constraints inherent to the problem. Because of this uniqueness, only the independent variables have to be fixed by the discretization points, while the dependent variables are added as additional variables to the upper-level problems, together with the coupling equality constraints. Previously, Stuber et al. [18] had proposed a different approach under a similar assumption. Instead of adding additional variables to the upper-level problems, they use a parametric interval-Newton method to bound the range of the implicit variables and then employ a global optimization algorithm that uses a similar method internally to handle the implicit functions directly. A first step in extending this approach to a finite number of solutions for the dependent lower-level variables is found in [19], again based on parametric interval-Newton methods, for enclosing finitely many real solution branches of parameter-dependent nonlinear systems of equations. However, it remains unclear how these results can be used to solve the problem problem (GSIP-eq). In summary, existing algorithms for solving GSIPs with nonconvex lower levels and coupling equality constraints rely on an a priori given partition into dependent and independent lower-level variables and the assumption that the dependent lower-level variables can be defined as an (implicit) function that is unique for given values of the upper- and independent lower-level variables.

In this work, we identify additional situations under which coupling equality constraints can be handled by modifying the existing discretization approach for generalized semi-infinite programming. More specifically, we propose a modification of the discretized upper-level problem. The key idea is to allow a perturbation of the lower-level variable values from the discretization points in response to changes in the upper-level variables. The perturbed values are selected to satisfy the coupling equality constraints while ensuring they remain close to the original discretization points, provided we can guarantee stability in the sense of the existence of a nearby solution as the upper-level variables change. In this way, we avoid an a priori partition into dependent and independent variables by considering the coupling equality constraints as underdetermined systems of equations. As we will see, this allows us to solve problems where a constant partition is invalid.

In the remainder of this section, we introduce our notation. In Section 2, we first discuss existing discretization-based approaches for solving generalized semi-infinite optimization problems that do not consider coupling equality constraints and analyze the shortcomings of these approaches when coupling equalities are introduced. We also illustrate that assuming some lower-level variables can be interpreted as (implicit) functions can be limiting. We then present our idea to amend these shortcomings in Section 3. Afterward, we formalize our approach under the assumption that the solution set of the coupling equality constraints is locally lower Lipschitz semi-continuous. In Section 4, we identify special cases where this assumption holds and specialize our proposed method. Numerical experiments and implementation remarks are presented in Section 5.

Throughout, we use bold letters to denote vectors and matrices and calligraphic font for sets. We use the convention that the size of a vector $\boldsymbol{x}$ is denoted by n_x and that the elements of the vector $\boldsymbol{x}$ are denoted by x_i, $i \in \{1, \ldots, n_x\}$. Further, we use $\|\cdot\|$ to denote an arbitrary vector norm or its induced matrix norm and denote a ball with radius r around the point $\hat{\boldsymbol{x}}$ with $\mathcal{B}_r(\hat{\boldsymbol{x}}) := \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leqslant r\right\}$. Considering a vector-valued function with k vector-valued arguments $\boldsymbol{f}\colon \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_k} \to \mathbb{R}^m$, we use the notation $\boldsymbol{D}_i \boldsymbol{f}\colon \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_k} \to \mathbb{R}^m \times \mathbb{R}^{n_i}$ for $i \in \{1, \ldots, k\}$ to denote the Jacobian of $\boldsymbol{f}$ with respect to its i-th argument. We use $\boldsymbol{D}\boldsymbol{f}$ instead if the function has only a single argument. For second-order derivatives, we analogously use $\boldsymbol{D}_{i,j} \boldsymbol{f}\colon \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_k} \to \mathbb{R}^m \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_j}$.

2 Existing Discretization-based algorithms for (G)SIP

The general idea in the discretization-based algorithms for (G)SIPs in [14, 17, 18, 20, 21] goes back to a procedure by [22]. Therein, the standard semi-infinite optimization problem

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & G(\boldsymbol{x}, \boldsymbol{y}) \leqslant 0, \quad \forall \boldsymbol{y} \in \mathcal{Y} \end{aligned} \tag{SIP}$$

is solved by adaptive discretization, resulting in outer approximation, i.e., the lower bounding problem

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & G(\boldsymbol{x}, \boldsymbol{y}^d) \leqslant 0, \quad \forall \boldsymbol{y}^d \in \mathcal{Y}^D \end{aligned} \tag{SIP-LBP}$$

is solved for a finite set of discretization points $\mathcal{Y}_k^D \subset \mathcal{Y}$. This constitutes a relaxation of problem (SIP) and provides a lower-bound and an iterate $\bar{\boldsymbol{x}}$ for the upper-level variable $\boldsymbol{x}$. Then a solution $\boldsymbol{y}^*$ of the lower-level problem $\max\limits_{\boldsymbol{y} \in \mathcal{Y}} G(\bar{\boldsymbol{x}}, \boldsymbol{y})$ is calculated. If $G(\bar{\boldsymbol{x}}, \boldsymbol{y}^*) \leqslant 0$, the iterate $\bar{\boldsymbol{x}}$ is feasible and thus optimal. Otherwise, $\boldsymbol{y}^*$ is added to $\mathcal{Y}^D$, which guarantees that $\bar{\boldsymbol{x}}$ is no longer feasible in the next iteration of problem (SIP-LBP). Convergence to a feasible point is only guaranteed in the limit. Several authors [16, 20, 23] have extended this approach to generate feasible iterates in finite iterations.

2.1 Discretization based algorithms for GSIP without coupling equality constraints

In [14, 21], problem (GSIP-eq) is considered without coupling equality constraints and is reformulated to

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}, \boldsymbol{y}) \leqslant 0 \bigvee_{i=1}^{n_{ineq}} h_i^{ineq}(\boldsymbol{x}, \boldsymbol{y}) > 0, \quad \forall \boldsymbol{y} \in \mathcal{Y} \end{aligned} \tag{GSIP-REF}$$

and then relaxed to

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}, \boldsymbol{y}) \leqslant 0 \bigvee_{i=1}^{n_{ineq}} h_i^{ineq}(\boldsymbol{x}, \boldsymbol{y}) \geqslant 0, \quad \forall \boldsymbol{y} \in \mathcal{Y}. \end{aligned} \tag{GSIP-REL}$$

It is further assumed that the infimum of problem (GSIP-REF) is equal to the minimum of problem (GSIP-REL) (see [14, Assumption 2] and [17, Assumption 5]). Given this assumption, the lower-bounding approach of [22] can be used with

$$G(\boldsymbol{x}, \boldsymbol{y}) := \min \left\{ g(\boldsymbol{x}, \boldsymbol{y}), \min_{i=1 \ldots n_{ineq}} - h_i^{ineq}(\boldsymbol{x}, \boldsymbol{y}) \right\} \tag{2}$$

in problem (SIP). The resulting lower-bounding problem can be written as

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & G(\boldsymbol{x}, \boldsymbol{y}^d) \leqslant 0, \quad \forall \boldsymbol{y}^d \in \mathcal{Y}^d \ . \end{aligned} \tag{LBP}$$

In [14, 17, 21], a so-called lower-level Slater point is searched and used as the discretization point to be added to $\mathcal{Y}^D$ in problem (SIP-LBP). In [17, 21] this is achieved by solving the problem

$$\max_{\boldsymbol{y}} G(\boldsymbol{x}, \boldsymbol{y}) \text{ s.t. } \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}). \tag{GSIP-LLP}$$

In [14], we show that the assumption of equivalence of problem (GSIP-REL) and problem (GSIP-REF) is slightly weaker than the following assumption on the existence of such a lower-level Slater point:

Assumption 2 Let f^* be the infimum of problem (GSIP-REF). Each infeasible upper-level point $\hat{\boldsymbol{x}} \in \mathcal{X}$ with $f(\hat{\boldsymbol{x}}) < f^*$ can be excluded by a lower-level Slater point with respect to the inequality constraints, i.e,

$$\exists \hat{\boldsymbol{y}} \in \mathcal{Y} : g(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) > 0 \wedge \boldsymbol{h}^{ineq}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) < \boldsymbol{0}.$$

It is paramount that the discretization point is chosen as a Slater point with respect to the lower-level constraints because this ensures that the discretization point remains lower-level feasible in the neighborhood of the last upper-level iterate. This is not possible in the context of coupling equality constraints. As the following example shows, this prohibits a naive application of the idea of [22].

Example 1 ([24]) Consider the GSIP

$$\begin{aligned} \min_{\boldsymbol{x} \in [0,1]^2} \quad & x_1 \\ \text{s.t.} \quad & x_1 \geqslant 1, \quad \forall y \in [0, 1] : y = x_2 \end{aligned} \tag{3}$$

and any upper-level point $\bar{\boldsymbol{x}}$ with $x_1 = 0$. The only solution to the lower-level problem is $y^1 = \bar{x}_2$. When we consider the discretization point y^1 fixed, the resulting discretized problem is

$$\begin{aligned} \min_{\boldsymbol{x} \in [0,1]^2} \quad & x_1 \\ \text{s.t.} \quad & x_1 \geqslant 1, \quad \forall y^d \in \{y^1\} : y^d = x_2. \end{aligned} \tag{4}$$

In this problem all upper-level points $\boldsymbol{x} \in [0, 1]^2$ with $x_2 \neq \bar{x}_2$ are feasible. Thus, $\bar{\boldsymbol{x}}$ with an arbitrarily small change in x_2 is feasible in the discretized problem. In other words, no neighborhood of $\bar{\boldsymbol{x}}$ is rendered infeasible by the discretization point. As a consequence, $x_1 = 0$ is feasible for all finite discretizations, while the optimal value is 1. This means that an improvement of the resulting lower-bound will not occur in finite steps even when ignoring the numerical difficulties of representing the constraint $x_2 \neq y^d$.

2.2 Discretization based algorithms for GSIP with coupling equality constraints

When considering coupling equality constraints, it becomes evident that treating discretization points is problematic, as illustrated in Example 1. An alternative approach would be to consider some lower-level variables as a function of the upper-level variables. For example, in Example 1, the coupling equality constraint uniquely determines y as a function of x_2 and can thus be used to eliminate the lower-level variable y as a degree of freedom. This is the key idea of the approaches in [17, 18].

These approaches rely on a decomposition of the lower-level variables into dependent and independent variables $\boldsymbol{y} = [\boldsymbol{y}^{dep,T}, \boldsymbol{y}^{indep,T}]$, where the coupling equality constraints $\boldsymbol{h}^{eq}$ uniquely determine the dependent lower-level variables $\boldsymbol{y}^{dep}$ as a function $\tilde{\boldsymbol{y}}^{dep}(\boldsymbol{x}, \boldsymbol{y}^{indep})$ of upper-level variables $\boldsymbol{x}$ and independent lower-level variables $\boldsymbol{y}^{indep}$. In other words, only the independent lower-level variables constitute degrees of freedom. Consequently, only the independent lower-level variables are fixed to discretization points in the discretized upper-level problem, i.e., in the lower bounding subproblem problem (SIP-LBP), while the dependent variables are adjusted such that the coupling equality constraints are satisfied. Such a decomposition does not always exist, as the following example shows:

Example 2 The GSIP with coupling equality constraints

$$
\begin{aligned}
\min_{\boldsymbol{x} \in [-0.75, 0.75]} \quad & -x_1 \\
\text{s.t.} \quad & (y_1 - x_1)(1 + y_2) + x_1 \leqslant 0, \quad \forall \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})
\end{aligned}
\tag{5}
$$

with

$$
\mathcal{Y}(\boldsymbol{x}) := \{\boldsymbol{y} \in [-1, 1]^2 \mid h^{eq}(\boldsymbol{x}, \boldsymbol{y}) = (x_1 - y_1)(x_1 - y_2) = 0\}
\tag{6}
$$

has the coupling equality constraint $h^{eq}(\boldsymbol{x}, \boldsymbol{y}) = (x_1 - y_1)(x_1 - y_2) = 0$ which does not allow for a unique interpretation as an implicit function from $\boldsymbol{x}$ and parts of $\boldsymbol{y}$ to the rest of $\boldsymbol{y}$. Indeed for given x_1, x_2, all $\boldsymbol{y} \in [-1, 1]^2$ with $y_1 = x_1$ or $y_2 = x_1$ are lower-level feasible. Choosing one of the lower-level variables as the dependent variable would ignore one of the two solution branches, constituting a lower-level restriction. Indeed, fixing y_1 as a function of y_2 and $\boldsymbol{x}$ would lead to $(1 + y_2)(x - x) = 0 \leqslant 0$, which changes the solution drastically.

In conclusion, expressing some lower-level variables in terms of others, such that the equality constraint is fulfilled, is not always possible. Further, finding such a decomposition can be difficult, and it is unclear how to find such a decomposition in general, even if it exists.

3 Handling coupling equality constraints with Lipschitz lower semi-continuous solution set

Example 1 shows that we must find a modification to problem (SIP-LBP) so that at least an open neighborhood of $\boldsymbol{x}^{*,(0)}$ will be rendered infeasible. As long as the resulting problem is still a relaxation of problem (GSIP-eq), we can expect to obtain convergence in the limit (this will be shown for our approach in the upcoming Theorem Theorem 1).

The key idea under the assumption of a unique solution discussed in Section 2.2 is to fix some lower-level variables by discretization and vary the others to fulfill the coupling equality constraints. If the solution to the coupling equality constraints is not unique, it is unclear how exactly the variables should be varied. In this case, we propose to select solutions close to the past discretization point, assuming the solution set of the coupling equality constraints is sufficiently stable so that such a solution exists after small changes in the upper-level variables. Conceptually, we would like to emulate embedding a local solver for systems of equations that converges to a nearby solution when started at the discretization point. A similar idea is used in [25], where the equivalent to our upper-level variable $\boldsymbol{x}$ is fixed in a sampling scheme. Instead of embedding such a local solver into the global optimization procedures, we would like to utilize the existing capability of optimizers to fulfill equality constraints. Thus, similar to [17], we propose to add a variable vector $\boldsymbol{y}^{d,'}$ to the relaxation of the upper level with the constraint that it fulfills the coupling equality constraints. So far, the approach is similar to so-called high-point relaxation in bilevel optimization [26]. On its own, this modification will not be advantageous. The following two problems will arise.

Firstly, allowing the added copy of the lower-level variables $\boldsymbol{y}^{d,'} \in \bar{\mathcal{Y}}$ to be freely chosen without considering the corresponding discretization point $\boldsymbol{y}^d$ will not guarantee that the current iterate $\hat{\boldsymbol{x}}$ will be rendered infeasible, let alone a neighborhood. This is because a second solution to the coupling equality constraints for the current iterate $\hat{\boldsymbol{x}}$ could be a less restrictive choice for the lower-level variables $\boldsymbol{y}^{d,'}$. To address this problem, we aim to force a selection of $\boldsymbol{y}^{d,'}$ close to $\boldsymbol{y}^d$ when $\boldsymbol{x}$ has not moved far from $\hat{\boldsymbol{x}}$. To this end, we limit the distance of $\boldsymbol{y}^{d,'}$ from $\boldsymbol{y}^d$ proportionally to the distance of $\boldsymbol{x}$ from $\hat{\boldsymbol{x}}$.

Secondly, adding the coupling equality constraints might make points $\boldsymbol{x} \in \mathcal{X}$ infeasible for which there exists no solution $\boldsymbol{y}^{d,'} \in \bar{\mathcal{Y}}$ which fulfills the coupling equality constraints $\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^{d,'}) = \boldsymbol{0}$. This is even more likely after we add the proportionality constraint of the last paragraph. To avoid this, we need the following assumption regarding the local stability of the solution set of the coupling equality constraints. The overall idea of our approach and Assumption 3 are visualized in Fig. 1.

