Optimization
A Journal of Mathematical Programming and Operations Research

ISSN: 0233-1934 (Print) 1029-4945 (Online) Journal homepage: www.tandfonline.com/journals/gopt20

Derivative-enhanced lower-Bounding in adaptive discretization for the global solution of semi-infinite optimization problems

Aron Zingler & Alexander Mitsos

To cite this article: Aron Zingler & Alexander Mitsos (2026) Derivative-enhanced lower-Bounding in adaptive discretization for the global solution of semi-infinite optimization problems, Optimization, 75:10, 2667-2700, DOI: 10.1080/02331934.2025.2571756

To link to this article: https://doi.org/10.1080/02331934.2025.2571756

Published online: 25 Oct 2025.

OPTIMIZATION
2026, VOL. 75, NO. 10, 2667–2700
https://doi.org/10.1080/02331934.2025.2571756

Taylor & Francis
Taylor & Francis Group

Derivative-enhanced lower-Bounding in adaptive discretization for the global solution of semi-infinite optimization problems

Aron Zingler [a] and Alexander Mitsos [b,a,c]

[a]Process Systems Engineering (AVT.SVT), RWTH Aachen University, Aachen, Germany; [b]JARA-CSD, Aachen, Germany; [c]Institute of Climate and Energy Systems: Energy Systems Engineering (ICE-1), Forschungszentrum Jülich GmbH, Jülich, Germany

ABSTRACT

We investigate the use of derivative information in the lower-bounding of adaptive discretization algorithms for semi-infinite optimization problems with linear constraints at the lower level. We propose a new discretization-based relaxation, enhanced with a convex restriction of the lower level and utilizing the first- and bounds for the second-order derivative of the semi-infinite constraint. The resulting subproblem maintains several advantageous characteristics from the original problem. By adaptively subdividing the lower-level feasible region, the subproblem is further refined. The resulting algorithm is shown to be superlinearly convergent (either locally or with respect to the width of relevant refined boxes) under suitable assumptions when the Hessian of the semi-infinite constraint is independent of the upper-level variables. We compare our proposed relaxation against existing lower-bounding approaches based on derivative information and adaptive discretization and show that excessive numbers of discretization points can be avoided.

ARTICLE HISTORY
Received 9 January 2025
Accepted 22 September 2025

KEYWORDS
Semi-infinite optimization; convex restriction; adaptive discretization

2020 MATHEMATICS SUBJECT CLASSIFICATIONS
90C26; 90C31

1. Introduction

We aim to solve the semi-infinite program (SIP)

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

globally where $\boldsymbol{x}$ are the upper- and $\boldsymbol{y}$ are the lower-level variables, with the respective host sets $\mathcal{X}$, $\mathcal{Y}$, in general both of infinite cardinality. Further, f and g are the objective and SIP constraint, respectively. The difficulty in solving SIP is that the SIP constraint is enforced for all feasible, and thus infinitely many, values of the lower-level variables $\boldsymbol{y}$. SIPs appear in various applications such as robust optimization, design centering, and function approximation, see [1–3] and references therein.

CONTACT Alexander Mitsos amitsos@alum.mit.edu

Classical approaches to solve SIP include reduction-based and discretization-based methods, see, e.g. [2]. The former aims at finding and tracking the lower-level variable values that correspond to binding constraints while the upper-level variables change. The latter enforces the SIP constraint at discretization points, that is, at a finite subset $\mathcal{Y}^D$ of the host set $\mathcal{Y}$, creating the relaxation of problem (SIP), that is, the lower bounding problem (LBP)

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

which can be improved by adding additional points to the discretization. Instead of choosing the discretization points uniformly, adaptive discretization-based methods select discretization points iteratively. Following the approach of [4], this is achieved by refining the discretization by a lower-level point $\boldsymbol{y}$ that corresponds to the maximum violation of the SIP constraint, i.e. solves the so-called lower-level problem

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

Several authors have combined the lower bound obtained by the discretization approach with methods to generate upper bounds for problem (SIP). In [5, 6] we added iteratively reduced restrictions to the right-hand side of the discretized SIP constraint. In [7], an oracle problem for deciding whether a given objective value is obtainable is combined with a bisection approach and discretization.

Another approach for upper-bounding is based on overestimation. In [8, 9], the authors use an overestimation of the SIP constraint obtained by interval extensions. Alternatively, an overestimation is derived from a convex relaxation of the lower-level problem in [9, 10].

Because SIPs with convex lower-level problems can be reformulated to a single-level optimization problem [11–13], convexification of the lower-level problem is also used outside of methods using discretization. In [14] the lower-level problem is convexified adaptively by subdivision of the lower-level feasible set. The approach is extended from boxes to arbitrarily shaped lower-level feasible sets in [15].

In summary, several approaches exist for upper-bounding, while, except for the interval-based bounds in [16], the main approach for generating lower bounds in the presence of nonconvex functions in the lower level remains the adaptive discretization approach from [4]. This lower-bounding approach has been proposed to be enhanced by embedding necessary optimality conditions for the lower level in [10], but the results were not tested with the proposed subdivision of the lower-level set. Another enhancement was proposed in [17, Chapter 3.4] based on postoptimal sensitivity analysis of the lower level. In both proposals, derivative information regarding the SIP constraint g is required.

In this manuscript, we propose a new method for lower-bounding in adaptive discretization-based methods similarly using derivative information. The aim is to use more information than just the discretization points to reduce the number of iterations. Since each iteration, in general, requires the global solution of the two optimization problems (LBP) and (LLP), reducing the number of iterations is a promising way to reduce overall computation time. This is especially true in our extensions of discretization

methods to existence-constrained SIPs [18] and generalized SIPs with coupling equality constraints [19, 20], because there, the number of variables considered in each subproblem increases with the number of discretization points. We believe that, in principle, the discussed methods for using derivative information could also be applied to these extensions. However, even for standard SIPs treated here, later iterations frequently become more expensive as the number of constraints in the subproblems grows with the number of discretization points.

The proposed method uses an adaptive convexification, similar to [10, 14, 15], but instead of deriving a convex relaxation of the lower level, leading to an upper-bound, a convex *restriction* is constructed. Within the proposed method, the lower-bounding problem is formulated in a way that avoids introducing nonconvex constraints in many cases when the corresponding constraints in problem (LBP) are convex. In particular, the lower-bounding subproblems will be equivalent to a duality-based reformulation in the special case where problem (LLP) are convex quadratic optimization problems with linear constraints. Indeed, we focus on cases where the lower-level constraints are linear, meaning that we make the additional assumption:

Assumption 1.1: The lower-level feasible set $\mathcal{Y}$ is a compact, nonempty polytope, i.e. $\mathcal{Y} := \{\boldsymbol{y} \in \mathbb{R}^{n_y} \mid \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\}$.

We note that this assumption holds for the majority of the academic problems we collected in our library in [21]. Further, many algorithms make the stronger assumption of box constraints. The case that the lower level is infeasible, i.e. $\mathcal{Y}$ is empty, means that the SIP constraint can be ignored. The discussed algorithms all converge after the first iteration in this case. Therefore, assuming $\mathcal{Y}$ to be non-empty is not restrictive, but simplifies some of the proofs.

Since we are interested in using derivative information and will later use bounds on the second-order derivatives of g calculated using interval arithmetic, we also assume that g is differentiable on a known box:

Assumption 1.2: The SIP constraint g is twice continuously differentiable with respect to its second argument on $\mathcal{X} \times \bar{\mathcal{Y}}$ where $\bar{\mathcal{Y}} := [\boldsymbol{y}^L, \boldsymbol{y}^U]$ is a box bounding the lower-level feasible set, i.e. $\bar{\mathcal{Y}} \supseteq \mathcal{Y}$.

Throughout, we make the following standard assumption about problem (SIP):

Assumption 1.3: The host set $\mathcal{X}$ is compact and the function f is continuous on the host set $\mathcal{X}$.

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 $\mathcal{B}_r(\hat{\boldsymbol{x}}) := \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leq r\}$. Considering a vector-valued function with k vector-valued arguments $\boldsymbol{f} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_k} \to \mathbb{R}^m$, we use the notation

$\boldsymbol{D}_i\boldsymbol{f} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_k} \to \mathbb{R}^m \times \mathbb{R}^{n_i}$ for $i \in \{1, \ldots, k\}$ to denote the Jacobian of $\boldsymbol{f}$ with respect to its ith argument. We use $\boldsymbol{Df}$ instead if the function has only a single argument. For second-order derivatives, we analogously use $\boldsymbol{D}_{i,j}\boldsymbol{f} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_k} \to \mathbb{R}^m \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_j}$.

We define the width of a box $\mathcal{V} := [\boldsymbol{v}^l, \boldsymbol{v}^u] \subseteq \mathbb{R}^{n_y}$, as $\text{width}(\mathcal{V}) := \max_{i \in \{1,\ldots,n_y\}} v_i^u - v_i^l$ and the width along a certain dimension as $\text{width}_i(\mathcal{V}) := v_i^u - v_i^l$.

The rest of this manuscript is structured as follows: In Section 2, we discuss earlier work on using derivative information in lower-bounding. We introduce our approach and compare the resulting subproblems with existing proposals to utilize derivative information for lower-bounding. We then summarize our proposed algorithm and detail the necessary steps. In Section 3, we examine the convergence of the algorithm and additionally discuss results regarding the convergence rate under additional assumptions. We report numerical experiments in Section 4 before giving a conclusion in Section 5.

2. Discretization-based lower-bounding utilizing derivative information

Our work can be seen as a continuation of the efforts in [17, 22–25] to improve problem (LBP) by adding utilizing more information at discretization points. To put our work in perspective, we give a summary of these references.

In [17, Chapter 3.4], we consider the case of box constraints, i.e. $\mathcal{Y} = [\boldsymbol{y}^L, \boldsymbol{y}^U]$ with $\boldsymbol{y}^L, \boldsymbol{y}^U \in \mathbb{R}^{n_y}$. Assuming the conditions required for standard postoptimal sensitivity analysis are satisfied, we view the solution of problem (LLP) locally as a function $\boldsymbol{y}^*(\boldsymbol{x})$ with respect to the value of the upper-level variables and compute the derivative of this function at the last iterate $\boldsymbol{D}\boldsymbol{y}^*(\boldsymbol{x}^d)$. The idea is that replacing the fixed value of $\boldsymbol{y}^d$ with a linear approximation of the parametric function $\boldsymbol{y}^*$ around the corresponding $\boldsymbol{x}^d$ will lead to a better approximation of problem (SIP) by problem (LBP). This is approach is generalized in [22] by deriving the linear approximation not by sensitivity analysis but based on heuristic optimization. To ensure the lower-bounding property of the resulting problem, the result of this linearization is mapped onto $\mathcal{Y}$, which has a closed-form solution for the considered case of box constraints. More specifically, the nonconvex and nonsmooth function $\mathbf{mid}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$, returning the element-wise median of $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, is used. In summary, the following problem is solved in [17]

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g\left(\boldsymbol{x}, \mathbf{mid}(\boldsymbol{l}, \boldsymbol{y}^d + \boldsymbol{D}\boldsymbol{y}^*(\boldsymbol{x}^d)(\boldsymbol{x} - \boldsymbol{x}^d), \boldsymbol{u})\right) \leq 0, \quad \forall\, \boldsymbol{y}^d \in \mathcal{Y}^D \end{aligned} \tag{LBP-SENS-BOX}$$

for a given discretization $\mathcal{Y}^D$. In the experiments in [17, Chapter 3.4], we had mixed results using problem (LBP-SENS-BOX) instead of problem (LBP), i.e. a reduction in iterations (required subproblems) but an increase in time for a single subproblem. One explanation is the complexity added by introducing additional nonlinearities to the problem, notably by the mid function. For instance, consider a case where f is convex and g is linear in $\boldsymbol{x}$ for fixed $\boldsymbol{y}$. Then problem (LBP) would be a convex optimization problem, while the mid-function causes problem (LBP-SENS-BOX) to be nonconvex and requires handling of nonsmooth constraints or introducing discrete variables.

A potential additional drawback of using a linearization around $\boldsymbol{x}^d$ is that it focuses on strengthening the relaxation in a neighborhood of this linearization point. When the standard constraint $g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0$ only excludes a very small neighborhood around $\boldsymbol{x}^d$, this can be very beneficial. However, in early iterations, the standard addition of the constraint $g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0$ usually already succeeds in excluding a reasonably large neighborhood. On the other hand, for upper-level points farther away, the modification in problem (LBP-SENS-BOX) compared to problem (LBP) might not be useful. Indeed, it can happen that problem (LBP-SENS-BOX) is a weaker relaxation of problem (SIP) than problem (LBP). This can be overcome by combining the constraints from problem (SIP) and problem (LBP-SENS-BOX). However, similar to adding an additional discretization point every iteration, this will increase the number of potentially nonlinear constraints significantly. We expect these to add additional overhead in the solution of the subproblems. On the other hand, if the SIP constraint g has an advantageous form, such as being linear or convex for fixed lower-level variables, this is likely lost in the modified constraint in problem (LBP-SENS-BOX). As described later, our proposed approach results in a relaxation that is always at least as strict as problem (LBP) for the same discretization points without the need also to include the constraint $g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0$. At the same time, it does not add nonconvex constraints compared to problem (LBP) in several interesting cases where the constraint $g(\boldsymbol{x}, \boldsymbol{y}^d)$ would result in a linear or convex constraint.

In [23], the authors consider a similar idea to [17], but with a more general constraint set $\mathcal{Y} := \{y \in \mathbb{R}^{n_y} | \boldsymbol{v}(\boldsymbol{y}) \leq 0\}$. With additional sensitivity information about the Lagrange multipliers $\boldsymbol{\mu}$ corresponding to the constraints $\boldsymbol{v}$ of the lower-level problem, they use the following approximation to problem (SIP)

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & L\left(\boldsymbol{x}, \boldsymbol{y}^d + \boldsymbol{D}\boldsymbol{y}^*(\boldsymbol{x}^d)(\boldsymbol{x} - \boldsymbol{x}^d), (\boldsymbol{\mu}^d + \boldsymbol{D}\boldsymbol{\mu}^*(\boldsymbol{x}^d)(\boldsymbol{x} - \boldsymbol{x}^d))\right) \leq 0, \quad \forall\, \boldsymbol{y}^d \in \mathcal{Y}^D. \end{aligned} \tag{LBP-SENS}$$

where $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}) := g(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{\mu}^T \boldsymbol{v}(\boldsymbol{y})$ is the Lagrange function of the lower-level problem. This approach is extended to lower-level constraints depending on the upper-level variables in [24]. Under suitable assumptions, including the so-called Reduction Ansatz, second-order local convergence of upper-level iterates is shown when the solution of problem (SIP) is a local solution of order two, while the algorithm using problem (LBP) requires the more strict assumption of solution order of one [24] for quadratic convergence. In contrast to problem (LBP-SENS-BOX) and our proposed approach, problem (LBP-SENS) does not always constitute a valid relaxation of problem (SIP) (see [23]). As a result, only convergence to a local solution can be shown. By building valid relaxations, our approach can guarantee convergence to global solutions. In terms of local convergence order, we will prove that assuming the Hessian of the lower-level objective g is independent of $\boldsymbol{x}$ and further suitable assumptions, similar to those in [23], our approach has superlinear convergence to a local solution of order two.

A different approach to strengthen problem (LBP) can be found in [25, 26]. The authors considered problem (SIP) with f and g convex for fixed lower-level variables $\boldsymbol{y}$ and where the lower-level feasible set is a compact polytope $\mathcal{Y} := \{\boldsymbol{y} \in \mathbb{R}^{n_y} | \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\}$. Among other

specializations, they utilized the subproblem

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x},\boldsymbol{y}^d) + \max_{\boldsymbol{w}\in\mathcal{Y}} \boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)(\boldsymbol{w}-\boldsymbol{y}^d) - \frac{L}{2}\|\boldsymbol{w}-\boldsymbol{y}^d\|_2^2 \le 0, \quad \forall\, \boldsymbol{y}^d \in \mathcal{Y}^D \end{aligned} \tag{LBP-REFINE-L}$$

where L should be a Lipschitz constant for the gradient of g, but is chosen experimentally in [25] and heuristically increased in [26] when a sufficient condition for inadequacy of L is found.

In our approach, we extend this as follows: We replace the scalar L by matrices $\boldsymbol{B}^d$ connected to the Hessian $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}^d)$ and instead of choosing it experimentally or heuristically, we compute these matrices using interval enclosures. We see a similar connection between our approach to the approach of [26] as between steepest descent with fixed step-size and Quasi-Newton approaches.

2.1. Overview of the proposed approach

We propose to associate each discretization point with a box around that discretization point and refine these boxes as discretization points are added. In contrast to relaxing the lower level on these boxes, which leads to an upper bound [11, 15], we construct a convex quadratic restriction of the lower-level problem, resulting in a lower bound.

The idea can be illustrated intuitively when the lower-level variable is a scalar. Assume that bounds on the second-order derivative, i.e. $[b,c] \ni D_2 g(\boldsymbol{x},y),\ \forall y \in \bar{\mathcal{Y}}^d$ are known. In [10, 11], $-\max\{0,c\}$ is used to construct a concave overestimator using the α-BB technique [27], resulting in a convex *relaxation*. Similarly, our approach uses $\min\{b,0\}$ to construct a concave underestimator of the SIP constraint and thus a convex *restriction* of the lower level. Instead of the (edge-) concave underestimator constructed in [28], we construct a *quadratic* convex restriction of the lower-level problem. By using a restriction that is quadratic, we aim to make the resulting subproblems easier to solve. We use

$$g(\boldsymbol{x},y^d) + D_2 g(\boldsymbol{x},y^d)(y-y^d) + b'(y-y^d)^2 \le g(\boldsymbol{x},y), \quad \forall\, y \in \bar{\mathcal{Y}}^d \tag{1}$$

implied by a Taylor-expansion around y^d with $b' = \min\{0,b\}$. This is visualized in a constructed example in Figure 1.

In higher dimensions and the context of global (single-level) optimization, most research [29] using the α-BB technique usually reduces the information obtained by the bounds on the second-order derivative to a scalar α or a diagonal matrix since this information can be computed efficiently. Similar to the recent work introducing nondiagonal terms in these relaxations [30], our convex restriction aims to keep as much information as possible from the bounds on the second-order derivative. Thus, we work with a nondiagonal matrix $\boldsymbol{B}$. We deem this effort worthwhile because, in our case, the resulting information contributes to the *formulation* of a global optimization subproblem, which is likely computationally expensive to solve. In contrast, when using the information *inside* a branch-and-bound algorithm, the effort of computing this information (potentially at every node) would be a major consideration [29].

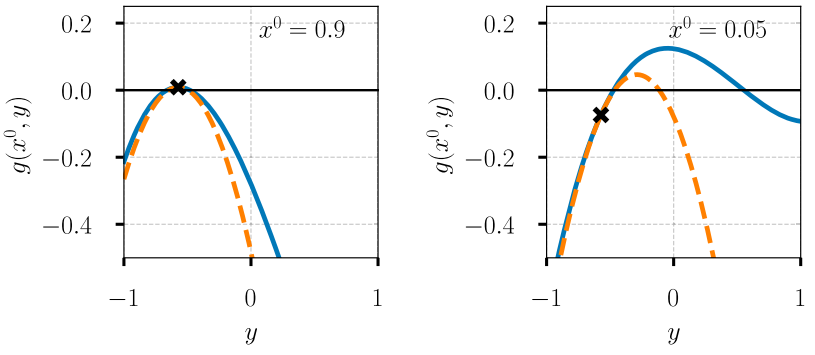

Figure 1. Illustration of the concave quadratic underestimator (dashed, orange) from Equation (1) for $g(x,y) = -\frac{1}{2}(x+y)^2 + \frac{1}{3}y^3 + \frac{1}{8}$ (solid, blue), for the discretization point $y^d \approx -0.57$, leading to a convex restriction of the lower-level. For $y \in [-1,1]$, the second derivative with respect to y is contained in the interval $[-3,1]$, leading to $b' = -3$ in Equation (1). On the left: for $x^0 = 0.9$, y^d is the maximizer of both the lower-level problem and the maximizer of the quadratic restriction. On the right: for another upper-level value, specifically $x^0 = 0.05$, the restriction underestimates the original function, but the maximum over the underestimation is larger than the function value at the discretization point $g(x^0, y^d)$.

More concretely, we propose to replace $g(\boldsymbol{x}, \boldsymbol{y}^d)$ in problem (LBP) with

$$g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) := g(\boldsymbol{x}, \boldsymbol{y}^d) + \max_{\boldsymbol{w} \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d} \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d)(\boldsymbol{w} - \boldsymbol{y}^d) - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{y}^d)^T \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d)(\boldsymbol{w} - \boldsymbol{y}^d) \quad (2)$$

where $\bar{\mathcal{Y}}^d$ is a box around $\boldsymbol{y}^d$ that is subsequently reduced in size. Here, $\boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d)$ is a matrix chosen such that

$$g(\boldsymbol{x}, \boldsymbol{y}^d) \leq g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq \max_{\boldsymbol{y} \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d} g(\boldsymbol{x}, \boldsymbol{y}) \quad (3)$$

so that the resulting optimization problem is a relaxation of problem (SIP) and is always at least as tight as problem (LBP). Note that the first inequality holds for arbitrary $\boldsymbol{B}$. To avoid additional nonconvexities, we will construct a positive semidefinite matrix $\boldsymbol{B}$ that is independent of the upper-level variables $\boldsymbol{x}$. For brevity, we will mostly suppress the dependency on $\bar{\mathcal{Y}}^d$ and just write $\boldsymbol{B}^d$.

Consider the optimization problem in Equation (2) with such a positive semidefinite matrix $\boldsymbol{B}^d$. We note that $g^\dagger$ is the optimal-value function of the following parametric quadratic programming problem:

$$\begin{aligned} \max_{\Delta \boldsymbol{y} \in R^{n_y}} \quad & g(\boldsymbol{x}, \boldsymbol{y}^d) + \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d) \Delta \boldsymbol{y} - \frac{1}{2} \Delta \boldsymbol{y}^T \boldsymbol{B}^d \Delta \boldsymbol{y} \\ \text{s.t.} \quad & \tilde{\boldsymbol{A}}^d \Delta \boldsymbol{y} \leq \tilde{\boldsymbol{b}}^d \end{aligned} \quad \text{(QP-Primal)}$$

with $\tilde{\boldsymbol{A}}^d := [\boldsymbol{A}; \boldsymbol{I}; -\boldsymbol{I}]$ and $\tilde{\boldsymbol{b}} := [\boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}^d; \boldsymbol{y}^{U,d} - \boldsymbol{y}^d; -\boldsymbol{y}^{L,d} + \boldsymbol{y}^d]$ and $\bar{\mathcal{Y}}^d = [\boldsymbol{y}^{L,d}, \boldsymbol{y}^{U,d}]$

Following the duality approach of [31], this has the same objective value as the convex quadratic minimization problem

$$\begin{aligned} \min_{\Delta \boldsymbol{y} \in R^{n_y}, \boldsymbol{\mu} \in \mathbb{R}^{n_b}} \quad & g(\boldsymbol{x}, \boldsymbol{y}^d) + \tilde{\boldsymbol{b}}^T \boldsymbol{\mu} + \frac{1}{2} \Delta \boldsymbol{y}^T \boldsymbol{B}^d \Delta \boldsymbol{y} \\ \text{s.t.} \quad & \tilde{\boldsymbol{A}}^{d,T} \boldsymbol{\mu} + \boldsymbol{B}^d \Delta \boldsymbol{y} = \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d)^T \\ & \boldsymbol{\mu} \geq \boldsymbol{0}. \end{aligned} \tag{QP-Dual}$$

Since we assume $\mathcal{Y}$ to be nonempty, problem (QP-Primal) is feasible, and thus strong duality holds [31]. Therefore, we know that

$$\nu^d_{(\text{QP}-\text{Primal})}(\boldsymbol{x}) = \nu^d_{(QP-Dual)}(\boldsymbol{x})$$

where ν^d represents the optimal value function for fixed $d \in \mathcal{D}$ with $\mathcal{D} := \{1, \ldots, |\mathcal{Y}^D|\}$. As a result, we can formulate the constraint $\boldsymbol{g}^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq 0$ as $\nu^d_{(\text{QP}-\text{Dual})}(\boldsymbol{x}) \leq 0$. Since problem (QP-Dual) is a minimization problem, we can avoid solving a bilevel optimization problem. Instead, we explicitly include the constraints and variables from problem (QP-Dual) and arrive at our formulation

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\mu}^d \in \mathbb{R}^{n_b}, \Delta \boldsymbol{y}^d \in \mathbb{R}^{n_y}, d \in \hat{\mathcal{D}}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0, \quad \forall\, d \in \mathcal{D} \setminus \hat{\mathcal{D}} \\ & g(\boldsymbol{x}, \boldsymbol{y}^d) + \hat{\boldsymbol{b}}^T \boldsymbol{\mu} + \frac{1}{2} \Delta \boldsymbol{y}^{d,T} \boldsymbol{B}^d \Delta \boldsymbol{y}^d \leq 0, \quad \forall\, d \in \hat{\mathcal{D}} \\ & \hat{\boldsymbol{A}}^{d,T} \boldsymbol{\mu}^d + \boldsymbol{B}^d \Delta \boldsymbol{y}^d = \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d)^T, \quad \forall\, d \in \hat{\mathcal{D}} \\ & \hat{\boldsymbol{A}}^d \Delta \boldsymbol{y}^d \leq \hat{\boldsymbol{b}}^d, \quad \forall\, d \in \hat{\mathcal{D}} \\ & \boldsymbol{\mu}^d \geq \boldsymbol{0}, \quad \forall\, d \in \hat{\mathcal{D}} \end{aligned} \tag{LBP-REFINE}$$

and $\boldsymbol{y}^d \in \mathcal{Y}^D$. Note that we replace the constraint $g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0$ only for a subset $\hat{\mathcal{D}} \subseteq \mathcal{D}$ of all discretization points. In contrast to problem (LBP-SENS-BOX), we introduce the additional variables $\boldsymbol{\mu}^d$ and $\Delta \boldsymbol{y}^d$ for each of these discretization points $d \in \hat{\mathcal{D}}$. However, the former affect the problem only linearly and thus do not require branching when problem (LBP-REFINE) is solved with a branch-and-bound algorithm. The latter only appear in the form of linear or convex quadratic terms. Some solvers for deterministic global optimization are able to detect and exploit this [32]. For others, using $\boldsymbol{z} = \boldsymbol{L}^{T,d} \Delta \boldsymbol{y}$ with $\boldsymbol{B}^d = \boldsymbol{L}\boldsymbol{L}^T$ and $g(\boldsymbol{x}, \boldsymbol{y}^d) + \tilde{\boldsymbol{b}}^T \boldsymbol{\mu} + \frac{1}{2} \boldsymbol{z}^T \boldsymbol{z} \leq 0$ can enable the use of specialized relaxations for $(\cdot)^2$. Nevertheless, we may decide not to apply the refined constraint for certain discretization points, i.e. for $d \in \mathcal{D} \setminus \hat{\mathcal{D}}$.

The relaxation obtained by the constructed subproblem problem (LBP-REFINE) is at least as strict as problem (LBP) with the same discretization points. Additionally, there are certain situations where this formulation is especially effective:

- If the lower-level problem is a LP and $\boldsymbol{B}^d = \boldsymbol{0}$, problem (LBP-REFINE) with $\bar{\mathcal{Y}}^d \supseteq \mathcal{Y}$ is an exact reformulation of problem (SIP).

- If the lower-level problem is a convex QP, i.e. $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{H} \preceq \boldsymbol{0}$ and $\boldsymbol{B}^d = -\boldsymbol{H}$, problem (LBP-REFINE) with $\bar{\mathcal{Y}}^d \supseteq \mathcal{Y}$ is an exact reformulation of problem (SIP).
- If the Hessian of the lower-level problem is independent of the upper-level variable, e.g. $g(\boldsymbol{x},\boldsymbol{y}) := a(\boldsymbol{y}) + \boldsymbol{b}(x)^T\boldsymbol{y} + c(\boldsymbol{x})$, then our method will converge superlinearly under assumptions similar to the assumptions taken in [23] (see upcoming Theorem 3.1).

Moreover, there are several cases where the proposed lower-bounding problem maintains advantageous characteristics from problem (LBP) and can be efficiently solved:

- If problem (LBP) is linear for fixed $\boldsymbol{y}$, e.g. $g(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{a}(\boldsymbol{y})^T\boldsymbol{x} + \boldsymbol{b}(\boldsymbol{y})$, we obtain a convex QCQP in (LBP-REFINE) because $\boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)$ is then a linear in $\boldsymbol{x}$.
- Similarly, if problem (LBP) is convex for fixed $\boldsymbol{y}$ with the form $g(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{a}(\boldsymbol{y})^T\boldsymbol{x} + \boldsymbol{b}(y) + c(\boldsymbol{x})$, we obtain a convex problem in problem (LBP-REFINE) because $\boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)$ is then linear in $\boldsymbol{x}$.

In the following, we compare the proposed subproblem with subproblems obtained in other approaches using derivative information to improve lower-bounding

2.2. Comparision with other derivative-based lower-bounding formulations

For a more direct comparison, we aim to extend the approach from [17], from box-constrained lower-level problems to the case of general linear constraints. The basic idea therein is to derive a local linear approximation of the parametric solution $\boldsymbol{y}^*(\boldsymbol{x})$ of the lower-level by using the sensitivity matrix $\boldsymbol{S}^d := -(\boldsymbol{D}_{2,2}g(\boldsymbol{x}^d,\boldsymbol{y}^d))^{-1}\boldsymbol{D}_{1,2}g(\boldsymbol{x}^d,\boldsymbol{y}^d)$ of the lower-level variables and use the constraint $g(\boldsymbol{x},\mathbf{mid}(\boldsymbol{y}^l,\boldsymbol{y}^d + \boldsymbol{S}(\boldsymbol{x}-\boldsymbol{x}^d),\boldsymbol{y}^u)) \leq 0$ instead of $g(\boldsymbol{x},\boldsymbol{y}^d) \leq 0$. In the case of bound constraints, the mid-function is used to ensure the resulting constraint corresponds to a relaxation by mapping the output of the linear function into the feasible region of the lower-level problem. In the general linear case, the equivalent mapping is harder to represent. In our generalization, we follow the proposal in [17] to always generate the constraint $g(\boldsymbol{x},\boldsymbol{y}^d) \leq 0$ and only add the improved constraints for discretization points $\boldsymbol{y}^d$ that are Slater points in problem (LLP). We propose to use $g(\boldsymbol{x},\boldsymbol{y}^d + \Delta\boldsymbol{y}) \leq 0$ where $\Delta\boldsymbol{y}$ is the solution of the QP

$$\begin{aligned} \max_{\Delta\boldsymbol{y}\in R^{n_y}} \quad & (\boldsymbol{x}-\boldsymbol{x}^d)^T\boldsymbol{D}_{1,2}g(\boldsymbol{x}^d,\boldsymbol{y}^d)\Delta\boldsymbol{y} - \frac{1}{2}\Delta\boldsymbol{y}^T(-\boldsymbol{D}_{2,2}g(\boldsymbol{x}^d,\boldsymbol{y}^d))\Delta\boldsymbol{y} \\ \text{s.t.} \quad & \tilde{\boldsymbol{A}}^d\Delta\boldsymbol{y} \leq \tilde{\boldsymbol{b}}^d. \end{aligned} \tag{Sens-QP-Primal}$$

Note that $\Delta\boldsymbol{y}^d = S^d(\boldsymbol{x}-\boldsymbol{x}^d)$ is the unconstrained maximizer of problem (Sens-QP-Primal), which means that using $g(\boldsymbol{x},\boldsymbol{y}^d + \Delta\boldsymbol{y}) \leq 0$ is equivalent to the constraint $g(\boldsymbol{x},\mathbf{mid}(\boldsymbol{y}^l,\boldsymbol{y}^d + \boldsymbol{S}(\boldsymbol{x}-\boldsymbol{x}^d),\boldsymbol{y}^u)) \leq 0$ proposed in [17] if the lower-level constraints are not active. Furthermore, note that $\boldsymbol{D}_{2,2}g(\boldsymbol{x}^d,\boldsymbol{y}^d)$ is negative semidefinite because of the assumption that the discretization point is a Slater point. There are multiple ways to embed the constraint that $\Delta\boldsymbol{y}^d$ is a maximizer of problem (Sens-QP-Primal) into the lower-bounding problem. Instead of enforcing KKT conditions, which would lead to complementarity conditions, we propose to use strong duality. Analogous to the discussion above, we can derive

the dual of this convex QP as

$$\begin{aligned} \min_{\Delta \boldsymbol{y} \in R^{n_y}, \boldsymbol{\mu} \in \mathbb{R}^{n_b}} \quad & \tilde{\boldsymbol{b}}^T \boldsymbol{\mu} + \frac{1}{2} \Delta \boldsymbol{y}^T (-\boldsymbol{D}_{2,2} g(\boldsymbol{x}^d, \boldsymbol{y}^d)) \Delta \boldsymbol{y} \\ \text{s.t.} \quad & \tilde{\boldsymbol{A}}^{d,T} \boldsymbol{\mu} - \boldsymbol{D}_{2,2} g(\boldsymbol{x}^d, \boldsymbol{y}^d) \Delta \boldsymbol{y} = \boldsymbol{D}_{2,1} g(\boldsymbol{x}^d, \boldsymbol{y}^d))(\boldsymbol{x} - \boldsymbol{x}^d) \\ & \boldsymbol{\mu} \geq \boldsymbol{0}. \end{aligned} \tag{Sens-QP-Dual}$$

We enforce strong duality by setting the objective functions of problems (Sens-QP-Primal) and (Sens-QP-Dual) equal and embedding the constraints of both problems.