Assumption 3 For each infeasible upper-level point $\hat{\boldsymbol{x}} \in \mathcal{X}$ and lower-level feasible point $\hat{\boldsymbol{y}} \in \mathcal{Y}(\hat{\boldsymbol{x}})$, the solution set of the lower-level equality constraints $\mathcal{S}(\boldsymbol{x}) := \{\boldsymbol{y} \in \bar{\mathcal{Y}} \mid \boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}\}$ fulfills the following Lipschitz lower semi-continuity condition [27]

$$\exists \hat{\delta}_{\hat{x}}, L_{\hat{x}} > \boldsymbol{0} : \forall \boldsymbol{x} \in \mathcal{B}_{\hat{\delta}_{\hat{x}}}(\hat{\boldsymbol{x}}) \left[\left(\mathcal{S}(\boldsymbol{x}) \cap \mathcal{B}_{L_{\hat{x}} \|\boldsymbol{x}-\hat{\boldsymbol{x}}\|}(\hat{\boldsymbol{y}}) \right) \neq \emptyset \right], \tag{7}$$

and the radii $\hat{\delta}_{\hat{x}}$ in problem (7) are bounded from below with a positive number δ_x.

Assumption 3 allows for the solution to be nonunique. It requires local stability of the solution set in the sense that at least one solution does not suddenly vanish when the upper-level variables are changed. Indeed, if a unique solution exists, Assumption 3 is implied by the Lipschitz continuity of this solution in regard to the upper-level variables.

In the following, we will additionally make the following assumption, which is identical to Assumption 4 in the absence of coupling equality constraints but requires that the Slater point also fulfills the equality constraints.

Assumption 4 Let f^* be the infimum of problem (GSIP-REF). Each infeasible upper-level point $\hat{\boldsymbol{x}} \in \mathcal{X}$ with $f(\hat{\boldsymbol{x}}) < f^*$ can be excluded by a lower-level Slater point with respect to the inequality constraints of the lower-level problem, i.e.,

$$\forall \hat{\boldsymbol{x}} \in \mathcal{X} : f(\hat{\boldsymbol{x}}) < f^* \exists \boldsymbol{y} \in \mathcal{Y}(\hat{\boldsymbol{x}}) : \left[g(\hat{\boldsymbol{x}}, \boldsymbol{y}) > 0 \wedge \boldsymbol{h}^{ineq}(\hat{\boldsymbol{x}}, \boldsymbol{y}) < \boldsymbol{0} \right] \tag{8}$$

In the next section, we formalize the sketched solution idea. We first state and discuss the modified lower-bounding problem, followed by the resulting algorithm, and show that the latter converges under the preceding assumptions. We present the algorithm and the

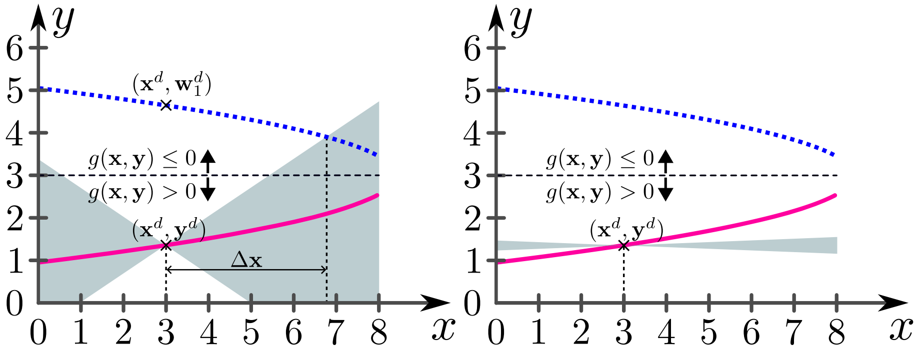

Fig. 1 Effect of constraining the deviation from the discretization point by the deviation from the corresponding upper-level iterate. The thick curves show two solution branches for the coupling equality constraints $\boldsymbol{h}^{eq}$. Only points on the solid magenta branch show a violation of the SIP constraint. On the left: We allow a deviation from the discretization point $\boldsymbol{y}^d$, but limit the change to the set $\mathcal{T}$ (represented by the gray area). In this example, the allowed deviation grows proportionally to the distance in the upper-level variables. Thus, the second (blue, square-dotted) solution branch can only be selected after $\boldsymbol{x}$ has moved at least $\Delta\boldsymbol{x}$. Since Assumption 3 holds, this is possible without cutting off the solution branch the discretization point was on. Without limiting the deviation from the deviation point, the optimizer could select $\boldsymbol{w}_1^d$ without changing from $\boldsymbol{x}^d$. On the right: If the allowed change of the lower-level variable from $\boldsymbol{y}^d$ is too restrictive, we might artificially introduce infeasibility; thus the need for Assumption 3

associated proofs in terms of a set-valued function $\mathcal{T}$. Intuitively, the role of this function is to restrict the deviation of lower-level variables values from the discretization points in the discretized upper-level problem, see Fig. 1. We will require $\mathcal{T}$ to fulfill the following properties for any $\boldsymbol{x}^d \in \mathcal{X}$ and $\boldsymbol{y}^d$ with $\boldsymbol{y}^d \in \mathcal{Y}(\boldsymbol{x}^d)$:

P.1 $\mathcal{T}(\boldsymbol{x}^d, \boldsymbol{x}^d, \boldsymbol{y}^d) = \{\boldsymbol{y}^d\}$ (consistency)

P.2 $\forall \boldsymbol{x} \in \mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)\left[\left(\mathcal{S}(\boldsymbol{x}) \cap \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)\right) \neq \emptyset\right]$ (satisfiability)

P.3 $\forall \delta_w > 0 \ \exists \delta_t > 0 : \forall \boldsymbol{x}^d \in \mathcal{X}\left[\|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_t \implies \forall \boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)\left[\|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \delta_w\right]\right]$ (continuity).

where $\mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)$ denotes a validity neighborhood for property **P.2** which we assume includes at least a ball around $\boldsymbol{x}^d$ with fixed radius $\delta_{\mathcal{T}}$, i.e., $\mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d) \supseteq \mathcal{B}_{\delta_{\mathcal{T}}}(\boldsymbol{x}^d)$ for each infeasible upper-level point $\boldsymbol{x}^d \in \mathcal{X}$. The set $\mathcal{T}$ can be seen as a generalization of the ball $\mathcal{B}_{L_{x^d}\|\boldsymbol{x}-\boldsymbol{x}^d\|}(\boldsymbol{y}^d)$ in Assumption 3. Indeed, note that as a result of Assumption 3, the choice

$$\mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d) := \mathcal{B}_{L_{x^d}\|\boldsymbol{x}-\boldsymbol{x}^d\|}(\boldsymbol{y}^d)$$

always fulfills all the listed properties. However, in this case, knowledge of the Lipschitz constant L_{x^d} would be needed to implement our proposed procedure in practice. We later discuss different choices for $\mathcal{T}$ where this knowledge is not required. Therefore, we will present the results regarding the lower-bounding problem and the resulting algorithm for arbitrary $\mathcal{T}$ with properties **P.1**, **P.2** and **P.3**.

3.1 Lower-bounding problem

As in the reformulation of GSIP to SIP discussed in the introduction, we utilize

$$G(\boldsymbol{x}, \boldsymbol{y}) := \min\left\{g(\boldsymbol{x}, \boldsymbol{y}), \min_i\{-h_i^{ineq}(\boldsymbol{x}, \boldsymbol{y})\}\right\}$$

to combine the objective and inequality constraints of the lower level into a single function G and shorten the notation. To handle coupling equality constraints, we propose the following lower-bounding problem instead of problem (LBP)

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{w}^d\in\mathbb{R}^{\bar{n}_y}, d\in\mathcal{D}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \left\{G(\boldsymbol{x}, \boldsymbol{w}^d) \leqslant 0 \wedge \left(\boldsymbol{w}^d \in \left(\mathcal{S}(\boldsymbol{x}) \cap \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)\right)\right)\right\} \\ & \vee \left\{\boldsymbol{x} \notin \mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)\right\}, \quad \forall d \in \mathcal{D}\,. \end{aligned} \tag{LBP-eq}$$

Therein, the additional variables $\boldsymbol{w}^d, d \in \mathcal{D}$ replace fixed values of the discretization variables in G compared to problem (LBP). They are constrained to fulfill the coupling equality constraints ($\boldsymbol{w}^d \in \mathcal{S}(\boldsymbol{x})$) and to stay within a controlled distance from the discretization points $\boldsymbol{y}^d$, $d \in \mathcal{D}$, ($\boldsymbol{w}^d \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)$), unless the upper variables $\boldsymbol{x}$ are outside the neighborhood $\mathcal{U}_{\mathcal{T}}$. We obtain the following results regarding problem (LBP-eq).

Lemma 1 *Given Assumption 1, and a SIP-infeasible upper-level point $\boldsymbol{x}^d \in \mathcal{X}$, a Slater point $\boldsymbol{y}^d$ with respect to the inequalities of the lower-level, i.e., $\boldsymbol{h}^{ineq}(\boldsymbol{x}^d, \boldsymbol{y}^d) < \boldsymbol{0}$, that proves the SIP-infeasiblity of the upper-level point $\boldsymbol{x}^d$, i.e., $\boldsymbol{y}^d \in \mathcal{Y}(\boldsymbol{x}^d)$ and $g(\boldsymbol{x}^d, \boldsymbol{y}^d) > 0$, renders all points in a neighborhood $\mathcal{B}_\delta(\boldsymbol{x}^d)$ of that upper-level point with radius $\delta > 0$ infeasible in problem (LBP-eq).*

Proof Note that the constraint in problem (LBP-eq) is a disjunction. For the second part of the disjunction, i.e., $\boldsymbol{x} \notin \mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)$, the claim follows trivially.

For the first part of the disjunction, we show that

$$\exists \delta > 0 : \forall \boldsymbol{x} \in \mathcal{B}_\delta(\boldsymbol{x}^d), \forall \boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)\, [G(\boldsymbol{x}, \boldsymbol{w}) > 0]\,.$$

Since the function G is uniformly continuous (continuous on a compact set), we obtain:

$$\begin{aligned} \forall \delta_1 > 0\, \exists \delta_2 > 0 : \forall \boldsymbol{w}, \boldsymbol{x} : \|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \delta_2 \text{ and } \|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_2 \\ \left[\|G(\boldsymbol{x}, \boldsymbol{w}) - G(\boldsymbol{x}^d, \boldsymbol{y}^d)\| \leqslant \delta_1\right]. \end{aligned} \tag{9}$$

Since the discretization point $\boldsymbol{y}^d$ is a lower-level Slater point with respect to the inequalities, we have $G(\boldsymbol{x}^d, \boldsymbol{y}^d) = \epsilon^d > 0$ (otherwise $g(\boldsymbol{x}^d, \boldsymbol{y}^d) \leqslant 0$ means that the upper-level point $\boldsymbol{x}^d$ was SIP-feasible). Set d_1 in problem (9) to $\frac{\epsilon^d}{2}$. From problem (9), we obtain the existence of a δ_2 such that

$$\forall \boldsymbol{w}, \boldsymbol{x} : \|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \delta_2 \text{ and } \|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_2 : \left[G(\boldsymbol{x}, \boldsymbol{w}) \geqslant \frac{\epsilon^d}{2} > 0\right].$$

Since we enforce $\boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)$, the property **P.3** of $\mathcal{T}$ means that there is a $\delta_t > 0$ such that $\|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_{\mathcal{T}} \implies \|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \delta_2$. As a result, the claim holds with $\delta = \min\{\delta_2, \delta_t\}$. □

Lemma 2 *Given a convergent sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ of an SIP-infeasible upper-level points converging to $\bar{\boldsymbol{x}}$, i.e., $\boldsymbol{x}^k \xrightarrow{k\to\infty} \bar{\boldsymbol{x}}$, and a corresponding sequence of lower-level points $\{\boldsymbol{y}^k\}_{k\in\mathbb{N}}$ as in Theorem 1. If there exists a fixed lower bound $\bar{\epsilon} > 0$ for the function values $G(\boldsymbol{x}^k, \boldsymbol{y}^k)$ for all k, there exists a fixed lower bound $\bar{\delta}$ for the radius δ in Theorem 1 for all k. In other words*

$$\left[\exists\bar{\epsilon} > 0 : \forall k \in \mathbb{N} \left[G(\boldsymbol{x}^k, \boldsymbol{y}^k) \geqslant \bar{\epsilon}\right]\right] \Longrightarrow \left[\exists\bar{\delta} > 0 : \forall \boldsymbol{x} \in \mathcal{B}_{\bar{\delta}}(\boldsymbol{x}^k), \forall \boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^k, \boldsymbol{y}^k)\ [G(\boldsymbol{x}, \boldsymbol{w}) > 0]\right]. \tag{10}$$

Proof Follows from the proof of Theorem 1 after replacing ϵ^d with $\bar{\epsilon}$ and noting that the uniform continuity of G and the corresponding property **P.3** of $\mathcal{T}$ guarantee the existence of δ_2 and δ_T that are independent of $\boldsymbol{x}^d$ and $\boldsymbol{y}^d$. □

Lemma 3 *Given Assumption 3, the problem (LBP-eq) is a relaxation of the original GSIP in problem (GSIP-eq).*

Proof First, denote the feasible set of problem (GSIP-eq) as $\mathcal{F}_{GSIP}$ and the neighborhood around the point $\boldsymbol{x}^d \in \mathcal{X}$ and its complement with $\mathcal{B}^d := \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{x} \in \mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)\}$ and $\bar{\mathcal{B}}^d := \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{x} \notin \mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d)\}$, respectively.

Further, denote with $\mathcal{F}^{HP}$ the projection of the feasible region of the constructed lower-bounding problem (LBP-eq) into the x-coordinate. The set $\mathcal{F}^{HP}$ is obtained by the intersection of the feasible sets of upper-level variables for each discretization point index d, i.e.,

$$\mathcal{F}^{HP} = \bigcap_{d\in\mathcal{D}} \left\{\mathcal{F}^{HP,d} \cup \bar{\mathcal{B}}^d\right\} \tag{11}$$

with

$$\mathcal{F}^{HP,d} := \left\{\boldsymbol{x} \in \mathcal{B}^d \mid \exists \boldsymbol{w}^d \in (\mathcal{S}(\boldsymbol{x}) \cap \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)) : G(\boldsymbol{x}, \boldsymbol{w}^d) \leqslant 0\right\}.$$

Since the objective functions of problem (GSIP-eq) and problem (LBP-eq) are identical, we need to show that replacing the feasible region of the original problem $\mathcal{F}_{\text{GSIP}}$ with the set $\mathcal{F}^{HP}$ constitutes a relaxation, i.e., $\mathcal{F}_{\text{GSIP}} \subseteq \mathcal{F}^{HP}$. For each discretization point index $d \in \mathcal{D}$, define the set of points feasible in the neighborhood around a given point $\boldsymbol{x}^d$, $d \in \mathcal{D}$ as

$$\mathcal{F}^{M,d} := \left\{\boldsymbol{x} \in \mathcal{B}^d \mid \max_{\boldsymbol{w}\in\mathcal{S}(\boldsymbol{x})} G(\boldsymbol{x}, \boldsymbol{w}) \leqslant 0\right\}.$$

If we combine these sets with the complement of the neighborhood $\bar{\mathcal{B}}^d$, where we ignore the semi-infinite constraint, we obtain a relaxation:

$$\mathcal{F}_{GSIP} \subseteq \bigcap_{d\in\mathcal{D}} \left\{\mathcal{F}^{M,d} \cup \bar{\mathcal{B}}^d\right\}$$

This follows from the fact that this combination is a relaxation for each discretization index d

$$\begin{aligned}\mathcal{F}_{GSIP} &= \left(\mathcal{F}_{GSIP} \cap \bar{\mathcal{B}}^d\right) \cup \left(\mathcal{F}_{GSIP} \cap \mathcal{B}^d\right)\\ &\subseteq \bar{\mathcal{B}}^d \cup \left(\mathcal{F}_{GSIP} \cap \mathcal{B}^d\right)\\ &\subseteq \bar{\mathcal{B}}^d \cup \mathcal{F}^{M,d}\end{aligned} \tag{12}$$

and that the intersection of relaxations is also a relaxation.