The resulting problem is:

$$\begin{aligned} \min_{\substack{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\mu}^d \in \mathbb{R}^{n_b}, \\ \Delta \boldsymbol{y}^d \in \mathbb{R}^{n_y}, d \in \tilde{\mathcal{D}}}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0, \quad \forall\, d \in \mathcal{D} \\ & g(\boldsymbol{x}, \boldsymbol{y}^d + \Delta \boldsymbol{y}^d) \leq 0, \quad \forall\, d \in \tilde{\mathcal{D}} \\ & \tilde{\boldsymbol{b}}^T \boldsymbol{\mu} = (\boldsymbol{x} - \boldsymbol{x}^d)^T \boldsymbol{D}_{1,2} g(\boldsymbol{x}^d, \boldsymbol{y}^d) \Delta \boldsymbol{y} + \Delta \boldsymbol{y}^T \boldsymbol{D}_{2,2} g(\boldsymbol{x}^d, \boldsymbol{y}^d) \Delta \boldsymbol{y}, \quad \forall\, d \in \tilde{\mathcal{D}} \\ & \tilde{\boldsymbol{A}}^{d,T} \boldsymbol{\mu} - \boldsymbol{D}_{2,2} g(\boldsymbol{x}^d, \boldsymbol{y}^d) \Delta \boldsymbol{y}^d = \boldsymbol{D}_{2,1} g(\boldsymbol{x}^d, \boldsymbol{y}^d)(\boldsymbol{x} - \boldsymbol{x}^d), \quad \forall\, d \in \tilde{\mathcal{D}} \\ & \tilde{\boldsymbol{A}}^d \Delta \boldsymbol{y}^d \leq \tilde{\boldsymbol{b}}^d, \quad \forall\, d \in \tilde{\mathcal{D}} \\ & \boldsymbol{\mu}^d \geq \boldsymbol{0}, \quad \forall\, d \in \tilde{\mathcal{D}}, \end{aligned} \tag{LBP-SENS-GEN}$$

where $\tilde{\mathcal{D}} \subseteq \mathcal{D}$ is the subset of discretization points that are Slater points. Compared to problem (LBP-REFINE), there are bilinear terms containing upper-level variables $\boldsymbol{x}$ and the auxiliary variables $\Delta \boldsymbol{y}^d$. Note that the change of the lower-level variables $\Delta \boldsymbol{y}^d$ is used in the arguments in the potential nonconvex SIP constraint g. This potentially adds additional nonconvex terms but guarantees no additional error is introduced. Indeed, since we use a quadratic approxmation in problem (LBP-REFINE), $\boldsymbol{y}^d + \Delta \boldsymbol{y}^d = \boldsymbol{y}^*(\boldsymbol{x})$ does not guarantee $g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) = g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$. As a result, problem (LBP-SENS-GEN) can be expected to approximate the constraint $g(\boldsymbol{x}, \boldsymbol{y}^*(x)) \leq 0$ more accurately close to $\boldsymbol{x}^d$. However, the following constructed example shows that further away, the formulation in problem (LBP-REFINE) can be advantageous not only in terms of fewer added nonconvex terms in the formulated subproblem but also with regard to approximation of the feasible set of problem (SIP) for a given set of discretization points.

Example 2.1: Consider $f(x) = x$ and $g(x, y) = -\frac{1}{8} + (x + \frac{1}{2})^2 + (2.5(x^2 + 1)x^2 - \frac{1}{9}x^4)y + 0.5(-2.5(x^2 + 1))y^2 + \frac{1}{27}y^3$ for $y \in \mathcal{Y} = [-\frac{3}{2}, \frac{3}{2}]$ and $x \in \mathcal{X} = [-1, 1]$, i.e. a third-order polynomial in y with a single minimizer $y(x) = x^2$ in $\mathcal{Y}$ for all $x \in \mathcal{X}$. Bounds on the second-order derivative are given by $[-\frac{16}{3}, -\frac{13}{6}]$. When linearizing the function $y^*(x) = x^2$ at positive upper-level iterates $x^0 > 0$, the approximation will become inaccurate for negative upper-level points $x < 0$. The resulting constraints corresponding to the different formulations are shown in Figure 2 for two initial iterates x^0. The approximation error in the linearization causes the additional constraint in problem (LBP-SENS-GEN) to become redundant compared to the traditional constraint $g(x, y) \leq 0$. In comparison,

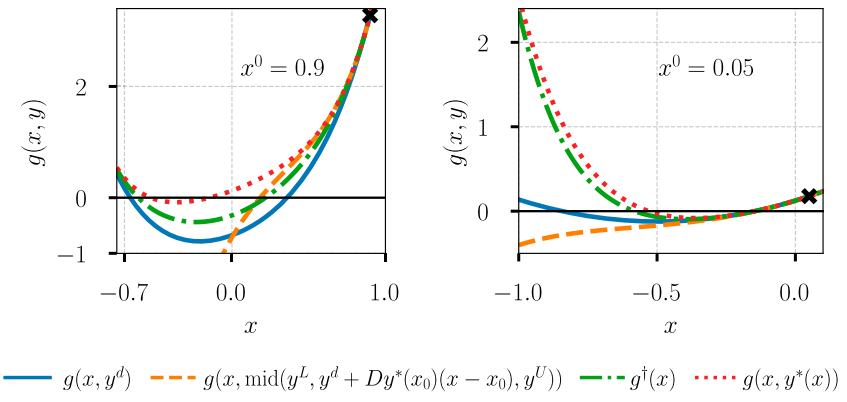

Figure 2. Comparison of the different lower-bounding approaches in Example 2.1 visualized in terms of resulting constraints in the discretized optimization problem. The square-dotted red line is based on the optimal solution of the lower-level $y^*(x)$ and coincides with the result obtained for the KKT-based approach in problem (LBP-KKT) because for this example, the lower-level problem is convex. Far away from the previous iteration x^0, the constraint added by problems (LBP-SENS-BOX) and (LBP-SENS-GEN) (orange, dashed) is significantly worse than the usual constraint added in problem (LBP) (solid blue). Meanwhile, the proposed formulation problem (LBP-REFINE) (dash-dotted green) adjusts better with large changes in the upper-level variable.

the formulation in problem (LBP-REFINE) approximates $g(x, y^*(x))$ better and always improves on the traditional constraint without including it explicitly.

We also proposed a different improvement to the lower-bounding in [10]. Here, the discretization point is not used in the improved constraint. Instead, the KKT conditions of the lower-level problem constrained to each box $\bar{\mathcal{Y}}^d$ are embedded, which results in the following lower-bounding problem:

$$
\begin{aligned}
\min_{\substack{\boldsymbol{x}\in\mathcal{X},\boldsymbol{\mu}^d\in\mathbb{R}^{n_b},\\ \boldsymbol{w}^d\in\mathbb{R}^{n_y},d\in\tilde{\mathcal{D}}}} \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0, \quad \forall\, d \in \mathcal{D} \\
& g(\boldsymbol{x}, \boldsymbol{w}^d) \leq 0, \quad \forall\, d \in \mathcal{D} \\
& \tilde{\boldsymbol{A}}^{d,T}\boldsymbol{\mu} = \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{w}^d)), \quad \forall\, d \in \mathcal{D} \\
& \tilde{\boldsymbol{A}}^d \boldsymbol{w}^d \leq \hat{\boldsymbol{b}}^d, \quad \forall\, d \in \mathcal{D} \\
& \boldsymbol{\mu}^d \geq \boldsymbol{0}, \quad \forall\, d \in \mathcal{D} \\
& \boldsymbol{\mu}^{d,T}(\tilde{\boldsymbol{A}}^d \boldsymbol{w}^d - \hat{\boldsymbol{b}}^d) = \boldsymbol{0}, \quad \forall\, d \in \mathcal{D}
\end{aligned}
\tag{LBP-KKT}
$$

with $\hat{\boldsymbol{b}}^d := [\boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}^d; \boldsymbol{y}^{U,d}; -\boldsymbol{y}^{L,d}]$. This relaxes the constraint that $\boldsymbol{w}^d$ should be a global maximizer of problem (LLP) over the box $\bar{Y}^d = [\boldsymbol{y}^{L,d}, \boldsymbol{y}^{U,d}]$ to the constraint that $\boldsymbol{w}^d$ is a stationary point. This formulation is expected to be even more challenging to solve than problem (LBP-SENS-GEN) since the Lagrange multipliers appear in a challenging nonlinear equilibrium constraint.

The additional constraints in LBP-KKT compared to problem (LBP) can become redundant for a discretization point $\boldsymbol{y}^d$ if there exists another KKT point in the box $\bar{\mathcal{Y}}^d$ that is not a maximizer. Such undesired KKT points are eventually excluded with the sufficient refinement of the respective box [10]. Alternativly, we could replace the constraint $g(\boldsymbol{x}, \boldsymbol{w}^d) \leq 0$ with $g(\boldsymbol{x}, \boldsymbol{w}^d) \leq g(\boldsymbol{x}, \boldsymbol{y}^d)$. We discuss further constraints to exclude some KKT points that are not maximizers in [10].

In the next section, we first summarize our algorithm and then give more details about the substeps.

2.3. Summary of the algorithm

The algorithm iteratively solves the lower-bounding problem (LBP-REFINE). Starting from an initial bounding box, the feasible region $\mathcal{Y}$ is covered by boxes, each containing an associated discretization point. We check the maximal violation of the SIP constraint by solving problem (LLP) and generate the next discretization point $\boldsymbol{y}^d$. We then introduce a new box containing the new discretization point, adaptively refining the set of boxes and calculate a valid matrix $\boldsymbol{B}^d$ for the refined boxes, improving the relaxation in problem (LBP-REFINE). Finally, we exclude some of the discretization points d from the set $\hat{\mathcal{D}}$ of discretization points where the constraint $g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq 0$ is used. The resulting algorithm is listed in Algorithm 1.

It remains to detail how we refine the boxes and how $\hat{D} \subseteq \mathcal{D}$ is chosen. Finally, we need to specify how the restriction is constructed by deriving a valid matrix $\boldsymbol{B}^d$. This restriction is discussed in the following, as well as a way to use the information computed for the convex restriction to construct a convex relaxation and, therefore, an upper-bounding heuristic.

2.4. Construction of the convex quadratic restriction

In Section 2.1, it was already shown how a quadratic convex restriction can be created in the scalar case from bounds on the second-order derivative of g on $\mathcal{X} \times \bar{\mathcal{Y}}^d$. Recall that in this scalar case, i.e. for bounds $D_{2,2}g(\boldsymbol{x}, y) \in [b, c],\ \forall \boldsymbol{x} \in \mathcal{X}, y \in \mathcal{Y}$, the quantities $-\max\{0, c\}$ and $\min\{b, 0\}$ were of interest for building a concave under- and overestimator, resulting in a convex restriction and relaxation of problem (LLP), respectively. The first is the minimum perturbation to make all values in the interval negative, while the second is the biggest nonpositive number that is smaller than all values in the interval.

Here, we consider a generalization to multiple dimensions that consists of finding (symmetric) bounding matrices $\boldsymbol{L}^d$ and $\boldsymbol{U}^d$ [33], meaning that $\boldsymbol{L}^d \preceq \boldsymbol{D}_{2,2}g(\boldsymbol{x}, \boldsymbol{y}) \preceq \boldsymbol{U}^d,\ \forall \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \bar{\mathcal{Y}}^d$. Here, we employ the notation $\boldsymbol{A} \preceq \boldsymbol{B} \iff \boldsymbol{0} \preceq \boldsymbol{B} - \boldsymbol{A}$, i.e. $\boldsymbol{B} - \boldsymbol{A}$ is positive semidefinite.

We use the negative semidefinite part of the lower-bounding matrix $\boldsymbol{L}^d$, i.e. $(\boldsymbol{L}^d)^-$ as a generalization of $\min\{b, 0\}$. This can be computed as $(\boldsymbol{L}^d)^- = \sum_i \min\{0, e_i\} \boldsymbol{v}_i \boldsymbol{v}_i^T$ [34], where $\boldsymbol{v}^i$ and e_i are the eigenvectors and eigenvalues of $\boldsymbol{L}^d$, respectively. The positive semidefinite part of a matrix, i.e. $(\boldsymbol{L}^d)^+$ can be defined analogously. To compute a valid lower-bounding matrix $\boldsymbol{L}^d$, we first compute the element-wise interval enclosure of the Hessian of g, i.e. matrices $\boldsymbol{H}^{L,d}$ and $\boldsymbol{H}^{U,d}$ where $\boldsymbol{D}_{2,2}g(\boldsymbol{x}, \boldsymbol{y}) \in [\boldsymbol{H}^{L,d}, \boldsymbol{H}^{U,d}],\ \forall \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \bar{\mathcal{Y}}^d$.

Algorithm 1: REFINE-B: Adaptive discretization algorithm based on convex-quadratic restriction over refined boxes

Data: bounding box $\mathcal{E}$ of $\mathcal{Y}$, finite conservatism limit $\sigma_{\max} > 0$

$\mathcal{D} \leftarrow \{\}$;
$i \leftarrow 0$;
$LBD \leftarrow -\infty$;
while *true* **do**
  Solve problem (LBP-REFINE) globally;
  if *Infeasible* **then**
    Problem (SIP) is infeasible. **return** Infeasible;
  Save the solution $\boldsymbol{x}^i$, the objective value $f^i = f(\boldsymbol{x}^i)$, and the lower bound LBD^i;
  $LBD := LBD^i$;
  Solve problem (LLP) globally for given $\boldsymbol{x}^i$, saving the solution $\boldsymbol{y}^i$ and the objective value G^i;
  if *Infeasible or* $G^i \leq 0$ **then**
    Found a feasible point. **return** $\boldsymbol{x}^i$;
  $d \leftarrow d^\dagger \in \underset{d' \in \mathcal{D}}{\operatorname{argmin}}\ \operatorname{width}(\mathcal{V}^{d'})$ s.t. $\boldsymbol{y}^i \in \mathcal{V}^{d'}$;
  `// adaptively refine box containing two points`
  if $i > 0$ **then**
    $\mathcal{V}^\dagger, \mathcal{V}^i \leftarrow$ `Refine`$(\boldsymbol{y}^d, \boldsymbol{y}^i, \mathcal{V}^d)$; `// see Function 1`
    $\bar{\mathcal{Y}}^i \leftarrow \mathcal{V}^i \cap \mathcal{E}$;
    $\bar{\mathcal{Y}}^d \leftarrow \mathcal{V}^d \cap \mathcal{E}$;
    $\boldsymbol{B}^i \leftarrow$ valid lower-bounding Hessian over $\mathcal{X} \times \bar{\mathcal{Y}}^i$;
    $\boldsymbol{B}^d \leftarrow$ valid lower-bounding Hessian over $\mathcal{X} \times \bar{\mathcal{Y}}^d$;
    `// estimate conservatism`
    $\sigma^i \leftarrow \psi(\boldsymbol{B}^i, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^i, \boldsymbol{y}^i))$;
    $\sigma^d \leftarrow \psi(\boldsymbol{B}^d, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d))$;
  else
    $\bar{\mathcal{Y}}^i \leftarrow \mathcal{E}$;
    $B^i \leftarrow$ valid lower-bounding Hessian over $\mathcal{X} \times \bar{\mathcal{Y}}^i$;
    $\sigma^i \leftarrow \psi(\boldsymbol{B}^i, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^i, \boldsymbol{y}^i))$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{i\}$;
  $\hat{\mathcal{D}} \leftarrow \{d \in \mathcal{D} \mid \sigma^d \leq \sigma_{\max}\}$

We require that the interval enclosures fulfill the following property regarding their convergence.

Property 2.1: *Let the* $f : \mathcal{S} \subset \mathbb{R}^n \to \mathbb{R}$ *be a continuous function and let* $\bar{\mathcal{V}}$ *and* $\bar{\mathcal{V}}^0$ *be boxes with* $\bar{\mathcal{V}} \subseteq \bar{\mathcal{V}}^0 \subseteq \mathcal{V}$. *Then an interval enclosure is Lipschitz if it results in a box* $\bar{\mathcal{F}} \subset \mathbb{R}$ *with* $\operatorname{width}(\bar{\mathcal{F}}) \leq K \operatorname{width}(\bar{\mathcal{V}})$ *where K depends on the box* $\bar{\mathcal{V}}^0$ *but not on* $\bar{\mathcal{V}}$.

We note that this property holds for enclosures computed by interval arithmetic under mild assumptions [35, Theorem 1].

We then obtain bounding matrices by the following proposition.

Proposition 2.1 ([33, Theorem 4.2.2]): *Given symmetric matrices $\boldsymbol{H}^{L,d}, \boldsymbol{H}^{U,d}$ enclosing $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y})$ on $\mathcal{X}\times\bar{\mathcal{Y}}^d$, we obtain bounding matrices $\boldsymbol{L}^d, \boldsymbol{U}^d$ by*

$$\boldsymbol{U}^d_{i,j} = \begin{cases} \boldsymbol{H}^{U,d}_{ii} + \frac{1}{4}\sum_{k=1}^{n_y}(\boldsymbol{H}^{U,d}_{k,i} - \boldsymbol{H}^{L,d}_{k,i}) + (\boldsymbol{H}^{U,d}_{i,k} - \boldsymbol{H}^{L,d}_{i,k}), & \text{if } i=j \\ (\boldsymbol{H}^{U,d}_{k,i} + \boldsymbol{H}^{L,d}_{k,i})/2, & \text{else} \end{cases}$$

and

$$\boldsymbol{L}^d_{i,j} = \begin{cases} \boldsymbol{H}^{L,d}_{ii} - \frac{1}{4}\sum_{k=1}^{n_y}(\boldsymbol{H}^{U,d}_{k,i} - \boldsymbol{H}^{L,d}_{k,i}) + (\boldsymbol{H}^{U,d}_{i,k} - \boldsymbol{H}^{L,d}_{i,k}), & \text{if } i=j \\ (\boldsymbol{H}^{U,d}_{k,i} + \boldsymbol{H}^{L,d}_{k,i})/2, & \text{else} \end{cases}$$

which satisfy $\boldsymbol{L}^d \preceq \boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}) \preceq \boldsymbol{U}^d$, and

$$\begin{aligned} g(\boldsymbol{x},\boldsymbol{y}) &\le g(\boldsymbol{x},\boldsymbol{y}^d) + \boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)(\boldsymbol{y}-\boldsymbol{y}^d) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{y}^d)^T\boldsymbol{U}^d(\boldsymbol{y}-\boldsymbol{y}^d) \\ g(\boldsymbol{x},\boldsymbol{y}) &\ge g(\boldsymbol{x},\boldsymbol{y}^d) + \boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)(\boldsymbol{y}-\boldsymbol{y}^d) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{y}^d)^T\boldsymbol{L}^d(\boldsymbol{y}-\boldsymbol{y}^d) \end{aligned} \tag{4}$$

on $\mathcal{X}\times\bar{\mathcal{Y}}^d$.

As stated in the following proposition, we use these results to derive a valid positive definite matrix for $\boldsymbol{B}^d$.

Proposition 2.2: *Consider a lower-bounding matrix $\boldsymbol{L}^d$ on $\mathcal{X}\times\bar{\mathcal{Y}}^d$ as described in Proposition 2.1. The matrix $\boldsymbol{B}^d = -(\boldsymbol{L}^d)^-$ is positive semidefinite and fulfills the inequality*

$$g(\boldsymbol{x},\boldsymbol{y}) \ge g(\boldsymbol{x},\boldsymbol{y}^d) + \boldsymbol{D}_2 g(\boldsymbol{x},\boldsymbol{y}^d)(\boldsymbol{y}-\boldsymbol{y}^d) - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{y}^d)^T\boldsymbol{B}^d(\boldsymbol{y}-\boldsymbol{y}^d)$$

for all $\boldsymbol{x}\in\mathcal{X}$ and $\boldsymbol{y}\in\bar{\mathcal{Y}}^d$.

Proof: The claim follows from Equation (4) in Proposition 2.1 and $-\boldsymbol{B} = (\boldsymbol{L}^d)^-$ since $(\boldsymbol{L}^d)^- \preceq \boldsymbol{L}^d$ which implies that $\boldsymbol{w}^T(\boldsymbol{L}^d)^-\boldsymbol{w}^T \le \boldsymbol{w}^T(\boldsymbol{L}^d)\boldsymbol{w}^T$ holds for all $\boldsymbol{w}\in\mathbb{R}^{n_y}$. ■

We note that in contrast to using an approach based on diagonal dominance, such as the approach taken in the α-BB method for constructing a convex relaxation or in [28] for constructing a concave relaxation, we obtain a restriction that is exact if the interval enclosure of the Hessian contains a single matrix. This follows from Equation (4) as $\boldsymbol{H}^{U,d} = \boldsymbol{H}^{L,d}$ implies $\boldsymbol{L}^d = \boldsymbol{U}^d$. As noted in Section 2.1, the additional computational cost compared to deriving a scalar or vector is considered small compared to the situation inside a branch-and-bound solver for global optimization.

Instead of constructing the convex quadratic restriction based on bounding Hessian matrices derived by interval enclosures, they could also be derived from specialized methods to bound Taylor-remainder series implemented by Streeter and Dillon [36]. However, at present, the available implementation of this approach is aimed at bounding the output of neural networks, and support for general multivariate nonlinear operations is currently limited. Another alternative would be to construct a quadratic restriction based on concave underestimation and convex overestimation for all encountered elementary unitary functions and nonlinear binary operations such as multiplication and division. For sin, cos and the logistic function, as well as multiplication, the respective envelopes have already been constructed in [37, 38]. Potentially, by extending these derivations to additional functions, a better convex quadratic restriction could be constructed. Another approach closely related to our construction of a quadratic convex restriction is the construction of convex quadratic relaxations in [39]. Therein, the quadratic relaxation is also based on a second-order Taylor-expansion, but the Hessian matrix $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y})$ is not replaced based on interval enclosures but rather is assumed to be positive definite at the expansion point $\boldsymbol{y}^d$ and scaled, for example by a scalar α. To use this approach in the current context, it would be necessary to assume that the Hessian of the SIP constraint $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}^d)$ remains negative definite for all upper-level variables $\boldsymbol{x} \in \mathcal{X}$.

Remark 2.1: We can also use the derived bounding Hessians to generate a procedure for upper-bounding, similar to [11, 15], by finding an optimal diagonal perturbation $\boldsymbol{\alpha}^d$, that is sufficient to make the upper bounding Hessian $\boldsymbol{U}^d$ negative definite. This has been considered for building convex relaxations inside a branch-and-bound algorithm in [40, 41] with favorable results in terms of the required number of iterations. Within a branch-and-bound algorithm for solving nonconvex optimization problems, the computational cost of solving a semidefinite optimization problem can be prohibitive [41]. However, similar to the reasoning behind using the lower-bounding Hessian for the restriction, when the relaxation is employed to *formulate* a global optimization problem, the additional cost of solving a (linear) semidefinite optimization problem is likely to be less significant.

Utilizing Wolfe duality, we can formulate a SIP with the convexified lower level as a single-level problem [12]. We can therefore utilize the derived coefficients $\boldsymbol{\alpha}^d$, to construct a convex relaxation of the lower-level problem and formulate the corresponding single-level optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}^d\in\mathcal{Y}\cap\bar{\mathcal{Y}}^d,\boldsymbol{\mu}^d\in\mathbb{R}^{n_b}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \ell^d(\boldsymbol{x},\boldsymbol{y}^d) \leq 0, \quad \forall\, d \in \mathcal{D} \\ & \boldsymbol{D}_2\ell^d(\boldsymbol{x},\boldsymbol{y}^d) = \boldsymbol{0}, \quad \forall\, d \in \mathcal{D} \\ & \boldsymbol{\mu}^d \geq 0, \quad \forall\, d \in \mathcal{D} \end{aligned} \tag{UBP-Heur}$$

where we define $\ell^d(\boldsymbol{x},\boldsymbol{y}) = \tilde{g}^d(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{\lambda}^T(\hat{\boldsymbol{A}}\boldsymbol{y} - \hat{\boldsymbol{b}})$ with $\hat{\boldsymbol{A}} = [\boldsymbol{A};\boldsymbol{I};-\boldsymbol{I}], \hat{\boldsymbol{b}} = [\boldsymbol{b};\boldsymbol{y}^{U,d};-\boldsymbol{y}^{L,d}]$. We note that problem (UBP-Heur) is a restriction of problem (SIP) if $\mathcal{Y} \subseteq \bigcup_{d\in\mathcal{D}} \bar{\mathcal{Y}}^d$. Correspondingly, every feasible point in problem (UBP-Heur) is feasible in problem (SIP). However, in contrast to the work in [11, 15], we use a box refinement that is chosen to facilitate the lower-bounding and is restricted to associate a discretization point with every box.

As a result, we allow for overlapping boxes, which means that we cannot guarantee that problem (UBP-Heur) ever becomes feasible. In preliminary tests, applying problem (UBP-Heur) with the boxes used for lower-bounding rarely identified a feasible point with an objective value close to optimal, except when the lower-level problem was determined to be convex. Alternatively, a separate box refinement, e.g. the box refinement of $\mathcal{Y}$ from [15], could be used in combination with problem (UBP-Heur) for upper-bounding and compared with the upper-bounding approaches mentioned in the introduction, but this is outside the scope of this work, which focuses on lower-bounding.

For our result on superlinear convergence, we will require that $-\boldsymbol{B}^d$ converges to the Hessian of the SIP constraint g in the following sense, which closely resembles the generalization of the Dennis-Moré condition for superlinear convergence of the BFGS method for constrained optimization [42]:

Property 2.2: *We call the Hessian approximation $\boldsymbol{B}^d = \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}})$ terminally exact if is convergent in the following sense:*

$$\max_{(\boldsymbol{y}^*+\boldsymbol{w})\in\mathcal{Y}^{\mathrm{act}}(\boldsymbol{y}^*)} \frac{\|\boldsymbol{w}^T\left(-\boldsymbol{B}(\boldsymbol{x},\bar{\mathcal{Y}}^d) - \boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}^*)\right)\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2} \xrightarrow{\bar{\mathcal{Y}}^d\to\{\boldsymbol{y}^*\}} 0$$

where $\mathcal{Y}^{\mathrm{act}}(\boldsymbol{y}^*) := \{\boldsymbol{y}\in\mathcal{Y} | C(\boldsymbol{y}-\boldsymbol{y}^*) = \boldsymbol{0}\}$ with $\boldsymbol{C}$ consisting of the active rows of problem (LLP) at $\boldsymbol{y}^*$, i.e. the rows where $\boldsymbol{A}\boldsymbol{y}^* \leq \boldsymbol{b}$ is fulfilled with equality.

We show that the presented construction fulfills this assumption if the Hessian of the lower level is independent of the upper-level variables $\boldsymbol{x}$.

Proposition 2.3: *Assume that $\boldsymbol{D}_{2,2}g$ does not depend on the upper-level variables $\boldsymbol{x}$. Then Property 2.2 is fulfilled by $\boldsymbol{B}^d = -(\boldsymbol{L}^d)$ with $\boldsymbol{L}^d$ constructed according to Proposition 2.1 if LICQ holds at the corresponding optimum $\boldsymbol{y}^*$.*

Proof: We assume that the Hessian $\boldsymbol{D}_{2,2}g$ is independent of $\boldsymbol{x}$ and can thus denote the constant matrix $\boldsymbol{D}_{2,2}g(\boldsymbol{x},\boldsymbol{y}^*)$ as $\boldsymbol{H}$.

We decompose the matrix $\boldsymbol{H}$ into its positive semidefinite part $(\boldsymbol{H})^+$ and its negative semidefinite part $(\boldsymbol{H})^-$, i.e. $\boldsymbol{H} = (\boldsymbol{H})^+ + (\boldsymbol{H})^-$. As an intermediate step, we obtain

$$\begin{aligned}\max_{(\hat{\boldsymbol{y}}+\boldsymbol{w})\in\mathcal{Y}^{\mathrm{act}}(\boldsymbol{y}^*)} & \frac{\|\boldsymbol{w}^T\left(\boldsymbol{H}-(-\boldsymbol{B}^d)\right)\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2} \\ = \max_{(\hat{\boldsymbol{y}}+\boldsymbol{w})\in\mathcal{Y}^{\mathrm{act}}(\boldsymbol{y}^*)} & \frac{\|\boldsymbol{w}^T\left((\boldsymbol{H})^- - (-\boldsymbol{B}^d)\right)\boldsymbol{w} + \boldsymbol{w}^T(\boldsymbol{H})^+\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2}\end{aligned}$$

using second-order necessary conditions (which follow from LICQ [43]), we obtain

$$= \max_{(\hat{\boldsymbol{y}}+\boldsymbol{w})\in\cap\mathcal{Y}^{\mathrm{act}}(\boldsymbol{y}^*)} \frac{\|\boldsymbol{w}^T\left((\boldsymbol{H})^- - (-\boldsymbol{B}^d)\right)\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2}$$

Finally, the claim follows from the fact that $-\boldsymbol{B}^d$ converges to $\boldsymbol{H}^-$ as $\bar{\mathcal{Y}}^d$ converges to $\{\boldsymbol{y}^*\}$ which is a direct consequence of the formula for the lower-bounding Hessian in Proposition 2.1, and the convergence of the interval enclosure (see Property 2.1). ■

In the next section, we discuss how the set $\hat{\mathcal{D}} \subseteq \mathcal{D}$ is chosen.

2.5. Identifying overly conservative convex quadratic restrictions

For some problems and boxes $\bar{\mathcal{Y}}^d$, the interval enclosure of the Hessian of the SIP constraint g might be extremely wide, and the underestimation by using $\boldsymbol{B}^d$ extremely conservative. In these situations, there will likely be little benefit from replacing the constraint $g(\boldsymbol{x}, \boldsymbol{y}^d) \leq 0$ with $g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq 0$, while introducing additional variables and constraints. We, therefore, only use the latter constraint at discretization indices $d \in \hat{\mathcal{D}}$. As a simple means to exclude indices where the used quadratic restriction is likely to be extremely conservative, we propose to exclude indices $d \in \mathcal{D}$ where the difference between $\boldsymbol{B}^d$ and the negative definite part of $\boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d)$ is large. We use $\hat{\mathcal{D}} = \{d' \in \mathcal{D} \mid \psi(\boldsymbol{B}^d, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d)) \leq \sigma_{\max}\}$ with $\psi(\boldsymbol{B}, \boldsymbol{H}) = \frac{\|-\boldsymbol{B}-(\boldsymbol{H})^-\|_F}{1+\|(\boldsymbol{H})^-\|_F}$ and a large conservatism limit, e.g. $\sigma_{\max} = 10^4$. The following proposition states that if the Hessian of the SIP constraint is independent of the upper-level variables and the boxes are sufficiently small, the corresponding discretization index d will be included in $\hat{\mathcal{D}}$.

Proposition 2.4: *Assume that $\boldsymbol{D}_{2,2}g$ does not depend on the upper-level variables $\boldsymbol{x}$. Then there exists $\epsilon^w > 0$ for all $\sigma_{\max} > 0$ such that* $\text{width}(\bar{\boldsymbol{Y}}^d) \leq \epsilon^w \implies \psi(\boldsymbol{B}^d, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d)) \leq \sigma_{\max}$ *with $\boldsymbol{B}^d = -(\boldsymbol{L}^d)$ where $\boldsymbol{L}^d$ is constructed according to Proposition 2.1.*

Proof: Note that for $\boldsymbol{B}^d = -(\boldsymbol{L}^d)^-$, $\psi(\boldsymbol{B}^d, \boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d)) = 0$ if the interval enclosure of the Hessian matrix is exact, i.e. if $\boldsymbol{H}^{U,d} = \boldsymbol{H}^{L,d}$ holds in Proposition 2.1. Since the Hessian does not depend on the upper-level variables, there exists $K > 0$ with $\|\boldsymbol{H}^{U,d} - \boldsymbol{H}^{L,d}\|_\infty \leq K\,\text{width}(\bar{\boldsymbol{Y}}^d)$ where $\boldsymbol{H}^{U,d}$ and $\boldsymbol{H}^{L,d}$ are the matrices in the enclosure of the Hessian (see Property 2.1). The claim follows as, according to Proposition 2.1, $\boldsymbol{L}^d$ depends continuously on the matrices $\boldsymbol{D}_{2,2}g(\boldsymbol{x}^d, \boldsymbol{y}^d)$ and the projection $(\cdot)^-$ to the negative semidefinite part is also continuous [34, 44]. ■

We discuss the box refinement used in Algorithm 1 in the following section.