We note that the sets $\mathcal{F}^{M,d}$ can be relaxed to the feasible set for a relaxed SIP constraint, where the lower-level variable is restricted to a small area around the discretization point, i.e., $\mathcal{F}^{M,d} \subseteq \mathcal{F}^{MM,d}$ with

$$\mathcal{F}^{MM,d} := \left\{ \boldsymbol{x} \in \mathcal{B}^d \mid \max_{\boldsymbol{w} \in \bar{\mathcal{Y}} \cap \mathcal{S}(\boldsymbol{x}) \cap \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)} G(\boldsymbol{x}, \boldsymbol{w}) \leqslant 0 \right\}.$$

Because of Assumption 3, we know that $\bar{\mathcal{Y}} \cap \mathcal{S}(\boldsymbol{x}) \cap \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)$ does not become empty and thus $\mathcal{F}^{MM,d} \subseteq \mathcal{F}^{HP,d}$.

The inclusion $\mathcal{F}^{M,d} \subseteq \mathcal{F}^{HP,d}$ is sufficient to prove the claim of the theorem $\mathcal{F}_{\text{GSIP}} \subseteq \mathcal{F}^{HP}$ because the inclusion relationship in preserved after taking the union of $\mathcal{F}^{HP,d}$ and $\mathcal{F}^{MM,d}$ with $\bar{\mathcal{B}}^d$, respectively, and taking the intersection over all $d \in \mathcal{D}$:

$$\begin{aligned}
& \mathcal{F}_{GSIP} \subseteq \bigcap_{d \in \mathcal{D}} \left\{ \mathcal{F}^{M,d} \cup \bar{\mathcal{B}}^d \right\} \\
\overset{\mathcal{F}^{M,d} \subseteq \mathcal{F}^{MM,d}}{\Longrightarrow} \quad & \mathcal{F}_{GSIP} \subseteq \bigcap_{d \in \mathcal{D}} \left\{ \mathcal{F}^{MM,d} \cup \bar{\mathcal{B}}^d \right\} \\
\overset{\mathcal{F}^{MM,d} \subseteq \mathcal{F}^{HP,d}}{\Longrightarrow} \quad & \mathcal{F}_{GSIP} \subseteq \bigcap_{d \in \mathcal{D}} \left\{ \mathcal{F}^{HP,d} \cup \bar{\mathcal{B}}^d \right\} = \mathcal{F}^{HP}
\end{aligned} \tag{13}$$

□

3.2 Lower-bounding algorithm

We use the lower-bounding problem in the adaptive discretization algorithm listed in Algorithm 1. The following theorem discusses the convergence of this algorithm.

Theorem 1 *Given Assumptions 3, 4 and 1, the lower bounding procedure in Algorithm 1 either finitely proves the problem to be infeasible, finitely returns a global solution or produces a sequence of iterates* $\left\{\boldsymbol{x}^i\right\}_{i \in \mathbb{N}}$ *such that each accumulation point is a global solution of problem (GSIP-eq).*

Proof (similar to Lemma 4.3 in [24]) Assume that the algorithm terminates finitely. Denote with i the final iteration. Then, either $G^i \leqslant 0$ problem (GSIP-LLP) was infeasible or problem (LBP-eq) was infeasible. In the latter case, since problem (LBP-eq) is a relaxation according to Theorem 3, the original problem (GSIP-eq) must be infeasible. In the former two cases, the current iterate $\boldsymbol{x}^i$ is SIP-feasible according to Assumption 4.

If Algorithm 1 does not terminate finitely, there is an infinite sequence of iterates. Because of the compactness of $\mathcal{X}$, there is at least one converging subsequence. Without loss of generality, pick a single subsequence denoted by $\{\boldsymbol{x}^{n_k}\}_{k \in \mathbb{N}}$ with $n_k \in \mathbb{N}$ and $n_k + 1 > n_k$ for $k > 0$ that converges to a point $\bar{\boldsymbol{x}}$. If $G^{n_k} \xrightarrow{k \to \infty} 0$, the accumulation point $\bar{\boldsymbol{x}}$ is SIP-feasible. Otherwise, there must exist an $\bar{\epsilon} > 0$ such that $G^{n_k} > \bar{\epsilon}$ (since the algorithm does not terminate, we must have $G^{n_k} > 0$ for all $k \in \mathbb{N}$). According to Theorem 1, this would mean that in each step of the subsequence k a neighborhood $\mathcal{B}_\delta(\boldsymbol{x}^{n_k})$ of the last iterate $\boldsymbol{x}^{n_k}$ becomes infeasible in problem (LBP-eq). According to Theorem 2 the radius of this neighborhood δ does not fall below the fixed value $\hat{\delta} := \min\{\bar{\delta}, \delta_x\}$. Since $\mathcal{X}$ is compact, problem (LBP-eq) would become infeasible after a finite number of steps k, which leads to a contradiction. □

```
// Initialize
D ← {};
i ← 0;
LBD ← −∞;
while true do
    Solve problem (LBP-eq);
    if Infeasible then
        Problem (GSIP-eq) is infeasible. return Infeasible;
    else
        Save the solution x^i and the objective value f(x^i);
    end
    // Update lower bound
    LBD := f(x^i);
    Solve problem (GSIP-LLP) given x^i saving the solution y^i and the objective value G^i;
    if Infeasible or G^i ⩽ 0 then
        Found a feasible point. return x^i ;
    end
    // Update index set with index of new discretization point y^i and
       upper-level iterate x^i
    D := D ∪ {i};
    i := i + 1;
end
```

Algorithm 1: The proposed algorithm for the solution of generalized semi-infinite programs (GSIPs) with coupling equality constraints.

3.3 Sufficient conditions for Stability of the lower-level solution set

We are interested in sufficient conditions for the stability of the lower-level solution set, more specifically for Assumption 3 to hold. The following example shows that without further assumptions, a problem with arbitrary coupling equality constraints $\boldsymbol{h}^{eq}$ can be reformulated such that the characteristics of $\boldsymbol{h}^{eq}$ are essentially hidden within the host set $\bar{\mathcal{Y}}$.

Example 3 Consider $\mathcal{X} = [\boldsymbol{x}^L, \boldsymbol{x}^U]$ and $\bar{\mathcal{Y}}_1 = [\boldsymbol{y}^L, \boldsymbol{y}^U]$ and define the lower-level feasible set as $\mathcal{Y}_1(\boldsymbol{x}) = \{\boldsymbol{y} \in \bar{\mathcal{Y}}_1 \mid \boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}\}$. Using a more complicated host set, we can reformulate this as a problem where all coupling equality constraints are linear.

Indeed, combine the old lower-level variables $\boldsymbol{y}$ with the new variables $\boldsymbol{z}$ according to $\boldsymbol{w}^T = [\boldsymbol{y}^T, \boldsymbol{z}^T]$. Then, we can use the more complicated host set $\bar{\mathcal{W}} = \{\boldsymbol{w} \in \bar{\mathcal{Y}}_1 \times \mathcal{X} \mid \boldsymbol{h}^{eq}(\boldsymbol{z}, \boldsymbol{y}) = \boldsymbol{0}\}$ to formulate the feasible set of the lower-level as $\mathcal{Y}_2(\boldsymbol{x}) = \{\boldsymbol{w} \in \bar{\mathcal{W}} \mid \boldsymbol{z} - \boldsymbol{x} = \boldsymbol{0}\}$ which has a simple linear coupling equality constraint. More concretely, the problem

$$\begin{aligned} \max_{\boldsymbol{y} \in [\boldsymbol{y}^L, \boldsymbol{y}^U]} \quad & g(\bar{\boldsymbol{x}}, \boldsymbol{y}) \\ \text{s.t.} \quad & \boldsymbol{h}^{eq}(\bar{\boldsymbol{x}}, \boldsymbol{y}) = \boldsymbol{0} \end{aligned} \tag{14}$$

can also be written as

$$\begin{aligned} \max_{(\boldsymbol{y}, \boldsymbol{z}) \in \bar{\mathcal{W}}} \quad & g(\bar{\boldsymbol{x}}, \boldsymbol{y}) \\ \text{s.t.} \quad & \bar{\boldsymbol{x}} - \boldsymbol{z} = \boldsymbol{0} \end{aligned} \tag{15}$$

with $\bar{\mathcal{W}} = \{(\boldsymbol{y}, \boldsymbol{z}) \in [\boldsymbol{y}^L, \boldsymbol{y}^U] \times \mathcal{X} \mid \boldsymbol{h}^{eq}(\boldsymbol{z}, \boldsymbol{y}) = \boldsymbol{0}\}$. The case where inequalities defining $\bar{\mathcal{Y}}$ become active is similarly problematic.

In order to simplify the theoretical analysis and the statement of the algorithm, we consider a formulation of the problem where all relevant details are explicitly stated in terms of the lower-level constraints. To formalize this, we introduce the following assumption that asserts the redundancy of the host set $\bar{\mathcal{Y}}$.

Assumption 5 The host set $\bar{\mathcal{Y}}$ is redundant in the following sense: For any (visited) infeasible upper-level point $\hat{\boldsymbol{x}}$, any solution $\boldsymbol{y}^*$ to problem (GSIP-LLP) is strictly in the interior of the host-set:

$$\exists \tau > 0 : \forall \boldsymbol{y}^* \in \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}(\hat{\boldsymbol{x}})} G(\boldsymbol{x}, \boldsymbol{y}) \left[\mathcal{B}_\tau(\boldsymbol{y}^*) \subset \bar{\mathcal{Y}}\right].$$

If the original host set was defined by a combination of bound constraints and continuous constraints, Assumption 5 can easily be satisfied with a simple reformulation. Indeed, let the original host set be defined as

$$\bar{\mathcal{Y}}_1 = \{\boldsymbol{y} \in [\boldsymbol{y}^L, \boldsymbol{y}^U] \mid \boldsymbol{v}^{eq}(\boldsymbol{y}) = \boldsymbol{0} \wedge \boldsymbol{v}^{ineq}(\boldsymbol{y}) \leqslant \boldsymbol{0}\}$$

then we can add $\boldsymbol{v}^{eq}$ to the coupling equality constraints, add $\boldsymbol{v}^{ub}(\boldsymbol{y}) := \boldsymbol{y} - \boldsymbol{y}^U$, $\boldsymbol{v}^{lb}(\boldsymbol{y}) := -\boldsymbol{y} + \boldsymbol{y}^L$, as well as $\boldsymbol{v}^{ineq}$ to the coupling inequality constraints $\boldsymbol{h}^{ineq}$. As a result, the new host set with any $\tau > 0$

$$\bar{\mathcal{Y}}_2 = [\boldsymbol{y}^L - \tau, \boldsymbol{y}^U + \tau]$$

will satisfy Assumption 5.

Assuming that the coupling equality constraints $\boldsymbol{h}^{eq}$ are continuously differentiable, the rank of the Jacobian with respect to the lower-level variables $\boldsymbol{y}$ can potentially be used to validate Assumption 3. Indeed, given Assumption 5, the constant rank of the Jacobian of the coupling equality constraints in the neighborhood of $\hat{\boldsymbol{x}}$ and $\boldsymbol{y}^*$ as defined in Assumption 5 is a sufficient condition [28, 29] for Assumption 3. Because of the lower semi-continuity of the rank of a matrix, full rank at $\hat{\boldsymbol{x}}$ and $\boldsymbol{y}^*$ also suffices. Most constraint qualifications for the lower-level problem, such as MFCQ, imply full rank of the Jacobian of the coupling equality constraints. Thus, one might consider this full-rank condition a weak assumption. According to [30], we can usually expect this condition to hold for most upper-level variables $\boldsymbol{x}$. However, this assumption is not trivial since we would need the condition to hold for all superoptimal upper-level variables. The following example illustrates that Assumption 3 can fail to hold, even for relatively simple examples.

Example 4 Define $q(\boldsymbol{y}) = (y_1 + 0.75)(y_1 + 0.6)(y_1 - 0.3)$ and $h^{eq}(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{y}) - x_1$. As illustrated in Fig. 2, $q(\boldsymbol{y})$ has a local minimum near 0. Let the discretization point $\boldsymbol{y}^d$ be chosen as the local minimizer and take $x_1^d = q(\boldsymbol{y}^d)$. Then $h^{eq}(\boldsymbol{x}^d, \boldsymbol{y}^d) = 0$, but for $x_1 < x_1^d$, there exist no solution $\boldsymbol{y}$ to $h^{eq}(\boldsymbol{x}, \boldsymbol{y}) = 0$ that is close. Clearly, Assumption 3 does not hold.

Although the rank of the Jacobian can provide valuable insights into the validity of Assumption 3, it does not provide a complete understanding of the applicability of Algorithm 1. Specifically, although full or constant rank of the Jacobian is a sufficient condition, it is not necessary for Assumption 3. Indeed, the coupling equality $h^{eq}(\boldsymbol{x}, \boldsymbol{y}) = (x_1 - y_1)(x_1 - y_2)$ from Example 2 has a singular Jacobian for any $t \in \mathbb{R}$ and $x_1 = t$, $y_1 = t$, $y_2 = t$, but there clearly always exist a solution $\boldsymbol{w}$ with $\|\boldsymbol{y} - \boldsymbol{w}\| \leqslant 1|x_1 - t|$ and $h^{eq}(\boldsymbol{x}, \boldsymbol{w}) = 0$. Furthermore, while full rank of the Jacobian of the coupling equality constraint is sufficient for Assumption 3, to implement the presented algorithm we still need a way to calculate the corresponding Lipschitz constant L or find an alternative way to construct a restrictive set $\mathcal{T}$ satisfying properties **P.1** through **P.3**. We discuss this in the next section.

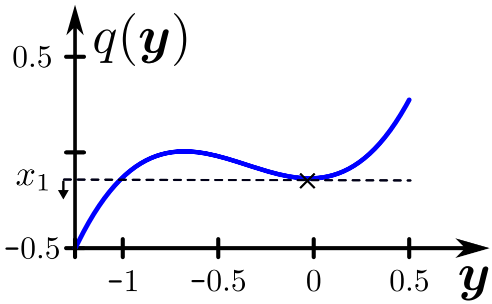

Fig. 2 Illustration of Example 4 where Assumption 3 does not hold. Here, $\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x}) - x_1$. When reducing x_1 further, no solutions exist close to the marked point

4 Specializations under full-rank assumption

In practice, we cannot always expect the knowledge of a valid approximation for the Lipschitz constant of the set $\mathcal{S}(\boldsymbol{x})$ that is reasonably tight. In the following, we specialize the approach to cases where knowledge of the Lipschitz constant is not explicitly required. Since it involves derivatives of the coupling equality constraints, we make the following assumption for the remainder of this section:

Assumption 6 The coupling equality constraints $\boldsymbol{h}^{eq}$ are Lipschitz continuously differentiable on $\mathcal{X} \times \bar{\mathcal{Y}}$ with respect to their second argument.

4.1 Specialization to Linear coupling equality constraints by Euclidean projection

In certain situations, we might have a way to directly express our preference for choosing a nearby solution to the coupling equality constraints directly by using the Euclidean projection to the solution set $S(\boldsymbol{x})$. Assume that we can, for a given upper-level point $\boldsymbol{x}^d$, easily compute one of the closest lower-level feasible points $\boldsymbol{y}$ to a given discretization point $\boldsymbol{y}^d$ in the Euclidean norm. If such a projection is available, we can solve the problem by choosing $\mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d) = \underset{\boldsymbol{y} \in \mathcal{S}(\boldsymbol{x})}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{y}^d\|_2^2$ since properties **P.1** and **P.2** hold trivially, and property **P.3** holds under Assumption 3 or if the Jacobian of the coupling equality constraints $\boldsymbol{D}_2\boldsymbol{h}^{\boldsymbol{eq}}(\boldsymbol{x}^d, \boldsymbol{y}^d)$ has full rank [31]. In the following, we discuss how to use the Euclidean projection for the particular case of linear coupling equality constraints, i.e.,

$$\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{A}(\boldsymbol{x})\boldsymbol{y} - \boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{0} \,. \tag{16}$$

The mapping to the closest solution of problem (16) to the previous solution $\boldsymbol{y}^d$ in the Euclidean norm is given by

$$\begin{aligned} \min_{\boldsymbol{w}^d \in \mathbb{R}^{n_y}} \quad & \frac{1}{2}\|\boldsymbol{w}^d - \boldsymbol{y}^d\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}^d = \boldsymbol{b}(\boldsymbol{x}) \,. \end{aligned} \tag{17}$$

This problem can be embedded using its KKT conditions

$$\begin{aligned} \boldsymbol{I}(\boldsymbol{w}^d - \boldsymbol{y}^d) + \boldsymbol{A}(\boldsymbol{x})^T\boldsymbol{\lambda} = \boldsymbol{0} \\ \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}^d = \boldsymbol{b}(\boldsymbol{x}) \end{aligned} \tag{18}$$

which only constitutes linear equality constraints without any equilibrium constraints. If the matrix $\boldsymbol{A}(\boldsymbol{x})$ has full rank for all $\boldsymbol{x} \in \mathcal{X}$, our lower-bounding problem for this case can be stated as

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{w}^d\in\mathbb{R}^{n_y},\boldsymbol{\lambda}^d\in\mathbb{R}^{n_{eq}},d\in\mathcal{D}} \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & \boldsymbol{I}(\boldsymbol{w}^d - \boldsymbol{y}^d) + \boldsymbol{A}(\boldsymbol{x})^T\boldsymbol{\lambda}^d = \boldsymbol{0}, && \forall d \in \mathcal{D} \\
& \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}^d = \boldsymbol{b}(\boldsymbol{x}), && \forall d \in \mathcal{D} \\
& G(\boldsymbol{x}, \boldsymbol{w}^d) \leqslant 0, && \forall d \in \mathcal{D} \ .
\end{aligned} \tag{LBP-lin}$$

Remark 1 This formulation is especially efficient in the case where the linear coupling equality constraints take on the form $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}(\boldsymbol{x})$, meaning that the coefficients of the coupling equality constraint are constant. In this case, we only introduce linear equality constraints.

Remark 2 Note that we cannot constrain the copy of the lower-level variables $\boldsymbol{w}^d$, $d \in \mathcal{D}$ to the lower-level host-set $\bar{\mathcal{Y}}$ in problem (LBP-lin). When numerically solving problem (LBP-lin) with solvers requiring bounded domains, one needs to be careful that bounds on $\boldsymbol{w}^d$ and $\boldsymbol{\lambda}$ are chosen without causing an unwanted restriction.

Note that this corresponds to applying the approach of [17] to the problem

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & G(\boldsymbol{x}, \boldsymbol{w}) \leqslant 0, \qquad \forall(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{y}) \in \mathcal{Y}(\boldsymbol{x})
\end{aligned} \tag{19}$$

with

$$\mathcal{Y}(\boldsymbol{x}) = \{(\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{\lambda}) \in \mathbb{R}^{n_y} \times \bar{\mathcal{Y}} \times \mathbb{R}^m \mid \boldsymbol{I}(\boldsymbol{w}^d - \boldsymbol{y}^d) + \boldsymbol{A}(\boldsymbol{x})^T\boldsymbol{\lambda}^d = \boldsymbol{0}, \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}^d = \boldsymbol{b}(\boldsymbol{x})\}$$

where $\boldsymbol{w}$ and $\boldsymbol{\lambda}$ are considered as the dependent variables. As a result, the algorithm presented in [17] and implemented as *RRHS* in our framework for discretization-based hierarchical optimization libDIPS [32] can be applied. The algorithm uses the steps in Algorithm 1 to find a lower bound but also introduces an upper-bounding procedure.

Alternatively, if the matrix $\boldsymbol{A}$ is constant and has full rank, a decomposition of $\boldsymbol{y}$ into a linear combination of dependent and independent variables is possible using a QR decomposition or a (pivoted) reduced row echelon form. Thus, it is possible to apply the approaches from Section 2.2 directly in this case. This is not necessarily true if the matrix $\boldsymbol{A}$ depends on the upper-level variables $\boldsymbol{x}$. Even if $A(\boldsymbol{x})$ has full rank, an iterative re-assignment of entries of $\boldsymbol{y}$ to dependent and independent variables can be necessary. The following example shows that there are indeed such cases where the approach from [17] is not applicable, but the introduced projection is.

Example 5 Consider the lower-level problem

$$\begin{aligned}
\max_{\boldsymbol{y}\in[-5,5]^2} \quad & g(x, \boldsymbol{y}) \\
\text{s.t.} \quad & h^{eq}(x, \boldsymbol{y}) = \max\{0, x+1\}y_1 + \max\{0, 1-x\}y_2 - 1 = 0
\end{aligned} \tag{LLP}$$

Where coupling equality constraints might also be written as

$$h^{eq}(x, \boldsymbol{y}) = \begin{cases} (1-x)y_2 - 1 & \text{for } x < -1 \\ (1+x)y_1 - 1 & \text{for } x > 1 \\ (1+x)y_1 + (1-x)y_2 & \text{else} \ . \end{cases}$$

The resulting matrix

$$\boldsymbol{A}(\boldsymbol{x}) = [\max\{0, x+1\}, \max\{0, 1-x\}]$$

has full rank for any $\boldsymbol{x}$. Since for $x \geqslant 1$ it has the form

$$\boldsymbol{A}(\boldsymbol{x}) = [x+1, 0]$$

and for $x \leqslant -1$ the form

$$\boldsymbol{A}(\boldsymbol{x}) = [0, 1-x]\,,$$

no constant partition of lower-level variables will be valid.

As mentioned in Section 4.1, we need Assumption 3 (which is implied by full rank of the matrix $\boldsymbol{A}$) for the projection to fulfill property **P.3**. We want to motivate this assumption for linear coupling equality constraints: Assume that the functions $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{b}(\boldsymbol{x})$ are locally Lipschitz continuous on $\mathcal{X}$. Consider any iterate $\hat{\boldsymbol{x}}$, where we have obtained a solution to the lower level. Since the lower-level problem was not infeasible for that iterate $\hat{\boldsymbol{x}}$, we can in general assume that $\boldsymbol{A}(\hat{\boldsymbol{x}})$ had full rank (otherwise $\boldsymbol{b}(\hat{\boldsymbol{x}})$ is chosen very precisely, this is unlikely enough to declare this a degenerate case). Then, the full rank assumption will hold locally (due to the lower semi-continuity of the rank).

4.2 Specialization with error bound from an extension of the Kantorovich theorem

Assume that we can obtain a so-called error bound, i.e., we find c_1 and c_2 so that we can guarantee: if $\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})\| \leqslant c_1$ then there exists a solution $\boldsymbol{y}^*$ around $\boldsymbol{y}^i$ with $\|\boldsymbol{y}^i - \boldsymbol{y}^*\| \leqslant \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})\| c_2$. Previously, we used the relation

$$\|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant L_x \|\boldsymbol{x} - \boldsymbol{x}^d\|.$$

However, we can also use

$$\|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant c_2 \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|.$$

The new relation is easily integrated into the lower-bounding problem problem (LBP-eq) as

$$\mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d) = \left\{\boldsymbol{w} \in \mathcal{Y} \mid \|\boldsymbol{y}^d - \boldsymbol{w}\| \leqslant c_2 \|\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^d)\|\right\}$$

and

$$\mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d) = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\| \leqslant c_1\right\}\,.$$

It may be easier to obtain the parameters c_1 and c_2 than L, assuming that the coupling equality constraints $\boldsymbol{h}^{eq}$ are continuously differentiable and that their Jacobian has full rank at $(\boldsymbol{x}^d, \boldsymbol{y}^d)$. We discuss how an existence theorem for solutions to nonlinear equations can be used in the described manner.

4.2.1 Obtaining the parameters

For square systems, a possible way to generate such error bounds and thus obtain the parameters mentioned in the last section is by the Kantorovich theorem (see, e.g., [33, Theorem 2.1]). In [34], the authors used a similar idea in the context of robust robotics. An extension to the Kantorovich theorem to the underdetermined case is given in [35] but involves pseudoinverses. The similar result given in [36, 37] is used in the following because it is easier to implement within a formulation suitable for deterministic global optimization.

Theorem 2 *[36] Define* $\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^m$ *as* $\boldsymbol{f}: \boldsymbol{y} \mapsto \boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})$ *for fixed, but arbitrary* $\boldsymbol{x}$*. Assume* $\boldsymbol{f}(\boldsymbol{y})$ *is differentiable in the ball* $\mathcal{B}_\rho(\boldsymbol{y_0})$ *and* $\boldsymbol{Df}(\boldsymbol{y})$ *satisfies the following Lipschitz condition in* $\mathcal{B}_\rho(\boldsymbol{y_0})$*:*

$$\|\boldsymbol{Df}(\boldsymbol{y_1}) - \boldsymbol{Df}(\boldsymbol{y_2})\| \leq K\,\|\boldsymbol{y_1} - \boldsymbol{y_2}\|\,, \quad \forall \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{B}_\rho(\boldsymbol{y}_0).$$

Let

$$\left\|\boldsymbol{Df}(\boldsymbol{y_0})^T \boldsymbol{\zeta}\right\|_* \geq \mu_0 \|\boldsymbol{\zeta}\|_*, \quad \forall \boldsymbol{\zeta} \in \mathbb{R}^m \tag{20}$$

hold for some $\mu_0 > 0$ *where* $\|\boldsymbol{c}\|_* := \sup_{\boldsymbol{b}:\|\boldsymbol{b}\|=1} \boldsymbol{c}^T\boldsymbol{b}$ *is the dual norm. Denote* $r^* = \min\{\rho, \frac{\mu_0}{2K}\}$ *and* $\mu^* = \mu_0 - Kr^*$*. If*

$$\|\boldsymbol{f}(\boldsymbol{y_0})\| < \mu^* r^* \tag{21}$$

then there exists a solution $\boldsymbol{y}^*$ *with* $\boldsymbol{f}(\boldsymbol{y^*}) = \boldsymbol{0}$*, and* $\|\boldsymbol{y^*} - \boldsymbol{y_0}\| \leq \frac{\|\boldsymbol{f}(\boldsymbol{y_0})\|}{\mu^*}$*.*

Proof The theorem is a restatement of [36, Theorem 2] using the estimates μ^* and r^* in [36, Collary 3]. □

Note how $\mu^* r^*$ and $\frac{1}{\mu^*}$ play the role of c_1 and c_2, respectively. For their computation, we need three quantities: the Lipschitz constant K, μ_0 (which in the Euclidean norm corresponds to the smallest singular value of $\boldsymbol{D}_2\boldsymbol{h}^{eq}$) and the minimal radius ρ. The radius ρ can easily be calculated as the distance of $\boldsymbol{y}^d$ to the boundary of $\bar{\mathcal{Y}}$ (which is at least τ according to assumption Assumption 5). Assuming the coupling equality constraints are twice continuously differentiable in their second argument, K can be estimated [38] by any lower bound to the optimization problem

$$\min_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{y}\in\bar{\mathcal{Y}}} -\|\boldsymbol{D}_{2,2}\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})\|_V \tag{22}$$

where $\|\cdot\|_V$ is any norm compatible with the operator norm $\|\cdot\|$ which is applied to matrices in Theorem 2.

For μ_0, we suggest two alternatives: Firstly, we can compute a value that is valid for all possible pairs of upper- and resulting feasible lower-level variables with

$$\begin{aligned} \underline{\mu_0} = \min_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x}), \boldsymbol{s}\in\mathbb{R}^{n_{heq}}} \quad & \|\boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})\|_* \\ \text{s.t.} \quad & \|\boldsymbol{s}\|_* = 1\ . \end{aligned} \tag{23}$$

Secondly, we can compute μ_0 as a function of the discretization point $\boldsymbol{y}^d$ and the upper-level variables $\boldsymbol{x}$ by embedding

$$\mu_0^d(\boldsymbol{x}) = \min_{\mu_0 \geqslant 0, \boldsymbol{s}\in\mathbb{R}^{n_{heq}}} \quad \mu_0 \tag{24a}$$

$$\text{s.t.} \quad \|\boldsymbol{s}\|_* = 1 \tag{24b}$$

$$\|\boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\boldsymbol{s}\|_* = \mu_0 \tag{24c}$$

into the lower-bounding optimization problem. Fortunately, we can avoid forming a bilevel optimization problem. This is because we will use $\mu_0(\boldsymbol{x})^d$ in the constraint

$$\|\boldsymbol{w}^d - \boldsymbol{y}^d\| \leqslant \frac{\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|}{\mu^*(\mu_0(\boldsymbol{x})^d)} \vee \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\| \geqslant \mu^*(\mu_0(\boldsymbol{x})^d) r^*(\mu_0(\boldsymbol{x})^d) \tag{25}$$

which is less restrictive for smaller values of μ_0. We can observe that both μ^* and r^* increase as a function of μ_0 and can thus embed problem (24b) and problem (24c) instead of problem (24). This is formalized in Lemma 4.

The following proposition states that problem (25) can be used in Algorithm 1 according to Theorem 1.

Proposition 1 *Denote with $\tilde{\mu}_0^d(\boldsymbol{x})$ a valid value of μ_0 in Theorem 2 for the upper-level point $\boldsymbol{x}$ and the discretization point $\boldsymbol{y}^d$, obtained either as $\tilde{\mu}_0^d(\boldsymbol{x}) := \mu_0^d(\boldsymbol{x})$ according to problem (24) or as $\tilde{\mu}_0^d(\boldsymbol{x}) := \underline{\mu}_0$ according to problem (23). Assume that the full rank assumption holds for each infeasible upper-level point $\boldsymbol{x}^d \in \mathcal{X}$ and corresponding feasible lower-level point $\boldsymbol{y}^d \in \mathcal{Y}(\boldsymbol{x})$. Further, assume that $\boldsymbol{h}^{eq}$ is Lipschitz continuously differentiable in its second argument. Then the lower bounding procedure in Algorithm 1 finitely proves the problem to be infeasible, returns a global solution or each accumulation point of the sequence of iterates $\{\boldsymbol{x}^i\}_{i\in\mathbb{N}}$ converges to a global solution of problem (GSIP-eq) if*

$$\mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d) := \|\boldsymbol{w}^d - \boldsymbol{y}^d\| \leqslant \frac{\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|}{\mu^*(\mu_0^d(\boldsymbol{x}))}$$

with

$$\mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d) := \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\| \leqslant \mu^*(\tilde{\mu}_0^d(\boldsymbol{x}))r^*(\tilde{\mu}_0^d(\boldsymbol{x}))\right\}$$

is used in Algorithm 1.