2.6. Construction of boxes around discretization points

The refinement in problem (LBP-REFINE) is based on boxes around each discretization point. These boxes can be chosen arbitrarily around each discretization point. In general, small boxes have the advantage of providing sharper interval enclosures of the Hessian, thus enabling less conservative restrictions. On the other hand, the semi-infinite constraint is approximated only on a small subset of the host set $\bar{\mathcal{Y}}$. As a result, we suggest refining the boxes used in the discretization adaptively. Intuitively, we prefer that boxes become smaller in regions where more discretization points are added. The following property of our box selection will be required in our proof of superlinear convergence.

Property 2.3: *In each iteration i with discretization index set $\mathcal{D}^i$, the considered boxes $\bar{\mathcal{Y}}^d$, $d \in \mathcal{D}^i$ each have an associated discretization point $\boldsymbol{y}^d \in \bar{\mathcal{Y}}^d$, $d \in \mathcal{D}^i$.*

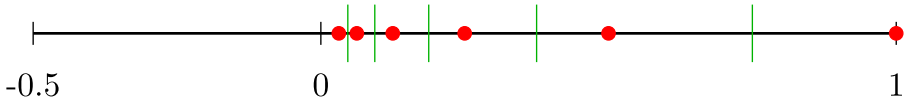

Figure 3. Only using disjunctive splits, it is impossible to ensure Proposition 2.3: for the sequence of discretization points $y^i = \frac{1}{2}^{i-1}, i \in \mathbb{N}$, disjunct splits will lead to the last discretization point always being contained in an interval of width larger than 0.5.

Consider any converging sequence of lower-level points $\{\boldsymbol{y}^{\boldsymbol{i}}\}_{i\in\mathbb{N}} \to \boldsymbol{y}^*$. *For any fixed* $\epsilon > 0$, *there exist* $K > 0$ *such that for all iterations* $i > K$, *the iterates* $\boldsymbol{y}^i, i > K$ *are inside boxes* $\bar{\mathcal{Y}}^d$, $d \in \mathcal{D}^i$ *with* $\text{width}(\bar{\mathcal{Y}}^d) < \epsilon$.

In continuous branch-and-bound methods, as well as within the upper-bounding approach of [15], a given box is refined by splitting the box along a specific variable dimension. Consider a box $\mathcal{V} = [\boldsymbol{v}^l, \boldsymbol{v}^u] \subseteq \bar{\mathcal{Y}}$ and two points $\boldsymbol{y}^1, \boldsymbol{y}^2 \in \mathcal{V}$. The box $\mathcal{V}$ can be partitioned into $\mathcal{V}_1 \ni \boldsymbol{y}^1$ and $\mathcal{V}_2 \ni \boldsymbol{y}^2$, for example, by splitting it at the midpoint $y_i = \frac{y_i^1 + y_i^2}{2}, i \in 1, \ldots, n_y$. We call this a disjunctive split because the resulting boxes are disjunct. However, as noted in [15], without countermeasures, this can result in boxes that have a smaller and smaller volume but a constant width. This is undesirable, considering the worst-case error in interval methods decreases with the width. In [15], an upper bound on the aspect ratio (the ratio of the longest and shortest side of the box) is enforced. In our setting, a box should be associated with one of the discretization points, which are not chosen freely but given by the solutions of the lower-level problems. As a result, it is generally impossible to enforce a specific aspect ratio only using disjunctive splits along the variable dimensions. This is illustrated in the following example.

Example 2.2: Consider the box $[-4, 4] \times [-1, 1]$ with the two points $(-4, 1)$ and $(-4, -1)$. One cannot divide the box using a disjunctive split without exceeding a maximum aspect ratio of 4 in the resulting boxes.

Even in the one-dimensional case, where aspect ratios are not an issue, the iterative splitting of a box into two boxes, both containing one of the discretization points, does not satisfy Proposition 2.3. This is evidenced by the counterexample $y^i = \frac{1}{2}^{i-1}, i \in \mathbb{N}$ on $\bar{\mathcal{Y}} = [-0.5, 1]$, adapted from [15, Example 3.2.3] and illustrated in Figure 3. Indeed, in this example, the boxes $\mathcal{V}$ containing the newest members of this sequence always have $\text{width}(\mathcal{V}) \geq 0.5$.

Instead of only using disjunctive splits, we propose to use them if possible. Otherwise, we keep the existing box and add a scaled version centered around the new discretization point. This approach is detailed in Function 1.

The following proposition shows that Proposition 2.3 is fulfilled if we use Function 1 for the box-refinement step in Algorithm 1.

Proposition 2.5: *Consider an initial box* $\mathcal{V}^{\text{init}} \supseteq \bar{\mathcal{Y}}$ *with an aspect ratio smaller than* α_{init} *and the initial point* $\boldsymbol{y}^{\text{init}} \in \mathcal{V}^{\text{init}}$. *For a sequence* $\{\boldsymbol{y}^{\boldsymbol{i}}\}_{i\in\mathbb{N}} \to \boldsymbol{y}^* \in \mathcal{V}^{\text{init}}$, *boxes are constructed iteratively: for each new point* $\boldsymbol{y}^i$, *find the box* $\mathcal{V}$ *with smallest width containing* $\boldsymbol{y}^i$ *and the*

Function 1: RefineBox: The proposed box refinement

1 **Function** `RefineBox` ($\boldsymbol{y}^1 \in \mathbb{R}^{n_y}$, $\boldsymbol{y}^2 \in \mathbb{R}^{n_y}$, $\mathcal{V} \ni \boldsymbol{y}^1, \boldsymbol{y}^2$) **:**
  Data: reduction factor $\omega_{\max} \in (0.5, 1)$, finite maximal aspect ratio $\alpha_{\max} \in (2, \infty)$, scaling factor $\upsilon \in (1, \infty)$;
2 **foreach** $i \in \{1, \ldots, n_y\}$ **do**
3 $y_{split,i} \leftarrow \frac{y_i^1 + y_i^2}{2}$;
  `// reduction factor`
4 $\omega_i \leftarrow 1 - \min\{\frac{y_{split,i} - y_i^l}{y_i^u - y_i^l}, \frac{y_i^u - y_{split,i}}{y_i^u - y_i^l}\}$;
  `// aspect ratio of box 1 (if increased by split)`
5 $\alpha^1 \leftarrow \max_{j \in \{1,\ldots,n_y\} \setminus \{i\}} \frac{y_j^u - y_j^l}{y_i^u - y_{split,i}}$;
  `// aspect ratio of box 2 (if increased by split)`
6 $\alpha^2 \leftarrow \max_{j \in \{1,\ldots,n_y\} \setminus \{i\}} \frac{y_j^u - y_j^l}{y_i^u - y_{split,i}}$;
7 $\alpha_i \leftarrow \max\{\alpha^1, \alpha^2\}$
  `// Apply disjunctive split if possible`
8 **if** $\Omega := \{j \in \{1, \ldots, n_y\} : \omega_j \leq \omega_{\max} \wedge \alpha_j \leq \alpha_{\max}\} \neq \emptyset$ **then**
9 select $j \in \operatorname{argmax}_{i \in \Omega} \omega_i$;
10 $\mathcal{V}^1, \mathcal{V}^2 \leftarrow$ split $\mathcal{V}$ along $y_j = y_{split,j}$;
11 **return** $\mathcal{V}^1, \mathcal{V}^2$;
  `// Apply centered reduction: add scaled box centered around` $\boldsymbol{y}^2$
12 **else**
13 $\mathcal{V}^1 \leftarrow \mathcal{V}$;
14 $\boldsymbol{r} = \frac{\boldsymbol{y}^u - \boldsymbol{y}^l}{\upsilon}$;
15 $\mathcal{V}^2 \leftarrow [\boldsymbol{y}^2 - \boldsymbol{r}, \boldsymbol{y}^2 + \boldsymbol{r}]$;
16 **return** $\mathcal{V}^1, \mathcal{V}^2$;

associated point $\boldsymbol{y}^\dagger$. *Generate two new boxes* $\mathcal{V}^\dagger$ *and* $\mathcal{V}^\star$, *associated with* $\boldsymbol{y}^\dagger$ *and* $\boldsymbol{y}^i$, *respectively, replacing* $\mathcal{V}$. *Proposition* 2.3 *holds for each sequence converging in* $\mathcal{V}^{\text{init}}$ *when* $\mathcal{V}^\dagger$ *and* $\mathcal{V}^\star$ *are created from* $\mathcal{V}$ *per* `Refine`$(\boldsymbol{y}^\dagger, \boldsymbol{y}^i, \mathcal{V})$ *in Function* 1.

Proof: Each element y^i of the sequence $\{\boldsymbol{y}^i\}_{i \in \mathbb{N}}$ will, by construction, be contained within an existing box. We assume that this box has a width greater than a fixed $\epsilon > 0$ for an infinite number of (not necessarily sequential) iterations.

We consider the following tree: the root node of the tree is the initial box. In the centered reduction in Line 12, the input box stays the same $\mathcal{V} = \mathcal{V}^1$, but the other box $\mathcal{V}^2$ is created with reduced width. We call the first the parent and the latter the child box. In the case of the disjunctive split of a box $\mathcal{V}$, it will also be called a parent, while the two resulting boxes $\mathcal{V}^1, \mathcal{V}^2$ are the children. We assign each box a depth recursively, i.e. the initial

box has depth 0 and children of depth s boxes are boxes of depth $s+1$. We note that the volume of children is reduced to at most $\max\{w_{\max}, \frac{1}{v}\}$ of the volume of the parent. Since a maximum aspect ratio of $\max\{\alpha_{\text{init}}, \alpha_{\max}\}$ is enforced, this means that for fixed δ there is a finite depth s^δ where all nodes $\mathcal{V}$ with depth $s \geq s^\delta$ have $\text{width}(\mathcal{V}) < \delta$. In the following, we show that any box can only have a finite number of children with width greater than δ. This implies that the number of boxes in the tree with width greater than δ is finite. However, since new sequence elements that are contained in an existing box of width greater than ϵ create at least one child of width at least $\epsilon \max\{\frac{1}{2}, \frac{1}{v}\}$, this leads to a contradiction with $\delta < \epsilon \max\{\frac{1}{2}, \frac{1}{v}\}$.

If a box is split by disjunctive splits (Line 10), the box is replaced by its two children and no new children are added. The alternative to a disjunctive cut is the centered reduction in Line 12. We show that a box $\mathcal{V}$ with $\text{width}(\mathcal{V}) > \delta$ cannot have infinitely many children caused by the centered reduction: Let w be the smallest width of a box $\mathcal{V}$. Since the maximum aspect ratio $\alpha'_{\max} = \max\{\alpha_{\text{init}}, \alpha_{\max}\}$ is maintained and $\text{width}(\mathcal{V}) > \delta$, we have $w \geq \frac{\delta}{\alpha'_{\max}} > 0$. Each child box $\mathcal{V}^{\text{child}}$ of box $\mathcal{V}$ is centered around a sequence element. This element is not contained in the other child boxes $\mathcal{V}^{\text{child},'}$ of $\mathcal{V}$, otherwise $\mathcal{V}^{\text{child}}$ would be a child of $\mathcal{V}^{\text{child},'}$ instead. Since each child box contains a ball of radius $\frac{w}{v} > 0$ and $\mathcal{V}$ is compact, the number of child boxes of $\mathcal{V}$ is finite. ■

3. Convergence

In this section, we investigate the convergence of Algorithm 1. First, we formally state that all accumulation points are SIP-feasible.

Proposition 3.1 (Feasibility of accumulation points): *Consider the sequence of iterates* $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ *generated by Algorithm* 1. *Any accumulation point* $\boldsymbol{x}^*$ *of this sequence is SIP-feasible, i.e.* $\boldsymbol{x}^* \in \mathcal{F}_{SIP}$.

Proof: According to Proposition 2.2, $g^\dagger(\boldsymbol{x}, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq 0$ is a relaxation of $\max_{\mathcal{Y}\cap\bar{\mathcal{Y}}^d} g(\boldsymbol{x}, \boldsymbol{y}) \leq 0$ and thus of the constraint $g(\boldsymbol{x}, \boldsymbol{y}) \leq 0,\ \forall \boldsymbol{y} \in \mathcal{Y}$. Accordingly, problem (LBP-REFINE) is a relaxation of problem (SIP). At the same time, the feasible set of problem (LBP-REFINE) is contained inside the feasible set of problem (LBP) (see Equation (3)). Thus, the claim follows from the corresponding result for the algorithm from [4]. ■

We also obtain the following quadratic bound for the SIP-constraint violation that holds even if the chosen function for $\boldsymbol{B}$ depends on the upper-level variables.

Lemma 3.1: *Let* $\boldsymbol{x}^k \in \mathcal{X}$ *be the current iterate of Algorithm* 1. *Denote with* $\boldsymbol{y}^k$ *any global solution to problem (LLP). Consider any discretization index* $d \in \hat{\mathcal{D}}$ *with discretization point* $\boldsymbol{y}^d$ *and associated box* $\bar{\mathcal{Y}}^d \ni \boldsymbol{y}^k$. *The SIP-constraint violation* $g(\boldsymbol{x}^k, \boldsymbol{y}^k)$ *is bounded by*

$$g(\boldsymbol{x}^k, \boldsymbol{y}^k) \leq \frac{1}{2}(\boldsymbol{y}^k - \boldsymbol{y}^d)^T(\boldsymbol{D}_{2,2}g(\boldsymbol{x}, \bar{\boldsymbol{y}}) + \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d))(\boldsymbol{y}^k - \boldsymbol{y}^d), \tag{5}$$

where $\bar{\boldsymbol{y}} = (1-\xi)\boldsymbol{y}^k + \xi\boldsymbol{y}^d \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d$ *with* $\xi \in [0, 1]$.

Proof: Since $g^\dagger(\boldsymbol{x}^k, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \leq 0$ is fulfilled by $\boldsymbol{x}^k$, we obtain

$$\begin{aligned} g(\boldsymbol{x}^k, \boldsymbol{y}^k) &\leq g(\boldsymbol{x}^k, \boldsymbol{y}^k) - g^\dagger(\boldsymbol{x}^k, \boldsymbol{y}^d, \bar{\mathcal{Y}}^d) \\ &= g(\boldsymbol{x}^k, \boldsymbol{y}^k) - \Bigg(g(\boldsymbol{x}, \boldsymbol{y}^d) + \max_{\boldsymbol{w} \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d} \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d)(\boldsymbol{w} - \boldsymbol{y}^d) \\ &\quad - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{y}^d)^T \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d)(\boldsymbol{w} - \boldsymbol{y}^d) \Bigg) \end{aligned} \tag{6}$$

and since $\boldsymbol{y}^k$ is a feasible point in the maximization

$$\leq g(\boldsymbol{x}^k, \boldsymbol{y}^k) - \left(g(\boldsymbol{x}, \boldsymbol{y}^d) + \boldsymbol{D}_2 g(\boldsymbol{x}, \boldsymbol{y}^d)(\boldsymbol{y}^k - \boldsymbol{y}^d) - \frac{1}{2}(\boldsymbol{y}^k - \boldsymbol{y}^d)^T \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d)(\boldsymbol{y}^k - \boldsymbol{y}^d) \right)$$

by Taylor expansion of g, there exists a $\bar{\boldsymbol{y}} = (1 - \xi)\boldsymbol{y}^k + \xi \boldsymbol{y}^d \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d$ with $\xi \in [0, 1]$ such that

$$= \frac{1}{2}(\boldsymbol{y}^k - \boldsymbol{y}^d)^T (\boldsymbol{D}_{2,2} g(\boldsymbol{x}, \bar{\boldsymbol{y}}) + \boldsymbol{B}(\boldsymbol{x}, \bar{\mathcal{Y}}^d))(\boldsymbol{y}^k - \boldsymbol{y}^d). \tag{7}$$

■

Additionally, we analyze the convergence behavior of Algorithm 1 assuming that the Hessian $\boldsymbol{D}_{2,2} g$ does not depend on the upper-level variables $\boldsymbol{x}$. We also assume that the so-called Reduction Ansatz (see Definition A.3) holds. While we aim to solve problem (SIP) globally, we study the local convergence order of converging subsequences. The local convergence order for the algorithm from [4] was analyzed in [24] under the Reduction Ansatz and the following assumption introduced in [24]. This assumption ensures that accumulation points of local minima are local minima [24] and is needed in the proof of Theorem A.1. We will also rely on this assumption for our results. To be self-contained, we repeat some of the definitions from [24] used in this assumption in the appendix.