Proof We show that $\mathcal{T}$ fulfills properties **P.1** through **P.3** and thus Algorithm 1 converges according to Theorem 1. Note that K is bounded by assumption, and further that assuming full rank of the Jacobian $\boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y})$ for $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})$ implies $\tilde{\mu}_0^d(\boldsymbol{x}) \geqslant \underline{\mu}_0 > 0$. From $\mu^*(\mu_0) \geqslant \frac{\mu_0}{2}$ and $\rho \geqslant \tau > 0$ one can in turn derive that there exist with $\underline{\mu^*} > 0$ and $\underline{r^*} > 0$ with $\mu^*(\tilde{\mu}_0^d(\boldsymbol{x})) \geqslant \underline{\mu^*}$ and $r^*(\tilde{\mu}_0^d(\boldsymbol{x})) \geqslant \underline{r^*}$.

Concerning $\mathcal{T}$, property **P.1** holds, since as discussed above $\mu^*(\tilde{\mu}_0^d(\boldsymbol{x})) \geqslant \underline{\mu^*} > 0$ and $\boldsymbol{h}^{eq}(\boldsymbol{x}^d, \boldsymbol{y}^d) = \boldsymbol{0}$. Property **P.2** holds according to Theorem 2. Finally, property **P.3** follows from uniform continuity of $\boldsymbol{h}^{eq}$ on $\mathcal{X} \times \bar{\mathcal{Y}}$ as follows: Property **P.3** requires the existence of $\delta_T > 0$ such that $\|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_T$ and $\boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)$ imply $\|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \delta_w$ for arbitrary small $\delta_w > 0$. Uniform continuity of $\boldsymbol{h}^{eq}$ on $\mathcal{X} \times \bar{\mathcal{Y}}$ guarantees the existence of $\delta_T > 0$ with

$$\left[\|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_T \implies \|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d) - \cancelto{0}{\boldsymbol{h}^{eq}(\boldsymbol{x}^d, \boldsymbol{y}^d)}\| \leqslant \delta_h\right], \forall \boldsymbol{x}^d \in \mathcal{X}.$$

For $\boldsymbol{w} \in \mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d)$ and $\|\boldsymbol{x} - \boldsymbol{x}^d\| \leqslant \delta_T$, we note that

$$\|\boldsymbol{w} - \boldsymbol{y}^d\| \leqslant \frac{\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|}{\mu^*(\mu_0^d(\boldsymbol{x}))} \leqslant \frac{\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|}{\underline{\mu^*}} \leqslant \frac{\delta_h}{\underline{\mu^*}}.$$

Thus, property **P.3** follows from choosing $\delta_w = \frac{\delta_h}{\underline{\mu^*}}$. The condition $\mathcal{U}_{\mathcal{T}}(\boldsymbol{x}^d, \boldsymbol{y}^d) \supseteq \mathcal{B}_{\delta_{\mathcal{T}}}(\boldsymbol{x}^d)$ for some constant $\delta_{\mathcal{T}} > 0$ follows analogously from $\boldsymbol{h}^{eq}(\boldsymbol{x}^d, \boldsymbol{y}^d) = \boldsymbol{0}$ and uniform continuity of $\boldsymbol{h}^{eq}$ on $\mathcal{X} \times \bar{\mathcal{Y}}$. □

Theorem 2 directly handles underdetermined systems of equations. Many theoretical results only cover the square case. In the next section, we briefly discuss a possible approach for using these results instead of Theorem 2, as well as disadvantages compared to using Theorem 2.

4.2.2 Avoiding an underdetermined equation system

Instead of using Theorem 2, we can extend the system of equations $\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$ to apply theoretical results for square systems: Taking inspiration from the linear case, we form the extended system

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\lambda}) := \begin{bmatrix} \boldsymbol{I}(\boldsymbol{w} - \boldsymbol{y}^d) + \boldsymbol{A}(\boldsymbol{x})^T \boldsymbol{\lambda} \\ \boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{w}) \end{bmatrix} = \boldsymbol{0} \tag{26}$$

where $\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)^T$. This is identical to the KKT system of the Euclidean projection to the discretization point $\boldsymbol{y}^d$ for fixed $\boldsymbol{x}$, except that the Jacobian of the coupling equality constraints $\boldsymbol{D}_2\boldsymbol{h}^{eq}$ is evaluated at the discretization point $\boldsymbol{y}^d$ instead of $\boldsymbol{w}$. A related idea is used in [39], where the Jacobian is used to find a coordinate transformation to enable the use of a theorem for square systems. For brevity, denote the extended lower-level variable vector with $\boldsymbol{z}^T := [\boldsymbol{w}^T, \boldsymbol{\lambda}^T]$, the extended discretization point with $\boldsymbol{z_0}^T := [\boldsymbol{y}^{d,T}, \boldsymbol{0}^T]$ and $\tilde{\boldsymbol{F}}(\boldsymbol{x}, \boldsymbol{z}) := \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\lambda})$. Computing the Jacobian with respect to the extended lower-level variables at $\boldsymbol{z_0}$ we get:

$$\boldsymbol{D}_2\tilde{\boldsymbol{F}}(\boldsymbol{x}, \boldsymbol{z_0}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)^T \\ \boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d) & \boldsymbol{0} \end{bmatrix}. \tag{27}$$

If we assume full rank of $\boldsymbol{D}_2\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)$, this matrix has also full rank for all $\boldsymbol{x} \in \mathcal{X}$. The extension of the implicit-function theorem in [40, Theorem 3.1] validates Assumption 3 with explicit expressions for $L_{\hat{x}}$ and $\delta_{\hat{x}}$ and can theoretically be used to compute the constants from Assumption 3.

Alternatively, results for quantitative results on the inverse and implicit function theorems in [41], results from homotopy continuation [25, Prop. 6.2] or variants of the Newton-Kantorovich theorem as in [33, Theorem 2.1] can be applied to the extended system. However, an implementation in the Algorithm 1 faces the following difficulties: Firstly, one has to contend with additional variables to represent $\boldsymbol{\lambda}$. Secondly, we expect increased conservatism in these results, making a smaller neighborhood of $\boldsymbol{x}^d$ infeasible in the lower-bounding problem. This expectation is based on estimations made during the derivation of such results, which are often based on matrix norms and use a single radius for the function inputs $\boldsymbol{z}$ while the scaling of $\boldsymbol{\lambda}$ and $\boldsymbol{w}$ could differ significantly.

Instead of using the extended system, one could also eliminate $\boldsymbol{w}$ and consider the function $\hat{F}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d + \boldsymbol{A}(\boldsymbol{x})^T\boldsymbol{\lambda})$. However, in this case, computing ρ such that $\boldsymbol{y}^d + \boldsymbol{A}(\boldsymbol{x})^T\boldsymbol{\lambda}$ remains in $\bar{\mathcal{Y}}$ is difficult without again introducing large conservatism. We believe that further research could overcome these challenges and use the extended system to utilize results for the more intensively studied square case, but we do not pursue these ideas further in this manuscript.

5 Numerical implementation and examples

In this section, we discuss challenges in the numerical implementation and present numerical experiments on small-scale problems for the nonlinear case and an application to robust optimal power flow, for which we apply the approach presented in Section 4.1 for the case of linear coupling equations.

5.1 Implementation

There are several challenges in implementing Algorithm 1 with the different specializations. We highlight the most important considerations:

Firstly, we formulated many subproblems that contain logical disjunctions. Especially with nonlinear terms, most global solvers cannot handle such disjunctions directly. In our experiments, we formulate them using the big-M reformulation, i.e., using binary variables (see [42] for details). Other reformulations or methods that directly handle the disjunctions could be used [43]. However, in the tests conducted in [44], for nonconvex problems, employing a big-M formulation alongside a general MINLP solver proved to be competitive with more sophisticated methods. Similarly, we use binary variables $\boldsymbol{v} \in \{0, 1\}^{n_{ineq}+1}$ to formulate

$$G(\boldsymbol{x}, \boldsymbol{y}) \leqslant 0$$

as

$$\begin{aligned}(1 - v_{n_{ineq}+1}) g(\boldsymbol{x}, \boldsymbol{y}) - \sum_{i=1}^{n_{ineq}} v_i h_i^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &\leqslant 0 \\ \sum_{i=1}^{n_{ineq}+1} v_i &= 1\end{aligned}$$

except in Section 5.3, where we use a big-M formulation to avoid introducing nonconvex bilinear terms.

Secondly, it was necessary to tighten the feasibility and integrality tolerances of the subsolvers to avoid cycling in later iterations. One reason for needing tight feasibility tolerances is also present in other discretization-based algorithms: in the limit, the constraint violation $\max_{\boldsymbol{y} \in Y(\boldsymbol{x})} G(\boldsymbol{x}, \boldsymbol{y})$ tends to zero as the iterates for $\boldsymbol{x}$ approach the solution. Additionally, certain constraints used in the discussed formulations tend to be sensitive to tolerances. For example, using the Euclidean norm in problem (25), we have a constraint of the form $\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|_2 \geqslant \tilde{c}$ where $\tilde{c} > 0$ is usually small. For this reason, we do not use the reformulation to $\|\boldsymbol{h}^{eq}(\boldsymbol{x}, \boldsymbol{y}^d)\|_2^2 \geqslant \tilde{c}^2$, which is usually preferred because it avoids the use of a square root, because it would necessitate even tighter tolerances.

5.2 Comparing approaches for nonlinear coupling equality constraints

Algorithm 1 employs problem (LBP-eq), which is parametric in $\mathcal{T}$ and $\mathcal{U}_{\mathcal{T}}$. In this section, we compare three approaches suitable for nonlinear coupling equality constraints resulting from different choices for $\mathcal{T}$. Two use the parametrization introduced in Section 4.2, while the other assumes knowledge of the Lipschitz constant and was introduced in Section 3. In the following, we differentiate between:

- Lipschitz-based: using $\mathcal{T}(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{y}^d) = \mathcal{B}_{L\|\boldsymbol{x}-\boldsymbol{x}^d\|}(\boldsymbol{y}^d)$ with known Lipschitz constant L;
- Polyak-fixed: the approach from Section 4.2 with a fixed μ_0 computed via problem (23);
- Polyak-variable: the approach from Section 4.2 with embedding problem (24) for the computation of μ_0.

We outline the details choices in the concrete testing of the proposed approaches for nonlinear coupling equality constraints:

We use the Euclidean norm throughout. For the latter two approaches, we calculate K by solving the optimization problem problem (22) with a relative optimality tolerance of 0.01 where $\|\boldsymbol{A}\|_V = \|\boldsymbol{A}\|_2 := \sqrt{\sum_{i=1}^{m} \|\boldsymbol{A}_i\|_2^2}$ for $\boldsymbol{A} \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ which is compatible with the Euclidean norm [45]. Therein, we denote with $\boldsymbol{A}_i$ the i-th $n \times n$ matrix component of $\boldsymbol{A}$. An alternative would be to use interval arithmetic and bound the 2-norm by $\|\boldsymbol{A}_i\|_2 \leqslant \|\boldsymbol{A}_i\|_F$, but the expected overestimation would likely be detrimental to the approaches relying on bounds on K.

For termination, we selected the criteria as follows: We test the algorithm as stated in Algorithm 1 for each approach, but introduce a feasibility tolerance of 10^{-4} so that the algorithm terminates with status `TOL` in iteration i when $G^i < 10^{-4}$. We also terminate with `TIME` if the time limit of 30 minutes is exceeded, with `ITER` after 500 iterations, and with `STUCK` if $\|x^i - x^{i-1}\| < 10^{-6}$ and $|LBD^i - LBD^{i-1}| \leqslant 10^{-6}$. The latter signals insufficient progress or looping, which can occur, for example, if Assumption 3 is not satisfied or because the subsolver only enforces the constraints within a given numerical tolerance.

We also test the effect of reducing the number of discretization points by adding $LBD - f(\boldsymbol{x}) \leqslant 0$ to problem (LBP-eq) and filtering the discretization point according to $\mathcal{D} \leftarrow \{i \in \mathcal{D} \mid f^i + \epsilon_{filter} \geqslant LBD\}$ with $\epsilon_{filter} > 0$. This filtering does not interfere with the convergence guarantees of Theorem 1. Intuitively, this is clear since similarly to the behavior of G in Theorem 1, the constraint $LBD - f(x^i) \geqslant \epsilon_{filter} > 0$ cuts of a neighborhood for all discretization points discarded by this filter. Still, we prove this more rigorously in Theorem 1.

For the tests in this section, we implemented the algorithm in Julia 1.10 utilizing JuMP [46] to formulate the optimization problems. We chose this approach for the prototype implementation instead of our library for discretization problems libDIPS [32], primarily because it allows us to leverage Symbolics.jl [47] for flexible higher-order symbolic differentiation and symbolic manipulation. This capability enables the automatic generation of expressions needed in problem (23) and problem (24). The subproblems are solved globally using Gurobi 11. In order to access the nonlinear capabilities of Gurobi 11, which at the time of writing are not directly available inside JuMP, we make use of the JuMP interface to GAMS: subproblems are passed from JuMP to GAMS 47, which then calls Gurobi 11 to solve the problems. An absolute optimality tolerance of 10^{-6}, a relative optimality tolerance of 10^{-5}, an integer tolerance of 10^{-9} and a feasibility tolerance of 10^{-8} and an aggressive filter tolerance ϵ_{filter} of 10^{-5} are used. The tests are run using an Intel i5-9500 CPU and 16GB of RAM.

We constructed several small test problems to compare the approaches. Details can be found in the appendix under Examples 6 to 11. For Examples 6 to 9 Assumption 3 holds. We also test the behavior of the algorithm under the failure of this assumption with Examples 10 and 11.

The results for the examples fulfilling Assumption 3 are presented in Table 1. Even for these small test problems, each approach runs into the time limit for at least one test problem when no filtering is used. This can be attributed to the fact that additional constraints and variables are added for each discretization point, which makes the subproblems challenging to solve globally.