Assumption 3.1 ([24, Assumption 3.8]): Let $\{\boldsymbol{x}^k\}_{k \in \mathbb{N}}$ be a sequence of local minima generated by the lower-bounding procedure converging to $\boldsymbol{x}^*$, i.e.

$$\lim_{k \to \infty} \boldsymbol{x}^k = \boldsymbol{x}^*.$$

Assume that $\boldsymbol{x}^*$ is a local solution of order ρ (see Definition A.2) and that EMFCQ (see Definition A.4) holds at $\boldsymbol{x}^*$ for problem (SIP). Moreover, assume that the radius r_k as defined in Definition A.1 of the local minima generated by the lower-bounding procedure is strictly positive

$$\inf_{k \in \mathbb{N}} r_k > 0.$$

Note that although Assumption 3.1 only requires local minima, the subproblems in Algorithm 1 are solved to global optimality. Consequently, the technical assumption that the radii of the generated minima have a non-vanishing limit, which might be difficult to verify otherwise, is satisfied trivially.

To prove our result regarding superlinear convergence, we need some consequences of the Reduction Ansatz, which are collected in the following lemma:

Lemma 3.2: *Denote the set of feasible points of problem* (SIP) *as* $\mathcal{F}_{SIP}$. *Consider any convergent (sub)sequence* $\{\boldsymbol{x}^{k_i}\}_{i\in\mathbb{N}}$ *of iterates which converges to a feasible point* $\boldsymbol{x}^* \in \mathcal{F}_{SIP}$. *Assume that the Reduction Ansatz (see Definition A.3) holds at* $\boldsymbol{x}^*$. *Then*

- *There is a finite set of isolated maxima* $\mathcal{Y}^*(\boldsymbol{x}*) \ni \boldsymbol{y}^*$ *of problem (LLP) at* $\boldsymbol{x}^*$ *with* $g(\boldsymbol{x}^*, \boldsymbol{y}^*) = 0$.
- *In a neighborhood* $\mathcal{N}$ *of* $\boldsymbol{x}^*$, *there exists a continuous function* $\boldsymbol{y}^f : \mathcal{N} \to \mathcal{Y}$ *for each* $\boldsymbol{y}^* \in \mathcal{Y}^*(\boldsymbol{x}^*)$ *with* $\boldsymbol{y}^f(\boldsymbol{x}^*) = \boldsymbol{y}^*$ *and* $g(\boldsymbol{x}, \boldsymbol{y}^f(\boldsymbol{x})) = 0$.
- *Each convergent subsequence of the lower-level solutions* $\{\boldsymbol{y}^{k_j}\}_{j\in\mathbb{N}}$ *converges to one of these optima* $\boldsymbol{y}^*$.
- *For each convergent subsequence of the lower-level solutions, the active set does not change for sufficiently large* j.

Proof: The first two points are well-known results of the Reduction Ansatz [45]. The third and fourth points follow from the second and continuity (see 24, Lemma 3.12]) ■

For our result of superlinear convergence, we use an additional assumption. This assumption concerns the box that the next lower-level solution will fall into. To discuss this assumption, we introduce the concept of a predictive box:

Definition 3.1 (Predictive box): We call $\bar{\mathcal{Y}}^d$ a predictive box and $\boldsymbol{y}^d$ a predictive point of iteration k if $d \in \hat{\mathcal{D}}$ is the index of one of the smallest boxes $\bar{\mathcal{Y}}^d, d \in \hat{\mathcal{D}}$ at iteration k containing the point $\boldsymbol{y}^{k+1}$, i.e. $d \in \underset{d'\in\hat{\mathcal{D}}:\hat{\boldsymbol{y}}\in\bar{\mathcal{Y}}^{d'}}{\operatorname{argmin}} \operatorname{width}(\bar{\mathcal{Y}}^{d'})$.

Assumption 3.2: For every convergent subsequence of lower-level solutions $\{\boldsymbol{y}^{k_i}\}_{i\in\mathbb{N}}$, there exists a constant L_α and j_α such that

$$\forall\, j > j_\alpha : \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^d\| \leq L_\alpha \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^{k_{j-1}}\|,$$

for at least one predictive point $\boldsymbol{y}^d$ of iteration $k_j - 1$.

The following theorem states the superlinear convergence of subsequences of Algorithm 1 under the Reduction Ansatz (see Definition A.3), Assumptions 3.1, 3.2 and the assumption that $\boldsymbol{D}_{2,2}g$ does not depend on upper-level variables $\boldsymbol{x}$.

Theorem 3.1: *Assume that the Reduction Ansatz holds. Consider any convergent subsequence* $\{\boldsymbol{x}^{k_i}\}_{i\in\mathcal{I}\subseteq\mathbb{N}}$ *of iterates in Algorithm* 1. *Without loss of generality, assume that this sequence converges to* $\boldsymbol{x}^* \in \mathcal{F}_{SIP}$. *Assume that Assumption 3.1 holds with order* $\rho \leq 2$ *at* $\boldsymbol{x}^*$ *and that the objective function* f *is Lipschitz-continuous near* $\boldsymbol{x}^*$. *Consider a partition of the subsequence* $\{\boldsymbol{x}^{k_i}\}_{i\in\mathbb{N}}$ *into further subsequences* $\{\boldsymbol{x}^{k_j}\}_{j\in\mathcal{J}\subseteq\mathcal{I}}$, *such that the corresponding sequences of lower-level solutions* $\{\boldsymbol{y}^{k_j}\}_{j\in\mathcal{J}}$ *converge to one of the finitely many points*

$\boldsymbol{y}^*$ from Lemma 3.2. If the Hessian $\boldsymbol{D}_{2,2}g$ is independent of the upper-level variables $\boldsymbol{x}$ and Assumption 3.2 holds, each of these subsequences converges at least superlinearly, i.e.

$$\frac{\|\boldsymbol{x}^* - \boldsymbol{x}^{k_j}\|}{\|\boldsymbol{x}^* - \boldsymbol{x}^{k_{j-1}}\|} \xrightarrow{j\to\infty} 0.$$

Proof: Without loss of generality, consider a single convergent subsequence of upper-level iterates $\{\boldsymbol{x}^{k_j}\}_{j\in\mathcal{J}\subseteq\mathcal{I}}$. Denote with $\boldsymbol{y}^{k_j}$ the solution obtained by solving problem (LLP). Note that Proposition 2.3 ensures $\boldsymbol{y}^d \to \boldsymbol{y}^*$ and width$(\bar{\mathcal{Y}}^d) \to 0$ with $y^* \in \bar{\mathcal{Y}}^d$. Thus, by Proposition 2.4, after finitely many iterations, a new discretization point is always contained in one of the existing boxes $\bar{\mathcal{Y}}^d$ with $d \in \hat{\mathcal{D}}$.

Let $\bar{\mathcal{Y}}^d$ and $\boldsymbol{y}^d \in \bar{\mathcal{Y}}^d, d \in \hat{\mathcal{D}}$ be a predictive box and the corresponding predictive point. According to Lemma 3.1, the violation in the current iteration is bounded by

$$g(\boldsymbol{x}^k, \boldsymbol{y}^k) \leq \frac{1}{2}(\boldsymbol{y}^{k_j} - \boldsymbol{y}^d)^T(\boldsymbol{D}_{2,2}g(\boldsymbol{x}, \bar{\boldsymbol{y}}) + B(\boldsymbol{x}, \bar{\mathcal{Y}}^d))(\boldsymbol{y}^{k_j} - \boldsymbol{y}^d), \tag{8}$$

where $\bar{\boldsymbol{y}} = (1-\xi)\boldsymbol{y}^{k_j} + \xi\boldsymbol{y}^d \in \mathcal{Y} \cap \bar{\mathcal{Y}}^d$ with $\xi \in [0,1]$.

According to Lemma 3.2, as both $\boldsymbol{y}^d$ and $\boldsymbol{y}^{k_j}$ converge to $\boldsymbol{y}^*$ the active set of constraints in problem (LLP) does not change after some iterations. Thus, we obtain $\mathcal{Y}^{\text{act}}(\boldsymbol{y}^*) = \mathcal{Y}^{\text{act}}(\boldsymbol{y}^d) = \mathcal{Y}^{\text{act}}(\boldsymbol{y}^{k_j})$ and thus $\bar{\boldsymbol{y}} \in \mathcal{Y}^{\text{act}}(\boldsymbol{y}^*)$.

Consequently, we can combine Equation (8), continuity of $\boldsymbol{D}_{2,2}g(\boldsymbol{x}, \cdot)$ and Property 2.2 to ensure that there exist $\beta^j \xrightarrow{j\to\infty} 0$ with

$$g(\boldsymbol{x}^{k_j}, \boldsymbol{y}^{k_j}) \leq \beta^j \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^d\|^2. \tag{9}$$

According to Assumption 3.2, there exists $L_\alpha > 0$ and j_α with

$$\forall j > j_\alpha : \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^d\|^2 \leq L_\alpha^2 \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^{k_{j-1}}\|^2. \tag{10}$$

By Lemma 3.2, for a sufficiently large iteration index j, the lower-level solutions $\boldsymbol{y}^{k_j}$ converging to the same point $\boldsymbol{y}^*$ depend Lipschitz-continuously on the upper-level variable:

$$\forall j > \tilde{j} : \|\boldsymbol{y}^{k_j} - \boldsymbol{y}^{k_{j-1}}\| \leq L_0 \|\boldsymbol{x}^{k_j} - \boldsymbol{x}^{k_{j-1}}\| \tag{11}$$

with a constant $L_0 > 0$.

By Theorem A.1, we have $L_2 > 0$ and $l'' \in \mathbb{N}$ with

$$\forall l > l'' : \|\boldsymbol{x}^* - \boldsymbol{x}^k\| \leq L_2 \alpha_k^{\frac{1}{\rho}}.$$

Inserting Equations (9)–(11), we obtain with $\rho \leq 2$ and for sufficiently large index j.

$$\begin{aligned}
\|\boldsymbol{x}^* - \boldsymbol{x}^{k_j}\| &\leq \underbrace{\gamma^j}_{L_\alpha L_2 L_0 \sqrt{\beta^j}} \|\boldsymbol{x}^{k_j} - \boldsymbol{x}^{k_{j-1}}\| \\
&= \gamma^j \|(\boldsymbol{x}^{k_j} - \boldsymbol{x}^*) + (\boldsymbol{x}^* - \boldsymbol{x}^{k_{j-1}})\| \\
&\leq \gamma^j \|(\boldsymbol{x}^{k_j} - \boldsymbol{x}^*)\| + \gamma^j \|(\boldsymbol{x}^* - \boldsymbol{x}^{k_{j-1}})\| \\
\overset{\gamma^j < 1}{\Longleftrightarrow} \quad \|\boldsymbol{x}^* - \boldsymbol{x}^{k_j}\| &\leq \frac{\gamma^j}{1-\gamma^j} \|\boldsymbol{x}^* - \boldsymbol{x}^{k_{j-1}}\| \\
\Longleftrightarrow \quad \frac{\|\boldsymbol{x}^* - \boldsymbol{x}^{k_j}\|}{\|\boldsymbol{x}^* - \boldsymbol{x}^{k_{j-1}}\|} &\leq \frac{\gamma^j}{1-\gamma^j} \xrightarrow{j \to \infty} 0
\end{aligned}$$

which proves the claim. ■

Assumption 3.2 is not trivial and requires additional discussion.

If the last iterate of the subsequence is a predictive point, i.e. $\boldsymbol{y}^{k_{j-1}} = \boldsymbol{y}^d$, $\forall j$, the assumption holds trivially.

However, the assumption can be violated, for example, when discretization points $\boldsymbol{y}^{k_j}$ and $\boldsymbol{y}^{k_{j-1}}$ are close to, but on different sides of the border of the predictive box. As an example, consider that the scalar sequence $y^{k_j} = \frac{(-1)^{k_j}}{(k_j+1)!}$ on the box $[-1, 1]$ with only disjunctive splits, i.e. without the centered reduction (which is not required for this sequence).

An interpretation of the violation of Assumption 3.2 is that the convergence of the lower-level solution points is arbitrarily faster than the convergence of the predictive box according to Proposition 2.3 and thus to the predictive point. This is formalized in the following lemma.

Lemma 3.3: *Let $\mathcal{L}_\alpha$ be the indices of the subsequence of iterates where Assumption 3.2 is violated for a given L_α. Then*

$$\forall \epsilon > 0 \exists L_\alpha : \forall j \in \mathcal{L}_\alpha \left[\frac{\|\boldsymbol{y}^{k_j} - \boldsymbol{y}^{k_{j-1}}\|}{\|\boldsymbol{y}^{k_j} - \boldsymbol{y}^d\|} \leq \epsilon \right]. \tag{12}$$

If we assume that the predictive point $\boldsymbol{y}^d$ is always the second-to-last in the subsequence, i.e. $\boldsymbol{y}^d = \boldsymbol{y}^{k_{j-2}}$ this would correspond to superlinear convergence of changes in discretization points. However, in general, it is not known that the descriptive point will be either the last or second-to-last discretization point. Instead, as we state in the following proposition, this leads to a different kind of superlinear convergence, namely in terms of the width of the predictive box width($\bar{\mathcal{Y}}^d$), which converges to zero as $\bar{\mathcal{Y}}^d$ converges to $\{\boldsymbol{y}^*\}$.

Proposition 3.2: *Assume that the requirements of Theorem 3.1 hold except for Assumption 3.2. Let $\mathcal{L}_\alpha$ be the set of iterates where Assumption 3.2 is violated for a given L_α. Consider the corresponding subsequences of the upper-level and lower-level iterates $\{\boldsymbol{x}^{k_l}\}_{l \in \mathcal{L}_\alpha \subseteq \mathcal{J}}$ and*

$\{\boldsymbol{y}^{k_l}\}_{l\in\mathcal{L}_\alpha\subseteq\mathcal{J}}$. *For any* $K_\alpha > 0$ *there exists* $L_\alpha > 1$ *such that the error bound*

$$\|\boldsymbol{x}^* - \boldsymbol{x}^{k_l}\| \leq K_\alpha \operatorname{width}(\bar{\mathcal{Y}}^d)$$

holds for sufficiently large iteration k_l *where* $\boldsymbol{y}^d$ *and* $\bar{\mathcal{Y}}^d$ *are any predictive point and box at iteration* $k_l - 1$,

Proof: Since Assumption 3.2 is violated the inequality

$$\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^{k_{l-1}}\| \leq \frac{1}{L_\alpha}\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^d\| \tag{13}$$

holds for all $l \in \mathcal{L}_\alpha$

According to [24, p.56], for sufficiently large iteration l there is a constant $K_1 > 0$ with

$$g(\boldsymbol{x}^{k_l}, \boldsymbol{y}^{k_l}) \leq K_1\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^{k_{l-1}}\|^2. \tag{14}$$

Assuming $g(\boldsymbol{x}^{k_l}, \boldsymbol{y}^{k_l}) > 0$, according to Theorem A.1 there exists L_2 with

$$\|\boldsymbol{x}^* - \boldsymbol{x}^{k_l}\| \leq L_2\sqrt{g(\boldsymbol{x}^{k_l}, \boldsymbol{y}^{k_l})}.$$

As a result

$$\|\boldsymbol{x}^* - \boldsymbol{x}^{k_l}\| \overset{\text{Equation (14)}}{\leq} L_2\sqrt{K_1}\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^{k_{l-1}}\| \overset{\text{Equation (13)}}{\leq} \frac{L_2\sqrt{K_1}}{L_\alpha}\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^d\|$$

There exists a constant c such that $\|\boldsymbol{y}^{k_l} - \boldsymbol{y}^d\| \leq c\operatorname{width}(\bar{\mathcal{Y}}^d)$. Since K_1 and L_2 are independent of L_α, the claim follows with $L_\alpha = \frac{cL_2\sqrt{K_1}}{K_\alpha}$. ■

Remark 3.1: In comparison, the algorithm from [4] converges quadratically for local solutions of order $\rho = 1$ under the conditions of Theorem 3.1 [24, Theorem 3.11], but there exist cases where it converges linearly with constant factor for local solutions of order $\rho = 2$. [24, Example 3.14]. According to [46], for most problem classes, especially when all involved functions are smooth, we can expect optimizers of order $\rho = 2$.

Remark 3.2: If one presumes convergence of the iterates $\boldsymbol{x}^k$ instead of the convergence of subsequences and that there is only a single global solution to problem (LLP), one can show superlinear convergence of the iterates instead of subsequences (see [24, Remark 3.13]).

4. Numerical experiments

We compare different approaches for generating lower bounds for problem (SIP). More specifically, we want to compare different subproblem formulations, i.e. using our proposed subproblem problem (LBP-REFINE), the usual approach problem (LBP) or the other derivative-based subproblems, problem (LBP-SENS-GEN) and problem (LBP-KKT) in Algorithm 1.

We implemented Algorithm 1 in our library for discretization-based algorithms libDIPS [21]. We use the conservatism limit $\sigma_{\max} = 10^4$, maximal reduction factor $\omega_{\max} =$

$\frac{15}{16}$, a scaling factor $\nu = 2$ and maximal aspect ratio $\alpha_{\max} = 16$. The interval enclosure for the Hessian matrix utilizes Chebyshev model arithmetic [47] in MC++ [48]. If interval extensions cannot be calculated using Chebyshev model arithmetic, interval arithmetic implemented in MC++ is used.

The subproblems are globally solved using MAiNGO 0.8.1 [49] using CPLEX 20.1.0 and Knitro 11.1.2 as subsolvers. The absolute and relative optimality tolerance for solving the lower-bounding problems was set to 10^{-4}. The optimality tolerances for the lower-level problems are adaptively chosen in libDIPS [21] (see the discussion in [50] and [6, Section 3.1.2]). We utilize default settings, except for a tightened inequality tolerance of 10^{-8} and selecting the incumbent as a linearization point if possible (`LBP_linPoints = 1`). The latter seemed to improve performance for problems containing variables only occurring in convex constraints. In the initial iteration, we find optimized bounds on $\mathcal{X}$ and $\mathcal{Y}$ by performing relatively coarse optimality-based bound tightening using MAiNGO with branching and local searches disabled (`BAB_maxIterations = 1` and `PRE_maxLocalSearches = 0`) based on the feasible sets of problem (LBP) and problem (LLP), respectively. This is repeated every 10 iterations for problem (LBP).

To test the different subproblems, we either use problem (LBP-REFINE) as detailed in Algorithm 1 or replace its use in Line 5 of Algorithm 1 by solving problem (LBP), problem (LBP-SENS-GEN) or problem (LBP-KKT). The resulting algorithms are tested on the subset of problems in [51] that fulfill Assumptions 1.2 and 1.1. The problems *Mitsos_2009-D1-0-1*, *Mitsos_2009-D1-0-2*, *Mitsos_2009-D1-0-1* and *Mitsos_2009-D1-0-1* contain a SIP constraint that is not differentiable on the boundary of $\mathcal{Y}$. Despite this, we can apply problem (LBP-REFINE) by ensuring all boxes containing the boundary are excluded from the set $\hat{\mathcal{D}}$. Similarly, problem (LBP-SENS-GEN) can be applied as derivatives are only evaluated at Slater points. In contrast, for problem (LBP-KKT), the formulation leads to problems where MAiNGO could not construct a relaxation, in particular, because zero as an argument to an inverse could not be ruled out.

We additionally included all problems from our library of test problems [21], where the symbolic differentiation implemented in libDIPS could be successfully applied, e.g. do not contain $\max(a, b)$, $|\cdot|$ or auxiliary variables in the definitions of the SIP constraint g, only contain linear constraints in the lower level and do not contain integer variables in the lower level. We conjecture that the lower-bounding approaches can be extended to allow for integer variables in the lower level by fixing the integer variables to their value at the discretization points.

We ran all algorithms for a maximum of 1000 iterations or 1200 s and a feasibility tolerance on the SIP constraint of 10^{-6}. Note that we do not implement an upper-bounding procedure and thus allow for termination with a nonzero feasibility tolerance. Therefore the points obtained by the lower-bounding procedures can be infeasible and thus superoptimal. In general, such a feasibility tolerance can lead to arbitrarily superoptimal results (see [52, Section 2.2] and [53] for a more detailed discussion). In our experiments, we did not observe this behavior in any instance where the feasibility tolerance was satisfied. The results for problems that could be solved in this sense by at least one approach and required more than one second for at least one approach are shown in Table 1.

Table 1. Comparison of lower-bounding approaches with feasibility tolerance of 10^{-6} on SIP constraint. Problems are sorted by the solution time when using problem (LBP). LBP: using problem (LBP), REF: using problem (LBP-REFINE), SENS: using problem (LBP-SENS-GEN), KKT: using problem (LBP-KKT). *T*: time limit of 1200 s was exceeded. *E*: Error in computing interval enclosures or relaxations for the derivatives in problem (LBP-KKT). †: Hessian $\boldsymbol{D}_{2,2}g$ is independent of upper-level variables.

bounding problem problem label in [21]	problem type ULP-LLP	solution time (rounded)				iterations			
		LBP	REF	SENS	KKT	LBP	REF	SENS	KKT
Mehrotra_2014-1	QCQP-NLP	< 1 s	< 1 s	< 1 s	E	2	2	2	1
Tichatschke_1988-1	QP-NLP	< 1 s	< 1 s	< 1 s	E	2	2	2	1
Tichatschke_1988-2	QCQP-NLP	< 1 s	< 1 s	< 1 s	E	2	2	2	1
Guo_2023-5-1-4	QCQP-QP	< 1 s	< 1 s	< 1 s	T	2	2	2	2
Gustafson_1973-2-2	LP-NLP†	< 1 s	1 s	< 1 s	E	6	7	6	1
Liu_1999-6-2	QP-NLP	< 1 s	1 s	2 s	< 1 s	7	7	3	2
RudnickCohen_2020-1	QCQP-QP†	< 1 s	< 1 s	< 1 s	15 s	5	5	5	5
Glashoff_1983-1-3-12	LP-NLP†	< 1 s	< 1 s	< 1 s	E	11	9	11	1
RudnickCohen_2020-2	NLP-NLP	< 1 s	< 1 s	< 1 s	36 s	5	5	4	5
Bonnans_2000-5-103	LP-NLP	< 1 s	< 1 s	< 1 s	7 s	8	8	5	7
Anderson_1989-2	QCQP-NLP†	< 1 s	< 1 s	< 1 s	2 s	4	3	4	3
Goberna_1998-10-13	LP-QP†	< 1 s	< 1 s	22 s	64 s	20	6	14	3
Zakovic_2003-17	QCQP-QP†	< 1 s	1 s	5 s	6 s	5	4	9	4
Coope_1998-1-1	LP-NLP	< 1 s	1 s	< 1 s	1 s	13	13	6	4
Mitsos_2009-MINLP1	NLP-NLP	< 1 s	T	< 1 s	T	3	2	3	2
Bonnans_2000-5-102	LP-NLP	< 1 s	1 s	2 s	10 s	14	11	9	10
Glashoff_1983-4-8-12	LP-NLP	< 1 s	1 s	1 s	3 s	11	11	8	7
Glashoff_1983-4-8-13	LP-NLP	< 1 s	1 s	2 s	T	15	10	12	3
Mitsos_2009-MINLP2	NLP-NLP	< 1 s	T	< 1 s	T	3	3	3	3
LoBianco_2001-3	NLP-NLP	< 1 s	< 1 s	< 1 s	2 s	3	3	2	2
Mitsos_2009-D1-0-1	NLP-NLP†	< 1 s	2 s	3 s	E	12	12	11	1
Mitsos_2009-D1-1-1	NLP-NLP†	1 s	1 s	3 s	E	13	11	10	1
Floudas_2008-CA	LP-NLP	1 s	4 s	4 s	77 s	14	13	12	10
Zakovic_2003-18	QCQP-QP†	1 s	6 s	1 s	137 s	13	10	13	9
Floudas_2008-DC	NLP-NLP	1 s	61 s	51 s	T	16	14	13	5
Mitsos_2009-DP	NLP-NLP	1 s	1 s	< 1 s	< 1 s	28	28	2	2
Guo_2020-3-19	NLP-NLP	1 s	3 s	14 s	4 s	17	17	12	11
Streit_1982-A8	QCQP-NLP	1 s	86 s	46 s	423 s	15	18	18	16
Lopez_2007-2	QCQP-QP†	2 s	< 1 s	3 s	< 1 s	42	2	9	2
Bhattacharjee_2005_1-H	QCQP-QP†	2 s	< 1 s	< 1 s	< 1 s	42	2	2	2
Kostyukova_2010-3-6	QCQP-NLP	3 s	18 s	T	T	26	25	7	7
Mitsos_2009-S	NLP-QP†	3 s	< 1 s	15 s	3 s	11	2	2	2
Mitsos_2009-D1-0-2	NLP-NLP	4 s	6 s	8 s	E	6	8	4	1
Pang_2022-20	NLP-NLP	22 s	22 s	22 s	22 s	2	2	2	2
Pang_2022-18	NLP-NLP	23 s	28 s	23 s	46 s	2	2	2	2
Pang_2022-19	NLP-NLP	26 s	98 s	28 s	161 s	2	2	2	2
Polak_1999-1	QCQP-NLP	28 s	T	406 s	T	22	17	15	4
Mitsos_2009-D1-1-2	NLP-NLP	96 s	268 s	T	E	21	26	9	1
Kostyukova_2010-3-7	QCQP-NLP	144 s	T	T	T	32	23	7	2
Lemonidis_2008-1-12	QCQP-QP†	250 s	239 s	259 s	250 s	1	1	1	1
Mitsos_2009-4-3-impl	LP-NLP	318 s	2 s	1 s	2 s	1000	16	11	8
Mitsos_2009-4-3	LP-NLP	357 s	2 s	1 s	1 s	1000	15	10	7
Gustafson_1979-2	LP-NLP	413 s	3 s	1 s	3 s	1000	16	10	8
Rueckmann_2009-3	QP-QP†	560 s	< 1 s	510 s	< 1 s	3	2	3	2
Mitsos_2009-4-6	LP-NLP	794 s	6 s	33 s	T	1000	25	25	4
Mitsos_2009-4-6-impl	LP-NLP	1079 s	9 s	35 s	T	1000	25	26	4
Watson_1983-8	LP-NLP	1101 s	639 s	1061 s	T	1000	25	1000	6
Gustafson_1973-2-4	LP-NLP	T	137 s	8 s	T	943	20	27	5
LoBianco_2001-2	LP-NLP	T	1066 s	T	T	105	55	14	4
Watson_1983-9	LP-NLP	T	107 s	20 s	T	500	32	17	5
Tanaka_1988-8-n10	LP-NLP	T	360 s	T	T	884	25	874	10

When comparing the traditional approach of using problem (LBP) with methods that utilize derivative information, we observe a trade-off. For cases where problem (LBP) requires only a small to medium amount of iterations, solving the more complicated subproblems in the derivative-based methods is either counterproductive in terms of solution time or provides no significant improvement. However, for problems where using problem (LBP) results in many iterations, derivative-based approaches tend to be advantageous, significantly reducing both the iteration count and overall computation time.

We observe that, as expected, using Algorithm 1 with problem (LBP-REFINE) terminates in the first or second iteration when the lower-level problem is a linear optimization problem or a convex quadratic problem with upper-level independent Hessian. However, many of the linear problems can usually be solved in less than 1 second by the other approaches as well. In general, when the Hessian of the SIP constraint $\boldsymbol{D}_{2,2}$ is independent of the upper-level variables, the number of required iterations remains low (below 15).

A deeper analysis of the problem instances where using Algorithm 1 did not perform well reveals that these problem instances frequently have very large bounds on the upper-level variables, even after bound tightening. Large parts of the upper-level feasible set remain feasible but are suboptimal. However, these are still considered in the calculation for the interval enclosure of the Hessian, for example, in the frequently occurring expression xy^2. This can be magnified if the dependency on the upper-level variable is nonlinear, as in expressions such as $\exp(x - y)$. Perhaps this could be improved by considering the restriction of the lower-level only on a limited region around the last upper-level iterate similar to the proposal in [15], but in the context of global solution of problem (SIP) by using disjunctions.

Using the additional constraints in problem (LBP-KKT) leads to the least number of iterations in most problems. This is despite the additional constraints using the discretization points only indirectly through the more focused refinement of boxes close to discretization points. Still, the solution times obtained with this lower-bounding approach are usually higher than with the others, and several subproblems could not be solved within the time limit, even for a relatively moderate iteration count. This underscores the difficulty of solving problem (LBP-KKT). Additionally, for several problems, this approach produced an error because potential domain violations were detected when calculating interval enclosures or relaxations inside MAiNGO. A major contribution to this is that the first derivative is considered over the whole box $\bar{\mathcal{Y}}^d, d \in \mathcal{D}$ instead of only at the discretization point. Similar to the treatment of non-Slater points in problem (LBP-SENS-GEN) and boxes with conservative restriction in problem (LBP-REFINE), a more sophisticated implementation could potentially avoid applying the derivative information on problematic boxes.

The lower-bounding approach using problem (LBP-SENS-GEN) also improves on the approach using problem (LBP) in most cases where using the latter requires more than 5 minutes to solve the problem. When compared to using problem (LBP-REFINE), the results are less consistent, with each approach being faster for a subset of problems. However, in instances where using problem (LBP-SENS-GEN) resulted in reaching the time limit while using problem (LBP-REFINE) successfully solved the problem, the number of iterations completed before reaching the limit was lower than the iterations

required by the successful approach. In contrast, when the approach using problem (LBP-REFINE) terminates due to the time limit, it had usually performed more iterations than were needed when using problem (LBP-SENS-GEN) to solve the problem successfully. This could indicate that, indeed, the subproblems tend to be easier to solve when using problem (LBP-SENS-GEN).

5. Conclusion

We propose a modified lower-bounding problem, which is adaptively refined as more discretization points are added. The approach is particularly efficient for problems with a certain structure, such as a convex lower-level problem or having a SIP constraint with a Hessian matrix that is independent of the upper-level variables.

We focus on using a reduced number of discretization points by including derivative information in the lower-bounding. This seems indeed beneficial for problems that are hard to solve by the traditional approach problem (LBP) because many iterations are required to reach the feasibility tolerance. This is despite introducing additional variables and constraints that make the subproblems more difficult to solve. It would be interesting to investigate if the usefulness of the discussed derivative-based relaxations can be predicted and adaptively applied during the solution process. Overall, the traditional approach seems to be well complemented by the derivative-based approaches using problem (LBP-REFINE) and problem (LBP-SENS-GEN).

We obtain the discretization point by the traditional approach for adaptive discretization algorithms, that is, by solving problem (LLP). Alternatively, one could change the way discretization points are selected in addition to applying additional derivative information to reduce the number of iterations. In [22], the discretization points are optimized locally with respect to the resulting lower bound. Perhaps the two approaches of choosing better discretization points and using tighter relaxation could be applied synergistically.

There are several ways to extend the presented work to more general problems. The first is to consider p lower-level inequality constraints that depend on the upper-level variables. The resulting GSIP can, under certain assumptions, be reformulated to a SIP with a semi-infinite constraint of the form $g(\boldsymbol{x}, \boldsymbol{y}) = \min_{i \in \{1,\dots,p+1\}} \tilde{g}_i(\boldsymbol{x}, \boldsymbol{y})$. The nonsmooth min operation complicates a direct extension of the presented approach to this SIP. In the specific case where the lower level has linear constraints for fixed upper-level variables and linear dependency on the upper-level variables, a direct extension should be possible by following the derivation in [54]. However, this would lead to the introduction of several bilinear terms.

The extensions to SIPs with convex quadratic constraints in the lower level would suffer from a similar issue. Instead of embedding the dual of a QP, we would need to embed the dual of a QCQP. A dual of a convex QCQP that is itself a convex QCQP is not known; thus, embedding the dual would likely necessitate the inclusion of second-order cone constraints in the resulting lower-bounding problem. In comparison to linear constraints, these constraints are not handled efficiently by current deterministic global optimizers for nonconvex optimization. As another extension, we could mirror the approach in [15] to handle an arbitrary lower-level feasible set by discarding parts of refined boxes that have an empty intersection with the feasible set.