As intended, the filtering reduces the computation times. One should note that the filtering is especially effective in the test problems used. This is because it turns out that for these problems (in the absence of numerical tolerances) frequently only the most recent discretization point is binding in the sense that discarding every discretization point except the latest would mostly yield the same solution to the subproblems. Even though the proposed filtering will not be as effective for general problems, it allows us to estimate how many iterations

Table 1 Results for the test problems fulfilling Assumption 3. `TIME` and `ITER` denote abortion due to time and iteration limit, respectively. `TOL` denotes successful termination within tolerance, while `STUCK` signals looping or insufficient progress. Note that in Example 8 the full rank assumption is violated. †: stuck, but already close to the solution

		Ex. 6	Ex. 7	Ex. 8	Ex. 9
Lipschitz-based	time	14s	1800s	10s	1s
	iter	2	9	39	8
	status	TOL	TIME	STUCK†	TOL
Lipschitz-based + filter	time	1s	2s	1s	1s
	iter	2	19	9	7
	status	TOL	TOL	TOL	TOL
Polyak-fixed	time	1800s	6s	1s	1800s
	iter	117	6	0	74
	status	TIME	TOL	STUCK	TIME
Polyak-fixed + filter	time	20s	1s	1s	26s
	iter	283	6	0	500
	status	TOL	TOL	STUCK	ITER
Polyak-variable	time	1800s	6s	1s	37s
	iter	5	5	2	60
	status	TIME	STUCK†	STUCK	TOL
Polyak-variable + filter	time	5s	1s	1s	3s
	iter	15	5	2	60
	status	TOL	TOL	STUCK	TOL

the different approaches would have needed, had we not set any time limit. Except for the Lipschitz-based approach in Example 8, this estimation seems to hold in cases where the time limit does not take effect. Examining the number of necessary iterations is interesting for two reasons: Firstly, if an approach is efficient in that only a few iterations are required, but the subproblems are hard to solve, it could be interesting to try to formulate the subproblem in a way that allows for a faster solution. Secondly, it indicates how sensitive our results are regarding the run-time limit. Most notably, the time limit is reached for Polyak-fixed for Example 6 and Example 9, but still more than double the iterations would have been needed.

The results in these two problems also illustrate the trade-off between the different approaches of computing μ_0 in Polyak-fixed and Polyak-variable. If μ_0 is fixed, the subproblems tend to be solved faster. However, in these two problems, the adaptive value $\mu_0(\boldsymbol{x})$ is small in certain regions of the upper-level domain $\mathcal{X}$. Thus, using a single value for μ_0 that is valid for the whole domain $\mathcal{X}$ makes many iterations necessary. This can also be seen in Example 9, where the computation of $\mu_0 = 0$ directly reveals that the full rank assumption is violated, which is why Polyak-fixed directly terminates.

The Lipschitz-based approaches require knowledge of the constant L. In exchange, it is the only approach capable of solving Example 8. Except for Example 7, which is linear and thus allows $K = 10^{-8}$, it is also the fastest approach.

We also test the behavior of the approaches on two examples that violate Assumption 3. The results for Examples 10 and 11 are shown in Table 2.

As mentioned, Polyak-fixed detects the violation of the full rank assumption and terminates directly. Since Assumption 3 is violated, there exists no valid value for the Lipschitz constant

Table 2 Result for the test problems not fulfilling Assumption 3. `TIME` and `ITER` denote abortion due to time and iteration limit, respectively. `TOL` denotes successful termination within tolerance, while `STUCK` signals looping or insufficient progress. †: stuck, but already close to the solution

		Ex. 10	Ex. 11
Lipschitz-based	time	1800s	6s
	iter	29	35
	status	TIME	TOL
Lipschitz-based + filter	time	4s	2s
	iter	36	35
	status	TOL	TOL
Polyak-fixed	time	1s	1s
	iter	0	0
	status	STUCK	STUCK
Polyak-fixed + filter	time	1s	1s
	iter	0	0
	status	STUCK	STUCK
Polyak-variable	time	1800s	128s
	iter	41	156
	status	TIME†	STUCK
Polyak-variable + filter	time	8s	6s
	iter	58	110
	status	TOL	STUCK

L. When using a large value of $L = 100$ heuristically with the Lipschitz-based approach, the algorithm converges to the correct solution. However, we cannot guarantee that the iterates are valid relaxations, so we cannot guarantee convergence to the global solution or any feasible point in general.

In contrast to Example 8, in Examples 10 and 11 the full rank assumption is violated only at isolated upper-level points. We conjecture that Polyak-variable will converge to a solution or one of these isolated points. Indeed, in both problems, the iterates for the Polyak-variable approach one of these isolated points but the rate of change between iterations diminishes until it falls below our numerical tolerance or the time limit is reached. For Example 10, the solution is one of these points, which is why Polyak-variable terminates close to it.

The experiments in this section considered toy problems with nonlinear coupling constraints. In the next section, we consider an application with linear coupling equality constraints.

5.3 Robustness of relaxed power flow - an application with linear coupling equality constraints

In Section 4.1, we discussed a specialization for linear coupling equality constraints. In this section, we apply this specialization to a power-flow problem, focusing on robustness in a power grid under load uncertainty. The robustness of power grids under uncertainty is an important topic with many different formulations discussed in the literature (see [48] for an overview). The considered problem exemplifies a scenario where coupling equality constraints are essential for a practical application and decomposing dependent and independent variables is not straightforward.

A power network consists of buses, connected by lines, as well as generators and loads attached to buses. Given an initial set-point for the generators, called a dispatch point, we are interested in the robustness of the grid to uncertainty in the loads, i.e., how much the loads can change from a forecast so that the power network can operate from this dispatch point if the loads change. We formulate this as a generalized semi-infinite optimization problem using the power flow formulation based on the so-called SOCP power flow, which is a relaxation to AC power flow. We give only a high-level overview of the problem, more details are provided in Section A.2.

We consider the following optimization problem: Let $\mathcal{G}$ be the set of generators, $\mathcal{N}$ be the set of buses, and $\mathcal{E}$ be the set of lines in the power grid. Consider fixed values for the generator set points $\boldsymbol{P}^g \in \mathbb{R}^{|\mathcal{G}|}$ and $\boldsymbol{Q}^g \in \mathbb{R}^{|\mathcal{G}|}$. Further, let $\boldsymbol{P}^{d,nom} \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{Q}^{d,nom} \in \mathbb{R}^{|\mathcal{N}|}$ be the nominal values for the real and reactive loads at buses, $n \in \mathcal{N}$, respectively. We search for a minimal change of the real and reactive loads $\boldsymbol{x} = [\boldsymbol{P}^d; \boldsymbol{Q}^d] \in \mathbb{R}^{2|\mathcal{N}|}$ from these nominal values such that no safe grid state $\boldsymbol{y} = [\boldsymbol{v}; \boldsymbol{c}; \boldsymbol{s}; \Delta P; \Delta Q] \in \mathbb{R}^{|\mathcal{N}|+2|\mathcal{E}|+2}$ exits, where $\boldsymbol{v}, \boldsymbol{c}$ and $\boldsymbol{s}$ represent the complex voltages of the network and where ΔP and ΔQ represent the overall deficit in real and reactive power. Concretely, the objective function is given by $f(\boldsymbol{x}) = \frac{1}{2|\mathcal{N}|}\sum_{n\in\mathcal{N}}(P_n^d - P_n^{d,nom})^2 + (Q_n^d - Q_n^{d,nom})^2$. Meanwhile, the coupling equality constraints $\boldsymbol{h}^{eq}$ are linear and have the form of an underdetermined system of linear equations

$$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{x} \tag{28}$$

(see problem (29), problem (30)) where $\boldsymbol{A} \in \mathbb{R}^{2|\mathcal{N}|\times|\mathcal{N}|+2|\mathcal{E}|+2}$ is a sparse matrix, with size and sparsity pattern depending on the grid instance. Thus, a general valid decomposition into dependent and independent lower-level variables is unknown as it depends on the concrete problem parameters. The coupling inequality constraints only depend on the lower-level variables $\boldsymbol{y}$, i.e., $\boldsymbol{h}^{ineq}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{q}(\boldsymbol{y})$, and are linear or quadratic and concave (see Eqs. (32) and (35) to (34)). The semi-infinite constraint is linear and only depends on the lower-level variables. Overall, the problem formulation leads to a GSIP with linear coupling equality constraints and nonconvex lower level. When formulating disjunctions using big-M as explained in Section 5, the discretized upper-level problem is a MIQCQP with convex continuous relaxation.

We solve the resulting GSIP using the specialization for linear coupling equality constraints by using problem (LBP-lin) as the lower-bounding problem. In addition to the lower bounding procedure detailed in Algorithm 1, we also search for an upper bound by using the approach of restricting the right-hand side [23], i.e., replacing $G(\boldsymbol{x}, \boldsymbol{y}) \leqslant 0$ with $G(\boldsymbol{x}, \boldsymbol{y}) \leqslant -\varepsilon^r$ in problem (LBP-lin), with an adaptively reduced restriction ε^r. While we have not considered upper-bounding explicitly in this work, as discussed in Section 4.1 the analysis of [17] is applicable in this case.

In contrast to the experiments in Section 5.2, where we required symbolic tools to formulate some of the subproblems, we implemented Algorithm 1 with the upper-bounding procedure using the existing solver *RRHS* in our software libDIPS [32] by specifying problem (LBP-lin) as the lower-bounding program instead of the usual problem (SIP-LBP).

All tests are run on an Intel Xeon 8468 Sapphire CPU with 4GB of RAM. We use the default parameters of initial restriction $\varepsilon^{r,0} = 0.1$ and reduction factor $\varepsilon^{red} = 10$, as well as absolute and relative optimality tolerance of 0.1 and 10^{-4}, respectively. To solve the subproblems, we use Gurobi 11.0.0 using a feasibility tolerance of 10^{-7}, a dual feasibility tolerance of 10^{-7}, an integer tolerance of 10^{-7}, and the NumericFocus and IntegralityFocus parameters set to their highest values of 3 and 1, respectively. In the upper-level problems, we use absolute and relative optimality tolerances of 10^{-4} and set the MIPFocus parameter to 3. In the lower level problem, we use absolute and relative optimality tolerances of 10^{-5} and set the Presolve parameter to 0. The latter was necessary to avoid numerical problems caused by presolve. We scale the objective function of both the upper and lower-level problems by 10^5. While this should be equivalent to an unscaled version with correspondingly tighter optimality tolerances, our experiments showed significantly increased run times without the scaling due to the algorithm requiring many more iterations. We speculate that this may be related to the unscaled objective function values being close to 0, causing numerical issues.

We apply the algorithm to the 5-bus power network instance "case5" established as a test case by MATPOWER [49, 50]. Even though the problem formulation leads to MIQCQPs with convex continuous relaxation, only small instances could be reliably solved within a reasonable time. We computed two different dispatch points $\boldsymbol{P}^g$ and $\boldsymbol{Q}^g$ that are feasible for 13 samples for the loads $\boldsymbol{P}^d$ and $\boldsymbol{Q}^d$. For the first one (dispatch point 1), we sample an uncertainty of 5% around the nominal load and 10% uncertainty for the second one (dispatch point 2). Ten of the samples are random. Two samples are obtained by setting all loads to their upper and lower bounds, respectively. The last sample is the nominal load, as given in the case file. We solve the GSIP resulting from two distinct dispatch points for different maximum uncertainty levels, given as percentages of the nominal load.

Among multiple runs, we observed small deviations (up to 5 seconds) for some instances, even though the iterations are identical. We therefore solve each instance five times to obtain more representative CPU times. The resulting iteration counts and average CPU times are shown in Table 3. We observe that instances involving dispatch point 1 generally required higher computational effort to solve than those involving dispatch point 2. Overall, for non-trivial instances, dispatch point 1 exhibits stronger variability in computational effort. In particular, the closer to the sampling design of 5% an instance is, the more effort is required to solve it. We observe a correlation between CPU times and iteration counts, though the differences between instances are more pronounced regarding CPU times. This indicates the expected result that subproblems in later iterations become more challenging to solve, due to their increased problem size.

For both dispatch points, the instances corresponding to uncertainty ranges inside the sampled uncertainty ranges used to generate the dispatch points terminated with infeasibility, indicating that these dispatch points are indeed robust (under the relaxed power flow model) against the load variations they were designed for. In all other instances, the returned objective values indicate that loads with greater variation can cause grid insecurities, even under the relaxed power flow model. The upper and lower bounds of all instances are shown in Table 3. As expected, dispatch point 2 can withstand greater load variations than dispatch point 1 without leaving a safe grid state. We observe that increasing the uncertainty set decreases the objective value, i.e., as the search space is expanded, points closer to the nominal demand (in the 2-norm) are added to the feasible region. This occurs until the uncertainty set is larger than the dispatch point was designed for by around 1% of the nominal load. Beyond this, increasing the uncertainty set further does not decrease the objective value.

We conclude that the proposed algorithm can be utilized to check the robustness of a dispatch point. A feasible point provides a load scenario that leads to a constraint violation in the grid. On the other hand, because the used model equations represent a relaxation compared to the underlying physical constraints, the infeasibility of the problem is a necessary condition for the robustness of the physical system. The results show that for this small instance, the heuristic approach of choosing the dispatch point based on relatively few samples leads to

Table 3 Upper and lower bounds on the objective value for all studied instances. Uncertainty range is the maximal deviation from the nominal load in % of nominal load. Values for the objective are unscaled

Dispatch Point	Uncertainty Range	LBD	UBD	Iter	Time
1	5.0%	*INF*	*INF*	27	51s
	5.2%	84.1MW^2	84.2MW^2	26	45s
	6.0%	83.3MW^2	83.3MW^2	22	30s
	10.0%	83.3MW^2	83.3MW^2	22	30s
2	5.0%	*INF*	*INF*	5	1s
	10.0%	*INF*	*INF*	21	31s
	10.2%	338.2MW^2	338.2MW^2	22	31s
	11.0%	333.4MW^2	333.4MW^2	22	32s
	15.0%	333.3MW^2	333.4MW^2	22	33s

a dispatch point that passes this additional check for the amount of uncertainty considered during sampling. It would be interesting to investigate whether this holds for larger grid instances, or if this would require many more samples due to the increased dimensionality of the uncertainty space. Since the presented algorithm does not require a decomposition into dependent and independent lower-level variables, it can, in theory, be applied to other grid instances by changing the instance data without having to derive such a decomposition. However, while the presented instance is solved robustly within a reasonable time, it is already challenging numerically, requiring tightened tolerances. Preliminary tests on larger instances revealed that in addition to the effort to solve the non-convex lower level, the MIQCQP in the upper level cannot be solved in a reasonable time in later iterations. However, the problem formulation can potentially be changed to improve performance, for example, by considering power limits only on a critical subset of lines.

6 Conclusion

We extended the applicability of discretization-based methods for generalized semi-infinite programming with coupling equality constraints. Our approach only requires local stability of the lower-levels solution characterized by the existence of a solution and Lipschitz behavior of the solution set instead of existence and uniqueness. It can be applied to underdetermined systems of coupling equality constraints without declaring appropriate entries of the lower-level variable $\boldsymbol{y}$ as dependent variables. We also show that a constant decomposition into dependent and independent variables can be impossible.

For most cases, we only discuss lower-bounding. Additionally, in the linear case one can transfer the results from [17]. Further, we conjecture that one can directly adapt existing strategies for generating upper bounds in adaptive discretization algorithms, e.g., [20], but this remains to be formally shown. Similarly, we only discussed continuous variables in the upper and lower levels, but handling discrete variables is well-established for adaptive discretization algorithms, and we conjecture that it does not cause any complications for the proposed approach.

We do not assume the uniqueness of the solution to the coupling equality constraints. This allows us to solve problems that do not satisfy this condition. However, the resulting subproblems are challenging. Despite this, we demonstrate that the algorithms can be implemented in practice and discuss implementation issues and details.

The challenging nature of the employed subproblems shows in the computation times in the numerical experiments. In the case of linear coupling equality constraints, we succeeded in applying the specialization of the algorithm to linear coupling equality constraints to a power-flow problem, determining robustness in a power grid under load uncertainty. While the considered problem formulation is practically relevant, the considered instance size is orders of magnitude smaller than that of realistic problem instances. With current solvers, applying the algorithm to these instances seems intractable. In the case of nonlinear coupling constraints, even for the small test problems used, none of the tested approaches can solve all within 30min when no filtering is used. Especially with more discretization points, the subproblems take a long time to solve. In other words, we are quite heavily limited by the current status of global deterministic subsolvers used to solve the subproblems, both in the nonlinear examples and for the application with linear coupling constraints. We see four possible avenues beyond waiting for subsolver or hardware improvements.

Firstly, other existence proofs could be used to construct $\mathcal{T}$ with the required properties and lead to a numerically more favorable formulation of the subproblems or be stronger in the sense that fewer iterations are required. In the square case, the existence theorem of Borsuk is quite strong and might be able to be implemented in the existing framework [33]. Similarly, an adaptation of [25, Proposition 6.2.] to the nonsquare case would give a way to calculate L numerically. Further, we see potential in existence theorems that give componentwise bounds instead of using a single vector norm [51]. While existence theorems for the nonsquare case are less deeply studied and using existence theorems for the square case comes with drawbacks, as discussed, further research could compare different choices and combinations of existence theorems.

Secondly, we see a promise in investigating the required number of discretization points, for example, by dropping some discretization points in a more sophisticated extension of our simple filtering approach. This can benefit discretization algorithms in general, but particularly in our current context, since each discretization point adds variables both for the lower-level variables and to formulate the logical disjunctions. Another way to avoid many discretization points would be to generate better cuts in terms of the inequality constraints. In our approach, we had to adjust the discretization points locally to handle the equality constraints. Transferring this idea of adjusting the discretization points in response to changes in upper-level variables to the coupling inequality constraints could reduce the number of iterations and thus the number of considered discretization points. A further extension of this idea could look at enforcing the equality constraints of the KKT conditions of the lower-level problem in this manner.

Thirdly, instead of reformulating the disjunctions so that they can be solved with established mixed-integer solvers, further research could investigate if solution approaches for solving the specific occurring disjunctions, similar to the work in [21], can improve efficiency.

Lastly, beyond the presented framework, one could also find a way to avoid introducing explicit copies of the lower-level variables. Perhaps the work in [19] to find boxes with a unique solution to the coupling equality constraint could be extended in this way.

A Appendix

A.1 Test problems

Here we give the details of the small test problems used. The small size allows us to compute a concrete value for L in the cases where Assumption 3 holds.

Example 6 In this example, the number of coupling equality constraints equals the number of lower-level variables, but there are two solutions for the coupling equality constraints.

$$\begin{aligned}
\mathcal{X} &= \left\{ x_1 \in [0.01, 1.2] \mid x^2 \geqslant \frac{1}{100} \right\} \\
\bar{\mathcal{Y}} &= [-1.1, 1.1] \times [-1.1, 1.1] \\
f(\boldsymbol{x}) &= -x_1 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 + y_2 - 1)^2 - \frac{1}{100} \\
h_1^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 - x_1)^2 + (y_2 - x_1)^2 - 1 \\
h_2^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= y_1^2 + y_2^2 - 1
\end{aligned} \tag{Ex-1}$$

here $L = 1$ is valid for the Euclidean norm and $\boldsymbol{x} = (1.0)$ is the analytical solution.

Example 7 This example uses the structure of the lower-level problem Example 5:

$$\begin{aligned}
\mathcal{X} &= [-2, 2] \\
\bar{\mathcal{Y}} &= [-1, 2] \times [-1, 2] \\
f(\boldsymbol{x}) &= (x_1 - \frac{1}{100})^2 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= y_1 - y_2 - 1 \\
h^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= \max\{0, 1 + x_1\} y_1 + \max\{0, 1 - x_1\} y_2 - 1 \\
h_1^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= y_1 - 1 \\
h_2^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= y_2 - 1 \\
h_3^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_1 \\
h_4^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_2
\end{aligned} \tag{Ex-2}$$

here $L = 1$ is valid for the Euclidean norm. We determined the solution graphically as $\boldsymbol{x} \approx (0.1117)$.

Example 8 This problem results from reformulating Example 5 to fulfill assumption Assumption 5:

$$
\begin{aligned}
\mathcal{X} &= [-0.75, 0.75] \\
\bar{\mathcal{Y}} &= [-1.1, 1.1] \times [-1.1, 1.1] \\
f(\boldsymbol{x}) &= -x_1 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 - x_1)(1 + y_2) + x_1 \\
h^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 - x_1)(y_2 - x_1) \\
h_1^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= y_1 - 1 \\
h_2^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= y_2 - 1 \\
h_3^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_1 - 1 \\
h_4^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_2 - 1
\end{aligned}
\tag{Ex-3}
$$

Here, $L = 1$ is optimal, and the analytical solution is $\boldsymbol{x}* = (\frac{1}{2}(1 - \sqrt{5})) \approx (-0.618)$.

Example 9 The coupling equality constraint in this example was used as a test case in [19] to find a collection of boxes with unique solutions.

$$
\begin{aligned}
\mathcal{X} &= [0.25, 20] \\
\bar{\mathcal{Y}} &= [-6, 6] \\
f(\boldsymbol{x}) &= -x_1 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 - 0.9) \\
h^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_1^3 + x_1 y_1 \\
h_1^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= y_1 - 5 \\
h_2^{ineq}(\boldsymbol{x}, \boldsymbol{y}) &= -y_1 - 5
\end{aligned}
\tag{Ex-4}
$$

It is easy to see that there are three different solutions of the coupling equality constraint with respect to the lower-level variable for each upper-level variable in the host set. Concretely, $y_1(\boldsymbol{x}) = 0 \vee y_1(\boldsymbol{x}) = \pm\sqrt{x_1}$. Here, $L = 1$ is optimal and the solution is $\boldsymbol{x}* = (0.81)$

Example 10 This example is constructed using the coupling equality constraint from Example 4:

$$
\begin{aligned}
\mathcal{X} &= [-0.2, 0.2] \\
\bar{\mathcal{Y}} &= [-2, 2] \\
f(\boldsymbol{x}) &= -x_1 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= (-y_1 + 0.5) \\
h^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 + 0.75)(y_1 + 0.6)(y_1 - 0.3) - x_1
\end{aligned}
\tag{Ex-5}
$$

In this example, no valid L exists (see Example 4). We will use this example to investigate the behavior of the algorithms if the assumptions fail. We will use $L = 100$ heuristically. The solution is approximately $\boldsymbol{x}* \approx (-0.135)$.

Example 11 This example is constructed similarly to Example 10, but with a slight modification of the equality constraints.

$$
\begin{aligned}
\mathcal{X} &= [-0.2, 0.2] \\
\bar{\mathcal{Y}} &= [-2, 2] \\
f(\boldsymbol{x}) &= -x_1 \\
g(\boldsymbol{x}, \boldsymbol{y}) &= (-y_1 + 0.5) \\
h^{eq}(\boldsymbol{x}, \boldsymbol{y}) &= (y_1 + 0.75)(y_1 + 0.6)(x_1 - 0.3)(y_1 - 0.9) - x_1
\end{aligned}
\tag{Ex-6}
$$

Here, Assumption 3 is violated. We will use $L = 100$ heuristically. The solution is roughly $\boldsymbol{x}* \approx (-0.11)$.

A.2 Details on the power-flow problem

The considered problem deals with robustness in power grids under load uncertainty. It exemplifies a practical scenario where coupling equality constraints are essential and decomposing dependent and independent variables is not straightforward. We summarize the required equations, but refer to [52, 53] for more details.

Let $\mathcal{N}, \mathcal{E}, \mathcal{G}, \mathcal{L}$ be the sets of buses, lines, generators, and loads. Further, let $n(e)$ and $m(e)$ denote originating and terminal bus for each line $e \in \mathcal{E}$ and conversely let $e^+(n)$ and $e^-(n)$ denote the in- and outgoing edges from bus $n \in \mathcal{N}$. Each bus $n \in \mathcal{N}$ has a (possibly empty) set of attached generators $g^+(n)$. The state of the grid is represented by the state variables $\boldsymbol{v}, \boldsymbol{c}$ and $\boldsymbol{s}$, where $\boldsymbol{v}$ represents the square of the node voltage $\boldsymbol{V}$. Following the usual formulation of optimal power flow, we consider that the grid state can be chosen optimistically from consistent grid states. Consistent grid states are those that fulfill the following constraints:

Given values for real and reactive loads $\boldsymbol{P}^d$ and $\boldsymbol{Q}^d$, as well as real and imaginary generator set-points $\boldsymbol{P}^g$ and $\boldsymbol{Q}^g$, the real and imaginary power balance at each bus are given by

$$
\begin{aligned}
&P_n^d + \sum_{g \in g^+(n)} P_n^g + \alpha_n^P \Delta P \\
&\quad = \hat{G}_n v_n + \sum_{e \in e^-(n)} G_e c_e + B_e s_e + \sum_{e \in e^+(n)} G_e c_e - B_e s_e, \ \forall n \in \mathcal{N}
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
&Q_n^d + \sum_{g \in g^+(n)} Q_n^g + \alpha_n^Q \Delta Q \\
&\quad = \hat{G}_n v_n + \sum_{e \in e^-(n)} G_e s_e - B_e c_e + \sum_{e \in e^+(n)} G_e s_e - B_e c_e, \ \forall n \in \mathcal{N}
\end{aligned}
\tag{30}
$$

where the injections at the generators can be adjusted proportionally from the setpoints according to the participation factors $\boldsymbol{\alpha}^P, \boldsymbol{\alpha}^Q$ with $\sum_{g \in \mathcal{G}} \alpha_g^Q = \sum_{g \in \mathcal{G}} \alpha_g^P = 1$ based on the overall deficit in real and imaginary power ΔP and ΔQ.

The state variables on each node are further interdependent. In SOCP power flow, the originally trigonometric relationship between the state variables

$$
\begin{aligned}
& c_e^2 + s_e^2 - v_{n(e)} v_{m(e)} = 0, && \forall e \in \mathcal{E} \\
\Longleftrightarrow\ & ||(c_e, s_e, \tfrac{1}{2}(v_{n(e)} - v_{m(e)}))||_2 - \frac{1}{2}(v_{n(e)} + v_{m(e)}) = 0, && \forall e \in \mathcal{E} \\
& \tan(\theta_e) = s_e / c_e, \ \forall e \in \mathcal{E}
\end{aligned}
\tag{31}
$$

is usually relaxed by changing the first equality to an inequality and discarding the second [53, 54].

Here, we instead relax it to

$$c_e^2 + s_e^2 - (V_{n(e)}^{min} V_{m(e)}^{min})^2 \geqslant 0, \qquad \forall e \in \mathcal{E} \tag{32a}$$

$$||(c_e, s_e, \frac{1}{2}(v_{n(e)} - v_{m(e)}))||_\infty - \frac{1}{2}(v_{n(e)} + v_{m(e)}) \leqslant 0, \qquad \forall e \in \mathcal{E} \tag{32b}$$

$$||(c_e, s_e, \frac{1}{2}(v_{n(e)} - v_{m(e)}))||_1 - \frac{\sqrt{3}}{2}(v_{n(e)} + v_{m(e)}) \leqslant 0, \qquad \forall e \in \mathcal{E} \tag{32c}$$

$$- s_e + \tan(-\theta_{max}) c_e \leqslant 0, \qquad \forall e \in \mathcal{E} \tag{32d}$$

$$- \tan(\theta_{max}) c_e + s_e \leqslant 0, \qquad \forall e \in \mathcal{E} \tag{32e}$$

where we take $\theta_{max} = \frac{\pi}{4}$ and V^{min} are lower-limits on the allowed voltage magnitude. Note that the approximation of $c_e^2 + s_e^2 - (V_{n(e)}^{min} V_{m(e)}^{min})^2 \leqslant 0$ in Eqs. (32b) and (32c) can be refined further using the approach in [55]. This relaxation yields convex MIQCQP for the upper-level problem. We note that an alternative approximation is used in [54], leading to a convex lower-level problem.

We further add the valid constraints

$$1 - c_e / \left(V_{n(e)}^{min} V_{m(e)}^{min} \cos(\theta_{max}) \right) \leqslant 0, \ \forall e \in \mathcal{E} \tag{33}$$

In addition to the constraints in Eqs. (29), (30) and (32), which model the physical behavior, the problem also contains engineering constraints that should not be violated for a stable operation of the grid. Here, we consider the following constraints (see [53] for these and further constraints). Note that we scaled all engineering constraints so that they are on similar scales.

The absolute value of active power through a line $e \in \mathcal{E}$ should stay below a certain maximum P_e^{max}

$$\begin{aligned} (G_e c_e + B_e s_e - G_e v_{n(e)})/P_e^{max} - 1 \leqslant 0, \ \forall e \in \mathcal{E} \\ -(G_e c_e + B_e s_e - G_e v_{n(e)})/P_e^{max} - 1 \leqslant 0, \ \forall e \in \mathcal{E} \\ (G_e c_e - B_e s_e - G_e v_{m(e)})/P_e^{max} - 1 \leqslant 0, \ \forall e \in \mathcal{E} \\ -(G_e c_e - B_e s_e - G_e v_{m(e)})/P_e^{max} - 1 \leqslant 0, \ \forall e \in \mathcal{E} \ . \end{aligned} \tag{34}$$

Each generator $g \in \mathcal{G}$ should keep their output inside a prescribed range given by $[P_g^{Gen,min}, P_g^{Gen,max}]$ and $[Q_g^{Gen,min}, Q_g^{Gen,max}]$ for the active and passive power generation, respectively. This is formulated as

$$\begin{aligned} (P_n^g + \alpha_n^P \Delta P)/P_g^{Gen,max} - 1 \leqslant 0, \ \forall g \in \mathcal{G} \\ (P_g^{Gen,min} - (P_n^g + \alpha_n^P \Delta P))/P_g^{Gen,max} \leqslant 0 \ \forall g \in \mathcal{G} \\ (Q_n^g + \alpha_n^Q \Delta Q)/Q_g^{Gen,max} - 1 \leqslant 0, \ \forall g \in \mathcal{G} \\ (Q_g^{Gen,min} - (Q_n^g + \alpha_n^Q \Delta Q))/Q_g^{Gen,max} \leqslant 0, \ \forall g \in \mathcal{G} \ . \end{aligned} \tag{35}$$

Lastly, the voltage at each bus $n \in \mathcal{B}$ should be within prescribed bounds. This yields

$$\begin{aligned} v_n/(V_n^{max})^2 - 1 \leqslant 0, \ \forall n \in \mathcal{N} \\ ((V_n^{min})^2 - v_n)/(V_n^{max})^2 \leqslant 0, \ \forall n \in \mathcal{N} \ . \end{aligned} \tag{36}$$

We consider fixed values for the nominal values for the real and imaginary loads at buses $\boldsymbol{P}^{d,nom} \in \mathbb{R}^{|\mathcal{N}|}$ and $\boldsymbol{Q}^{d,nom} \in \mathbb{R}^{|\mathcal{N}|}$ and for the generator set points $\boldsymbol{P}^g \in \mathbb{R}^{|\mathcal{G}|}$ and $\boldsymbol{Q}^g \in$

$\mathbb{R}^{|\mathcal{G}|}$. We search for a minimal change in real and imaginary loads $\boldsymbol{x} = [\boldsymbol{P}^d; \boldsymbol{Q}^d] \in \mathbb{R}^{2|\mathcal{N}|}$ from these nominal values such that no safe grid state $\boldsymbol{y} = [\boldsymbol{v}; \boldsymbol{c}; \boldsymbol{s}; \Delta P; \Delta Q] \in \mathbb{R}^{|\mathcal{N}|+2|\mathcal{E}|+2}$ exits, where $\boldsymbol{v}$, $\boldsymbol{c}$ and $\boldsymbol{s}$ represent the complex voltages of the network and where ΔP and ΔQ represent the overall deficit in real and imaginary power.