Further, we conjecture that the proposed approaches can be applied in the presence of discrete lower-level variables with additional implementation effort by considering the discrete variables as fixed to the values of the discretization points.

In the case of standard SIP, adding a discretization point in problem (LBP) only adds a constraint, while the derivative-based approaches add additional variables. In solution methods for generalizations of SIP, such as ESIP [18] and generalized SIP with coupling equality constraints [19, 20], where additional variables are added anyway, a reduction in the number of iterations could be even more impactful. For ESIP, extending the lower-bounding approach proposed by Djelassi and Mitsos [18] appears relatively straightforward. In the case of generalized SIPs with linear constraints in the lower level and coupling equality constraints satisfying a uniqueness assumption [19], exploring the use of the dual of a convex quadratic restriction, as outlined in problem (LBP-REFINE), could be promising. However, addressing the potential issue of an empty feasible set at the lower level would be required, potentially by adopting a strategy similar to that described in [12, Section 6].

We focused on lower-bounding. When a convergent upper-bounding procedure is employed, termination based on optimality tolerance is possible. On the one hand, this might prevent a large number of iterations seen in some of our experiments, potentially reducing the advantage of employing derivative information. On the other hand, the upper-bounding approaches in [5, 6] based on a modification of problem (LBP), could themselves be enhanced with derivative information.

As discussed, the box refinement described in Section 2.6 is not appropriate for upper-bounding with problem (UBP-Heur). A convergent upper-bounding procedure similar to the approach in [11, 15] could potentially be obtained by considering a separate set of boxes. Such an upper-bounding problem could further be extended to choose the relaxation coefficients adaptively, as proposed in [55]. This, however, necessitates robust solvers for solving highly nonconvex problems, including a relatively small SDP constraint.

Acknowledgements

We acknowledge the use of the GPT-3.5 and 4.0 language model developed by OpenAI as an implementation aid, specifically as a code snippet generator, for creating shell scripts to automate the execution of the benchmark tests on the RWTH High Performance Computing cluster and for generating ideas for language improvements in already written text.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Réseau de transport d' électricité (RTE) within the project 'Hierarchical Optimization for Worst-Case Analysis under Uncertainty with Flexibility Analysis for Power Grids and its Extension to SOCP Formulation'. Computations were performed with computing resources granted by RWTH Aachen University.

Data availability statement

The benchmark results generated and analyzed during the current study are available in a json file format on reasonable request.

ORCID

Aron Zingler http://orcid.org/0000-0003-1135-0236
Alexander Mitsos http://orcid.org/0000-0003-0335-6566

References

[1] Polak E. Semi-infinite optimization in engineering design. In: Semi-Infinite Programming and Applications: An International Symposium; 1981 Sep 8–10. Austin, Texas: Springer; 1981. p. 236–248. doi: 10.1007/978-3-642-46477-5_16.
[2] Hettich R, Kortanek KO. Semi-infinite programming: theory, methods, and applications. SIAM Rev. 1993;35(3):380–429. doi: 10.1137/1035089.
[3] Djelassi H, Mitsos A, Stein O. Recent advances in nonconvex semi-infinite programming: applications and algorithms. EURO J Comput Optim. 2021;9:100006. doi: 10.1016/j.ejco.2021.100006.
[4] Blankenship JW, Falk JE. Infinitely constrained optimization problems. J Optim Theory Appl. 1976;19(2):261–281. doi: 10.1007/bf00934096.
[5] Mitsos A. Global optimization of semi-infinite programs via restriction of the right-hand side. Optimization. 2011;60(10-11):1291–1308. doi: 10.1080/02331934.2010.527970.
[6] Djelassi H, Mitsos A. A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs. J Glob Optim. 2017;68(2):227–253. doi: 10.1007/s10898-016-0476-7.
[7] Tsoukalas A, Rustem B. A feasible point adaptation of the blankenship and falk algorithm for semi-infinite programming. Optim Lett. 2011;5(4):705–716. doi: 10.1007/s11590-010-0236-4.
[8] Bhattacharjee B, Lemonidis P, Green Jr WH, et al. Global solution of semi-infinite programs. Math Program. 2005;103:283–307. doi: 10.1007/s10107-005-0583-6.
[9] Lemonidis P. *Global optimization algorithms for semi-infinite and generalized semi-infinite programs* [PhD thesis]. Massachusetts Institute of Technology; 2008.
[10] Mitsos A, Lemonidis P, Lee CK, et al. Relaxation-based bounds for semi-infinite programs. SIAM J Optim. 2008;19(1):77–113. doi: 10.1137/060674685.
[11] Stein O, Winterfeld A. Feasible method for generalized semi-infinite programming. J Optim Theory Appl. 2010;146:419–443. doi: 10.1007/s10957-010-9674-5.
[12] Diehl M, Houska B, Stein O, et al. A lifting method for generalized semi-infinite programs based on lower level Wolfe duality. Comput Optim Appl. 2013;54:189–210. doi: 10.1007/s10589-012-9489-4.
[13] Stein O, Still G. Solving semi-infinite optimization problems with interior point techniques. SIAM J Control Optim. 2003;42(3):769–788. doi: 10.1137/s0363012901398393.
[14] Floudas CA, Stein O. The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM J Optim. 2008;18(4):1187–1208. doi: 10.1137/060657741.
[15] Steuermann H-P. *Adaptive algorithms for semi-infinite programming with arbitrary index sets* [PhD thesis]. 2011. doi: 10.5445/IR/1000023658.
[16] Marendet A, Goldsztejn A, Chabert G, et al. A standard branch-and-bound approach for nonlinear semi-infinite problems. Eur J Oper Res. 2020;282(2):438–452. doi: 10.1016/j.ejor.2019.10.025.
[17] Djelassi H. *Discretization-based algorithms for the global solution of hierarchical programs* [Dissertation]. Aachen: Rheinisch-Westfälische Technische Hochschule Aachen; 2020. doi: 10.18154/RWTH-2020-09163.
[18] Djelassi H, Mitsos A. Global solution of semi-infinite programs with existence constraints. J Optim Theory Appl. 2021;188(3):863–881. doi: 10.1007/s10957-021-01813-2.
[19] Djelassi H, Glass M, Mitsos A. Discretization-based algorithms for generalized semi-infinite and bilevel programs with coupling equality constraints. J Glob Optim. 2019;75(2):341–392. doi: 10.1007/s10898-019-00764-3.
[20] Zingler A, Lipow AW, Mitsos A Discretization algorithms for general semi-infinite programs with coupling equality constraints under local solution stability. J Glob Optim. 2024;93:27–61.

[21] Jungen D, Zingler A, Djelassi H, et al. libDIPS–discretization-based semi-infinite and bilevel programming solvers. Optimization Online preprint, 2023.
[22] Turan EM, Jäschke J, Kannan R. Optimality-based discretization methods for the global optimization of nonconvex semi-infinite programs. preprint, 2023. arXiv:2303.00219.
[23] Seidel T, Küfer K-H. An adaptive discretization method solving semi-infinite optimization problems with quadratic rate of convergence. Optimization. 2022;71(8):2211–2239. doi: 10.1080/02331934.2020.1804566.
[24] Seidel T. *Solving semi-infinite optimization problems with quadratic rate of convergence* [PhD thesis]. 2020. doi: 10.24406/publica-fhg-283154.
[25] Okuno T, Hayashi S, Yamashita N, et al. An exchange method with refined subproblems for convex semi-infinite programming problems. Optim Methods Softw. 2016;31(6):1305–1324. doi: 10.1080/10556788.2015.1124432.
[26] Gomoto K. *An exchange method with refined subproblems for convex semi-infinite programming problems* [PhD thesis]. Kyoto University; 2014.
[27] Androulakis IP, Maranas CD, Floudas CA. αBB: a global optimization method for general constrained nonconvex problems. J Glob Optim. 1995;7:337–363. doi: 10.1007/bf01099647.
[28] Hasan MMF. An edge-concave underestimator for the global optimization of twice-differentiable nonconvex problems. J Glob Optim. 2018;71(4):735–752. doi: 10.1007/s10898-018-0646-x.
[29] Skjäl A, Westerlund T. New methods for calculating αBB-type underestimators. J Glob Optim. 2014;58(3):411–427. doi: 10.1007/s10898-013-0057-y.
[30] Skjäl A, Westerlund T, Misener R, et al. A generalization of the classical αBB convex underestimation via diagonal and nondiagonal quadratic terms. J Optim Theory Appl. 2012;154:462–490. doi: 10.1007/s10957-012-0033-6.
[31] Dorn WS. Duality in quadratic programming. Q Appl Math. 1960;18(2):155–162. doi: 10.1090/qam/112751.
[32] Khajavirad A, Sahinidis NV. A hybrid Lp/nlp paradigm for global optimization relaxations. Math Program Comput. 2018;10(3):383–421. doi: 10.1007/s12532-018-0138-5.
[33] Stephens C. *Global optimization: theory versus practice* [en. PhD thesis]. 1997. doi: 10.26021/3523.
[34] Higham NJ. Computing a nearest symmetric positive semidefinite matrix. Linear Algebra Appl. 1988;103:103–118. doi: 10.1016/0024-3795(88)90223-6.
[35] Alefeld G, Mayer G. Interval analysis: theory and applications. J Comput Appl Math. 2000;121(1-2):421–464. doi: 10.1016/S0377-0427(00)00342-3
[36] Streeter M, Dillon JV. Automatically bounding the Taylor remainder series: tighter bounds and new applications. preprint, 2022. arXiv:2212.11429.
[37] Lee D, Turitsyn K, Slotine J-J. Sequential convex restriction and its applications in robust optimization. preprint, 2019. arXiv:1909.01778.
[38] Lee D, Nguyen HD, Dvijotham K, et al. Convex restriction of power flow feasibility sets. IEEE Trans Control Netw Syst. 2019;6(3):1235–1245. doi: 10.1109/tcns.2019.2930896.
[39] Strahl WR, Raghunathan AU, Sahinidis NV, et al. Constructing tight quadratic relaxations for global optimization: ii. underestimating difference-of-convex (d.c.) functions. preprint, 2024. arXiv:2408.13058.
[40] Adjiman CS, Androulakis IP, Floudas CA. A global optimization method, αBB, for general twice-Differentiable constrained NLPs–II. Implementation and computational results. Comput Chem Eng. 1998;22(9):1159–1179. doi: 10.1016/S0098-1354(98)00218-X.
[41] Adjiman CS, Floudas CA. The αBB global optimization algorithm for nonconvex problems: an overview. In: Migdalas A, Pardalos PM, Värbrand P, editors. From local to global optimization. Boston, MA: Springer; 2001. p. 155–186. doi: 10.1007/978-1-4757-5284-7_8.
[42] Boggs PT, Tolle JW, Wang P. On the local convergence of Quasi-Newton methods for constrained optimization. SIAM J Control Optim. 1982;20(2):161–171. doi: 10.1137/0320014.
[43] McCormick GP. Second order conditions for constrained minima. SIAM J Appl Math. 1967;15(3):641–652. doi: 10.1137/0115056.

[44] Kroó A. The continuity of best approximations. Acta Math Acad Sci Hung. 1977;30:175–188. doi: 10.1007/BF01895663.
[45] Hettich RP, Jongen HT. Semi-infinite programming: conditions of optimality and applications. In: Stoer J, editor, Optimization Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg; 1978. p. 1–11.
[46] Still G. Discretization in semi-infinite programming: the rate of convergence. Math Program. 2001;91:53–69. doi: 10.1007/s101070100239.
[47] Brisebarre N, Joldeş M. Chebyshev interpolation polynomial-based tools for rigorous computing Munich. Germany Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation; Munich, Germany; 2010. doi: 10.1145/1837934.1837966.
[48] Chachuat B. Mc++(version 2.0): A Toolkit for bounding factorable functions; 2019. Available from: https://github.com/omega-icl/mcpp.
[49] Bongartz D, Najman J, Sass S, et al. MAiNGO – McCormick-based algorithm for mixed-integer nonlinear global optimization [Tech. rep.]. Process Systems Engineering (AVT.SVT), RWTH Aachen University; 2018. [accessed 2023 Jul 20]. Available from: http://www.avt.rwth-aachen.de/global/show_document.asp?id = aaaaaaaaabclahw.
[50] Harwood SM, Papageorgiou DJ, Trespalacios F. A note on semi-infinite program bounding methods. Optim Lett. 2021;15(4):1485–1490. doi: 10.1007/s11590-020-01638-4.
[51] Mitsos A. Test set of semi-infinite programs. Aachen, Germany, 2016. [accessed 2024 Oct 17]. Available from: https://www.avt.rwth-aachen.de/cms/AVT/Forschung/Systemverfahrenstechnik/ kpdo/A-Test-Set-of-Semi-Infinite-Programs/?lidx = 1.
[52] Bienstock D, Del Pia A, Hildebrand R. Complexity, exactness, and rationality in polynomial optimization. Math Program. 2023;197(2):661–692. doi: 10.1007/s10107-022-01818-3
[53] Beck Y, Bienstock D, Schmidt M, et al. On a computationally ill-behaved bilevel problem with a continuous and nonconvex lower level. J Optim Theory Appl. 2023;198(1):428–447. doi: 10.1007/s10957-023-02238-9
[54] Levitin ES, Tichatschke R. A branch-and-bound approach for solving a class of generalized semi-infinite programming problems. J Glob Optim. 1998;13(3):299–315. doi: 10.1023/a:1008245113420.
[55] Houska B. Robust optimization of dynamic systems [PhD thesis]. KU Leuven; 2011. ISBN: 978-94-6018-394-2.

Appendix

A.1 Additional definitions and theorems from [24]

The following definitions are taken from [24] with minor adjustments for consistent notation.

Definition A.1 (Radius of a minimum): Let f and $\mathcal{M}$ denote the objective function and feasible region of an optimization problem. A feasible point $\boldsymbol{x}^*$ is called a local minimum with radius r if for every $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{B}_r(\boldsymbol{x}^*)$ the following holds:

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$$

Definition A.2 (Order of a minimum): Let f and $\mathcal{M}$ denote the objective function and feasible region of an optimization problem. A local minimum, $\boldsymbol{x}^*$, is said to be of order $\rho \in \mathbb{N}$, if there is a constant $L > 0$ and a radius $r > 0$ such that, for every $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{B}_r(\boldsymbol{x}^*)$, the following holds:

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \geq L\|\boldsymbol{x} - \boldsymbol{x}^*\|^\rho$$

Definition A.3 (Reduction Ansatz going back to [45]): Suppose LICQ holds for the lower-level problem (LLP) at every point in $\mathcal{Y}$ and let $\boldsymbol{x}^* \in \mathcal{M}_{SIP}$. The Reduction Ansatz is satisfied for problem (SIP) at $\boldsymbol{x}^*$ if strict complementary slackness and second-order sufficient condition hold for problem (LLP) for $\bar{\boldsymbol{x}} = \boldsymbol{x}^*$ and for every active lower-level point, i.e. for $\boldsymbol{y} \in \mathcal{Y} : g(\boldsymbol{x}^*, \boldsymbol{y}) = 0$.

Definition A.4: A feasible point $\boldsymbol{x} \in \mathcal{M}_{SIP}$, satisfies the Extended Mangasarian-Fromovitz Constraint Qualification (EMFCQ), if there is a vector $\boldsymbol{\epsilon} \in \mathbb{R}^n$ with

$$\boldsymbol{D}_1 g(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{\epsilon} \leq -1, \quad \forall\, \boldsymbol{y} \in \mathcal{Y} : g(\boldsymbol{x}, \boldsymbol{y}) = 0$$

Theorem A.1 ([24, Theorem 3.9]): *If Assumption 3.1 holds and the objective function f of the SIP is Lipschitz-continuous near $\boldsymbol{x}^*$, then*

(1) *There is a constant $L_1 > 0$ and a $k' \in \mathbb{N}$ such that*

$$0 \leq f(\boldsymbol{x}^*) - f(\boldsymbol{x}^k) \leq L_1 \alpha_k, \quad \forall\, k > k'$$

(2) *There is a constant $L_2 > 0$ and a $k'' \in \mathbb{N}$ such that*

$$\|\boldsymbol{x}^* - \boldsymbol{x}^k\| \leq L_2 \alpha_k^{\frac{1}{\rho}}, \quad \forall\, k > k''$$

where $\alpha_k := \max\limits_{\boldsymbol{y} \in \mathcal{Y}} \max\{0, g(\boldsymbol{x}^k, \boldsymbol{y})\}$ is the maximal violation of the iterate $\boldsymbol{x}^k$.