In summary, the objective of the generalized semi-infinite optimization problem is $f(\boldsymbol{x}) = \frac{1}{2|\mathcal{N}|}\sum_{n\in\mathcal{N}}(P_n^d - P_n^{d,nom})^2 + (Q_n^d - Q_n^{d,nom})^2$, the coupling equality constraints $\boldsymbol{h}^{eq}$ are linear and given by problem (29) and problem (30). The coupling inequality constraints $\boldsymbol{h}^{ineq}$ are given by Eqs. (32) and (36) to (34). An arbitrary linear inequality is designated as the semi-infinite constraint g to bring the problem into the format of problem (GSIP-eq) since the lower level is essentially a feasibility problem. This does not have any impact on the algorithm considering the symmetry between the semi-infinite constraint g and the coupling inequality constraints $\boldsymbol{h}^{ineq}$ in problem (2).

A.3 Additional proofs

Lemma 4 *Assume that $\boldsymbol{g}: \mathcal{X}\times\mathbb{R}\to\mathbb{R}^m$ is monotonic in each component with respect to its second argument. Consider the optimization problem*

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}, z^*\in\mathcal{Z}} \quad & f(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{g}(\boldsymbol{x}, f^L(z)) \leqslant \boldsymbol{0} \\ & z^* \in \underset{z\in\mathcal{Z}(\boldsymbol{x})}{argmin}\, f^L(z^*) \end{aligned} \tag{A.1}$$

with compact sets $\mathcal{X}$ and $\mathcal{Z}$.

The solution set of problem (A.1) and

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}, z\in\mathcal{Z}} \quad & f(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{g}(\boldsymbol{x}, f^L(z)) \leqslant \boldsymbol{0} \end{aligned} \tag{A.2}$$

projected onto the $\boldsymbol{x}$ component are identical.

Proof A point $\hat{\boldsymbol{x}}$ is feasible in problem (A.2) if and only if it is feasible in problem (A.1). Assume $\hat{\boldsymbol{x}}$ is feasible in problem (A.2). Thus, $\exists z \in \mathcal{Z} : \boldsymbol{g}(\hat{\boldsymbol{x}}, f^L(z)) \leqslant \boldsymbol{0}$. Since $\boldsymbol{g}$ is monotonic in the second argument, this implies $\boldsymbol{g}(\hat{\boldsymbol{x}}, \min_{z\in\mathcal{Z}} f^L(z)) \leqslant \boldsymbol{0}$, in other words, $\hat{\boldsymbol{x}}$ is feasible in problem (A.1).

Assume $\hat{\boldsymbol{x}}$ is infeasible in problem (A.2). Thus, $\nexists z \in \mathcal{Z} : \boldsymbol{g}(\hat{\boldsymbol{x}}, f^L(z)) \leqslant \boldsymbol{0}$. This implies $\boldsymbol{g}(\hat{\boldsymbol{x}}, \min_{z\in\mathcal{Z}} f^L(z)) > \boldsymbol{0}$ □

Corollary 1 *The convergence guarantees of Theorem 1 still hold when replacing $\mathcal{D}$ in problem (LBP-eq) in any iteration of Algorithm 1 with the filtered discretization set $\tilde{\mathcal{D}} := \{i \in \mathcal{D} \mid f(\boldsymbol{x}^i) + \epsilon_{filter} \leqslant LBD\}$ and adding the constraint $f(\boldsymbol{x}) \geqslant LBD$ for any valid lower bound for the optimal value f^* of problem (GSIP-eq), i.e., $LBD \leqslant f^*$.*

Proof First, consider the case where the filtering causes no discretization points to be discarded. Clearly, Theorem 1 still holds for the modified algorithm. Next, consider the case where the filtering causes discretization points to be discarded finitely often. Consider the last iteration where this happens. All subsequent iterations are equivalent to applying problem (GSIP-eq) with the additional constraints introduced by the filtered discretization set

$\tilde{\mathcal{D}}$ and the added constraint $f(\boldsymbol{x}) \geqslant LBD$. These constraints are redundant in that they do not change the feasible set compared to the original problem. Thus, convergence holds under the same conditions as in Theorem 1. Lastly, assume that discretization points are dropped for an infinite number of iterations. This leads to a contradiction since we enforce $f(\boldsymbol{x}) \geqslant LBD$ in all iterations following the filtering but only discard discretization points with $f(\boldsymbol{x}^i) + \epsilon_{filter} \leqslant LBD$. Thus, the lower bound LBD must have increased by at least ϵ_{filter} between these iterations. Since the host-set $\mathcal{X}$ is compact and f is continuous, this can only occur finitely often. □

Acknowledgements We thank Susanne Saß for helpful discussions regarding notation and Daniel Jungen for valuable feedback regarding the proofs.

Author Contributions All authors contributed to the study's conception and design. Formal analysis and investigation were conducted by Aron Zingler under the supervision of Alexander Mitsos. Adrian W. Lipow conducted the experiments in Section 5.3 and contributed to the description and analysis of this case study under the supervision of Aron Zingler. The first draft of the manuscript was written by Aron Zingler. The other authors contributed to subsequent revisions. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This project has received funding from RTE within the project "Hierarchical Optimization for Worst-Case Analysis under Uncertainty with Flexibility Analysis for Power Grids and its Extension to SOCP Formulation".

Data Availability Julia code and libDIPS problem files used in the experiment are available on reasonable request. libDIPS is available under https://git.rwth-aachen.de/avt-svt/public/libdips.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Vázquez, F.G., Rückmann, J.-J., Stein, O., Still, G.: Generalized semi-infinite programming: a tutorial. Journal of computational and applied mathematics **217**(2), 394–419 (2008)
2. Djelassi, H., Mitsos, A., Stein, O.: Recent advances in nonconvex semi-infinite programming: Applications and algorithms. EURO Journal on Computational Optimization **9**, 100006 (2021)
3. Stein, O., Still, G.: Solving semi-infinite optimization problems with interior point techniques. SIAM Journal on Control and Optimization **42**(3), 769–788 (2003)
4. Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM Journal on Optimization **18**(4), 1187–1208 (2008)
5. Steuermann, H.-P.: Adaptive algorithms for semi-infinite programming with arbitrary index sets. PhD thesis, Karlsruher Institut für Technologie (KIT) (2011). https://doi.org/10.5445/IR/1000023658
6. Marendet, A., Goldsztejn, A., Chabert, G., Jermann, C.: A standard branch-and-bound approach for nonlinear semi-infinite problems. European Journal of Operational Research **282**(2), 438–452 (2020)
7. Bhattacharjee, B., Green, W.H., Barton, P.I.: Interval methods for semi-infinite programs. Computational Optimization and applications **30**, 63–93 (2005)

8. Mitsos, A., Lemonidis, P., Lee, C.K., Barton, P.I.: Relaxation-based bounds for semi-infinite programs. SIAM Journal on Optimization **19**(1), 77–113 (2008)
9. Reemtsen, R., Görner, S.: In: Reemtsen, R., Rückmann, J.-J. (eds.) Numerical Methods for Semi-Infinite Programming: A Survey, pp. 195–275. Springer, Boston, MA (1998)
10. Stein, O., Still, G.: On generalized semi-infinite optimization and bilevel optimization. European Journal of Operational Research **142**(3), 444–462 (2002)
11. Kleniati, P.-M., Adjiman, C.S.: A generalization of the branch-and-sandwich algorithm: from continuous to mixed-integer nonlinear bilevel problems. Computers & Chemical Engineering **72**, 373–386 (2015)
12. Wiesemann, W., Tsoukalas, A., Kleniati, P.-M., Rustem, B.: Pessimistic bilevel optimization. SIAM Journal on Optimization **23**(1), 353–380 (2013)
13. Mitsos, A.: Global solution of nonlinear mixed-integer bilevel programs. Journal of Global Optimization **47**, 557–582 (2010)
14. Mitsos, A., Tsoukalas, A.: Global optimization of generalized semi-infinite programs via restriction of the right hand side. Journal of Global Optimization **61**(1), 1–17 (2015). https://doi.org/10.1007/s10898-014-0146-6
15. Tsoukalas, A., Rustem, B., Pistikopoulos, E.N.: A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. Journal of Global Optimization **44**, 235–250 (2009)
16. Tsoukalas, A., Rustem, B.: A feasible point adaptation of the Blankenship and Falk algorithm for semi-infinite programming. Optimization Letters **5**(4), 705–716 (2011). https://doi.org/10.1007/s11590-010-0236-4
17. Djelassi, H., Glass, M., Mitsos, A.: Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. Journal of Global Optimization **75**(2), 341–392 (2019). https://doi.org/10.1007/s10898-019-00764-3
18. Stuber, M.D., Barton, P.I.: Semi-infinite optimization with implicit functions. Industrial & Engineering Chemistry Research **54**(1), 307–317 (2015)
19. Stuber, M.D.: Evaluation of process systems operating envelopes. PhD thesis, Massachusetts Institute of Technology (2013)
20. Djelassi, H., Mitsos, A.: A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs. Journal of Global Optimization **68**(2), 227–253 (2017). https://doi.org/10.1007/s10898-016-0476-7
21. Kirst, P., Stein, O.: Global optimization of generalized semi-infinite programs using disjunctive programming. Journal of Global Optimization **73**, 1–25 (2019)
22. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. Journal of Optimization Theory and Applications **19**(2), 261–281 (1976). https://doi.org/10.1007/bf00934096
23. Mitsos, A.: Global optimization of semi-infinite programs via restriction of the right-hand side. Optimization 60(10-11), 1291–1308 (2011) https://doi.org/10.1080/02331934.2010.527970
24. Djelassi, H.: Discretization-based algorithms for the global solution of hierarchical programs. Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen (2020). https://doi.org/10.18154/RWTH-2020-09163
25. Oustry, A.: Global optimization of nonlinear semi-infinite programming problems: Applications in power systems and control. PhD thesis, Institut Polytechnique de Paris (2023)
26. Bard, J.F.: Linear Bilevel Programming: Discrete Variables, pp. 232–268. Springer, Boston, MA (1998)
27. Klatte, D., Kummer, B.: Optimization methods and stability of inclusions in Banach spaces. Mathematical Programming **117**, 305–330 (2009)
28. Bednarczuk, E.M., Minchenko, L.I., Rutkowski, K.E.: On Lipschitz-like continuity of a class of set-valued mappings. Optimization **69**(12), 2535–2549 (2020)
29. Still, G.: Generalized semi-infinite programming: numerical aspects. Optimization **49**(3), 223–242 (2001)
30. Spingarn, J.E., Rockafellar, R.T.: The generic nature of optimality conditions in nonlinear programming. Mathematics of Operations Research **4**(4), 425–430 (1979)
31. Izmailov, A.F., Kurennoy, A.S.: On the sensitivity of a Euclidean projection. Computational Mathematics and Mathematical Physics **54**, 407–417 (2014)
32. Jungen, D., Zingler, A., Djelassi, H., Mitsos, A.: libDIPS—Discretization-Based Semi-Infinite and Bilevel Programming Solvers. Optimization Online preprint (2023). https://optimization-online.org/?p=24914
33. Alefeld, G., Frommer, A., Heindl, G., Mayer, J.: On the existence theorems of Kantorovich, Miranda and Borsuk. Electronic Transactions on Numerical Analysis **17**, 102–111 (2004)
34. Goldsztejn, A., Caro, S., Chabert, G.: A three-step methodology for dimensional tolerance synthesis of parallel manipulators. Mechanism and Machine Theory **105**, 213–234 (2016)
35. Paardekooper, M.H.C.: An upper and a lower bound for the distance of a manifold to a nearby point. Journal of Mathematical Analysis and Applications **150**(1), 237–245 (1990)

36. Polyak, B., Tremba, A.: Solving underdetermined nonlinear equations by Newton-like method. arXiv preprint arXiv:1703.07810v1 (2017)
37. Arutyunov, A., Zhukovskiy, S.: Nonlocal generalized implicit function theorems in Hilbert spaces. Differential Equations **56**, 1525–1538 (2020)
38. Dennis, J., Jr.: On the Kantorovich hypothesis for Newton's method. SIAM Journal on Numerical Analysis **6**(3), 493–507 (1969)
39. Füllner, C., Kirst, P., Stein, O.: Convergent upper bounds in global minimization with nonlinear equality constraints. Mathematical Programming **187**, 617–651 (2021)
40. Papi, M.: On the domain of the implicit function and applications. Journal of Inequalities and Applications **2005**, 1–14 (2005)
41. Jindal, A., Chatterjee, D., Banavar, R.: Estimates of the size of the domain of the implicit function theorem: a mapping degree-based approach. Math. Control Signals Systems **36**(1), 139–175 (2024)
42. Raman, R., Grossmann, I.E.: Modelling and computational techniques for logic based integer programming. Computers & Chemical Engineering **18**(7), 563–578 (1994)
43. Ruiz, J.P., Grossmann, I.E.: Global optimization of non-convex generalized disjunctive programs: a review on reformulations and relaxation techniques. Journal of Global Optimization **67**, 43–58 (2017)
44. Chen, Q., Johnson, E.S., Bernal, D.E., Valentin, R., Kale, S., Bates, J., Siirola, J.D., Grossmann, I.E.: Pyomo. GDP: an ecosystem for logic based modeling and optimization development. Optimization and Engineering 23(1), 607–642 (2022)
45. Polyak, B.: Convexity of nonlinear image of a small ball with applications to optimization. Set-Valued Analysis **9**(1), 159–168 (2001)
46. Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., Vielma, J.P.: JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. Mathematical Programming Computation (2023) https://doi.org/10.1007/s12532-023-00239-3
47. Gowda, S., Ma, Y., Cheli, A., Gwóźzdź, M., Shah, V.B., Edelman, A., Rackauckas, C.: High-performance symbolic-numerics via multiple dispatch. ACM Commun. Comput. Algebra **55**(3), 92–96 (2022). https://doi.org/10.1145/3511528.3511535
48. Roald, L.A., Pozo, D., Papavasiliou, A., Molzahn, D.K., Kazempour, J., Conejo, A.: Power systems optimization under uncertainty: A review of methods and applications. Electric Power Systems Research **214**, 108725 (2023)
49. Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on Power Systems 26(1), 12–19 (2011) https://doi.org/10.1109/TPWRS.2010.2051168
50. Li, F., Bo, R.: Small test systems for power system economic studies. In: IEEE PES General Meeting, pp. 1–4 (2010). https://doi.org/10.1109/PES.2010.5589973
51. Yamamoto, T., Chen, X.: An existence and nonexistence theorem for solutions of nonlinear systems and its application to algebraic equations. Journal of computational and applied mathematics **30**(1), 87–97 (1990)
52. Jabr, R.A.: Radial distribution load flow using conic programming. IEEE transactions on power systems **21**(3), 1458–1459 (2006)
53. Kocuk, B., Dey, S.S., Sun, X.A.: Strong SOCP relaxations for the optimal power flow problem. Operations Research **64**(6), 1177–1196 (2016)
54. Lorca, A., Sun, X.A.: The adaptive robust multi-period alternating current optimal power flow problem. IEEE Transactions on Power Systems **33**(2), 1993–2003 (2017)
55. Ben-Tal, A., Nemirovski, A.: On Polyhedral Approximations of the Second-Order Cone. Mathematics of Operations Research **26**(2), 193–205 (2001)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.